KERNEL METHODS FOR THE INCORPORATION OF PRIOR-KNOWLEDGE INTO SUPPORT VECTOR MACHINES

ANTOINE VEILLARD

NATIONAL UNIVERSITY OF SINGAPORE

2012

KERNEL METHODS FOR THE INCORPORATION OF PRIOR-KNOWLEDGE INTO SUPPORT VECTOR MACHINES

ANTOINE VEILLARD

(M.Eng., École Polytechnique, Palaiseau, France)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2012

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Antoine Veillard

16 August 2012

Acknowledgments

I would like to express my indeptedness to my advisors, Dr Stéphane Bressan and Dr Daniel Racoceanu, whose experience and patience have been invaluable to me.

My appreciation also go to my colleagues at the National University of Singapore and at the Image and Pervasive Access Lab for our mutiple collaborations and for their informal support.

I am also particularly grateful to the team of anatomopathologists and engineers from the Groupement Hospitalier Pitié-Salpetrière of Paris for providing me with their expertise and assistance in the context of the MICO project.

Finally, I dedicate this thesis to my family and my best friend. This thesis would certainly not have existed without their moral and emotional support.

List of Author's Publications

[1] A. Veillard, M. S. Kulikova, S. Bressan and D. Racoceanu. SVM-based framework for the robust extraction of objects from histopathological images using color, texture, scale and geometry. Submitted.

[2] A. Veillard, D. Racoceanu and S. Bressan. pRBF kernels: a framework for the incorporation of task-specific properties into support vector methods. Submitted.

[3] A. Veillard, M. S. Kulikova, and D. Racoceanu. Cell nuclei extraction from breast cancer histopathology images using color, texture, scale and shape information. In *Proc. European Congress on Telepathology and International Congress on Virtual Microscopy*, 2012.

[4] M. S. Kulikova, A. Veillard, L. Roux and D. Racoceanu. Nuclei extraction from histopathological images using a marked point process approach. In *Proc. SPIE Medical Imaging*, 2012.

[5] A. Veillard, D. Racoceanu, and S. Bressan. Incorporating prior-knowledge in support vector machines by kernel adaptation. In *Proc. International Conference on Tools with Artificial Intelligence*, 2011.

[6] C-H. Huang, A. Veillard, L. Roux, N. Loménie, and D. Racoceanu. Time-efficient sparse analysis of histopathological whole slide images. *Computerized Medical Imaging and Graphics*, 35:579-591, 2011.

[7] A. Veillard, N. Loménie, and D. Racoceanu. An exploration scheme for large images: application to breast cancer grading. In *Proc. International Conference on Pattern Recognition*, 2010.

[8] A. Veillard, E. Melissa, C. Theodora, and S. Bressan. Learning to rank indonesian-english machine translations. In *Proc. International MALINDO Workshop*, 2010.

[9] A. Veillard, E. Melissa, C. Theodora, D. Racoceanu, and S. Bressan. Support vector methods for sentence level machine translation evaluation. In *Proc. Tools with Artificial Intelligence*, 2010.

[10] L. Roux, A. E. Tutac, A. Veillard, J.-R. Dalle, D. Racoceanu, N. Loménie and J. Klossa. A cognitive approach to microscopy analysis applied to automatic breast cancer grading. In *Proc. European Congress of Pathology*, 2009.

[11] L. Roux, A. E. Tutac, N. Loménie, D. Balensi, D. Racoceanu, A. Veillard, W.-K. Leow, J. Klossa and T. C. Putti. A cognitive virtual microscopic framework for knowlege-based exploration of large microscopic images in breast cancer histopathology. In *Proc. Engineering in Medicine and Biology Society*, 2009.

[12] D. Racoceanu, A. E. Tutac, W. Xiong, J.-R. Dalle, C.-H. Huang, L. Roux, W.-K. Leow, A. Veillard, J.-H. Lim and T. C. Putti. A virtual microscope framework for breast cancer grading. In *Proc. A-STAR CCo workshop in Computer Aided Diagnosis, Treatment and Prediction*, 2009.

Contents

Summary

This thesis is dedicated to a number of original methods for the incorporation of prior-knowledge into Support Vector Methods (SVM) based on modifications of the pervasively used Radial Basis Function (RBF) kernel. The methods proposed in this thesis are collectively referred to as the knowledge-enhanced RBF (KE-RBF) framework.

SVMs are a class of state-of-the-art supervised learning algorithm implementing the structural risk minimization principle first proposed by the mathematician Vladimir N. Vapnik. In combination with the general purpose RBF kernel, it has been applied to successfully solve many complex, real-life problems. However, the required amount of training data can be very high making the SVM option unavailable in many practical situations.

Often, prior-knowledge on the task is available and could be used together with labeled data for training. This requires specific methods to be developed since by its design, the SVM takes only labeled data points as input.

The KE-RBF framework is a set of original kernel methods for the incorporation of prior-knowledge into SVMs. It comprises 3 new kernels (the ξRBF, pRBF and gRBF kernels) based on transformations of the RBF kernel widely used in machine learning. It gives systematic methods for the incorporation of properties specific to the problem while retaining the versatility making the popularity of the RBF kernel.

The KE-RBF kernels allow for the incorporation of a wide array of commonly available problem-specific prior-knowledge including global properties such as monotonicity, pseudo-periodicity or characteristic correlation patterns and semi-global properties represented by unlabelled and labelled regions.

KE-RBF kernels are highly usable in practice and pave the way for several interesting

new possibilities with SVMs such as learning with very small or strongly biased datasets as shown in a benchmark based on 5 different applications using real-world and synthetic data from a wide variety of domains of application.

We show that the KE-RBF framework is highly usable in practice, has the potential to largely improve learning performances over the RBF kernel, and sharply reduces the requirements in training data.

In particular, the good results obtained with very small or strongly biased training sets pave the way for several interesting new possibilities of application of SVMs beyond their standard limits.

Finally, we propose a valorization of our contribution through a computer-aided breast cancer grading application able to satisfy the actual operational requirements of the pathologists. This application demonstrates how the KE-RBF framework can work as one of the numerous components or a complex, real-life engineering project an proves the operational readiness of the framework.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The study of biopsy micrographs from surgically extracted breast tumors is currently the gold standard for the assessment of breast cancer and is performed routinely in daily clinical practice. This task known as Breast Cancer Grading (BCG) provides essential prognostic and management information for the pathology. Therefore, a good grading has a great impact on the quality of the medical care and the reduction of human and financial costs due to misdiagnosis. Unfortunately, BCG is a highly qualified job requiring a large amount of work from experienced pathologists. Moreover, the tedious nature of the task makes it prone to frequent errors.

Many specialized and repetitive tasks such as BCG could greatly benefit from a partial or full automatization. Nevertheless, they often require the knowledge and experience of highly-qualified specialists which is not simple to model. Therefore, powerful methods able to extract and model the complex know-how of specialists accomplishing complex tasks are necessary.

Support Vector Machines (SVM) with their numerous variants for classification and regression tasks are state-of-the-art machine learning algorithms which can be used for this purpose. Some of their key features are the absence of local optima, the possibility to control over-fitting and the use of kernels. In combination with the nonlinear Radial Basis Function (RBF) kernel, they provide a powerful and versatile learning tool often used as a default choice in many real-world applications.

SVMs are supervised statistical learning algorithms: they work by extracting the

knowledge about the task implicitly contained in a training set of annotated samples. Therefore, as long as training data is available in sufficient quantity and quality, the SVM+RBF combination can be applied as a general-purpose learning black-box on the data and often produce good results. On complex problems, such methods can however lead to steep requirements in training data. Unfortunately, countless reasons (cost issues, time constraints, ethical reasons, etc...) make training data for real-world problems hard to obtain.

Meanwhile, real-world problems are seldom black-boxes as some general or specific knowledge about the task is often available. In some cases, specific information on the category or the range of the parameters may be available: *e.g.* "a non-smoking person less than 20 years of age is at very low risk of developing breast cancer". In other cases, specific patterns may be known: *e.g.* "the breaking distance of a car is quadratically correlated to its velocity". Although insufficient to fully characterize a particular task, such information can provide a very substantial help in modeling the problem. Thus, it seems natural to rely upon such additional *prior-knowledge* when training data is insufficient.

In most of the cases, the "learning-by-examples" paradigm embodied by supervised learning is not a natural analogy of the way concepts are defined in real life. For instance, histopathology textbooks describe a specific disease with text and a small amount of micrographs exemplifying typical cases rather than an exhaustive collection of micrographs covering possible positive and negative cases. Therefore, problems for which a limited amount of examples is available with some formalized knowledge are arguably more common in real-life than tasks for which examples are unlimited but nothing else in particular is known.

In this thesis, we propose the Knowledge-Enhanced RBF (KE-RBF) kernel framework, a family of kernel methods for the incorporation of prior-knowledge into SVMs. Based upon adaptations of the standard RBF kernel according to the prior-knowledge, they aim at incorporating properties highly characteristic of particular problems while preserving the versatility making the popularity of the RBF kernels.

The framework consists of three distinct types of kernels: ξRBF kernels, pRBF kernels and gRBF kernels. Our original KE-RBF framework allows for the incorpora-

tion of a wide array of commonly available problem-specific prior-knowledge including global properties such as monotonicity, pseudo-periodicity or characteristic correlation patterns; and semi-global properties represented by unlabeled and labeled regions.

The KE-RBF framework ally effectiveness with ease of use, and pave the way for several interesting new possibilities with SVMs such as learning with very small or strongly biased datasets. Accordingly, our work significantly contributes towards a shift of paradigm for a more practical use of SVMs: from an often unrealistic situation where lots of training data are required to a more practical situation where a limited amount of data in addition to some problem-specific advice is available.

1.2 Objectives

This thesis has three objectives: a didactic goal, a research goal and a valorization goal.

First, we will provide a didactic tutorial to the SVM from a statistical standpoint. Instead of describing it as a geometrical construction which is not able *per se* to justify its good average performances, the SVM will be presented as the implementation of the structual risk minimization principle, a theoretically validated strategy originally proposed by the Russian mathematician Valdimir N. Vapnik and able to achieve a specific statistical goal. The tutorial is intended for anybody who is not familiar with the statistical aspect behind the SVM and is interested in "why" the SVM works rather than just "how" it works.

The required specialized notions will be introduced in an concise and organized fashion including: the positive-definite kernels and the Moore-Aronszajn theorem, the reproducing kernel Hilbert spaces and the representer theorem, the use of strong duality in convex optimization, and the computation of statistical learning bounds in supervised learning. Only a basic mathematical background is pre-required from the reader.

A particular emphasis will be put upon the importance of choosing the right kernels and their associated reproducing kernel Hilbert space.

Then, sustained research work will be conducted on the central topic of this thesis: the incorporation of prior-knowledge into SVMs. Following a review of the current state-

of-the art identifying gaps and promising leads, we will present the KE-RBF framework, our original kernel-based solution to the problem.

The KE-RBF framework, based on adaptations of the standard RBF kernel, can be subdivided into 3 families of kernel methods: ξRBF kernels, pRBF kernels and gRBF kernels. Together, they enable the incorporation of a wide range of prior-knowledge specific to the task including global properties such as monotonicity, pseudo-periodicity or characteristic correlation patterns; and semi-global properties represented by unlabeled and labeled regions of the feature space.

Following their theoretical description and validation, a systematic empirical evaluation of the framework will be conducted on several applications using real-world and synthetic data covering fields as diverse as meteorology, oncology, signal processing and zootomy. We aim to show that the methods are easy to use in practice, have the potential to largely improve learning performances and are able to sharply reduces the requirements in training data by making use of the prior-knowledge. In particular, we will demonstrate that they enable learning with very small or strongly biased training sets significantly broadening the field of application of SVMs.

Finally, we will propose a valorization of our contribution through an application to BCG aimed at satisfying actual operational needs of pathologists. The BCG system is a central component of the MICO[1] project funded by the Agence Nationale pour la Recherche (France). It involves industrial partners and pathologists from a university hospital. Therefore, a strong emphasis is put on the validity of the approach from a medical standpoint and its operational viability in a real clinical environment

Our application will be a complete approach to BCG including a robust detection and extraction of histological structures from complex images combining a wide range of information including color, texture, scale and geometry in a machine learning framework; a local frame-level BCG using the gRBF kernel to combine annotated medical data and formalized medical knowledge; and an efficient strategy based on dynamic sampling and computational geometry tools to explore large images for the grading of entire slides within an operationally acceptable timeframe.

[1] `http://ipal.cnrs.fr/project/mico`

1.3 Outline

This thesis has the following structure.

In Chapter 2, we propose a statistical introduction to SVMs as an implementation of the structural risk minimization principle rather than the more common place geometrical approach.

In Chapter 3, we provide a structured and critical review of the state-of-the-art in prior-knowledge incorporation methods into SVMs. We identify the strengths and weaknesses of the respective methods in-line with the objective of dealing with small training sets and propose promising leads.

The KE-RBF kernel framework which constitutes the original contribution of this thesis is presented in Chapter 4. It comprises 3 new kernels (the ξRBF, pRBF and gRBF kernels) based on transformations of the RBF kernel pervasively used in machine learning.

Then, the KE-RBF kernels are validated in an extensive and detailed performance evaluation based on 5 different applications in Chapter 5.

Finally, our BCG system which includes an application of KE-RBF kernels is presented in Chapter 6.

Chapter 2

A Statistical Introduction to Support Vector Methods

2.1 Introduction

In this chapter, we propose a comprehensive tutorial on support vector methods (SVM), a class of state-of-the-art supervised learning algorithms which can be applied both to classification and regression tasks.

SVMs are often presented from a geometrical standpoint as the construction of a hyperplane in a real Hilbert space. The hyperplane is used to separate classes or as a regression model. An excellent tutorial adopting this perspective is available from Burges [4]. Although this geometrical approach fully describes the SVM, it does not provide a mathematical explanation for the good statistical performances of the SVM.

In fact, the SVM can be justified as the implementation of a statistically sound strategy known as the structural risk minimization (SRM) principle. In the present tutorial, we make the choice to follow this statistical approach by presenting SVMs as a natural implementation of the SRM principle.

Basic notions in differential analysis, linear algebra, Hilbertian geometry and probability theory are prerequired from the reader. More specialized notions are progressively introduced throughout this tutorial.

2.1.1 A brief History of the SVM

If the SRM principle was established by Vapnik and Chervonenkis as early as 1974 [83], it is only much later that the first SVM for classification tasks was poposed by Cortes and Vapnik [6] and recognized as an interesting alternative to the state-of-the-art statistical learning algorithms such as neural networks (NN). A version for scalar regression was also proposed the same year by Vapnik [81].

Although the SRM principle itself was anterior to the SVM, rigorous statistical learning bounds for the classification and regression cases were only proposed in 2000 by Shawe-Taylor and Cristianini [71] for the soft-margin SVM.

Today, Vapnik's SVM and its many variations are widely considered among the best supervised learning algorithms du to their learning power, generalizability and versatility.

2.1.2 Outline

First, an introduction to the theory of positive-definite kernels is given in Section 2.2. The notions convered in this presentation are the kernel trick, reproducing kernels and the representer theorem.

In Section 2.3, a few notions in convex optimization theory related to Lagrangians and the primal-dual reformulation of problems are presented and will be subsequently used to derive the SVM algorithm from the SRM principle.

The SRM principle itself is presented and theorecially justified in Section 2.4. The particular formulation of the SRM presented here is based on Rademacher's complexity theory.

Section 2.5.1 is dedicated to support vector classifiers (SVC), *i.e.* SVMs for classification tasks, constructed as an implementation of the SRM principle. Support vector regressions (SVR) are then presented in section 2.5.2 as an adaptation of SVCs to regression problems.

The link with the better-known geometrical interpretation is made in Section 2.5.3 and common variants of SVMs are presented in Section 2.5.4.

2.2 Kernel theory

Kernels are a simple mathematical notion with major applications which are both theoretical in the field of Hilbertian analysis and practical in computer science.

The positive-definite (PD) kernels described in Section 2.2.1 are of a particular importance as they can be used to manipulate data embedded into potentially complex Hilbert spaces (Section 2.2.3) through inner product evaluations.

PD kernels have very significant applications in computer science and statistical machine learning in particular for two major reasons. First, they enable a simple algorithmic strategy, known as the kernel trick (Section 2.2.2), which can tremendously improve the usefulness of many linear algorithms. Second, they allow for the reformulation of optimization problems into an efficiently solvable form, a result known as the representer theorem and presented in Section 2.2.4.

2.2.1 Positive definite kernels

This section defines a few notions related to PD kernels together with examples, and introduces a central result known as the Moore-Aronszajn theorem.

Definition 2.2.1. *Positive definite (PD) kernel*

Let $\mathcal{X}$ be a non-empty set. A positive definite kernel over $\mathcal{X}$ is a funtion $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that:

1. K is symmetric.
2. $\forall N \in \mathbb{N}$, $\forall (x_1, x_2, \ldots, x_N) \in \mathcal{X}^N$, $\forall (v_1, v_2, \ldots, v_N) \in \mathbb{R}^N$:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} v_i v_j K(x_i, x_j) \geq 0 \tag{2.1}$$

Definition 2.2.2. *Strictly PD kernel*

Let K be a PD kernel over $\mathcal{X}$.

If $\forall N \in \mathbb{N}$, $\forall (x_1, x_2, \ldots, x_N) \in \mathcal{X}^N$ **pairwise distinct**, $\forall (v_1, v_2, \ldots, v_N) \in \mathbb{R}^N$:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} v_i v_j K(x_i, x_j) = 0 \implies \forall i \in [\![1, N]\!],\ v_i = 0 \tag{2.2}$$

then, we say that K is *strictly positive*.

Definition 2.2.3. *Gram matrix of a PD kernel*

Let K be a PD kernel over $\mathcal{X}$. The Gram matrix of K with respect to a finite subset $\mathcal{A} = (a_1, a_2, \ldots, a_N)$ of $\mathcal{X}$ is the N-by-N symmetric matrix denoted $K_{\mathcal{A}}$ and defined as:

$$K_{\mathcal{A}} = (K(a_i, a_j))_{i=1\ldots N, j=1\ldots N}. \tag{2.3}$$

By extension, the Gram matrix of K with respect to two finite subsets $\mathcal{A} = (a_1, a_2, \ldots, a_N)$ and $\mathcal{B} = (b_1, b_2, \ldots, b_M)$ of $\mathcal{X}$ is the N-by-M symmetric matrix denoted $K_{\mathcal{A},\mathcal{B}}$ and defined as:

$$K_{\mathcal{A},\mathcal{B}} = (K(a_i, b_j))_{i=1\ldots N, j=1\ldots M}. \tag{2.4}$$

Remark 2.2.4. Equation (2.1) is equivalent to the Gram matrix $K_{\mathcal{A}}$ being positive semi-definite for any finite subset $\mathcal{A} \subset \mathcal{X}$ and equation (2.2) is equivalent to the Gram matrix $K_{\mathcal{A}}$ being positive definite for any finite subset $\mathcal{A} \subset \mathcal{X}$. Attention should be paid at the fact that the notions of positive definiteness and positive semi-definiteness do not coincide for kernels and matrices!

The linear kernel is one of the most simple non-trivial PD kernel.

Example 2.2.5. *The linear kernel*

$K_{\text{lin}}(x, y) = \langle x, y \rangle$ is a PD kernel over $\mathbb{R}^d$. Indeed, $\forall N \in \mathbb{N}$, $(x_1, x_2, \ldots, x_N) \in (\mathbb{R}^d)^N$, $(v_1, v_2, \ldots, v_N) \in \mathbb{R}^N$:

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j \langle x_i, x_j \rangle \\
&= \langle \sum_{i=1}^{N} v_i x_i, \sum_{j=1}^{N} v_j x_j \rangle \text{ by bilinearity of the inner product} \\
&= \| \sum_{i=1}^{N} v_i x_i \|^2 \geq 0
\end{aligned}$$

The linear kernel is not strictly PD. Taking $N = 2$, $x_1 \neq 0$, $x_2 = -x_1$ and $v_1 = v_2 = 1$ provides a simple counter example.

A less obvious PD kernel which is commonly used in machine learning is called the

Gaussian radial basis function (RBF) kernel.

Example 2.2.6. *The Gaussian radial basis function kernel*

The Gaussian RBF kernel (or simply RBF kernel) with parameter $\gamma \geq 0$ defined by:

$$\begin{aligned} K_{\text{rbf}} : (\mathbb{R}^d)^2 &\to \mathbb{R} \\ (x, y) &\mapsto \exp(-\gamma \|x - y\|^2) \end{aligned}$$

is a strictly PD kernel.

Proving the positive-definiteness of the Gaussian RBF kernel is not difficult but involves several steps requiring background notions in mathematical analysis (such as power series expansions) wich are not directly relevant to our prupose. The following is a sketch of the proof.

First, we introduce an auxiliary kernel function:

$$K_1(x, y) = \exp(2\gamma \langle x, y \rangle) \tag{2.5}$$

Its power series expansion is:

$$K_1(x, y) = \sum_{i=1,\ldots,\infty} \frac{(2\gamma \langle x, y \rangle)^i}{i!} \tag{2.6}$$

which is PD as a converging infinite sum of PD kernels (sums and products of PD kernels are PD).

Then we introduce another auxiliary kernel function:

$$K_2(x, y) = f(x) f(y) \tag{2.7}$$

which is trivially PD regardless of the expression of f. Then we pose:

$$f(x) = \exp(-\gamma \|x\|^2) \tag{2.8}$$

The proof is completed by noticing that $K_{\text{rbf}} = K_1 \times K_2$ is PD as a product of PD kernels.

Any PD kernel can be expressed in the following fashion.

Example 2.2.7. *The general case*

Let $(\mathcal{H}, \langle ., . \rangle_{\mathcal{H}})$ be a Hilbert space and $\Phi : \mathcal{X} \to \mathcal{H}$. Then,

$$K : \mathcal{X}^2 \to \mathbb{R}$$
$$(x, y) \mapsto \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}$$

is a PD kernel.

This result can be proved with a straightforward adaptation of the proof in example 2.2.5.

Reciprocically, such a Hilbert space $\mathcal{H}$ and an application $\Phi : \mathcal{X} \to \mathcal{H}$ exist for any PD kernel over $\mathcal{X}$. This major result is known as the Moore-Aronszajn theorem.

Theorem 2.2.8. The Moore-Aronszajn theorem

The two following assertions are equivalent:

1. *K is a PD kernel on $\mathcal{X}$.*
2. *There is a Hilbert space $(\mathcal{H}, \langle ., . \rangle_{\mathcal{H}})$ and a mapping $\Phi : \mathcal{X} \to \mathcal{H}$ such that:*

$$\forall x, y \in \mathcal{X},\ K(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}} \tag{2.9}$$

The reverse implication from assertion 2 to assertion 1 is easy as pointed out in example 2.2.7. Proving the non-trivial direct implication requires to have an insight on the nature of the application Φ and the Hilbert space $\mathcal{H}$. Therefore, the full proof of the theorem will be postponed to section 2.2.3 and the result is admitted for the moment.

2.2.2 Kernel methods and the kernel trick

A widespread utilization of PD kernels with countless practical applications is known as the *kernel trick*.

A large number of algorithms processing finite-dimensional vectors can be expressed in terms of pairwise inner products of the data. This class of algorithms requiring only Gram matrices as inputs is called *kernel methods*.

Besides, we established in theorem 2.2.8 (Moore-Aronszajn) that a PD kernel $K : \mathcal{X}^2 \to \mathbb{R}$ is quivalent to the inner product in a certain Hilbert space $(\mathcal{H}, \langle ., . \rangle_{\mathcal{H}})$. There-

fore, by replacing every inner product evaluation by a kernel evaluation, we can effectively apply the algorithm to the data embedded in space $\mathcal{H}$ instead of the data in the original space $\mathcal{X}$.

The kernel trick consists in the substitution of the Gram matrix of inner products in $\mathcal{X}$ by the kernel gram matrix (as defined in definition 2.2.3), which is in fact the Gram matrix of inner products in $\mathcal{H}$.

One of the most important aspects of the kernel trick is that this transposition of the problem from $\mathcal{X}$ to $\mathcal{H}$ is possible without knowing the application $\Phi : \mathcal{X} \to \mathcal{H}$ or without being able to compute it. The objects in $\mathcal{H}$ are manipulated implicitely through evaluations of the kernel function K.

For instance, the canonical distance (*i.e.* the inner-product distance) in $\mathcal{H}$ between the images $\Phi(\mathcal{X}) = \{\Phi(x)|x \in \mathcal{X}\}$ can be expressed using the kernel function alone.

Theorem 2.2.9. Kernel distance

Let K be a PD kernel over $\mathcal{X}$, $(\mathcal{H}, \langle .,. \rangle_{\mathcal{H}})$ be a Hilbert space, and $\Phi : \mathcal{X} \to \mathcal{H}$ such that $\forall (x_1, x_2) \in \mathcal{X}^2,\ K(x_1, x_2) = \langle \Phi(x_1), \Phi(x_2) \rangle_{\mathcal{H}}$.

Then, for $(x_1, x_2) \in \mathcal{X}^2$:

$$d_K(x_1, x_2) \overset{def}{=} \|\Phi(x_1) - \Phi(x_2)\|_{\mathcal{H}} = \sqrt{K(x_1, x_1) + K(x_2, x_2) - 2K(x_1, x_2)} \tag{2.10}$$

This restriction of the cannonical distance from $\mathcal{H}^2$ to $\Phi(\mathcal{X})^2$ is referred to as the kernel distance.

Proof.

$$\begin{aligned} d_K(x_1, x_2)^2 &\overset{def}{=} \|\Phi(x_1) - \Phi(x_2)\|_{\mathcal{H}}^2 \\ &= \langle \Phi(x_1) - \Phi(x_2), \Phi(x_1) - \Phi(x_2) \rangle_{\mathcal{H}} \text{ by definition} \\ &= \langle \Phi(x_1), \Phi(x_1) \rangle_{\mathcal{H}} + \langle \Phi(x_2), \Phi(x_2) \rangle_{\mathcal{H}} - 2\langle \Phi(x_1), \Phi(x_2) \rangle_{\mathcal{H}} \\ &\quad \text{by symmetry and bilinearity of the inner product} \\ &= K(x_1, x_1) + K(x_2, x_2) - 2K(x_1, x_2) \end{aligned}$$

□

Remark 2.2.10. Since K is a PD kernel, the existances of the Hilbert space $\mathcal{H}$ and the

application Φ are guaranteed by theorem 2.2.8 (Moore-Aronszajn).

In fact, a PD kernel induces a pseudometric on $\mathcal{X}$ by extension of the canonical metric on $\mathcal{H}$.

Theorem 2.2.11. Induced pseudometric space

Let K be a PD kernel over $\mathcal{X}$ and d_K be the corresponding kernel distance. Then, $(\mathcal{X}, d_K)$ is a pseudometric space.

Proof. The 4 properties of the definition of a pseudometric must be verified. d_K is a restriction of the cannonical distance in $\mathcal{H}$, thus non-negativity, symmetry and triangular inequality are given.

Therefore, we only need to verify that for $x \in \mathcal{X}$:

$$d_K(x,x) = \sqrt{K(x,x) + K(x,x) - 2K(x,x)} = \sqrt{0} = 0$$

□

Under what conditions the pseudometric space can be a metric space? This happens *iff* the property called "identity of discernibles" is verified. In other words, we additionally need that for $(x,y) \in \mathcal{X}^2$:

$$d_K(x,y) = 0 \implies x = y$$

we will show that this is equivalent to saying that K is strictly positive.

Theorem 2.2.12. Induced metric space

Let K be a strictly PD kernel over $\mathcal{X}$ and d_K be the corresponding kernel distance. Then, $(\mathcal{X}, d_K)$ is a metric space.

Proof. We propose a proof by contradiction.

Let $(x,y) \in \mathcal{X}^2$ and $d_K(x,y) = 0$. Therefore, by theorem 2.2.9,

$$\sqrt{K(x,x) + K(y,y) - 2K(x,y)} = 0$$

$$i.e.\ K(x,x) + K(y,y) - 2K(x,y) = 0 \tag{2.11}$$

Lets now assume $x \neq y$. Lets pose $N = 2$, $x_1 = x$, $x_2 = y$, $v_1 = -1$ and $v_2 = 1$. Since the x_i are pairwise distinct, the definition 2.2.2 of strictly PD kernels gives:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j K(x_i, x_j) = 0 \implies \forall i \in [\![1, N]\!],\ v_i = 0$$

But given that the right hand side of the implication is false ($v_1 \neq 0$ for instance), we get by contraposition:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j K(x_i, x_j) \neq 0$$

$$i.e.\ K(x, x) + K(y, y) - 2K(x, y) \neq 0$$

which contradicts statement (2.11). □

The intended benefits of performing this kernel trick are usually one of the following:

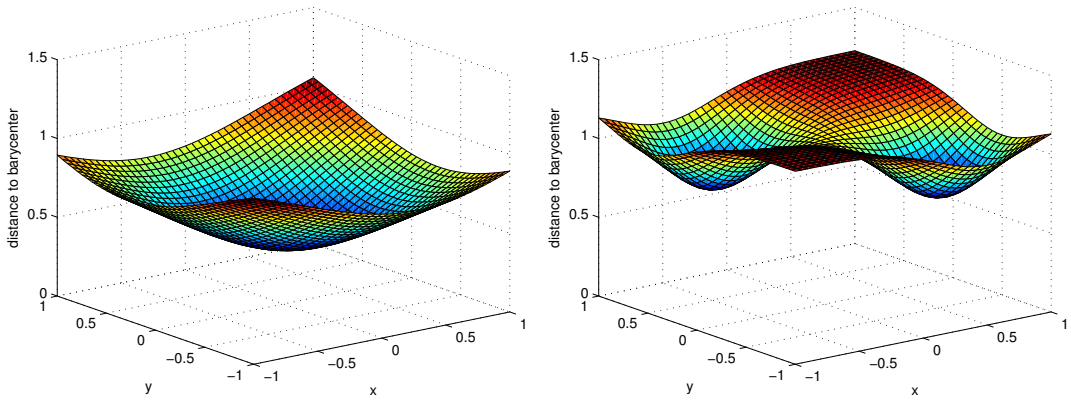
- Embedding the initial data into a *higher-dimensional (potentially infinite dimensional)* feature space involving points without an inverse image in $\mathcal{X}$.
- Obtaining *nonlinear* versions of linear algorithms.
- Applying vectorial algorithms to *non-vectorial* data such as strings or graphs.

Below is an example showing a situation when a point in the kernel space does not have any inverse image.

Example 2.2.13. *Barycenter in kernel space*

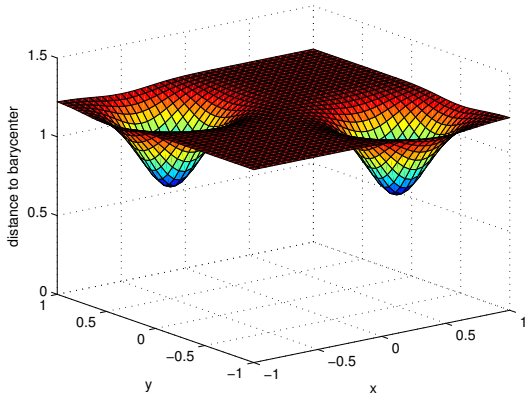
Let $\mathcal{S} = (x_1, x_2, \ldots, x_N) \in \mathcal{X}^N$. The barycenter in $\mathcal{H}$ of $\Phi(\mathcal{S})$ is defined as:

$$\text{bary}(\Phi(\mathcal{S})) = \frac{1}{N}\sum_{i=1}^{N} \Phi(x_i)$$

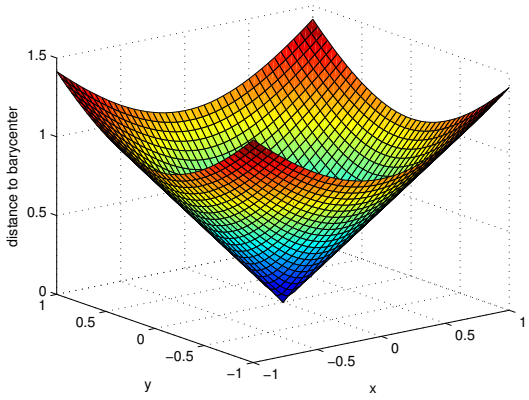


(a) RBF with $\gamma = 0.5$

(b) RBF with $\gamma = 3$



(c) RBF with $\gamma = 10$



(d) Barycenter in $\mathbb{R}^2$

Figure 2.1: For (a), (b) and (c): distance to the barycenter of $\Phi(\mathcal{S})$ with $\mathcal{S} = \{(-0.5, 0.5), (0.5, -0.5)\}$ in the RBF kernel space for different values of the γ parameter. For (d): distance to the barycenter of $\mathcal{S} = \{(-0.5, 0.5), (0.5, -0.5)\}$ in the standard Euclidean space $\mathbb{R}^2$.

The squared distance between the image by Φ of $x \in \mathcal{X}$ and $\text{bary}(\Phi(\mathcal{S}))$ is:

$$\begin{aligned}
d^2 &= \|\Phi(x) - \text{bary}(\Phi(\mathcal{S}))\|_{\mathcal{H}}^2 \\
&= \|\Phi(x) - \frac{1}{N}\sum_{i=1}^{N}\Phi(x_i)\|_{\mathcal{H}}^2 \\
&= \langle \Phi(x) - \frac{1}{N}\sum_{i=1}^{N}\Phi(x_i), \Phi(x) - \frac{1}{N}\sum_{i=1}^{N}\Phi(x_i)\rangle_{\mathcal{H}} \\
&= \langle \Phi(x), \Phi(x)\rangle_{\mathcal{H}} - \frac{2}{n}\sum_{i=1}^{N}\langle \Phi(x), \Phi(x_i)\rangle_{\mathcal{H}} + \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\langle \Phi(x_i), \Phi(x_j)\rangle_{\mathcal{H}} \\
&\quad \text{by bilinearity of } \langle .,. \rangle_{\mathcal{H}} \\
&= K(x,x) - \frac{2}{N}\sum_{i=1}^{N}K(x,x_i) + \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}K(x_i,x_j)
\end{aligned}$$

Figures 2.1a, 2.1b and 2.1c show the distance d to $\text{bary}(\Phi(\mathcal{S}))$ in the BRF kernel space with: $\mathcal{X} = \mathbb{R}^2$, $\mathcal{S} = \{(-0.5, 0.5), (0.5, -0.5)\}$ and $K = K_{\text{rbf}}$ with different values of the γ parameter. The distance d remains strictly positive showing that there is **no inverse image** of $\text{bary}(\Phi(\mathcal{S}))$ in $\mathbb{R}^2$. We can observe that with a small enough value of the parameter γ, there is a single point in $\Phi(\mathbb{R}^2)$ minimizing d (*i.e.* closest to the barycenter) whereas there are multiple minima when γ gets larger.

For reference, figure 2.1d shows the standard Euclidean distance to the barycentre of $\mathcal{S}$ in $\mathbb{R}$. The barycenter has coordinates $\text{bary}(\mathcal{S}) = (0, 0)$.

The next example shows how a simple linear classifier can be made nonlinear using the kernel trick.

Example 2.2.14. *Mean cosine classifier*

Let $\mathcal{S}_1 \in \mathbb{R}^n$ and $\mathcal{S}_2 \in \mathbb{R}^n$ be two finite and disjoint sets of points. Given a point $x \in \mathcal{X}$, the mean cosine between x and the points in $\mathcal{S}_1$ is:

$$\begin{aligned}
d_1(x) &= \frac{1}{|\mathcal{S}_1|}\sum_{y \in \mathcal{S}_1}\cos(x,y) \\
&= \frac{1}{|\mathcal{S}_1|}\sum_{y \in \mathcal{S}_1}\frac{\langle x, y\rangle}{\|x\|_2\|y\|_2}
\end{aligned}$$

In a similar fashion, the mean cosine between x and the points in $\mathcal{S}_2$ is:

$$\begin{aligned} d_2(x) &= \frac{1}{|\mathcal{S}_2|} \sum_{y \in \mathcal{S}_2} \cos(x, y) \\ &= \frac{1}{|\mathcal{S}_2|} \sum_{y \in \mathcal{S}_2} \frac{\langle x, y \rangle}{\|x\|_2 \|y\|_2} \end{aligned}$$

Lets δ be the difference:

$$\delta(x) = d_1(x) - d_2(x)$$

The classifier referred to as the "mean cosine classifier" (only for the purpose of this example) assigns a point x to class 1 if $\delta(x) \geq 0$ or to class 2 otherwise.

Figure 2.2 illustrates the mean cosine classifier for $n = 2$, $\mathcal{S}_1 = \{(-0.4, 0.5)\}$ and $\mathcal{S}_2 = \{(0.5, -0.2), (-0.4, -0.7)\}$. The curve separating the two classes is a straight line. Incidentally, there is a singularity at $(0, 0)$.

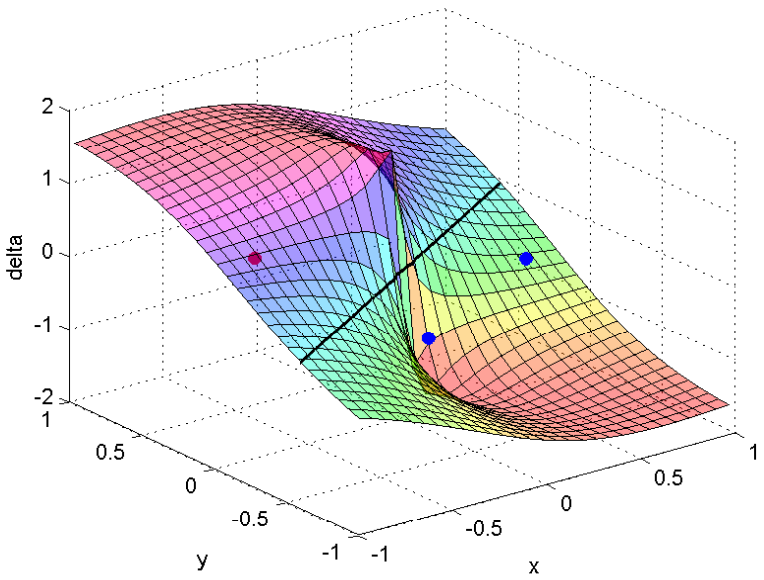


Figure 2.2: Separating curve of the mean cosine classifier.

The mean cosine classifier can be kernelized by simply replacing the inner product by the kernel function K_{rbf} and using theorem 2.2.9. Figure 2.3 illustrates the version of the classifier kernelized using the RBF kernel. Unlike previously, the separating surface is a non-straight curve.

Remark 2.2.15. In the case of the RBF kernel, the mean cosine defined in 2.2.14 is

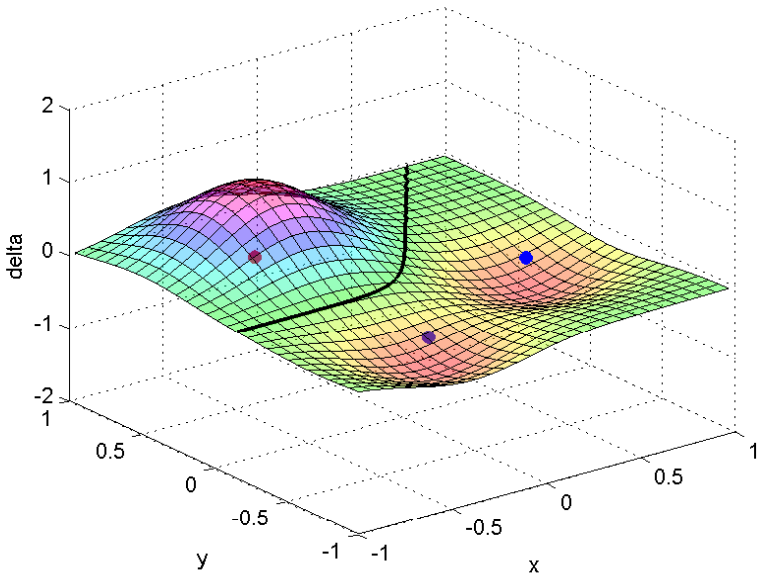


Figure 2.3: Separating curve of the kernelized mean cosine classifier. RBF kernel with $\gamma = 5$.

usually referred to as a *kernel density estimation*. It is a common way to estimate the probability density function of a variable. The resulting mean cosine classifier is therefore a simple Bayesian classifier.

Bhattacharyya's kernel for probability distributions is an example of kernel on non-vectorial data.

Example 2.2.16. *Bhattacharyya's kernel for probability distributions*

Let $\mathcal{P}$ the set of probability distributions over $\mathbb{R}$. Bhattacharyya's kernel, named after Bhattacharyy's affinity between distributions, is defined over $\mathcal{P}$ by:

$$\forall (p, p') \in \mathcal{P}^2,\ K(p, p') = \int_{\mathbb{R}} \sqrt{p}\sqrt{p'}$$

Bhattacharyya's kernel is a PD kernel because it is trivially symmetric and $\forall N \in \mathbb{N}$,

$\forall (p_1, p_2, \ldots, p_N) \in \mathcal{P}^N$, $\forall (v_1, v_2, \ldots, v_N) \in \mathbb{R}^N$:

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j K(p_i, p_j) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j \int_{\mathbb{R}} \sqrt{p_i}\sqrt{p_j} \\
&= \int_{\mathbb{R}} \sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j \sqrt{p_i}\sqrt{p_j} \text{ by linearity of } \int \\
&= \int_{\mathbb{R}} (\sum_{i=1}^{N} v_i \sqrt{p_i})^2 \text{ by linearity of } \sum \\
&\geq 0 \text{ because } (\sum_{i=1}^{N} v_i \sqrt{p_i})^2 \geq 0
\end{aligned}$$

Remark 2.2.17. Example 2.2.16 generalizes well to probability distributions on arbitrary Lebesgue measurable spaces. It can also adapt easily to the discrete case.

2.2.3 Reproducing kernel Hilbert spaces

The proof for theorem 2.2.8 (Moore-Aronszajn) in section 2.2.1 is still due. In this section, additional materials required to complete the proof as-well-as to understand the nature of the embedding $\Phi : \mathcal{X} \to \mathcal{H}$ will be presented.

Definition 2.2.18. *Reproducing kernel*

Let $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ be a vector subspace of real valued functions provided with an inner product $\langle ., . \rangle_{\mathcal{H}}$ (therefore, $(\mathcal{H}, \langle ., . \rangle_{\mathcal{H}})$ has a Hilbert space structure). A function $K : \mathcal{X}^2 \to \mathbb{R}$ is a reproducing kernel of $\mathcal{H}$ if the following two conditions hold:

1. $\forall x \in \mathcal{X}, K_x \in \mathcal{H}$
2. $\forall x \in \mathcal{X}$, $\forall f \in \mathcal{H}$, $f(x) = \langle f, K_x \rangle_{\mathcal{H}}$

where K_x is defined for every $x \in \mathcal{X}$ by:

$$\begin{aligned}
K_x : \mathcal{X} &\to \mathbb{R} \\
t &\mapsto K(x, t)
\end{aligned}$$

Property 2. is referred to as the *reproducing property.*

Definition 2.2.19. *Reproducing kernel Hilbert space (RKHS)*

A Hilbert space of real valued functions $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ is called a RKHS if it admits a reproducing kernel.

There is actually a strong relationship between PD kernels and reproducing kernels.

Theorem 2.2.20. A PD kernel is a reproducing kernel

Let $K : \mathcal{X}^2 \to \mathbb{R}$ be a PD kernel. Let $\mathcal{H}_K$ be the real vector space generated (spanned) by the functions $\{K_x | x \in \mathcal{X}\}$, also written as $span_{\mathbb{R}}\{K_x\}_{x\in\mathcal{X}}$, and let $\langle ., . \rangle_{\mathcal{H}_K}$ be defined on $\mathcal{H}_K \times \mathcal{H}_K$ by:

$$\langle \sum_{i=1}^{N} \alpha_i K_{x_i}, \sum_{j=1}^{M} \beta_j K_{y_j} \rangle_{\mathcal{H}_K} = \sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_i \beta_j K(x_i, y_j) \tag{2.12}$$

Then, $(\mathcal{H}_K, \langle ., . \rangle_{\mathcal{H}_K})$ is a (real) Hibert space and K is a reproducing kernel of the RKHS $\mathcal{H}_K$.

Proof. First, note that $\mathcal{H}_K$ is a vector subspace of $\mathbb{R}^{\mathcal{X}}$ and therefore a real vector space. Moreover, $\langle ., . \rangle_{\mathcal{H}_K}$ is well defined because it does not depend on a particular expansion of the terms. Indeed:

$$\begin{aligned}
\langle \sum_{i=1}^{N} \alpha_i K_{x_i}, \sum_{j=1}^{M} \beta_j K_{y_j} \rangle_{\mathcal{H}_K} &= \sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_i \beta_j K(x_i, y_j) \\
&= \sum_{j=1}^{M} \beta_j \sum_{i=1}^{N} \alpha_i K(x_i, y_j) \\
&= \sum_{j=1}^{M} \beta_j \sum_{i=1}^{N} \alpha_i K_{x_i}(y_j) \text{ by definition of } K_{x_i} \\
&= \sum_{j=1}^{M} \beta_j (\sum_{i=1}^{N} \alpha_i K_{x_i})(y_j)
\end{aligned}$$

which does not depend of a particular expansion of the left hand side term. The proof with expansions of the right hand side term is similar.

Then, we prove that $(\mathcal{H}_K, \langle ., . \rangle_{\mathcal{H}_K})$ is a Hilbert space which requires verifying that $\langle ., . \rangle_{\mathcal{H}_K}$ is symmetric, bilinear and positive-definite which trivially unfold from the symmetry and positive-definiteness of K, and the bilinearity of the sum.

Finally, K is a reproducing kernel of $\mathcal{H}_K$ because:

- $\forall x \in \mathcal{X},\ K_x \in \mathcal{H}_K$ by definition of $\mathcal{H}_K$.

- Moreover, for $f \in \mathcal{H}_K$ with $f = \sum_{i=1}^{N} \alpha_i K_{x_i}$ and $x \in \mathcal{X}$:

$$\begin{aligned}
\langle f, K_x \rangle_{\mathcal{H}_K} &= \langle \sum_{i=1}^{N} \alpha_i K_{x_i}, K_x \rangle_{\mathcal{H}_K} \\
&= \sum_{i=1}^{N} \alpha_i K(x_i, x) \text{ by definition of } \langle ., . \rangle_{\mathcal{H}_K} \\
&= \sum_{i=1}^{N} \alpha_i K_{x_i}(x) \text{ by definition of } K_{x_i} \\
&= (\sum_{i=1}^{N} \alpha_i K_{x_i})(x) \\
&= f(x)
\end{aligned}$$

□

Lets now prove the direct implication in the Moore-Aronszajn theorem enounced in section 2.2.1.

Proof of the Moore-Aronszajn theorem. Let $K : \mathcal{X}^2 \to \mathbb{R}$ be a PD kernel. By theorem 2.2.20, K is the reproducing kernel of a RKHS $\mathcal{H}$. Then for any $x \in \mathcal{X}$ and $y \in \mathcal{X}$:

$$\begin{aligned}
K(x, y) &= \langle K_x, K_y \rangle_{\mathcal{H}} \text{ by the reproducing property of } K \\
&= \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}
\end{aligned}$$

by defining Φ as:

$$\begin{aligned}
\Phi : \mathcal{X} &\to \mathcal{H} \\
x &\mapsto K_x
\end{aligned}$$

The reverse implication was trivially established in example 2.2.7. □

Remark 2.2.21. We proved in theorem 2.2.20 that a PD kernel is a reproducing kernel. In fact, the reciprocal is also true: a reproducing kernel is a PD kernel. In addition, every RKHS has a unique reproducing kernel and every PD kernel is the reproducing kernel of a single RKHS. Therefore, we can speak of "the" reproducing kernel of a RKHS or "the" RKHS of a PD kernel. As a consequence, the relationship between PD kernels

and RKHS is 1-to-1 and explicit, which implies that the nature of the embedding Φ is actually known. The interested reader will be able to find the relevant developments in appendix of this thesis.

In conclusion, a space of functions over $\mathcal{X}$ called the RKHS is a possible realization of the embedding given by the Moore-Aronszajn theorem (into a Hilbert space $\mathcal{H}$ in which the PD kernel is an inner product). This embedding is the result of a mapping by:

$$\Phi : \mathcal{X} \to \mathcal{H} \tag{2.13}$$

$$x \mapsto K_x \tag{2.14}$$

2.2.4 The representer theorem

The representer theorem is a powerful application of the theory of PD kernels. It allows to express the solution of a class of optimization problems with a (finite) linear combination of kernel terms.

Theorem 2.2.22. Representer theorem

Let:

- $\mathcal{X}$ *be a non-empty set*
- $K : \mathcal{X}^2 \to \mathbb{R}$ *be a PD kernel with RKHS* $\mathcal{H}_K$*.*
- $\mathcal{S} = \{x_1, \ldots, x_N\} \subset \mathcal{X}$ *be a finite subset of* $\mathcal{X}$
- $\Psi : \mathbb{R}^{N+1} \to \mathbb{R}$ *be a real function of* $N+1$ *variable stricly increasing with respect to the last variable.*

If $\hat{f}$ *is a solution of the optimization problem i.e. :*

$$\hat{f} = \underset{f \in \mathcal{H}_K}{\operatorname{argmin}}\ \Psi(f(x_1), \ldots, f(x_N), \|f\|_{\mathcal{H}_K}) \tag{2.15}$$

then $\hat{f}$ *admits a solution of the form:*

$$\hat{f} = \sum_{i=1}^{N} \alpha_i K_{x_i} \tag{2.16}$$

Proof. Let $\mathcal{H}_{K,\mathcal{S}} = span_{\mathbb{R}}\{K_{x_i}\}_{x_i \in \mathcal{S}}$ be the at most N-dimenstional subspace of $\mathcal{H}_K$ generated by the K_{x_i}.

Let $\hat{f}$ be a solution to the optimization problem. $\mathcal{H}_K$ begin a Hilbert space:

$$\mathcal{H}_K = \mathcal{H}_{K,\mathcal{S}} \oplus \mathcal{H}_{K,\mathcal{S}}^{\perp}$$

Therefore:

$$\hat{f} = \hat{f}_{\mathcal{S}} + \hat{f}_{\mathcal{S}^{\perp}} \tag{2.17}$$

with $\hat{f}_{\mathcal{S}} \in \mathcal{H}_{K,\mathcal{S}}$ and $\hat{f}_{\mathcal{S}^{\perp}} \in \mathcal{H}_{K,\mathcal{S}}^{\perp}$.

The next step is to prove that $\hat{f}_{\mathcal{S}^{\perp}} = 0$. For any $x_i \in \mathcal{S}$:

$$\begin{aligned}
\hat{f}(x_i) &= \hat{f}_{\mathcal{S}}(x_i) + \hat{f}_{\mathcal{S}^{\perp}}(x_i) \\
&= \hat{f}_{\mathcal{S}}(x_i) + \langle \hat{f}_{\mathcal{S}^{\perp}}, K_{x_i} \rangle_{\mathcal{H}_K} \text{ by the reproducing property of } K \\
&= \hat{f}_{\mathcal{S}}(x_i) + 0 \text{ because } \hat{f}_{\mathcal{S}^{\perp}} \in \mathcal{H}_{K,\mathcal{S}}^{\perp} \text{ and } K_{x_i} \in \mathcal{H}_{K,\mathcal{S}} \\
&= \hat{f}_{\mathcal{S}}(x_i)
\end{aligned}$$

Therefore:

$$\forall x_i \in \mathcal{S},\ \hat{f}(x_i) = \hat{f}_{\mathcal{S}}(x_i). \tag{2.18}$$

Moreover, Pythagora's theorem gives us:

$$\|\hat{f}\|_{\mathcal{H}_K}^2 = \|\hat{f}_{\mathcal{S}}\|_{\mathcal{H}_K}^2 + \|\hat{f}_{\mathcal{S}^{\perp}}\|_{\mathcal{H}_K}^2 \tag{2.19}$$

Which implies:

$$\|\hat{f}\|_{\mathcal{H}_K} \geq \|\hat{f}_{\mathcal{S}}\|_{\mathcal{H}_K} \tag{2.20}$$

As a consequence:

$$\begin{aligned}
&\Psi(\hat{f}(x_1), \ldots, \hat{f}(x_n), \|\hat{f}\|_{\mathcal{H}_K}) \\
&= \Psi(\hat{f}_{\mathcal{S}}(x_1), \ldots, \hat{f}_{\mathcal{S}}(x_n), \|\hat{f}\|_{\mathcal{H}_K}) \text{ using equation (2.18)} \\
&\geq \Psi(\hat{f}_{\mathcal{S}}(x_1), \ldots, \hat{f}_{\mathcal{S}}(x_n), \|\hat{f}_{\mathcal{S}}\|_{\mathcal{H}_K}) \text{ using equation (2.20)}
\end{aligned}$$

Since the monotonicity of Ψ with respect to the last variable is strict, the equality holds iff $\|\hat{f}\|_{\mathcal{H}_K} = \|\hat{f}_{\mathcal{S}}\|_{\mathcal{H}_K}$. This implies $\|\hat{f_{\mathcal{S}}^{\perp}}\|_{\mathcal{H}_K} = 0$ and therefore equation (2.19) yields $\hat{f_{\mathcal{S}^{\perp}}} = 0$.

From this and equation (2.17), we obtain $\hat{f} = \hat{f}_{\mathcal{S}}$ *i.e.* $\hat{f} \in \mathcal{H}_K^{\mathcal{S}}$. □

In practice, a more restrictive form of the representer theorem is often sufficient:

Corollary 2.2.23. Weak representer theorem

Let:

- $\mathcal{X}$ *be a non-empty set*
- $K : \mathcal{X}^2 \to \mathbb{R}$ *be a PD kernel with RKHS* $\mathcal{H}_K$*.*
- $\mathcal{S} = \{x_1, \ldots, x + N\} \subset \mathcal{X}$ *be a finite subset of* $\mathcal{X}$
- $\Lambda : \mathbb{R}^N \to \mathbb{R}$ *be a "loss" function*
- $\lambda > 0$
- $\Omega : \mathbb{R} \to \mathbb{R}$ *be a strictly increasing function*

If $\hat{f}$ *is a solution of the optimization problem:*

$$\hat{f} = \underset{f \in \mathcal{H}_K}{\operatorname{argmin}} \Lambda(f(x_1), \ldots, f(x_N)) + \lambda\Omega(\|f\|_{\mathcal{H}_K}) \tag{2.21}$$

then $\hat{f}$ *admits a solution of the form:*

$$\hat{f} = \sum_{i=1}^{n} \alpha_i K_{x_i} \tag{2.22}$$

Proof. The formula:

$$\Psi(f(x_1), \ldots, f(x_N), \|f\|_{\mathcal{H}_K}) = \Lambda(f(x_1), \ldots, f(x_N)) + \lambda\Omega(\|f\|_{\mathcal{H}_K})$$

defines a function from $\mathbb{R}^{n+1}$ to $\mathbb{R}$, strictly increasing with respect to the last variable. From this point, theorem 2.2.22 can be applied. □

Remark 2.2.24. In statistical machine learning, the two components Λ and Ω play a very distinct and specific role. On one hand, the loss function Λ fits the model f to the training data. On the other hand, the minimization of $\|f\|_{\mathcal{H}_K}$ ensures the smoothness of the solution and thus has a regularization effect. The balance between fitness and regularity is achieved by setting λ to the appropriate value, usually by tuning.

Most importantly, the expression of the solution given by the representer theorem lies in a subspace of finite dimention. This has huge practical consequences as it allows for the implementation of efficent optimization algorithms.

2.2.5 Kernels: Summary

Here is a summary of the essential points developed in this introduction to kernel theory.

1. A PD kernel K is an inner product after the data space $\mathcal{X}$ has been embedded into some Hilbert space $\mathcal{H}$ (Moore-Aronszajn theorem).

2. Therefore, the PD kernel induces the notion of kernel distance, a pseudometric on $\mathcal{X}$ by extension of the canonical Hilbertian metric in $\mathcal{H}$. The pseudometric is a metric if the kernel is strictly PD.

3. The kernel trick is an algorithmic strategy consisting in the substitution of the Gram matrix of inner-products by a kernel Gram matrix. The kernel trick exploits the metric induction in order to:

   - apply algorithms in data spaces of a larger dimension;
   - obtain nonlinear versions of linear algorithms;
   - or to extend vectorial algorithms to non-vectorial data.

4. Performing the kernel trick does not require information about the nature of the space $\mathcal{H}$ or the expression of the mapping $\Phi : \mathcal{X} \to \mathcal{H}$.

5. The RKHS associated to K, a space of functions over $\mathcal{X}$, is a realization of this embedding. An explicit formula of the embedding is given in theorem A.0.6 in appendix.

6. The reproducing theorem allows a certain type of optimization problems in RKHS to be implemented and solved efficiently.

2.3 Constrained optimization theory

Most statistical machine learning algorithms, including the SVM, involve the resolution of a constrained optimization problem. Optimization problems constrained by equalities and inequalities (introduced in Section 2.3.1) can be reformulated using Lagrangians in order to facilitate their resolution (Section 2.3.2). A set of necessary conditions on the solutions known as the Karush-Kuhn-Tucker (KKT) conditions is also often useful (Section 2.3.3).

For this whole section, let $\mathcal{E} = \mathbb{R}^n$; and f, $\{g_i | i \in [\![1, l]\!]\}$ and $\{h_j | j \in [\![1, m]\!]\}$ be real valued functions defined over $\mathcal{E}$.

2.3.1 Problem formulation

Definition 2.3.1. *Constrained optimization problem*

Optimization problems under equality and inequality constraints of the following type:

$$\begin{aligned} \underset{x \in \mathcal{E}}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, l \\ & h_j(x) = 0, \quad j = 1, \ldots, m \end{aligned} \tag{2.23}$$

are called *constrained optimization problems.*

Definition 2.3.2. *Feasible points of a constrained optimization problem*

An element $x \in \mathcal{E}$ is a *feasible point* of the constrained optimization problem (2.23) if it satisfies the following conditions:

1. $\forall i \in [\![1, l]\!],\ g_i(x) \leq 0$
2. $\forall j \in [\![1, m]\!],\ h_j(x) = 0$

An constrained optimization problem which admits at least one feasible point is said to be *feasible.*

A feasible point for which the inequalities of condition 1 are strict is a *strictly feasible point*. A problem which admits at least one strictly feasible point is said to be *strictly feasible*.

Definition 2.3.3. *Solution of a constrained optimization problem*

An element $\hat{x} \in \mathcal{E}$ is a *solution* of the constrained optimization problem (2.23) if it satisfies all the following conditions:

1. $\hat{x}$ is a feasible point of (2.23).
2. $\forall x \in \mathcal{E}, (x \text{ feasible } \implies f(\hat{x}) \leq f(x))$

Definition 2.3.4. *Optimal value of a constrained optimization problem*

The *optimal value* f^* of a **feasible** constrained optimization problem (2.23) is defined as:

$$f^* = \inf_{x \text{ feasible}} f(x) \tag{2.24}$$

Remark 2.3.5. All feasible problems have an optimal value (eventually $-\infty$) but not all feasible problems have solutions.

2.3.2 Weak and strong duality

Definition 2.3.6. *Lagrangian*

The *Lagrangian* of the constrained optimization problem (2.23) is the function:

$$\begin{array}{rccl} L: & \mathcal{E} \times \mathbb{R}^l \times \mathbb{R}^m & \to & \mathbb{R} \\ & (x, \mu, \nu) & \mapsto & f(x) + \sum_{i=1}^{l} \mu_i g_i(x) + \sum_{j=1}^{m} \nu_j h_j(x) \end{array} \tag{2.25}$$

with $\mu = (\mu_i)_{i \in [\![1,l]\!]}$ and $\nu = (\nu_i)_{i \in [\![1,m]\!]}$ known as the *Lagrange multipliers*.

Definition 2.3.7. *Lagrange dual function*

The *Lagrange function* of the Lagrangian (2.25) is the function:

$$\begin{array}{rccl} g: & \mathbb{R}^l \times \mathbb{R}^m & \to & \mathbb{R} \\ & (\mu, \nu) & \mapsto & \inf_{x \in \mathcal{E}} L(x, \mu, \nu) \end{array} \tag{2.26}$$

Definition 2.3.8. *Lagrange dual problem*

For the primal problem (2.23), the *Lagrange dual problem* is the following optimization problem:

$$\begin{aligned} &\underset{\mu,\nu}{\text{maximize}} \quad g(\lambda,\mu) \\ &\text{subject to} \quad \mu_i \geq 0, \quad i \in [\![1,l]\!] \end{aligned} \tag{2.27}$$

where g is the Lagrange dual function.

Subsequently, the original forumlation of an optimization problem as in equation (2.23) is referred to as the *primal problem.*

Remark 2.3.9. Lagrange dual problems are always feasible.

Weak duality is a relationship existing between the optimal values of the primal and dual problems without any additional conditions.

Theorem 2.3.10. Weak duality

Let f^* be the optimal value of the **feasable** *primal problem (2.23) and g^* be the optimal value of the corresponding dual Lagrange problem (2.27).*

Then:

$$g^* \leq f^* \tag{2.28}$$

Proof. Let $x \in \mathcal{E}$ be a feasible point, *i.e.* $\forall i \in [\![1,l]\!],\ g_i(x) \leq 0$ and $\forall j \in [\![1,m]\!],\ h_j(x) = 0$. Then, for $\mu_i \geq 0,\ i \in [\![1,l]\!]$ and $\nu_j,\ j \in [\![1,m]\!]$:

$$\sum_{i=1}^{l} \mu_i g_i(x) + \sum_{j=1}^{l} \nu_i h_j(x) \leq 0$$

which implies:

$$L(x,\mu,\nu) = f(x) + \sum_{i=1}^{l} \mu_i g_i(x) + \sum_{j=1}^{l} \nu_i h_j(x) \leq f(x)$$

Since:

$$g(\mu,\nu) = \inf_{x' \in \mathcal{E}} L(x',\mu,\nu) \leq L(x,\mu,\nu)$$

then:

$$g(\mu, \nu) \leq f(x)$$

which is valid for any feasible point x, $\mu_i \geq 0,\ i \in [\![1, l]\!]$ and $\nu_j,\ j \in [\![1, m]\!]$. Therefore, by taking the supremum and infimum:

$$g^* = \sup_{\mu \in (\mathbb{R}_+)^l, \nu \in \mathbb{R}^m} g(\mu, \nu) \leq \inf_{x \in \mathcal{E}} f(x) = f^*$$

□

Remark 2.3.11. f^* and g^* can eventually be equal to $-\infty$. When they are both finite, $f^* - g^*$ is called the *optimal duality gap.*

Weak duality only gives a lower bound of the primal problem. We say that *strong duality* is achieved when the inequality in theorem 2.3.10 is an equality. Strong duality is achieved if the problem is convex and strictly feasible.

Definition 2.3.12. *Convex optimization problem*

An optimization problem is *convex* if it has the following form:

$$\begin{aligned} \underset{x \in \mathcal{E}}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i \in [\![1, m]\!] \\ & Ax = b \end{aligned} \tag{2.29}$$

with f convex, g_i convex for all $i \in [\![1, l]\!]$, $A \in \mathrm{M}_{m,n}(\mathbb{R})$ and $b \in \mathbb{R}^m$.

Theorem 2.3.13. Strong duality

Let f^* be the optimal value of the **strictly feasible** *and* **convex** *primal problem (2.29) and g^* be the optimal value of the corresponding dual Lagrange problem (2.27).*

Then:

$$g^* = f^* \tag{2.30}$$

Proof. Notice that if the matrix A is not a full rank matrix, then 2 cases are possible:

1. The equation $Ax = b$ does not have any solution, which is exluded since the problem is assumed feasible.

2. The equation $Ax = b$ can be rewritten into an equation $A'x = b'$ admitting the same solutions and with A' being a full rank matrix.

Therefore, without any loss of generality, it is possible to assume that A is a full rank matrix.

First, we define the following set:

$$\begin{aligned}\mathcal{C}_1 =&\{(u_1, \dots, u_l, v_1, \dots, v_m, w) \in \mathbb{R}^{l+m+1} | \exists x \in \mathcal{E} : \ \forall i \in [\![1, m]\!], \ g_i(x) \leq u_i \\ &\wedge Ax - b = v \text{ with } v = (v_j)_{j \in [\![1,m]\!]} \wedge f(x) \leq w\}\end{aligned}$$

which is convex since f and all the g_i are convex functions. We can note that the optimal solution of the primal problem is:

$$f^* = \inf_{(0,\dots,0,0,\dots,0,w) \in \mathcal{C}_1} w$$

Then we define the following set:

$$\mathcal{C}_2 = \{(0, \dots, 0, 0, \dots, 0, w) \in \mathbb{R}^{l+m+1} | w < f^*\}$$

which is obviously convex.

Both sets are convex and $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$ by construction. Therefore, $\mathcal{C}_1$ and $\mathcal{C}_2$ are separated by a hyperplane, *i.e.* there exists $(\mu_1, \dots, \mu_l, \nu_1, \dots, \nu_m, \eta) \in \mathbb{R}^{l+m+1} \notin \{0\}$ and $\zeta \in \mathbb{R}$ such that:

$$\begin{cases} (u_1, \dots, u_l, v_1, \dots, v_m, w) \in \mathcal{C}_1 \implies \displaystyle\sum_{i=1}^{l} \mu_i u_i + \sum_{j=i}^{m} \nu_j v_j + \eta w \geq \zeta \\ (u_1, \dots, u_l, v_1, \dots, v_m, w) \in \mathcal{C}_2 \implies \displaystyle\sum_{i=1}^{l} \mu_i u_i + \sum_{j=i}^{m} \nu_j v_j + \eta w \leq \zeta \end{cases}$$

An element of $\mathcal{C}_1$ remains in $\mathcal{C}_1$ when any u_i, $i \in [\![1, l]\!]$ is increased. Therefore, the first

implication gives:

$$\forall i \in [\![1, l]\!],\ \mu_i \geq 0$$

In a similar fashion, the second implication gives:

$$\eta \geq 0$$

which in turn yields:

$$\forall w < f^*,\ \eta w \leq \zeta$$

thus:

$$\eta f^* \leq \zeta$$

For any $x \in \mathcal{E}$, $(g_1(x), \ldots, g_l(x), h_1(x), \ldots, h_m(x), f(x))$ belongs to $\mathcal{C}_1$ (for the recall, $h_j(x) = \langle A_j, x \rangle - b_j$ where A_j is the j-th line of A and b_j is the j-th element of the vector b). Therefore, the first implication gives:

$$\sum_{i=1}^{l} \mu_i g_i(x) + \langle \nu, Ax - b \rangle + \eta f(x) \geq \zeta$$

with $\nu = (\nu_j)_{j \in [\![1,m]\!]}$, which implies:

$$\sum_{i=1}^{l} \mu_i g_i(x) + \langle \nu, Ax - b \rangle + \eta f(x) \geq \eta f^* \tag{2.31}$$

Now, only two different situations can happen:

Case $\eta > 0$: After division by η equation (2.31) becomes: for all $x \in \mathcal{E}$, $\mu \in \mathbb{R}^l$ and

$\nu \in \mathbb{R}^m$,

$$\begin{aligned} & L(x, \frac{\mu}{\eta}, \frac{\nu}{\eta}) \geq f^* \\ \implies & \inf_{x \in \mathcal{E}} L(x, \mu, \nu) \geq f^* \\ \implies & \sup_{\mu \in [\![1,l]\!], \nu \in [\![1,m]\!]} g(\mu, \nu) \geq f^* \\ i.e. \quad & g^* \geq f^* \end{aligned}$$

By weak duality (theorem 2.3.10), we finally get:

$$f^* = g^*$$

Case $\eta = 0$: We will prove that this case is impossible.

Equation (2.31) becomes: for all $x \in \mathcal{E}$,

$$\sum_{i=1}^{l} \mu_i g_i(x) + \langle \nu, Ax - b \rangle \geq 0 \tag{2.32}$$

thus for $\tilde{x} \in \mathcal{E}$ strictly feasible,

$$\sum_{i=1}^{l} \mu_i g_i(\tilde{x}) \geq 0$$

which implies $\forall i \in [\![1, l]\!],\ \mu_i = 0$ because $\forall i \in [\![1, l]\!],\ g_i(\tilde{x}) < 0$. Moreover, since $(\mu, \nu, \eta) \neq 0$, we get $\nu \neq 0$.

Then (2.32) simplifies into: for all $x \in \mathcal{E}$,

$$\begin{aligned} & \langle \nu, Ax - b \rangle \geq 0 \\ i.e. \ & (\nu^{\mathrm{T}} A)x - \nu^{\mathrm{T}} b \geq 0 \end{aligned}$$

which is possible *iff* $\nu^{\mathrm{T}} A = 0$.

However $\nu \neq 0$ and A is of full rank, therefore the only possibility is that the matrix A is the empty 0-by-0 matix implying $\dim(b) = 0$ and $\dim(\nu) = 0$ which is impossible because we would get $(\mu, \nu, \eta) = 0$, which is excluded.

□

If strong duality can be achieved, a solution of the primal problem minimizes the Lagrangian for any solution of the corresponding dual problem.

Theorem 2.3.14. Primal-dual optimal pairs

Let $\hat{x}$ be a solution of the primal problem. If strong duality holds, then for any solution $(\hat{\mu}, \hat{\nu})$ of the corresponding dual problem:

$$f^* = g^* = L(\hat{x}, \hat{\mu}, \hat{\nu}) \tag{2.33}$$

$(\hat{x}, \hat{\mu}, \hat{\nu})$ is referred to as a primal-dual optimal pair*.*

Proof. Since $\hat{x}$ is a feasible point and $\forall i \in [\![1, l]\!],\ \hat{\mu}_i \leq 0$, on one hand:

$$\begin{aligned} & \sum_{i=1}^{l} \hat{\mu}_i g_i(\hat{x}) + \sum_{j=1}^{l} \hat{\nu}_i h_j(\hat{x}) \leq 0 \\ \implies & f(\hat{x}) + \sum_{i=1}^{l} \hat{\mu}_i g_i(\hat{x}) + \sum_{j=1}^{l} \hat{\nu}_i h_j(\hat{x}) \leq f(\hat{x}) \\ i.e. \quad & L(\hat{x}, \hat{\mu}, \hat{\nu}) \leq f^* \end{aligned} \tag{2.34}$$

On the other hand:

$$\begin{aligned} & \inf_{x \in \mathcal{E}} L(x, \hat{\mu}, \hat{\nu}) \leq L(\hat{x}, \hat{\mu}, \hat{\nu}) \\ i.e.\ & g^* \leq L(\hat{x}, \hat{\mu}, \hat{\nu}) \end{aligned} \tag{2.35}$$

Therefore, equations (2.34) and (2.35) give:

$$g^* \leq L(\hat{x}, \hat{\mu}, \hat{\nu}) \leq f^*$$

and strong duality completes the proof. □

2.3.3 Karush-Kuhn-Tucker conditions

The Karush-Kuhn-Tucker (KKT) conditions are a set of necessary conditions on the primal-dual optimal pairs.

Theorem 2.3.15. Karush-Kuhn-Tucker (KKT) conditions

Let the target function f and the constraint functions $\{g_i\}_{i\in[\![1,l]\!]}$ and $\{h_j\}_{j\in[\![1,m]\!]}$ be differentiable.

If $\hat{x}$ a **local minimum** *for the* **convex** *optimisation problem (2.29), then for any $(\hat{\mu},\hat{\nu})$ solution to the corresponding dual problem, the following conditions hold:*

Stationarity:

$$\vec{\nabla}_x f(\hat{x}) + \sum_{i=1}^{l} \hat{\mu}_i \vec{\nabla}_x g_i(\hat{x}) + \sum_{j=1}^{m} \hat{\nu}_i \vec{\nabla}_x h_i(\hat{x}) = 0 \tag{2.36}$$

Primal feasibility:

$$\forall i \in [\![1,l]\!],\ g_i(\hat{x}) \leq 0 \tag{2.37}$$

Dual feasibility:

$$\forall j \in [\![1,m]\!],\ h_j(\hat{x}) = 0 \tag{2.38}$$

Complementary slackness:

$$\forall i \in [\![1,l]\!],\ \hat{\mu} g_i(\hat{x}) = 0 \tag{2.39}$$

Proof. The problem is convex, therefore the local minimum $\hat{x}$ is a solution of the optimization problem. Moreover, convexity and theorem 2.3.13 entail strong duality from which theorem 2.3.14 entails that $\hat{x}$ minimizes the Lagrangian at $(\hat{\mu},\hat{\nu})$. Therefore:

$$\vec{\nabla}_x (f(\hat{x}) + \sum_{i=1}^{l} \hat{\mu}_i g_i(\hat{x}) + \sum_{j=1}^{m} \hat{\nu}_i h_i(\hat{x})) = 0$$

which gives the stationarity condition by linearity of the gradient.

The primal and dual feasability conditions are trivial consequences of $(\hat{x},\hat{\mu},\hat{\nu})$ begin a primal-dual optimal pair.

Complementary slackness conditions can also be easily established. Let $i \in [\![1,l]\!]$. $\hat{x}$ is a feasible point of the primal problem thus $g_i(\hat{x}) \leq 0$. $(\hat{\mu},\hat{\nu})$ is a feasible point of

the dual problem thus $\mu_i \geq 0$. If $\hat{\mu}_i g_i(\hat{x}) \neq 0$, *i.e.* $g_i(\hat{x}) < 0$ and $\mu_i > 0$, posing $\mu_i = 0$ improves the optimum of the dual problem which contradicts the fact that $(\hat{\mu}, \hat{\nu})$ is a solution of the dual problem. □

In summary, convex optimization problems have good properties entailed by strong duality. The initial primal problem can be transformed into a Lagrange dual problem which has additional variables and less constraints. The new problem can therefore be simpler to solve, and with the same optima (theorem 2.3.14). Additionally, KKT conditions can be used to further simplify the problem and compute the solutions of the primal problem from the solutions of the dual problem.

2.4 Structural risk minimization

In this section, we present the SRM principle, a theoretical strategy for the resolution of supervised learning problems based on the minimization of the statistical risk.

In Section 2.4.1, we first give a brief statistical introduction to supervised learning presented as the minimization of a statistical measure known as the "risk". In Section 2.4.2, we then show that although the risk cannot be directly computed, it can be statistically bounded under some specific conditions leading to the strategy known as the SRM principle .

2.4.1 Supervised learning in a nutshell

2.4.1.1 Definitions

This section introduces the basic terminology and notations relevant to supervised learning.

Let $\mathcal{X}$ be a set referred to as the *input (or feature) space* and $\mathcal{Y} \subset \mathbb{R}$ be a set referred to as the *output (or label) space*.

An *observation* $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is an input-output pair assumed to occur *i.i.d.* (independently and identically distributed) according to a probability distribution $\mathscr{P}$ referred to as the *problem* distribution.

A *labelling model* (or simply *model*) is any function $f : \mathcal{X} \to \mathcal{Y}$ defining how to associate the proper output to a given input.

How well a model f is able to associate a given input $x \in \mathcal{X}$ to an output $y \in \mathcal{Y}$ is defined by a *loss function*:

$$\begin{array}{rccc} \Lambda: & \mathcal{X} \times \mathcal{Y} \times \mathcal{D} & \to & \mathbb{R} \\ & (x, y, f) & \mapsto & \Lambda(x, y, f) \end{array} \tag{2.40}$$

Remark 2.4.1. Often, the problem is referred to as a *classification problem* when the support of $\mathscr{P}$ in $\mathcal{Y}$ is discrete, and as a *scalar regression problem* otherwise. In some other cases, the distinction does not depend on $\mathscr{P}$ but on the type of loss function considered.

The *theoretical risk* (or simply *risk*) is the expected value of the loss function with a given model f according to the probability distribution $\mathscr{P}$:

$$R_{\Lambda,\mathscr{P}}(f) = \mathbb{E}_{(X,Y)\sim\mathscr{P}}\left[\Lambda(X, Y, f))\right] \tag{2.41}$$

Given a finite set of observations $\mathcal{S}_N = (x_i, y_i)_{i\in[\![1,N]\!]} \in (\mathcal{X}, \mathcal{Y})^N$ *i.i.d.* according to $\mathscr{P}$, the *empirical risk* is the mean value realized by the loss function with a given model f on the set $\mathcal{S}_N$:

$$R_{\mathrm{emp}\,\Lambda,\mathcal{S}_N}(f) = \frac{1}{N}\sum_{i=1}^{N} \Lambda(x_i, y_i, f) \tag{2.42}$$

Remark 2.4.2. When there is no risk of confusion, subscripts referring to the loss function Λ, the problem $\mathscr{P}$ or the finite set $\mathcal{S}_N$ can be omitted.

2.4.1.2 Objective: risk minimization

The goal of supervised learning is to find a model f minimizing the risk $R(f)$. Unfortunately, the problem distribution $\mathscr{P}$ is not known in practice. Therefore, it is not possible to minimize the theoretical risk directly.

Instead, only finite sets of observations $\mathcal{S}_N$ are available. The use of finite *training sets* of observations in order to solve the problem is referred to as *supervised learning*. *Empirical risk minimization*, *i.e.* finding a model f minimizing the empirical risk $R_{\mathrm{emp}}(f)$ may therefore seem a natural strategy.

However, the labelling model f minimizing $R_{\mathrm{emp}}(f)$ can be far from a minimum of

$R(f)$, a problem described as *over-fitting* $\mathcal{S}_N$. In general, *empirical risk minimization* produces models with poor performances on instances not yet seen in the training set.

2.4.2 Learning bounds

The SRM principle is based upon a bounding of the theoretical risk under some hypothesis on the loss function Λ and $\mathcal{Y}$, and by restricting the choice of the models to a subset $\mathcal{D} \subset \mathcal{Y}^{\mathcal{X}}$.

2.4.2.1 Lipschitz loss functions

Definition 2.4.3. *Lipschitz ϕ-loss*

Let $\mathcal{Y} = \{-1, +1\}$ and $\mathcal{D} = \mathbb{R}^{\mathcal{X}}$. A *Lipschitz ϕ-loss function* is a loss function $\Lambda : \mathcal{X} \times \mathcal{Y} \times \mathcal{D}$ where:

$$\Lambda(x, y, f) = \phi(yf(x)) \tag{2.43}$$

with $\phi : \mathbb{R} \to \mathbb{R}$ Lipschitz, *i.e.* there is a $L_\phi > 0$ such that:

$$\forall (x_1, x_2) \in \mathcal{X}^2,\ |\phi(x_1) - \phi(x_2)| \leq L_\phi |x_1 - x_2| \tag{2.44}$$

The following are examples of commonly encountered Lipschitz ϕ-loss functions.

Example 2.4.4. *Hinge loss functions*

$$\phi_{\text{hinge}}(t) = \max(0, 1 - t)$$
$$\phi_{\text{s.hinge}}(t) = \max(0, 1 - t)^2$$

The *hinge loss* function ϕ_{hinge} is 1-Lipschitz. Strictly speaking, the *squared hinge loss* function $\phi_{\text{s.hinge}}$ is not Lipschitz, however it is Lipschitz on any bounded subset of $\mathbb{R}$.

Hinge loss functions force the quantity $yf(x)$ to be positive, *i.e.* $f(x)$ to have the same sign as y, and be at least greater that 1. The quantity $yf(x)$ is often referred to as the *margin*. Hinge loss function are used with certain SVMs which are adequately

referred to as *"large margin classifiers"*.

The *average Rademacher complexity* is a measure of richness of a set of functions $\mathcal{F}$ with respect to a probability distribution.

Definition 2.4.5. *Average Rademacher complexity*

Let $(X_i)_{i\in[\![1,n]\!]}$ be n random variables *i.i.d.* according to a probability distribution $\mathscr{P}$.

The *Rademacher complexity* for a set of real valued functions $\mathcal{F} \subset \mathbb{R}^{\mathbb{R}}$ is defined as:

$$\mathrm{Rad}_{\mathscr{P},n}(\mathcal{F}) = \mathbb{E}_{X,\sigma}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(X_i)\right] \tag{2.45}$$

with $(\sigma_i)_{i\in[\![1,n]\!]}$ being uniform *i.i.d.* ± 1-valued random variables (*a.k.a. Rademacher variables*).

When a Lipschitz ϕ-loss function is used, the difference between the theoretical risk and the empirical risk can be probabilistically bounded in terms of the Rademacher complexity.

Theorem 2.4.6. Learning bounds with Lipschitz ϕ-loss function

Let Λ a L_ϕ-Lipschitz ϕ-loss function, $\mathcal{D} \subset \mathbb{R}^{\mathcal{X}}$ a set of models, $f \in \mathcal{D}$ and $\mathcal{S}_n$ a set of n independent observations i.i.d. according to $\mathscr{P}$.

The following inequality holds:

$$\mathbb{E}_S\left[\sup_{f\in\mathcal{F}} R_{\Lambda,\mathscr{P}}(f) - R_{\mathrm{emp}_{\Lambda,\mathcal{S}_n}}(f)\right] \leq 2L_\phi \mathrm{Rad}_{\mathscr{P},n}(\mathcal{D}) \tag{2.46}$$

In addition, if Λ is bounded by ψ_Λ for any observation from $\mathscr{P}$, then with probability at least $1-\delta$ (for any $\delta \in [0,1]$):

$$R_{\Lambda,\mathscr{P}}(f) \leq R_{\mathrm{emp}_{\Lambda,\mathcal{S}_n}}(f) + 2L_\phi \mathrm{Rad}_{\mathscr{P},n}(\mathcal{D}) + \psi_\Lambda\sqrt{\frac{-\log\delta}{2n}} \tag{2.47}$$

By abuse of language, we summarize inequality (2.47) saying that with "high probability":

$$R_{\Lambda,\mathscr{P}}(f) \leq R_{\mathrm{emp}_{\Lambda,\mathcal{S}_n}}(f) + 2L_\phi \mathrm{Rad}_{\mathscr{P},n}(\mathcal{D}) \tag{2.48}$$

2.4.2.2 Structural risk minimization in RKHS

When the set of models $\mathcal{D}$ is a topological ball in a RKHS, $\mathrm{Rad}_{\mathscr{P},n}(\mathcal{D})$ can itself be bounded.

Theorem 2.4.7. Capacity control of RKHS balls

Let $\mathcal{H}$ be a RKHS with reproducing kernel K. The Rademacher complexity of $\mathcal{H}_B = \{f \in \mathcal{H} | \|f\|_{\mathcal{H}} \leq B\}$, i.e. the ball of radius B in H verifies:

$$\mathrm{Rad}_{\mathscr{P},n}(\mathcal{H}_B) \leq B\sqrt{\frac{\mathbb{E}_X\left[K(X,X)\right]}{n}} \tag{2.49}$$

Proof.

$$\begin{aligned}
\mathrm{Rad}_{\mathscr{P},n}(\mathcal{H}_B) &= \mathbb{E}_{X,\sigma}\left[\sup_{f\in\mathcal{H}_B}\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(X_i)\right] \\
&= \mathbb{E}_{X,\sigma}\left[\sup_{f\in\mathcal{H}_B}\frac{1}{n}\sum_{i=1}^{n}\langle f, \sigma_i K_{X_i}\rangle_{\mathcal{H}}\right] && \text{, by the reproducing property} \\
&\leq \mathbb{E}_{X,\sigma}\left[\sup_{f\in\mathcal{H}_B}\frac{1}{n}\sum_{i=1}^{n}\|f\|_{\mathcal{H}}\|\sigma_i K_{X_i}\|_{\mathcal{H}}\right] && \text{, by Cauchy-Schwartz inequality} \\
&\leq \mathbb{E}_{X,\sigma}\left[\frac{1}{n}\sum_{i=1}^{n}B\|\sigma_i K_{X_i}\|_{\mathcal{H}}\right] \\
&= \frac{B}{n}\sqrt{\mathbb{E}_{X,\sigma}\left[\|\sum_{i=1}^{n}\sigma_i K_{X_i}\|_{\mathcal{H}}^2\right]} \\
&= \frac{B}{n}\sqrt{\mathbb{E}_{X,\sigma}\left[\langle\sum_{i=1}^{n}\sigma_i K_{X_i}, \sum_{i=1}^{n}\sigma_i K_{X_i}\rangle_{\mathcal{H}}\right]} \\
&= \frac{B}{n}\sqrt{\mathbb{E}_{X,\sigma}\left[\sum_{i=1}^{n}\sum_{j=1}^{n}\sigma_i\sigma_j\langle K_{X_i}, K_{X_j}\rangle_{\mathcal{H}}\right]} \\
&= \frac{B}{n}\sqrt{\mathbb{E}_{X,\sigma}\left[\sum_{i=1}^{n}\sum_{j=1}^{n}\sigma_i\sigma_j K(X_i, X_j)\right]} && \text{, by the reproducing property} \\
&= \frac{B}{n}\sqrt{\mathbb{E}_{X}\left[\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}_{\sigma}\left[\sigma_i\sigma_j\right] K(X_i, X_j)\right]}
\end{aligned}$$

However, $\mathbb{E}_\sigma\left[\sigma_i\sigma_j\right] = \delta_{i,j}$, therefore:

$$= \frac{B}{n}\sqrt{\mathbb{E}_X\left[\sum_{i=1}^{n} K(X_i, X_i)\right]}$$
$$= B\sqrt{\frac{\mathbb{E}_X\left[K(X,X)\right]}{n}}$$

□

Therefore, the learning bound in theorem 2.4.6 becomes:

Theorem 2.4.8. Learning bounds in RKHS

Let Λ a L_ϕ-Lipschitz ϕ-loss function, $\mathcal{H}_B \subset \mathbb{R}^{\mathcal{X}}$ a RKHS ball with radius B, and $\mathcal{S}_n$ a set of n independent observations i.i.d. according to $\mathscr{P}$. Then, with "high probability":

$$\forall f \in \mathcal{H}_B,\ R_{\Lambda,\mathscr{P}}(f) \leq R_{\mathrm{emp}_{\Lambda,\mathcal{S}_n}}(f) + 2BL_\phi\sqrt{\frac{\mathbb{E}_X\left[K(X,X)\right]}{n}} \tag{2.50}$$

Proof. Corollary of theorem 2.4.6 and theorem 2.4.7. □

Now, assume that $K(X,X) \leq K_m^2$ is bounded. This is for instance the case with the RBF kernel with $K_m = 1$. In general, it is reasponable to assume that the data in bounded.

Then, inequality (2.50) becomes:

$$R_{\Lambda,\mathscr{P}}(f) \leq R_{\mathrm{emp}_{\Lambda,\mathcal{S}_n}}(f) + \frac{2BL_\phi K_m}{\sqrt{n}}$$
$$i.e.\ R_{\Lambda,\mathscr{P}}(f) \leq R_{\mathrm{emp}_{\Lambda,\mathcal{S}_n}}(f) + B\Delta \tag{2.51}$$

with $\Delta = \frac{2L_\phi K_m}{\sqrt{n}}$.

Therefore, rather than minimizing the empirical risk $R_{\mathrm{emp}}(f) = \frac{1}{n}\sum_{i=1}^{n}\phi(y_i f(x_i))$, one should strike the right balance between a minimization of $R_{\mathrm{emp}}(f)$ and a minimization of $B\Delta$ (hence of B) referred to as the *capacity term*, as illustrated on Figure 2.4.

The SRM principle can be formulated from inequality (2.51) as an optimization problem. Given a RKHS $\mathcal{H}$ and a training data set $\mathcal{S}_n = (x_i, y_i)_{i\in[\![1,n]\!]}$:

$$\underset{f\in\mathcal{H}}{\operatorname{argmin}}\ \frac{1}{n}\sum_{i=1}^{n}\phi(y_i f(x_i)) + \Delta\|f\|_{\mathcal{H}}^2 \tag{2.52}$$

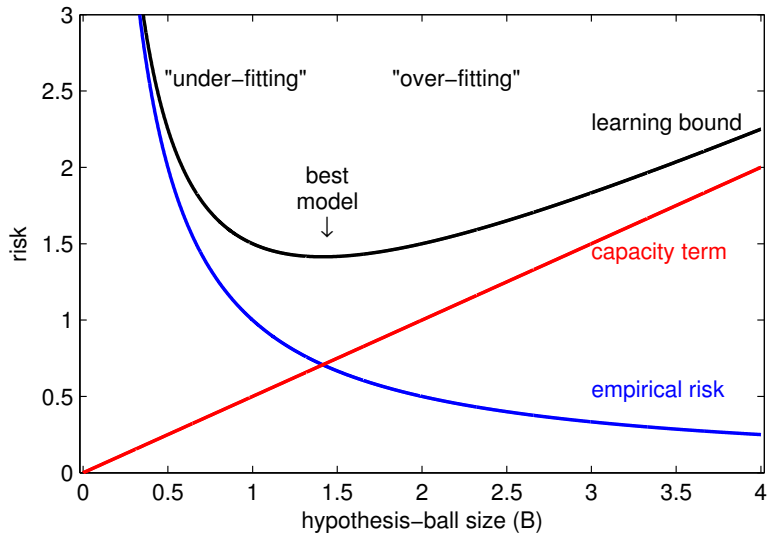


Figure 2.4: The theoretical risk $R(f)$ is bounded by the sum of the monotonically decreasing empirical risk $R_{\text{emp}}(f)$ and the monotonically increasing capacity term $B\Delta$.

2.5 Support vector machines

SVMs are direct applications of the SRM principle. They separate in two different categories distinguished by the type of loss function ϕ employed:

- *Support Vector Classifiers (SVC)* based on the *hinge loss* solve classification problems.
- *Support Vector Regressions (SVR)* based on the *ϵ-insensitive loss* solve regression problems.

Remark 2.5.1. In this thesis, the term SVM is used to designate either a classifier or a regression. The terms SVC and SVR will be used when we want to address them distinctively.

We first present how the SRM principle can be derived into the SVC for classification (Section 2.5.1). Then, the SVR will be presented as an adaptation of the SVC to regression tasks (Section 2.5.2). Section 2.5.3 bridges the gap between this statistical definition of SVMs and their better-known geometrical interpretation. Finally, the main differences between the most commonly used types of SVMs are presented in Section 2.5.4.

2.5.1 Support vector classification

When the hinge loss function $\phi_{\text{hinge}}(t) = \max(0, 1 - t)$ introduced in example 2.4.4 is used, the optimization problem (2.52) yields a classifier known as the SVC. A number of successive transformations are necessary in order to turn problem (2.52) into a computationally solvable and efficient form.

2.5.1.1 Primal form

Since the value of Δ is unknown, problem (2.52) is equivalent to solving for some parameter $\lambda \geq 0$:

$$\underset{f \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \phi_{\text{hinge}}(y_i f(x_i)) + \lambda \|f\|_{\mathcal{H}}^2 \tag{2.53}$$

In practice, the tradeoff parameter λ has to be adjusted using a tuning method such as a grid search.

The unconstrained and convex optimization problem satisfies the hypothesis of the weak representer theorem (theorem 2.2.23). Therefore, the solution to problem (2.53) has the following expression:

$$f(x) = \sum_{j=1}^{n} \alpha_j K_{x_j}(x) = \sum_{j=1}^{n} \alpha_j K(x, x_j) \tag{2.54}$$

By substitution into problem (2.53), we get:

$$\underset{(\alpha_i)_{i=1,\ldots,N} \in \mathbb{R}^N}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \phi_{\text{hinge}} \left(y_i \sum_{j=1}^{n} \alpha_j K(x_i, x_j) \right) + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) \tag{2.55}$$

The next step will be to apply the KKT conditions (theorem 2.3.15) to problem (2.55). Therefore, we require the target function to be differentiable. However, ϕ_{hinge} is not differentiable. This problem can be circumvented by a reformulation of problem (2.55) into an equivalent form introducing new variables $(\xi_i)_{i=1}^N$ known as the

slack variables:

$$\begin{aligned} \underset{(\alpha_i)_{i=1,\dots,N}\in\mathbb{R}^N}{\text{minimize}} \quad & \frac{1}{n}\sum_{i=1}^{n}\xi_i + \lambda\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j K(x_i,x_j) \\ \text{subject to} \quad & \phi_{\text{hinge}}\left(y_i\sum_{j=1}^{n}\alpha_j K(x_i,x_j)\right) \leq \xi_i, \quad i=1,\dots,N \end{aligned} \tag{2.56}$$

Using the definition of ϕ_{hinge}, this is in turn equivalent to:

$$\begin{aligned} \underset{(\alpha_i)_{i=1,\dots,N}\in\mathbb{R}^N}{\text{minimize}} \quad & \frac{1}{n}\sum_{i=1}^{n}\xi_i + \lambda\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j K(x_i,x_j) & \\ \text{subject to} \quad & y_i\sum_{j=1}^{n}\alpha_j K(x_i,x_j) - 1 + \xi_i \geq 0, & i=1,\dots,N \\ & \xi_i \geq 0, & i=1,\dots,N \end{aligned} \tag{2.57}$$

(2.57) is known as the *primal form* of the SVC.

2.5.1.2 Dual form

Problem (2.57) can be solved more efficiently using another equivalent formulation known as the *dual form* obtained by exploiting the primal-dual equivalence and the KKT conditions.

The Lagrangian of the primal form (2.57) is obtained by introducing the Lagrange multipliers $\mu_i \geq 0$ and $\nu_i \geq 0$:

$$\begin{aligned} \tilde{L}_{\text{SVC}}(\alpha,\xi,\mu,\nu) =& \frac{1}{n}\sum_{i=1}^{n}\xi_i + \lambda\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j K(x_i,x_j) \\ & - \sum_{i=1}^{n}\mu_i\left(y_i\sum_{j=1}^{n}\alpha_j K(x_i,x_j) - 1 + \xi_i\right) - \sum_{i=1}^{n}\nu_i\xi_i \end{aligned} \tag{2.58}$$

where $\alpha = (\alpha_i)_{i=1,\dots,N}$, $\xi = (\xi_i)_{i=1,\dots,N}$, $\mu = (\mu_i)_{i=1,\dots,N}$ and $\nu = (\nu_i)_{i=1,\dots,N}$.

(2.57) is a convex problem, therefore the stationarity condition of the KKT conditions (theorem 2.3.15) applies:

$$\vec{\nabla}_{\alpha,\xi}\tilde{L}_{\text{SVC}} = 0 \tag{2.59}$$

Thus, $\frac{\partial\tilde{L}_{\text{SVC}}}{\partial\alpha_i} = 0$ and $\frac{\partial\tilde{L}_{\text{SVC}}}{\partial\xi_i} = 0$ for $i = 1,\dots,N$.

On one hand, for $i = 1, \ldots, N$:

$$\frac{\partial \tilde{L}_{\text{SVC}}}{\partial \xi_i} = \frac{1}{n} - \mu_i - \nu_i \tag{2.60}$$

and thus:

$$\frac{\partial \tilde{L}_{\text{SVC}}}{\partial \xi_i} = 0 \implies \nu_i = \frac{1}{n} - \mu_i \tag{2.61}$$

On the other hand, for $i = 1, \ldots, N$:

$$\frac{\partial \tilde{L}_{\text{SVC}}}{\partial \alpha_i} = 2\lambda \sum_{j=1}^{n} \alpha_j K(x_i, x_j) - \sum_{j=1}^{n} y_j \mu_j K(x_i, x_j) \tag{2.62}$$

and thus:

$$\frac{\partial \tilde{L}_{\text{SVC}}}{\partial \alpha_i} = 0 \implies 2\lambda \sum_{j=1}^{n} \alpha_j K(x_i, x_j) - \sum_{j=1}^{n} y_j \mu_j K(x_i, x_j) = 0 \tag{2.63}$$

Assuming $\lambda \neq 0$, for $j = 1, \ldots, N$ we can pose:

$$\alpha_j = \frac{y_j \mu_j}{2\lambda} + \alpha'_j \tag{2.64}$$

with $\alpha'_j \in \mathbb{R}$. By substitution in (2.63), we obtain:

$$\forall i \in [\![1, N]\!], \ \sum_{j=1}^{n} \alpha'_j K(x_i, x_j) = 0 \tag{2.65}$$

We can remark that choosing any $\alpha' = (\alpha'_j)_{j=1,\ldots,N}$ satisfying condition (2.65) does not change the solution f. Therefore, we can simply pose $\alpha' = 0$ and:

$$\alpha_j = \frac{y_j \mu_j}{2\lambda} \tag{2.66}$$

Substituting (2.61) and (2.66) into the Lagrangian (2.58) yields:

$$\tilde{L}_{\text{SVC}}(\alpha, \xi, \mu, \nu) = \sum_{i=1}^{N} \mu_i - \frac{1}{4\lambda} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \mu_i \mu_j K(x_i, x_j) - \sum_{i=1}^{N} \mu_i \xi_i \tag{2.67}$$

Meanwhile, strong-duality (theorem 2.3.13) entails that the primal problem (2.57)

is equivalent to the dual problem:

$$\begin{aligned} &\underset{\mu\in\mathbb{R}^N,\,\nu\in\mathbb{R}^N}{\text{maximize}} \quad \inf_{\alpha\in\mathbb{R}^N,\,\xi\in\mathbb{R}^N} \tilde{L}_{\text{SVC}}(\alpha,\xi,\mu,\nu) \\ &\text{subject to} \quad \mu_i \geq 0, \qquad\qquad\qquad\qquad i = 1,\ldots,N \end{aligned} \tag{2.68}$$

$\tilde{L}_{\text{SVC}}$ is linear in each of the ξ_i and therefore:

$$\exists i : \mu_i\xi_i \neq 0 \implies \inf_{\alpha\in\mathbb{R}^N,\,\xi\in\mathbb{R}^N} \tilde{L}_{\text{SVC}}(\alpha,\xi,\mu,\nu) = -\infty \tag{2.69}$$

which implies that (2.68) is equivalent to:

$$\begin{aligned} &\underset{\mu\in\mathbb{R}^N}{\text{maximize}} \quad \sum_{i=1}^{N}\mu_i - \frac{1}{4\lambda}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \mu_i \mu_j K(x_i,x_j) \\ &\text{subject to} \quad \mu_i \geq 0, \qquad\qquad\qquad\qquad i = 1,\ldots,N \end{aligned} \tag{2.70}$$

which is known as the *dual form* of the SVC.

Note that the slack variables vanish from the dual formulation of the SVC which largely explains why it is more efficient to save the dual form than the primal form.

2.5.1.3 Decision function

Given a solution $\hat{f}$ to the optimization problem, the binary decision function of the SVC is given by:

$$\text{sgn} \circ \hat{f} \tag{2.71}$$

where sgn is the sign function such as $\text{sgn}(t) = 1$ if $t \geq 0$ and $\text{sgn}(t) = -1$ if $t < 0$.

2.5.1.4 The support vectors

The training points for which $\alpha_i \neq 0$ are known as the *support vectors*. Only the support vectors lead to active contraints ($\xi_i > 0$) in the optimization problem and have an impact on the solution. Therefore, the solution of an SVM is entirely determined by its support vectors.

2.5.2 Support vector regression

The SVR commonly used for regression tasks is obtained when the ϵ-insensitive loss $\phi_\epsilon(y_i, f(x_i))$ is used in place of the hinge loss $\phi_{\text{hinge}}(y_i f(x_i))$. The ϵ-insensitive loss for $\epsilon \geq 0$ is defined as:

$$\phi_\epsilon(t_1, t_2) = \begin{cases} 0 \text{ if } |t_1 - t_2| \leq \epsilon \\ |t_1 - t_2| - \epsilon \text{ otherwise} \end{cases} \tag{2.72}$$

Remark 2.5.2. The ϵ-insensitive loss is not a Lipschitz ϕ-loss function. Therefore, the SVR to is to be understood as an adaptation of the SVC to regression problems rather than a direct application of the SRM principle.

The primal form of the SVR obtained by replacing the hinge loss by the ϵ-insensitive loss in the formulation of the SVC is:

$$\begin{aligned} \underset{(\alpha_i)_{i=1,\ldots,N} \in \mathbb{R}^N}{\text{minimize}} \quad & \frac{1}{n} \sum_{i=1}^{n} \xi_i + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) \\ \text{subject to} \quad & y_i - \sum_{j=1}^{n} \alpha_j K(x_i, x_j) \leq \xi_i + \epsilon, \qquad i = 1, \ldots, N \\ & \sum_{j=1}^{n} \alpha_j K(x_i, x_j) - y_i \leq \xi_i + \epsilon, \qquad i = 1, \ldots, N \\ & \xi_i \geq 0, \qquad i = 1, \ldots, N \end{aligned} \tag{2.73}$$

The introduction of the slack variables result in the addition of two different constraints per training sample into the problem instead of only one with the SVC.

For efficiency reasons, different slack variables ξ_i and ξ_i^* should be used for each of

the constraints:

$$\begin{aligned}
\underset{(\alpha_i)_{i=1,\dots,N}\in\mathbb{R}^N}{\text{minimize}} \quad & \frac{1}{n}\sum_{i=1}^{n}\xi_i+\xi_i^*+\lambda\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j K(x_i,x_j) \\
\text{subject to} \quad & y_i-\sum_{j=1}^{n}\alpha_j K(x_i,x_j)\leq \xi_i+\epsilon, && i=1,\dots,N \\
& \sum_{j=1}^{n}\alpha_j K(x_i,x_j)-y_i\leq \xi_i^*+\epsilon, && i=1,\dots,N \\
& \xi_i\geq 0, && i=1,\dots,N \\
& \xi_i^*\geq 0, && i=1,\dots,N
\end{aligned} \tag{2.74}$$

(2.73) and (2.74) have the exact same solutions in α_i. The dual form can subsequently be obtained in a similar fashion as in Section 2.5.1.

2.5.3 Geometrical interpretation

SVMs are often approached from a geometrical angle as a construction of hyperplanes in the RKHS $\mathcal{H}$ of K. The connection with our statistical approach is easily made using the Moore-Aronszajn theorem (theorem 2.2.8) stating that a PD kernel is the inner product in the RKHS after applying a mapping Φ to the data from $\mathcal{X}$ to $\mathcal{H}$, *i.e.* :

$$K(x_1,x_2)=\langle\Phi(x_1),\Phi(x_2)\rangle_{\mathcal{H}} \tag{2.75}$$

Therefore, the solution (2.54) becomes:

$$\hat{f}(x)=\sum_{i=1}^{n}\alpha_i\langle\Phi(x_i),\Phi(x)\rangle_{\mathcal{H}}=\langle\sum_{i=1}^{n}\alpha_i\Phi(x_i),\Phi(x)\rangle_{\mathcal{H}} \tag{2.76}$$

which corresponds to the equation of the hyperplane orthogonal to $\sum_{i=1}^{n}\alpha_i\Phi(x_i)$ in $\mathcal{H}$.

Note that all hyperplanes defined in this fashion pass through the origin of $\mathcal{H}$. An offset variable $b\in\mathbb{R}$ is often added to the solution to allow for affine hyperplanes. Accordingly, rather than a Hilbert space, the solution is searched in an affine space:

$$\hat{f}=\sum_{i=1}^{n}\alpha_i K_{x_i}+b \tag{2.77}$$

with $\alpha_i\in\mathbb{R}$ and $b\in\mathbb{R}$.

An explanation on how similar problem forumlations can be obtained from geometrical considerations is given in appendix of this thesis. A full tutorial on SVMs from a geometrical standpoint is available in [4].

2.5.4 Popular variants of SVM

In this section, we briefly present the most popular types of SVMs, namely the 1-SVM and LPSVM, and explain the motivations behind the differences in their design.

2.5.4.1 1-SVM

It is the most common type of SVMs. The problem formulations given in Section 2.5.1 (using the hinge-loss function) and in Section 2.5.2 (using the ϵ-insensitive loss functions) are 1-SVMs. Notably, this category comprises the C-SVM and the ν-SVM presenting different control parameters offering slightly different control options. The equivalent of the C-SVM for scalar regression is known as the ϵ-SVR.

C-SVM Instead of the parameter λ used in (2.70), the control parameter is $C = \frac{1}{2N\lambda}$. The resulting formulation of the primal problem is then:

$$\begin{aligned}
\underset{(\beta_i)_{i=1,\dots,N}\in\mathbb{R}^N,\, b\in\mathbb{R}}{\text{minimize}} \quad & C\sum_{i=1}^{N}\xi_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \beta_i \beta_j K(x_i, x_j) & & \\
\text{subject to} \quad & y_i(\sum_{j=1}^{N} y_j \beta_j K(x_i, x_j) + b) - 1 + \xi_i \geq 0, & i = 1, \dots, N & \\
& \xi_i \geq 0, & i = 1, \dots, N & \\
& 0 \leq \beta_i \leq C, & i = 1, \dots, N &
\end{aligned} \tag{2.78}$$

Note that $\beta_i = y_i\alpha_i$. The parameter C is known as the *misclassification cost* parameter. A higher value of C will allow for a closer fit of the training data while a lower value of C will force the decision model to be more regular. Therefore, C is a rather direct way of controlling overfitting.

ν-SVM The ν-SVM is a popular alternative to the C-SVM. It replaces the control parameter C with a new parameter $\nu \in [0, 1]$. Its primal formulation also introducing a

new variable $\rho \geq 0$ is:

$$\begin{aligned}
\underset{(\beta_i)_{i=1,\ldots,N} \in \mathbb{R}^N,\, b \in \mathbb{R}}{\text{minimize}} \quad & \frac{1}{N}\sum_{i=1}^{N}\xi_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \beta_i \beta_j K(x_i, x_j) - \nu\rho \\
\text{subject to} \quad & y_i(\sum_{j=1}^{N} y_j \beta_j K(x_i, x_j) + b) - \rho + \xi_i \geq 0, \qquad i = 1, \ldots, N \\
& \xi_i \geq 0, \qquad i = 1, \ldots, N \\
& \rho \geq 0
\end{aligned} \tag{2.79}$$

Unlike the C parameter which has an implicit effect over the overfitting, the parameter ν has an explicit impact: it is the upper bound on the fraction of "margin errors", *i.e.* points for which $\xi_i > 0$. Full technical details are available in [5].

ϵ-SVR It is simply the C-SVM using the ϵ-insensitive loss instead of the hinge loss. The ϵ-SVR therefore has two control parameters: the misclassification cost parameter C playing the same role as for the C-SVM and the loss parameter ϵ specifying how much the model can deviate from the training samples without penalty.

2.5.4.2 LPSVM

The linear-programming SVM (LPSVM) is obtained by replacing the 2-norm in the target function of C-SVM by a 1-norm. The resulting primal form is:

$$\begin{aligned}
\underset{(\beta_i)_{i=1,\ldots,N} \in \mathbb{R}^N,\, b \in \mathbb{R}}{\text{minimize}} \quad & C\sum_{i=1}^{N}\xi_i + \frac{1}{2}\sum_{i=1}^{N} y_i \beta_i \\
\text{subject to} \quad & y_i(\sum_{j=1}^{N} y_j \beta_j K(x_i, x_j) + b) - 1 + \xi_i \geq 0, \quad i = 1, \ldots, N \\
& \xi_i \geq 0, \qquad i = 1, \ldots, N \\
& 0 \leq \beta_i \leq C, \qquad i = 1, \ldots, N
\end{aligned} \tag{2.80}$$

The resulting linear program can be solved much faster than the quadratic program of the 1-SVM.

Chapter 3

Incorporation of Prior-Knowledge into SVMs: the State-of-the-Art

3.1 Introduction

Supervised machine learning methods such as SVMs are based upon the "learning from example" paradigm: decision models are created from the information implicitly contained into labeled training data. An advantage associated to such an approach is that learning algorithms can be straightforwardly applied on the data without requiring specialized domain-knowledge from the user.

Nevertheless, the user often has a more or less specialized understanding of the domain when dealing with real-life problems. On complex problems, the best results are rarely obtained by blindly applying the learning algorithms, but rather by incorporating as many problem specific aspects as possible into the learning process.

Moreover, cases where labeled data is not available in sufficient amounts for adequate training are common. In those situations, using prior-knowledge to compensate for the missing data appears as the natural solution. Unfortunately, the standard SVMs do not provide a systematic way for incorporating such prior-knowledge and the user usually has to rely upon *ad hoc* methods.

In this review chapter, we propose a state-of-the-art review of systematic methods for the incorporation of prior-knowledge into SVMs. Incorporating formalized prior-knowledge in statistical learning is an increasingly popular way to improve the perfor-

mance of learning algorithms and the topic topic has attracted much interest from the research community in recent years. For reference, we can point out two recent review papers dealing with the incorporation of prior-knowledge into SVM by Lauer and Bloch [38] and Wang [87] underlining the ongoing interest of the research community for this topic.

Starting with a broad overview (Section 3.2), the review empathizes on the type of prior-knowledge (Section 3.3) rather than the incorporation method. A summary of the previous work by type and method is then available in Section 3.4 with a matrix representation in Table 3.1. Finally, a discussion on how well the current state-of-the-art addresses the issue of insufficient training data is provided in Section 3.5.

3.2 Overview of the related work

This review on prior-knowledge incorporation into SVMs has two possible angles of approach:

- a review by type of prior-knowledge (Section 3.2.1);
- a review by incorporation method (Section 3.2.2).

Therefore, the previous work can be summarized into a matrix representation (see Table 3.1 in Section 3.4) according to the type or prior-knowledge and the incorporation method.

3.2.1 Types of prior-knowledge

Generally, prior-knowledge refers to any information on the problem that cannot be inferred from the training data alone. This definition can cover a wide variety of aspects. We propose the following subdivisions to the notion of prior-knowledge:

- the domain-specific prior-knowledge;
- the data-specific prior-knowledge;
- the problem-specific prior-knowledge.

The *domain-specific knowledge* which represent a large fraction of the previous work corresponds to information about the domain of the application rather than specific

aspects of the problem. For instance, the string edit distance is relevant to text-based applications but not particularly to image-based applications. On the other hand, the interpretation of images can be invariant to transformations such as a rotation or a scaling which does not apply to text. This type of information is usually relevant to the domain in general rather than a specific problem.

The *data-specific knowledge* consists in additional information about the available data points. This includes qualitative information about the training data such as class imbalances or the reliability of various sources, and information about the distribution of the unlabeled data.

The *problem-specific (or task-specific) knowledge* corresponds to properties characterizing the problem itself. For instance, a phenomenon can be monotonic *w.r.t.* a parameter such as the "risk of breast cancer" of a person *w.r.t.* the "age" of that person. We may also have explicit information about the range of parameters such as: "a female under 20 years old is not at a significant risk of getting breast cancer". This kind of information is only meaningful in relation with a specific problem.

3.2.2 Prior-knowledge incorporation methods

The prior-knowledge can be incorporated into the SVM at virtually any stage of the learning process. Accordingly, we can distinguish the following types of incorporation methods:

- the sample-based methods;
- the optimization-based methods;
- the kernel-based methods.

The *sample-based methods* consist in modifications of the training samples often by adding artificially generated "virtual" samples. This is the most straightforward method for the incorporation of prior-knowledge in terms of implementation as no modification of the learning algorithm or kernel is required.

The *optimization-based methods* modify the formulation of the constrained optimization problem of a standard SVM. Technically, the resulting classifier can be considered as a new type of SVM in its own right. The prior-knowledge is incorporated as additional

constraints into the optimization problem and sometimes by an reformulating the target function. The optimal solution may change but its search space will usually remains the same. Optimization-based methods may represent substantial design and implementation work. Nevertheless, they present the advantage of incorporating prior-knowledge in the very explicit form of constraints.

The *kernel-based methods* consist in replacing the "generic" kernel with a kernel specifically designed to incorporate the prior-knowledge. Kernel-based methods are direct applications of the kernel trick and do not require a particular modification of the learning algorithm. Therefore, they can be used with any standard SVM of choice and benefit from all the corresponding optimizations already available. However, embedding explicit properties into kernels is not a straightforward task: the new kernel contains the prior-knowledge in an implicit fashion and the validity of the method is difficult to prove theoretically. In addition, the resulting kernel may not have the desirable mathematical properties such as being PD.

Some of the related work presented in this chapter is a mixture of two or more incorporation methods and will subsequently be referred to as "hybrid" methods. On the other hand, each of the works addresses a single type of prior-knowledge presented in Section 3.2.1. Therefore, a presentation according to the type of prior-knowledge is a less ambiguous choice which justifies the approach taken in this review. This formalization effort around the type of prior-knowledge is a main point of divergence compared to the other reviews in [38] and [87].

3.3 Review by type of prior-knowledge

In this section, we present the related works according to the classification proposed in Section 3.2.1.

3.3.1 Methods for domain-specific knowledge

The types of domain-specific knowledge addressed in previous works are: the invariance of the label to specific transformations in $\mathcal{X}$, and notions of distance specific to the particular type of objects.

3.3.1.1 Invariance to transformations

Certain types of objects are not affected by specific transformations of the input data.

Formally, we say a decision function $f : \mathcal{X} \to \mathcal{Y}$ is globally invariant to a set $\{T_\theta : \mathcal{X} \to \mathcal{Y} |\ \theta \in \mathcal{D}\}$ of transformations if:

$$\forall \theta \in \mathcal{D},\ f = f \circ T_\theta \tag{3.1}$$

The nature of the transformations can vary according to the application. For instance, in some computer vision applications, rotating an image may not affect its interpretation. Then, T_θ are rotations parametrized by their angle θ. Similarly, if the label is invariant to rescaling, T_θ will be homothecies parametrised by their scaling factor θ.

Sometimes the invariance to transformation is not global but instead local. Local invariance around θ_0 can be defined as:

$$\left.\frac{\partial f \circ T_\theta}{\partial \theta}\right|_{\theta=\theta_0} = 0 \tag{3.2}$$

For instance, this is the case in some character recognition applications where a slanted "u" as in "*u*" is still read a "u", but rotating it too far will transform it into an "n".

We can distinguish two different approaches to the problem of transformation invariances in the previous works: the incorporation into the training set of "virtual" samples artificially generated from the original training data, and a reduction of the problem to equivalent classes (most often their approximation).

Virtual samples The idea of generating new training samples from the ones pre-existing in the dataset has first been introduced by Poggio and Vetter [58] who used the symmetries present in the objects to generate additional samples.

This type of approach was later justified by Niyogi et al. [51] as a way to perform regularization through the incorporation of prior-knowledge. As presented in [51], the idea behind virtual samples is to incorporate label invariance under a set of transformations. In a nutshell, if the labels are invariant under T, then we generate virtual samples

for all input-output pairs (x_i, y_i) by applying the transformation:

$$(x_i, y_i) \mapsto (Tx_i, y_i) \tag{3.3}$$

The incorporation of virtual samples into SVMs has been proposed by Schölkopf et al. [66] with the virtual SVM framework which tackles a major problem associated with the use of virtual samples. Indeed, the additional virtual samples often result in a greatly inflated training set causing a significant increase in the time and space complexity of learning algorithms.

Meanwhile, the decision model of an SVM is fully determined by the support vectors, which are a subset of the training data. Therefore, instead of generating virtual samples for all the training set, a standard SVM is first used to select the support vectors and virtual samples are generated only for the support vectors. A second SVM is then trained from the support vectors and the aptly named virtual support vectors.

Nevertheless, the solution proposed by Schölkopf et al. is only a mitigation of the problem rather than a definite solution since the amount of support vectors is not bounded and can potentially remain very high.

Instead of inflating the problem with new samples, other methods are based on a reduction of the problem to equivalent classes.

Jittering kernels An improvement to the incorporation of virtual samples in the training set is to perform the transformation inside the kernel product itself. This idea of jittering kernels proposed by Decoste and Burl for the k-nearest-neighbor classifier and subsequently applied to SVMs in [12] consists in computing a number of "jitters" (the equivalent of virtual samples) for each of the training samples and using them in place of the original training samples when computing the kernel.

Given a kernel K and two data samples x_i and x_j, a jittered version $K^J(x_i, x_j)$ of the kernel product is computed in the following fashion:

1. Compute the N_J jitters $J(x_i)$ of the point x_i, including itself.

2. Select the jitter x_q that is the closest to x_j in the RKHS, *i.e.* minimizing the

kernel distance:

$$\begin{aligned} x_q &= \operatorname*{argmin}_{x \in J(x_i)} \|x - x_j\|_{\mathcal{H}} \\ &= \operatorname*{argmin}_{x \in J(x_i)} \sqrt{K(x,x) - 2k(x,x_j) + k(x_j,x_j)} \end{aligned} \tag{3.4}$$

3. Pose $K^J(x_i, x_j) = k(x_q, x_j)$.

Computing the jittered kernel is at least N_J times longer than computing the standard kernel K since N_J jitters are considered for each of the data samples. In return, the problem can be up to N_J times smaller compared to the use of virtual samples which corresponds to a quadratic gain in $O(N_J^2)$ on the size of the kernel matrix.

Jittered kernels (and the virtual sample method) are particularly indicated to use with transformations which produce a small, finite set of images such as symmetries. Attention should be given to the fact that the resulting jittered kernel may not always be PD depending on the type of jitters used.

Tangent distance kernels Unlike jittering kernels which approximate equivalent classes by an arbitrary amount of samples, tangent distance kernels opt for an analytical approach of the problem. Tangent distance kernels introduced for neural networks by Simard et al. [73] and implemented for SVMs by Haasdonk and Keysers [28] specifically deal with local invariances to transformations parametrized by a continuous parameter, for instance rotations parametrized by their angle.

Let $x \in \mathcal{X}$ be a training sample and $\{T_\theta | \theta \in \mathbb{R}\}$ a set of invariant transformations parametrized by $\theta \in \mathbb{R}$. We assume $T_0(x) = x$. The equivalence class of x is a parametric curve:

$$C_x(\theta) = T_\theta(x) \tag{3.5}$$

Assuming it is continuously differentiable at $\theta = 0$, C_x can be approximated in the neighborhood of $\theta = 0$, hence of $C_x(0) = x$, by its first order Taylor's development

around 0:

$$\begin{aligned} C_x(\theta) &= C_x(0) + \theta \frac{\partial C_x}{\partial \theta}(0) + O(\theta^2) \\ &\approx x + \theta \frac{\partial C_x}{\partial \theta}(0) \end{aligned} \tag{3.6}$$

which is the tangent to the curve C_x at the point x.

A tangent distance kernel is then obtained by replacing the distance between two points x_1 and x_2 in the RBF kernel by the distance d_T between the trajectories C_{x_1} and C_{x_2} approximated by their tangents:

$$d_T(x_1, x_2) = \min_{\theta_1, \theta_2} \left(x_1 + \theta_1 \frac{\partial C_{x_1}}{\partial \theta}(0) - x_2 - \theta_2 \frac{\partial C_{x_2}}{\partial \theta}(0) \right) \tag{3.7}$$

Instead of the object-to-object version in (3.7), a sample-to-object version where only one trajectory is considered is also possible. Note that TD kernel are usually not PD kernels which is obvious in the case of the non-symmetric sample-to-object version.

Tangent vector kernels The tangent vector kernels proposed by Pozdnoukhov and Bengio [59] can be viewed as the combination of the jittering kernel method and the tangent distance. Instead of representing the equivalent class with a single tangent vector, multiple tangent vectors are computed from multiple virtual support vectors without explicitly adding them in the training set (as for the jittering kernel).

Haar integration kernels The Haar integration was proposed in [69] for the construction of invariant features and the corresponding Haar-integration kernels were introduced in [29]. The idea is to compute the average kernel output on the set $\mathcal{T}$ of all the admissible invariant transformations. Formally, the Haar integration kernel is defined as:

$$K_{\mathcal{T}}(x_1, x_2) = \int_{\mathcal{T}} \int_{\mathcal{T}} K(T(x_1), T'(x_2)) dT dT' \tag{3.8}$$

If $\Phi : \mathcal{X} \to \mathcal{H}$ is the implicit embedding of the data from $\mathcal{X}$ to the RKHS $\mathcal{H}$ of K:

$$\begin{aligned} K_{\mathcal{T}}(x_1, x_2) &= \int_{\mathcal{T}} \int_{\mathcal{T}} \langle \Phi(T(x_1)), \Phi(T'(x_2)) \rangle_{\mathcal{H}} dT dT' \\ &= \langle \int_{\mathcal{T}} \Phi(T(x_1)) dT, \int_{\mathcal{T}} \Phi(T'(x_2)) dT' \rangle_{\mathcal{H}} \end{aligned} \tag{3.9}$$

Therefore, the Haar integration kernel is analytically equivalent to the inner product between the class averages in the kernel space (which may not have an reciprocal image in $\mathcal{X}$).

Unlike jittering kernels, tangent distance kernels and tangent vector kernels, the Haar integration kernels present the advantage to be positive definite.

The following previous methods use an optimization-based approach to deal with transformation invariances.

Permutation-invariant SVM The permutation-invariant SVM (π-SVM) has been introduces by Shivaswamy and Jebara [72] as a method to incorporate the invariance to the permutations of the components of the input vectors. The method can be considered a hybrid between a sample-based method and an optimization-based method.

The main idea is to find a permutation of the components for each of the inputs that minimizes the radius of the data and maximizes the margin of the SVM. It is an iterative optimization process repeating the two following steps:

1. Apply the SVM on the data and find the decision boundary and the margin.

2. For each input vector, find a permutation of its components using the Kuhn-Munkres alignment algorithm (*a.k.a.* "Hungarian method") bringing it closer to the centroid of the data ball while not decreasing the margin of the SVM.

The iterative process is stopped once a local minimum is reached.

Semi-definite programming machines Semi-Definite Programming Machines (SDPM) proposed by Graepel and Herbrich [25] find optimal hyperplanes between trajectories instead of between samples. In many regards, the SDPM is a close relative of the tangent distance kernel but follows an optimization based approach.

Given a set of invariant transformations $\{T_\theta | \theta \in \mathcal{D}\}$, we consider the trajectory $C_{x_i}(\theta) = T_\theta(x_i)$ for every data points x_i approximated by its k-th order Taylor expansion around $\theta = 0$:

$$\begin{aligned} C_{x_i}(\theta) &\approx \sum_{i=0}^{k} \frac{\theta^k}{k!} \frac{\partial^k C_{x_i}}{\partial \theta^k}(0) \\ &= X_i(\theta) \end{aligned} \tag{3.10}$$

(we assume $T_0(x_i) = x_i$).

The Taylor expansions are incorporated into the optimization problem in place of the data points:

$$\begin{aligned} &\underset{w \in \mathbb{R}^n}{\text{minimize}} \quad \|w\|_2^2 \\ &\text{subject to} \quad y_i \langle w, X_i(\theta) \rangle \leq 0, \quad \theta \in \mathcal{D},\ i = 1, \ldots, N \end{aligned}$$

Note that the semi-definite program above used in [25] is slightly different from an SVM but the idea is easily transposable to an SVM. For reference, semi-definite programming was proposed in [80].

An advantage of the SDPM over the tangent distance kernels its the possibility to use higher order Taylor expansions. The solution proposed by Graepel and Herbrich works for the linear kernel. Their paper suggests that it could work with other kernels provided that the Taylor expansion can be transposed to the kernel space, which is not a trivial problem.

Invariant hyperplanes Schölkopf et al. [67] also proposed a modification of the optimization problem to incorporate local invariances. The decision function:

$$f(x) = \sum_{i=1}^{N} y_i \langle x, x_i \rangle + b \tag{3.11}$$

is modified into:

$$\begin{aligned} g(x) &= \sum_{i=1}^{N} y_i \langle Bx, Bx_i \rangle + b \\ &= \sum_{i=1}^{N} y_i \langle x, B^T B x_i \rangle + b \end{aligned} \tag{3.12}$$

where the real valued N-by-N matrix B contains the information about a first order approximation of the local invariance.

The new decision function can be kernelized for nonlinear classification in the following fashion:

$$g(x) = \sum_{i=1}^{N} y_i K(Bx, Bx_i) + b \tag{3.13}$$

3.3.1.2 Object-specific distance

Kernels for particular types of objects other than real-valued vectors from $\mathbb{R}^n$ are increasingly popular. They entail a notion of distance (which is a valid mathematical metric when the kernel is PD) which takes into account the specificity of the object.

Kernels for objects are very abundant in the literature. Therefore, two representative examples are given rather than an exhaustive list of kernels.

Kernels for (finite) sets of vectors Kondor and Jebara [33] proposed a kernel for finite sets of vectors from $\mathbb{R}^n$. Sets of vectors are sometimes represented and treated as matrices where the columns represent individual vectors but the two objects are in fact quite different: with sets of vectors, the ordering of the objects (columns) is irrelevant and the amount of objects is not necessarily fixed.

Their analytical approach is based on Bhattacharyya's affinity between probability distributions over $\mathcal{X} = \mathbb{R}^n$ (verified to be a PD kernel in Chapter 2):

$$K(p_1, p_2) = \int_{x \in \mathcal{X}} \sqrt{p_1(x) p_2(x)} dx \tag{3.14}$$

The idea is to consider the underlying distribution of the components instead of the actual components.

A kernel principal component analysis [68] with the RBF kernel is first applied on the sets of vectors in order to obtain their best approximation by a multivariate normal distribution. Then, the distributions of the respective sets are used as inputs for Bhattacharyya's kernel.

A kernel for finite sets of vectors was also proposed by Wolf et al. [92] following a

different algebraic approach based on the Gram-Schmidt decomposition of usual kernel matrices and the computation of principal angles between them.

Local alignment kernel Sequences are encountered in many fields of application such as sentences in natural language processing or DNA sequences in genetics.

Let $\mathcal{A}$ be an alphabet of characters and x_1 and x_2 two sequences. For instance $\mathcal{A}$={X,Y,Z} and:

$$\begin{aligned} x_1 &= XYXZZX \\ x_2 &= XXXYYZ \end{aligned} \tag{3.15}$$

Given an alignment π of the sequences, for instance:

$$\begin{aligned} X--YXZZZ \\ XXXY--YZ \end{aligned} \tag{3.16}$$

the alignment score is computed as:

$$s(x_1, x_2, \pi) = S(X, X) + g(2) + S(Y, Y) + g(2) + S(Z, Y) + S(Z, Z) \tag{3.17}$$

where $S \in \mathbb{R}^{\mathcal{A}^2}$ is a substitution matrix and $g : \mathbb{N} \to \mathbb{R}$ a gap penalty function.

The widely-used Smith-Waterman local alignment score is given by:

$$SW(x_1, x_2) = \max_{\pi \in \Pi(x_1, x_2)} s(x_1, x_2, \pi) \tag{3.18}$$

where $\Pi(x_1, x_2)$ is the set of all possible alignments between x_1 and x_2.

The main idea behind the local alignment kernel is to replace the notion of Euclidean distance in the RBF kernel by the Smith-Waterman local alignment score. However, the result is not a positive definite kernel.

In order to solve the problem, Vert et al. [86] suggested the use of an alternative PD formulation of the local alignment kernel:

$$K_{LA}(x_1, x_2) = \sum_{\pi \in \Pi(x_1, x_2)} \exp(\gamma s(x_1, x_2, \pi)) \tag{3.19}$$

and showed that it achieves good performances on real-life biological problems.

3.3.2 Methods for data-specific knowledge

The prior-knowledge specific to the data can be divided into: additional information about the labeled training data, and information about the distribution of the unlabeled data.

3.3.2.1 Quality of the labeled data

Qualitative information about the labeled training data such as class imbalances *w.r.t.* the problem distribution $\mathscr{P}$ can be incorporated with the following methods.

Weighted samples In the standard soft margin C-SVM, a single misclassification cost parameter $C > 0$ is used for all the labeled data samples:

$$\begin{aligned} \underset{w\in\mathbb{R}^n,\, b\in\mathbb{R}}{\text{minimize}} \quad & \|w\|_2^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \quad i = 1,\ldots,N \\ & \xi_i \geq 0, \qquad\qquad\qquad\quad i = 1,\ldots,N \end{aligned}$$

Instead, a particular cost parameter $C_i > 0$ can be set for each individual sample, leading to the following re-formulation of the optimization function:

$$\begin{aligned} \underset{w\in\mathbb{R}^n,\, b\in\mathbb{R}}{\text{minimize}} \quad & \|w\|_2^2 + \sum_{i=1}^{N} C_i\xi_i \\ \text{subject to} \quad & y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \quad i = 1,\ldots,N \\ & \xi_i \geq 0, \qquad\qquad\qquad\quad i = 1,\ldots,N \end{aligned} \tag{3.20}$$

Using this framework, unbalanced training data can be dealt with by setting asymmetric margins, an approach proposed by Veropoulos et al. [85] who used 2 different misclassification cost parameters C_+ and C_- according to the class.

Uneven quality of the training data can be managed by setting a different misclassification cost C_i for each sample according to the degree of confidence on the sample. Wu and Srihari [95] define C_i as a monotonically decreasing function of the confidence (although the problem formulation is slightly different from equation (3.20)). Wang

et al. [88] also used a similar approach to attribute different weights to data obtained from different sources according to their reliability.

The weighted sample framework is actually a hybrid methods which can be viewed either as an optimization-based method (as in this description) or as a kernel-based method. This is because a soft-margin C-SVM is equivalent to a hard-margin SVM with a different kernel (see proposition 6.11 in [8]). More specifically, if $D = diag(d_1, d_2, \ldots, d_N)$ is the diagonal matrix such that $\frac{1}{d_i} = C_i$ where C_i is the misclassification cost corresponding to the i-th sample, the soft-margin problem with kernel matrix K is equivalent to the hard-margin problem with kernel $K + D$.

Knowledge-driven kernel selection Class imbalance issues can be particularly severe in classification tasks involving a specific class of "positive" cases and another unspecific class of "negative" cases. In such a situation, the unspecific class is usually under-represented considering the variety of object it can contain.

This often occurs with problems involving the recognition of a precise object among everything else. The "Car Evaluation Data Set" publicly available from the UCI machine learning repository[1] where images of cars must be distinguished from all other natural images is an example of such a problem.

A solution proposed by Wang et al. [89] consists in choosing a kernel that maximizes the ratio of the scatter of the negative samples over the scatter of the positive samples. This will cause the decision boundary to tightly fit the positive samples while largely avoiding the negative samples.

3.3.2.2 Distribution of the unlabeled data

In many cases, the unlabeled data is already available during training. The specific distribution of the unlabeled data can then be incorporated into the learning process, an approach known as transductive learning.

Transductive SVM On one hand, the classical SVM performs *inductive learning* by constructing a general decision model from the labels of specific training samples. On the other hand *transductive learning* proposed by Vapnik [82] consists in directly

[1] `http://archive.ics.uci.edu/ml/datasets/Car+Evaluation`

transposing the labels of specific training samples to specific unlabeled samples.

Transductive learning directly solves a particular problem whereas inductive learning tries to solve a general problem first before deriving a solution for the particular problem. Therefore, transductive learning which does not require generality is expected to be considerably easier than inductive learning.

The transductive version of the C-SVM extends the standard C-SVM by taking into account the distribution of the unlabeled data $D^* = \{x_i^*\}_{i=1}^{N^*}$. The idea is to train the SVM assuming labels for the data in D^* that maximize the resulting margin:

$$
\begin{aligned}
\underset{w\in\mathbb{R}^n,\, b\in\mathbb{R}}{\text{minimize}} \quad & \|w\|_2^2 + C\sum_{i=1}^{N}\xi_i + C^*\sum_{j=1}^{N^*}\xi_j^* \\
\text{subject to} \quad & y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, && i = 1,\ldots,N \\
& \xi_i \geq 0, && i = 1,\ldots,N \\
& y_j^*(\langle w, x_j^*\rangle + b) \geq 1 - \xi_j^*, && j = 1,\ldots,N^* \\
& y_j^* \in \{-1,+1\}, && j = 1,\ldots,N^* \\
& \xi_j^* \geq 0, && j = 1,\ldots,N^*
\end{aligned}
$$

$C > 0$ and $C^* > 0$ are the misclassification cost parameters for the labeled data and the unlabeled data respectively. In practice, $C^* \leq C$ is recommended in order to penalize less strongly the misclassification of the unlabeled samples which are given hypothetical labels.

3.3.3 Methods for problem-specific knowledge

Properties related to the task itself are usually the most specific and therefore the most useful as prior-knowledge. Among the previous work, labeled regions of $\mathcal{X}$, *i.e.* subsets of $\mathcal{X}$ with an infinite amount of elements, have been extensively considered in a framework known as the Knowledge-based Linear Programming (KBLP) from Mangasarian *et al.* and its various extensions. In this review, we collectively refer to them as the Knowledge-Based SVMs (KBSVMs).

3.3.3.1 Labeled regions

The expression *knowledge-based linear programming* coined by Mangasarian [45] covers a set of methods incorporating constraints in the form of logical implications into the optimization problem. Mangasarian *et al.* use the LPSVM, the linear programming version of the SVM presented in Chapter 1, hence the appellation of the framework. Nevertheless, their method is also applicable to the more usual quadratic programming versions.

The logical implications are obtained from prior-knowledge corresponding to labeled regions. A labeled region $(\mathcal{X}', y') \in \mathfrak{P}(\mathcal{X}) \times \mathcal{Y}$ where $\mathfrak{P}(\mathcal{X})$ are the parts of $\mathcal{X}$ suggests that the labeling function $\hat{f} : \mathcal{X} \to \mathcal{Y}$ should attribute the label y' to points from $\mathcal{X}$:

$$x \in \mathcal{X}' \implies \hat{f}(x) = y' \tag{3.21}$$

which gives the logical implication. They can be seen as an extension of the standard labeled samples.

Remark 3.3.1. At the attention of the reader familiar with the KBLP framework, the conventions and notations in this section are chosen to be consistent with the rest of the manuscript and are largely different from those employed by Mangasarian *et al.*

Knowledge-based SVC

Linear classification Knowledge-based linear programming was first introduced in the context of linear classification by Fung et al. [23, 45] as a modification of the LPSVM. The modification allows the introduction of prior-knowledge in the form of polyhedral labeled sets (referred to as *knowledge sets* in [23]) in the input domain.

The original LPSVM solves the following constrained linear optimization problem with parameter $C > 0$:

$$\begin{aligned} \underset{w \in \mathbb{R}^n,\, b \in \mathbb{R}}{\text{minimize}} \quad & \|w\|_1 + C \sum_{i=1}^{N} \xi_i \\ \text{subject to} \quad & y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N \\ & \xi_i \geq 0, \qquad\qquad\qquad\quad i = 1, \ldots, N \end{aligned} \tag{3.22}$$

A polyhedral knowledge set $\mathcal{P}$ can be defined by a set of $M_{\mathcal{P}}$ linear equalities:

$$\langle e_j, x \rangle \leq \epsilon_j \; j = 1, \ldots, M_{\mathcal{P}} \tag{3.23}$$

This can be summarized by the equivalent matrix notation:

$$Ex \leq e \tag{3.24}$$

with E begin the matrix with lines e_j^T for $j = 1, \ldots, M_{\mathcal{P}}$ and e the vector with coordinates ϵ_j for $j = 1, \ldots, M_{\mathcal{P}}$.

The prior-knowledge consists in defining polyhedral knowledge sets for which $y = 1$ or $y = -1$. Therefore, for each knowledge set defined as in (3.24), the following logical implication must hold (we choose $+1$ or -1 according to the class of the knowledge set):

$$Ex \leq e \implies \pm(\langle w, x \rangle + b) \geq 1 \tag{3.25}$$

However implications such as (3.25) cannot be directly incorporated as linear constraints into the optimization problem (3.22).

Fung et al. [23] proved that the logical implication (3.25) is equivalent to the existence of a solution u for the set of linear constraints (again, the sign is chosen according to the class):

$$\begin{cases} E^T u \pm w = 0 \\ \langle e, u \rangle \pm b + 1 \leq 0 \\ u \geq 0 \end{cases} \tag{3.26}$$

Lets consider the following knowledge sets:

- k sets $\{x | E_i x \leq e_i\}$ belonging to the class with label $+1$
- l sets $\{x | F_i x \leq f_i\}$ belonging to the class with label -1

Problem (3.22) can then be rewritten as the following valid linear program:

$$
\begin{aligned}
\underset{w\in\mathbb{R}^n,\, b\in\mathbb{R}}{\text{minimize}} \quad & \|w\|_1 + C\sum_{i_1=1}^{N}\xi_{i_1} & & \\
\text{subject to} \quad & y_{i_1}(\langle w, x_{i_1}\rangle + b) \geq 1 - \xi_{i_1}, & & i_1 = 1,\ldots,N \\
& \xi_{i_1} \geq 0, & & i_1 = 1,\ldots,N \\
& E_{i_2}^T u_{i_2} + w = 0, & & i_2 = 1,\ldots,k \\
& \langle e_{i_2}, u_{i_2}\rangle + b + 1 \leq 0, & & i_2 = 1,\ldots,k \\
& u_{i_2} \geq 0, & & i_2 = 1,\ldots,k \\
& F_{i_3}^T v_{i_3} - w = 0, & & i_3 = 1,\ldots,l \\
& \langle f_{i_3}, v_{i_3}\rangle - b + 1 \leq 0, & & i_3 = 1,\ldots,l \\
& v_{i_3} \geq 0, & & i_3 = 1,\ldots,l
\end{aligned}
\tag{3.27}
$$

The linear program (3.27) is a hard-margin problem for the knowledge sets and requires that every of them is classified correctly which is not always possible. Slack variables r_i ,ρ_i, s_i and σ_i are added to turn the hard constraints into soft constraints,

in a fashion very similar to the soft-margin SVM:

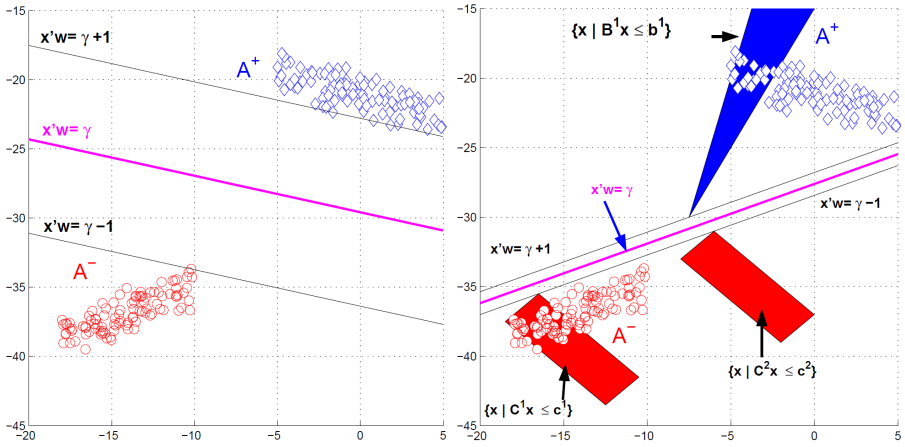
$$
\begin{aligned}
\underset{w\in\mathbb{R}^n,\, b\in\mathbb{R}}{\text{minimize}} \quad & \|w\|_1 + C\sum_{i_1=1}^{N}\xi_{i_1} \\
& +\mu\left(\textstyle\sum_{i_2=1}^{k}(\|r_{i_2}\|_1+\rho_{i_2}) + \sum_{i_3=1}^{l}(\|s_{i_3}\|_1+\sigma_{i_3})\right) \\
\text{subject to} \quad & y_{i_1}(\langle w, x_{i_1}\rangle + b) \geq 1-\xi_{i_1}, && i_1 = 1,\ldots,N \\
& \xi_{i_1} \geq 0, && i_1 = 1,\ldots,N \\
& -r_{i_2} \leq E_{i_2}^T u_{i_2} + w \leq r_{i_2}, && i_2 = 1,\ldots,k \\
& \langle e_{i_2}, u_{i_2}\rangle + b + 1 \leq \rho_{i_2}, && i_2 = 1,\ldots,k \\
& u_{i_2} \geq 0, && i_2 = 1,\ldots,k \\
& r_{i_2} \geq 0, && i_2 = 1,\ldots,k \\
& \rho_{i_2} \geq 0, && i_2 = 1,\ldots,k \\
& -s_{i_3} \leq F_{i_3}^T v_{i_3} - w \leq s_{i_3}, && i_3 = 1,\ldots,l \\
& \langle f_{i_3}, v_{i_3}\rangle - b + 1 \leq \sigma_{i_3}, && i_3 = 1,\ldots,l \\
& v_{i_3} \geq 0, && i_3 = 1,\ldots,l \\
& s_{i_3} \geq 0, && i_3 = 1,\ldots,l \\
& \sigma_{i_3} \geq 0, && i_3 = 1,\ldots,l
\end{aligned}
\tag{3.28}
$$

The parameter $\mu > 0$ is the misclassification cost associated with the knowledge sets.

Setting specific values for μ and C defines a balance between data and prior-knowledge. Choosing $\mu = 0$ results in (3.28) begin a standard LPSVM without knowledge sets. Conversely, choosing $C = 0$ corresponds to training the SVM without training data from the prior-knowledge only. μ and C must be adjusted by a tuning method such as grid search.

Figure 3.1 from [23] shows the impact of the polyhedral knowledge sets on the decision function.

Nonlinear classification Fung et al. [24] subsequently extended their framework to the use with a nonlinear kernel K. The authors use the following generalized support



(a) Linear LPSVM **(b)** Addition of three knowledge sets

Figure 3.1: Influence of knowledge sets on the decision function of the KBSVM (from [23]).

vector machine framework presented in [44]:

$$
\begin{aligned}
&\underset{(\alpha_i)_{i=1}^N \in \mathbb{R}^N}{\text{minimize}} && \sum_{i=1}^{N} \beta_i + C \sum_{i=1}^{N} \xi_i \\
&b \in \mathbb{R} \\
&\text{subject to} && y_i(\sum_{j=1}^{N} \alpha_j y_j K(x_j, x_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N \\
& && -\beta_i \leq \alpha_i \leq \beta_i, \qquad\qquad i = 1, \ldots, N \\
& && \xi_i \geq 0, \qquad\qquad i = 1, \ldots, N
\end{aligned}
\tag{3.29}
$$

Again, a linear program is used instead of the more standard quadratic programming formulation.

The logical implication (3.25) also needs to be "kernelized" correspondingly, which results in the following logical implication:

$$
Ex \leq e \implies \pm(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b) \geq 1 \tag{3.30}
$$

Unfortunately, (3.30) cannot be transformed into an equivalent set of linear constraints such as (3.26) due to non-linearity and non-convexity issues.

In order to bypass the difficulty, the authors propose to use a kernelized version of

the knowledge set. Instead of the polyhedral sets:

$$\{x|Ex \leq e\} \tag{3.31}$$

kernelized polyhedral sets are defined as:

$$\{z|K_{E,X}z \leq e\} \tag{3.32}$$

where X is the N-lines by n-columns matrix representing the training data set (lines correspond to instances and columns to features) and $K_{E,X} = (K(e_i, x_j))_{i=1,\ldots,M_{\mathcal{P}}}^{j=1,\ldots,N}$ is the kernel matrix between data set E and data set X.

The kernelization of the knowledge sets can be justified in the following fashion. Under the general assumption that the columns of X are linearly independent, the linear version of the logical implication (3.25) is equivalent to:

$$\begin{cases} Ex \leq e \\ x = X^T z \end{cases} \implies \begin{cases} \pm(w^T x + b) \geq 1 \\ w = X^T(y \otimes \alpha) \\ x = X^T z \end{cases} \tag{3.33}$$

where $\otimes$ designates the element-wise multiplication of vectors (resulting in a vector of the same dimension) and y (resp. α) is the vector with components $y_i,\ i = 1, \ldots, N$ (resp. $\alpha_i,\ i = 1, \ldots, N$). By substitution, this is equivalent to:

$$EX^T z \leq e \implies \pm((y \otimes \alpha)^T X X^T z + b) \geq 1 \tag{3.34}$$

Therefore, the kernelization of this implication yields:

$$K_{E,X}z \leq e \implies \pm((y \otimes \alpha)^T K_{X,X} z + b) \geq 1 \tag{3.35}$$

where we recognize the kernelized knowledge set (3.32). Subsequently, Fung et al. [24] proved that the kernelized logical implication (3.35) is equivalent to the existence of a

solution u satisfying the following set of linear constraints:

$$\begin{cases} K_{X,E}u \pm K_{X,X}(y \otimes \alpha) = 0 \\ \langle e, u \rangle \pm b + 1 \leq 0 \\ u \geq 0 \end{cases} \tag{3.36}$$

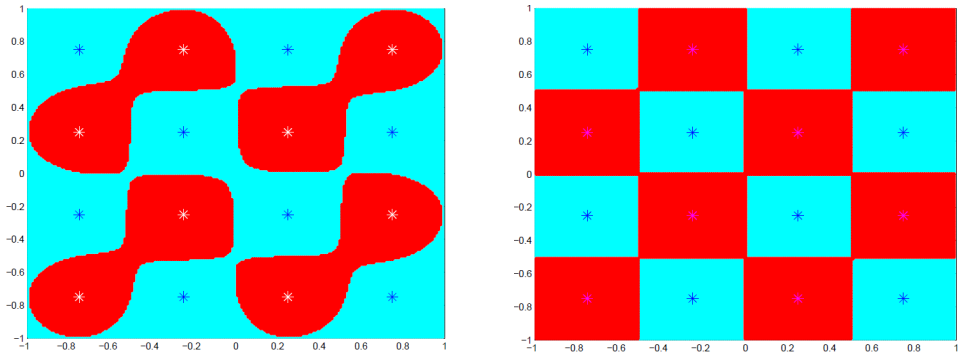
By preserving the notations introduced in the linear case for the knowledge sets and by introducing slack variables in a similar fashion as in (3.28), we finally obtain the following linear program formulation with parameters $C > 0$ and $\mu > 0$:

$$\begin{aligned} \underset{\substack{(\alpha_i)_{i=1}^N \in \mathbb{R}^N \\ b \in \mathbb{R}}}{\text{minimize}} \quad & \sum_{i=1}^{N} \beta_i + C \sum_{i_1=1}^{N} \xi_{i_1} \\ & + \mu \left(\textstyle\sum_{i_2=1}^{k} (\|r_{i_2}\|_1 + \rho_{i_2}) + \sum_{i_3=1}^{l} (\|s_{i_3}\|_1 + \sigma_{i_3}) \right) \\ \text{subject to} \quad & y_{i_1}(\sum_{j=1}^{N} \alpha_j y_j K(x_j, x_{i_1}) + b) \geq 1 - \xi_{i_1}, && i_1 = 1, \ldots, N \\ & -\beta_{i_1} \leq \alpha_{i_1} \leq \beta_{i_1}, && i_1 = 1, \ldots, N \\ & \xi_{i_1} \geq 0, && i_1 = 1, \ldots, N \\ & -r_{i_2} \leq K_{X,E_{i_2}} u_{i_2} + K_{X,X}(y \otimes \alpha) \leq r_{i_2}, && i_2 = 1, \ldots, k \\ & \langle e_{i_2}, u_{i_2} \rangle + b + 1 \leq \rho_{i_2}, && i_2 = 1, \ldots, k \\ & u_{i_2} \geq 0, && i_2 = 1, \ldots, k \\ & r_{i_2} \geq 0, && i_2 = 1, \ldots, k \\ & \rho_{i_2} \geq 0, && i_2 = 1, \ldots, k \\ & -s_{i_3} \leq K_{X,F_{i_3}} v_{i_3} - K_{X,X}(y \otimes \alpha) \leq s_{i_3}, && i_3 = 1, \ldots, l \\ & \langle f_{i_3}, v_{i_3} \rangle - b + 1 \leq \sigma_{i_3}, && i_3 = 1, \ldots, l \\ & v_{i_3} \geq 0, && i_3 = 1, \ldots, l \\ & s_{i_3} \geq 0, && i_3 = 1, \ldots, l \\ & \sigma_{i_3} \geq 0, && i_3 = 1, \ldots, l \end{aligned} \tag{3.37}$$

Unfortunately, this nonlinear knowledge-based linear programming framework suffers from a series of drawbacks due to the kernelization (3.32) of the prior-knowledge which

depends on the data X. This is undesirable because it is no longer possible to think about the prior-knowledge independently from the data.

Moreover, this kernelization process is non-intuitive and non-transparent. This results in the prior-knowledge having a largely unpredictable effect on the decision function. The illustration on the check-board data set in figure 3.2 shows that the prior-knowledge seems to spread to all the data, regardless of where the knowledge sets were actually located in $\mathcal{X}$.



(a) Without prior-knowledge **(b)** With prior-knowledge

Figure 3.2: Results on the check-board dataset from [24]. Only two knowledge sets corresponding to the two leftmost squares of the lowest line are defined. The prior-knowledge has an effect on all the squares of the check-board regardless of which ones actually contains prior-knowledge.

Mangasarian and Wild [47] later proposed an extension of this nonlinear KBLP to a different form of nonlinear prior-knowledge in which the polyhedral constraint on the knowledge sets is relaxed.

Knowledge-based SVR The KBLP framework for classification can also be used for regression. Early work using the initial model of kernelized knowledge is available in [49] and later work with the modified knowledge model in [46]. In addition, a fusion of the latest SVM and SVR frameworks can be found in [48].

The adaptation from classification problems to regression problems requires little modification. The the loss function needs to be adapted but the way in which the prior-knowledge is incorporated remains identical. Therefore, any kind of SVMs including linear and quadratic versions of SVCs and SVRs can be used instead of the linear programs initially proposed. Mangasarian et al. [50] propose themselves an adaptation of their framework to another type of SVM known as "proximal SVM".

Extensions and variations The following are previous work on the incorporation of labeled sets into SVMs proposing an alternative to the KBLP framework or extending it.

Simpler KBSVM Le and Smola [40] proposed a much simpler alternative to Mangasarian's knowledge-based linear programming framework. Instead of incorporating the prior-knowledge as additional constraints, Le and Smola opted to directly modify the decision function f by composing it with a function $\phi : \mathcal{Y} \to \mathcal{Y}$ containing the prior-knowledge.

For instance, in the case of binary classification:

$$\phi(y) = \begin{cases} \max(1, y) \text{ if } y \text{ belongs to a labeled region for the class } +1 \\ \min(-1, y) \text{ if } y \text{ belongs to a labeled region for the class } +1 \\ x \text{ otherwise} \end{cases} \tag{3.38}$$

Figure 3.3 shows that the new decision function $\phi \circ f$ itself integrates the prior-knowledge rather than its choice being constrained by the prior-knowledge as in Mangasarian's framework.

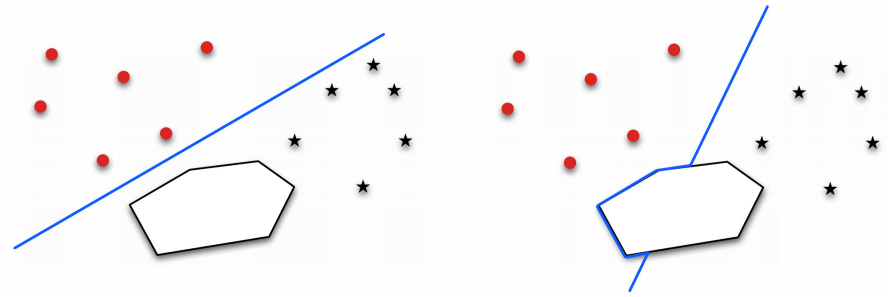


Figure 3.3: Left: Mangasarian's knowledge-based SVM, right: simplified knowledge-based SVM (from [40]).

This radically simple method circumvents all the difficulties encountered by Mangasarian et al. regarding the incorporation of prior-knowledge such as the kernelization of prior-knowledge, the opaqueness of the prior-knowledge once kernelized or the addition of numerous new parameters and variables to the problem.

However, these advantages do not come for free. Rather than really solving the prior-knowledge incorporation issue, this method transforms it into an optimization issue. Indeed, the minimization of the new regularized empirical risk corresponding to

$\phi \circ f$ (which is the underlying principle of the SVM as fully detailed in Chapter 1) does not guarantee a solvable convex problem. As a workaround, Le and Smola proposed an approximate resolution without any guarantees on the quality of the solution.

The authors claim that additional forms of prior-knowledge other than labeled sets such as monotonicity or parity can be incorporated with their method. Although functions ϕ modeling such properties exist, the problem of solving the resulting optimization problem remains entire and arguably without a simple solution.

Therefore, this method is more an interesting modeling idea than a fully workable alternative.

Extensional KBSVM Maclin et al. [42] proposed a simplification of the KBLP framework. In the extensional KBSVM, the knowledge sets are considered as an extension of the labeled data samples of the same class. It simplifies the fairly complex way imperfect advice is dealt with slack variables and additional training parameters in the original framework from Mangasarian et al.

When knowledge sets are in contradiction with the labeled data, instead of slacking the knowledge sets themselves, the knowledge sets are left unchanged and the constraints themselves are slacked.

Maclin et al. [43] also proposed a method for automatic refinement of the labeled regions.

Online KBSVM Kunapuli et al. [36] proposed an online learning version of knowledge-based support vector machines. A passive-aggressive framework is used to update the SVM with prior-knowledge when new samples are added.

Knowledge Initialisation Concurrently to the development of KBLP, Diederich and Barakat [15] proposed an alternative sample-based approach to the problem. It can be viewed as attempting to achieve the same objectives as the KBLP framework using the virtual sample method.

After a preliminary refinement phase using neural-networks, the logical implications are used to generate virtual samples which are added to the training set for the SVM.

Although not referred to as "knowledge initialization", [98] also proposed a related method for the incorporation of fuzzy IF-THEN rules via the generation of additional

virtual samples.

Despite being straightforward, those methods suffer from the severe drawbacks of the virtual sample method and can arguably considered as a less good approach than the optimization-based approach taken in other related works.

3.4 Matrix summary of the previous work

Table 3.1 summarizes the related work presented in Section 3.3 in a matrix representation according to the type of prior-knowledge and the incorporation method. It appears that almost any combination of type and method was tried.

The earliest works dating back from the 90s are sample-based methods dealing with transformation invariances (a type of domain-specific knowledge) through the generation of artificial, virtual samples. They implement a straightforward idea which proved effective but can significantly increase the size of the problem which is a crippling drawback.

Mostly for this reason, the sample-based methods were later replaced by kernel-based methods (jittering kernels, Haar integration kernels) and optimization-based methods (π-SVM) exploiting the same idea without explicitely adding virtual samples. Other kernel-based (tangent distance kernels, tangent vector kernels) and optimization-based (semi-definite programming, invariant hyperplanes) methods consider analytical approximations of the equivalent classes rather than virtual samples.

A number of kernel-baseds method were also developped not to deal with invariances but for specific datatypes (sets, sequences, etc...). They allow an extension of the SVMs from points in $\mathbb{R}^n$ to the objects they are designed for.

Data-specific prior-knowledge was mainly addressed with optimization-based methods (weighted samples and transductive SVMs). We noted that the weighted samples methods which were first developed as a reformulation of the optimization problem (adjustment of the misclassification costs) is equivalent to a kernel-based approach.

A family of optimization-based methods referred to as the KBSVM framework and its variations represent the main research effort on problem specific prior-knowledge and deal with the incorporation of labelled sets into the problem. A few sample-based approaches (knowledge initialization) pursuing the same objectives as the KBSVMs

were also proposed. Their much simpler design is an advantage but they suffer from the same drawbacks as the earlier sample-based approaches. Moreover, labelled sets usually contain an infinite amount of points and are difficult to discretize into virtual samples.

Table 3.1 shows that 2 combinations were not addressed by previous works:

- sample-based methods for data-specific knowledge;
- kernel-based methods for problem-specific knowledge.

The absence of work dealing with the first combination can be explained by the fact that knowledge on the data instances themselves does not naturally translate into additional data instances.

In contrast, kernel-based approaches to the incorporation of properties specific to the problem may have many latent qualities as developed in the following section.

3.5 Prior-knowledge and missing data: discussion and future work

The previous work on the incorporation of prior-knowledge presented in this chapter shows that various forms of knowledge can be incorporated into SVMs with various methods in order to successfully improve the learning results.

Nevertheless, an excessive focus on the improvement of results alone may steer us away from a more essential question which is usually sidestepped: *"does the method provide an adequate answer to precise needs of the user?"*

In practice, situations in which data is scare but some form of prior-knowledge about the problem is available are common place. In this context, it is clear that *the prior-knowledge is an alternative to the missing data* rather than a mean to improve upon already satifactory results.

Therefore, it is insufficient for the different methods to simply improve upon learning results on average. Instead, they should be able to substitute prior-knowledge to missing training data.

In this section, we provide a synthetic discussion on the related work in relation with this objective and identify the most important challenges for future works and the most

Table 3.1: Matrix view of the state-of-the-art on the incorporation of prior-knowledge into SVMs. Columns correspond to types of prior-knowledge and rows to incorporation methods. The hybrid methods appear in more than one row.

	Domain-specific	Data-specific	Problem-specific
Sample-based	• Virtual samples [51, 58, 66] • π-SVM [72]		• Knowledge initialization [15, 98]
Kernel-based	• Jittering kernels [11, 12] • Tangent distance kernels [28, 59, 73] • Tangent vector kernels [59] • Haar integration kernels [29, 69] • Kernels for finite sets [33, 92] • Local alignment kernel [86]	• Weighted samples [85, 88, 95] • Knowledge-driven kernel selection [89]	
Optimization-based	• π-SVM [72] • Semi-definite programming machines [25] • Invariant hyperplanes [67]	• Weighted samples [85, 88, 95] • Transductive SVM [82]	• KBSVM [23, 24, 45–50] • Extensional KBSVM [42, 43] • Simpler KBSVM [40] • Online KBSVM [36]

promising leads to address them.

3.5.1 Prior-knowledge as a substitute for data

Each of the 3 types of prior-knowledge presented in Section 3.2.1 including knowledge on the domain, the data and the problem has been addressed by some previous work as shown in the matrix representation in Table 3.1.

A majority of it relates to *domain-specific prior-knowledge* and in particular to invariances to transformations. Although contributing to improve learning results by playing an important regularisation role [51], this type of prior-knowledge provides the least amount of specific information on the problem itself.

In particular, domain-specific prior-knowledge is not expected to act as a substitute for missing data. Indeed, the methods work either by generating new "virtual" samples from the existing ones or by deriving equivalent classes (or an approximation) from them. Therefore, these methods cannot perform well without the preexistence of "good" samples in the data.

The works dealing with *data-specific knowledge* either correct class imbalances [85, 88, 89, 95] or exploit the distribution of the unlabelled data [82] and do not address the problem of missing data.

The only type of prior-knowledge adequately fulfilling this role is the *problem-specific prior-knowledge*. The previous works structured around the KBSVM framework [23, 24, 45–50] focuses on the incorporation of knowledge as labeled regions. These "knowledge sets" placed on regions containing few data can induce radical changes in the decision function that are not dictated by the data.

However, the prior-knowledge about the problem can take many other forms that just labeled regions. For instance, we may also think of be global properties of the model such as monotonicity, periodicity or correlation patterns of the output *w.r.t.* features. Those other types of prior-knowledge are yet to be addressed in a convincing way.

3.5.2 Soundness and potential of kernel methods

The present review also shows the advantages and drawbacks of the different incorporation methods namely the sample-based, optimization-based and kernel-based methods.

The *sample-based methods* involving the generation of "virtual" samples are the most straightforward to implement as no modification is required on the algorithm or the kernel. However, they suffer from clear drawbacks such as a potentially dramatic increase in the size of the problem (a problem only mitigeated by the restriction to virtual support vectors [66]) or problems posed by an arbitrary discretization of continuous properties. In practice, the sample-based methods mostly used for transformation invariances [51, 58, 66] have progressively been phased out in favor of kernel-based methods [11, 12, 28, 59, 59, 73, 86] and optimization-based methods [25, 67] fulfilling the same roles.

The *optimization-based methods* offer an explicit way to incorporate prior-knowledge through additional constraints. However, they suffer from the high-complexity of their design making them difficult to implement and use in practice as evidenced by the various attempts to simplify the KBSVM framework [40, 42, 43] often at the cost of decreased performances or new issues.

In addition, optimization-based methods alter the statistical meaning of the SVM by modifying the target function. This brings a theoretical dilemma: *ad hoc* modifications of the problem formulation denatures the essence of the SVM as an implementation of the structural risk minimization principle (see Chapter 2). In other words, large modifications of the optimization problem lead to giving up on the theoretically guaranteed advantages of the SVM.

Finally, they are not a good choice to deal with the issue of missing data: while displacing the optimum in the search space, they do not modify the search space itself. Indeed, the form of a solution f remains the one given by the application of representer theorem studied in Chapter 2:

$$f(x) = \sum_{i=1}^{N} K(x_i, x) + b \tag{3.39}$$

which quality directly depends on the training data.

Compared to other approaches, the *kernel-based methods* offer the most implicit and indirect way to deal with prior-knowledge. Therefore, they usually require more intuition to design them and more theoretical work to justify them. However, the "kernel trick" is the natural and theoretically valid way to modify the RKHS in which the solution is searched. Moreover, the search space will be adapted regardless of the available data.

Therefore, a kernel-based approach appears as the soundest and most promising option.

3.5.3 Future challenges and promising leads

The present review prompts several conclusions regarding the current state-of-the-art.

First, most of the current methods are not designed to perform well in situations where training data is severely lacking. Therefore, they do not allow the use of prior-knowledge as a substitute for training data.

Second, the type of prior-knowledge addressed in the current methods relates more to the general domain of application rather than the problem itself.

Third, although more difficult to design and justify, the kernel-based approaches do not suffer from the crippling drawbacks of sample-based methods and the limitations of optimization-based methods.

In the light of these conclusions, it appears necessary to focus the future efforts towards the incorporation of prior-knowledge more specific to the properties of the problem itself, for which a kernel-based approach seems the most indicated.

A framework enabling an effective substitution of the missing data with prior-knowledge would be an important stepping stone for a switch of paradigm in the current use of SVMs towards more realistic situations with limited data and a few global properties about the problem.

Chapter 4

KE-RBF: Augmenting the RBF Kernel with Prior-Knowledge

4.1 Introduction

In this chapter, we present our original framework for the incorporation of various forms of prior-knowledge into SVMs referred to as the Knowledge-Enhanced RBF (KE-RBF) framework.

KE-RBF kernels are modifications of the standard RBF kernel, widely regarded as the best general purpose kernel due to its power and versatility. They provide an framework enabling the incorporation of various type of prior-knowledge commonly available as expert advice on the problem. The idea behind KE-RBF kernels is to preserve the power and versatility of the standard RBF kernel, while allowing for the incorporation of problem-specific prior-knowledge. They can be used with the existing types SVMs, including all variants of SVCs and SVRs, with the same ease-of-use as the original RBF kernel and without significantly increasing the computational complexity of the optimization problem.

The objective in mind is to broaden the field of application of SVMs by enabling their use in situations where SVMs are usually considered ineffective.

4.1.1 Motivations

The main motivation behind the KE-RBF framework is to allow the use of the powerful SVM+RBF combination in more realistic contexts than what is currently possible.

The SVM+RBF combination is one of the most widely used class of suppervised learning algorithms. In particular, the nonlinear RBF kernel with adjustable kernel bandwidth offers the versatility necessary to adapt to a wide variety of situations.

However, the volume of training data required to take advantage of nonlinear classifiers such as the SVM+RBF combination can be very high. Several previous studies [2, 62] suggest that linear methods, usually considered much less powerful, are often a better choice than nonlinear methods when the available data is limited. Therefore, the practical use of the SVM+RBF combination is severely restricted by the requirement for quality training data in sufficient amounts.

In many real-life situations, training data is available only in limited quantities. Meanwhile, specific expert advice about the problem is often available. In fact, the "learning-by-example" paradigm which involves the creation of models from entirely implicit knowledge is not a natural analogy of the way concepts are defined in real life. For instance, histopathology textbooks describe a specific condition with text and a small amount of micrographs of typical cases rather than a huge collection of example micrographs covering possible positive and negative cases of the disease.

Accordingly, our objective is to enable a shift of paradigm towards a more practical use of SVMs: from an often unrealistic situation where lots of training data are required to a more practical situation where a limited amount of data in addition to some problem-specific advice is available.

4.1.2 Main features of the KE-RBF framework

The KE-RBF framework is able to deal with a large variety of problem-specific prior-knowledge such as specific correlation patterns present in the problem, the pseudo-periodicity or dominant frequencies of phenomena, or specific knowledge on regions from the feature space (more precise definitions are given in Section 4.2). In contrast, most of the previous works on the incorporation of prior-knowledge into SVMs deals with domain-specific knowledge such as invariances which do not provide specific informations

on the problem itself (see Chapter 3).

Another main characteristic of KE-RBF kernels is their affinity with small or biased training sets. As pointed out during the review in Chapter 3, the existing methods dealing with problem-specific knowledge are optimization-based approaches incorporating the prior-knowledge as additional constraints. Unfortunately, this approach is not able to yield a good solution when the original search space is inadequate due to the lack of data. In comparison, a SVM+KE-RBF combination will adapt the search space to the available prior-knowledge rather than just shift the optimum. gRBF kernels, a subtype of KE-RBF kernels presented in Section 4.5 can even be used just with prior-knowledge in the absence of any training data.

Finally, being a purely kernel-based approach, the KE-RBF framework has a number of advantages in terms of ease of use. In particular, it is compatible with standard SVMs and solvers without requiring modifications and it does not significantly increase the complexity of the problem.

4.1.3 Outline

Section 4.2 gives a general overview of which type of KE-RBF kernel to use with which type of prior knowledge. Then, the 3 different types of KE-RBF kernels are presented in their respective sections: ξRBF kernels incorporating the prior-knowledge via a dedicated knowledge function in Section 4.3; pRBF kernels based on tensor products of an RBF kernel with more specific kernels in Section 4.4; and gRBF kernels, a generalization of the RBF kernel from $\mathbb{R}^n$ to $\mathfrak{P}(\mathbb{R}^n)$, in Section 4.5. We conclude the chapter on a discussion on the complementary role of the prior-knowledge and the usual labeled training data in Section 4.6.

A thorough empirical validation of the KE-RBF framework on several real-life and synthetic problems is provided in Chapter 5.

This chapter uses the notations introduced in Chapter 2. In particular, $\mathcal{X}$ designates the *input space or feature space* and $\mathcal{Y} \subset \mathbb{R}$ the *output or label space*. We assume $\mathcal{X} \subset \mathbb{R}^n$ for some $n \in \mathbb{N}$.

4.2 Overview of the KE-RBF framework

The KE-RBF framework consists of 3 mathematically different types of modifications of the standard RBF kernel and are able to deal with several different types of prior-knowledge. Therefore, there are two natural angles of approach to the KE-RBF framework: the mathematical nature of the kernel and the type of prior-knowledge involved.

4.2.1 Types of KE-RBF kernels

The modified RBF kernels fall into one of the following mathematical categories.

ξRBF kernels: they correspond to the product of the standard RBF kernel K_{rbf} with a function ξ containing the prior-knowledge *i.e.* $K_a(x_1, x_2) = \xi(x_1, x_2)K_{\text{rbf}}(x_1, x_2)$;

pRBF kernels: they are tensor products of the standard RBF kernel with another kernel K having more characteristic properties (*e.g.* monotonicity) *i.e.* $K_a(x_1, x_2) = K_{\text{rbf}}(x_{1,1}, x_{2,1}) \times K(x_{1,2}, x_{2,2})$ with $x_1 = (x_{1,2}, x_{1,2})$ and $x_2 = (x_{2,2}, x_{2,2})$.

gRBF kernels: they are a generalization of the standard RBF kernel from $\mathbb{R}^n \times \mathbb{R}^n$ to $\mathfrak{P}(\mathbb{R}^n) \times \mathfrak{P}(\mathbb{R}^n)$, *i.e.* from points of $\mathbb{R}^n$ to sets of $\mathbb{R}^n$.

4.2.2 Types of prior-knowledge

The prior-knowledge involved in the KE-RBF framework can be divided into two broad categories: *semi-global* prior-knowledge influencing large regions of the feature space and *global* prior-knowledge influencing the entire feature space.

4.2.2.1 Semi-global prior-knowledge

Two subtypes of semi-global prior-knowledge can be incorporated with the KE-RBF framework.

Unlabeled regions $\mathcal{X}_0 \subset \mathcal{X}$: they can be viewed as an indicative clustering of points in $\mathcal{X}$ in order to underline their similarity, and do not require any explicit hypothesis on the label space $\mathcal{Y}$.

Labeled regions $(\mathcal{X}_0, y_0) \in \mathfrak{P}(\mathcal{X}) \times \mathcal{Y}$: they can be viewed as defining an average label value for the points in the region.

4.2.2.2 Global prior-knowledge

Four subtypes of global prior-knowledge are dealt with.

Monotonicity *w.r.t.* one or more feature: it refers to the increasing or decreasing behavior of the label *w.r.t.* a feature. For instance, the price of wine bottles can be considered as an increasing function of the age in years.

Pseudo-periodicity *w.r.t.* one or more features: it indicates that labels have a cyclic behavior *w.r.t.* a feature. An example is air temperature and the day-night cycle.

Frequency decomposition *w.r.t.* one or more features: sometimes, more than one dominant frequencies are involved. For instance air temperatures also follow a seasonal cycle in addition to the day-night cycle and therefore correspond to the combination of at least 2 dominant frequencies.

Explicit correlation pattern between the label and a specific set of features: for instance, explicit correlation patterns can be found between body volume and body mass which are linearly correlated or between car speed and breaking distance which are quadratically correlated.

4.2.3 Matrix representation of the KE-RBF framework

The matrix representation in Table 4.1 indicates which type of kernel can be used with which type or prior-knowledge. The matching is not one-to-one and may be a bit misleading: unlabeled regions and pseudo-periodicity which are seemingly unrelated types of prior-knowledge are incorporated with the same kernel (ξRBF kernel), whereas labeled regions are dealt with another kernel (gRBF kernel). Practical examples for the use for each method and type of prior-knowledge are given in Chapter 5.

4.3 ξRBF kernel

ξRBF kernels correspond to the functional product of the standard RBF kernel with a real-valued function ξ defined over $\mathcal{X}^2$ and containing the prior-knowledge. The most

		ξRBF	pRBF	gRBF
semi-global	unlabeled regions	×		
	labeled regions			×
global	monotonicity		×	
	pseudo-periodicity	×		
	frequency decomposition	×		
	explicit correlation		×	

Table 4.1: Matrix representation of the different types of KE-RBF kernels (top) with the different types of prior-knowledge (left). Crosses indicate kernels that can be used with a specific type of prior-knowledge.

generic expression of a ξRBF kernel is:

$$K_a(x_1, x_2) = \xi(x_1, x_2) K_{\text{rbf}}(x_1, x_2) \tag{4.1}$$

where $\xi : \mathcal{X}^2 \to \mathbb{R}$ is a symmetric function containing the prior-knowledge.

Assuming that the modified kernel K_a is a valid PD kernel, the idea is to alter the notion of kernel distance in order to influence the separability of points according to the prior-knowledge. On one hand, if the prior-knowledge suggests that two objects share similarities, then the objects should be moved closer and the kernel distance decreased. On the other hand, if it implies that those two objects are unrelated of dissimilar, the objects should be moved further apart and the kernel distance increased.

If desired, the amount of prior-knowledge incorporated into the kernel can be controlled with an additional parameter:

$$K_a(x_1, x_2) = (\lambda + \mu\xi(x_1, x_2)) K_{\text{rbf}}(x_1, x_2) \tag{4.2}$$

where $\mu = 1 - \lambda \in [0, 1]$ controls the the amount of prior-knowledge (note that (4.1) corresponds to the case $\mu = 1$).

In practice, the additional parameter μ should be set according to the degree of confidence regarding the prior-knowledge. $\mu = 1$ is a good default choice when the prior-knowledge comes from a reliable source. An empirical study on role of μ is available in an application of this ξRBF kernel to the diagnosis of breast cancer from morphological parameters of cell nuclei in Section 5.2.

The function ξ can be adapted to incorporate various forms of prior-knowledge. In the following sections, we deal with different types of prior-knowledge: unlabeled

regions of $\mathcal{X}$ without any explicit hypothesis on the label space $\mathcal{Y}$ in Section 4.3.1) and the frequency decomposition of the labeling model *w.r.t.* one or several features in Section 4.3.2. The latter can be a single pseudo-period or a combination of multiple dominant frequencies.

For reference, we provide a slightly different approach to the kernels presented in this section in [84].

4.3.1 Unlabeled regions

Unlabeled regions correspond to sets $\mathcal{A} \subset \mathcal{X}$ of the input space without explicit hypothesis regarding the label space $\mathcal{Y}$. This type of prior-knowledge can be viewed as an indicative clustering of the data points which emphasizes similarities and dissimilarities between the objects.

First, a version dealing with crisp sets (standard mathematical sets) is presented in Section 4.3.1.1. Then, the framework is extended to fuzzy sets in Section 4.3.1.2.

An application to digital histopathology using real medical data is given in Section 5.2.

4.3.1.1 Crisp unlabeled regions

Let $\mathcal{A} \subset \mathcal{X}$ be a subset (region) of the feature space. Let $\chi : \mathcal{X} \to \{-1, 1\}$ be an indicator function for the set $\mathcal{A}$ such that:

$$\chi(x) = \begin{cases} 1 & \text{if } x \in \mathcal{A} \\ -1 & \text{if } x \notin \mathcal{A} \end{cases} \tag{4.3}$$

The only restriction imposed on the set $\mathcal{A}$ is the existence of an indicator function. This very loose restriction allows for the use of virtually any set with an analytical description.

We propose the following ξRBF kernel:

$$K_a(x_1, x_2) = \xi(x_1, x_2) K_{\text{rbf}}(x_1, x_2) \tag{4.4}$$

where $\xi : \mathcal{X}^2 \to [0,1]$ containing the prior-knowledge is defined as follows:

$$\xi(x_1, x_2) = \frac{\chi(x_1)\chi(x_2) + 1}{2} \tag{4.5}$$

We verify that K_a has the good properties, *i.e.* K_a is PD. This is a straightforward consequence of the two following results on PD kernels.

Theorem 4.3.1.

Let $K_1 : \mathcal{X}^2 \to \mathbb{R}$ and $K_2 : \mathcal{X}^2 \to \mathbb{R}$ be PD, and $\lambda \in \mathbb{R}^+$. Then:

1. *$K_1 + K_2$ is PD*
2. *$K_1 \times K_2$ is PD*
3. *$K_1 + \lambda$ is PD*
4. *λK_1 is PD*

Proof. All four kernels are symmetric. Thus, we only need to verify that their Gram matrices are positive semi-definite. Let $N \in \mathbb{R}$, $(x_1, x_2, \ldots, x_N) \in \mathcal{X}^N$ and $(v_1, v_2, \ldots, v_N) \in \mathbb{R}^N$.

Proof of 1.

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j (K_1 + K_2)(x_i, x_j) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j (K_1(x_i, x_j) + K_2(x_i, x_j)) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j K_1(x_i, x_j) + \sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j K_2(x_i, x_j) \\
&\geq 0 \text{ as the sum of two positive terms } (K_1 \text{ and } K_2 \text{ are PD})
\end{aligned}$$

Therefore $K_1 + K_2$ is PD.

Proof of 2.

The Gram matrix $G_2 = (K_2(x_i, x_j))_{i,j=1\ldots N}$ is positive semi-definite. Therefore, there is an N-by-N matrix $M = (m_{i,j})_{i,j=1\ldots N}$ (we can for instance consider the Cholesky

decomposition of G_2) such that $G_2 = MM^T$. Then:

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j (K_1 \times K_2)(x_i, x_j) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j K_1(x_i, x_j) K_2(x_i, x_j) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j (K_1(x_i, x_j) \sum_{k=1}^{N} m_{i,k} m_{k,j} \\
&= \sum_{k=1}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j (K_1(x_i, x_j) m_{i,k} m_{k,j} \\
&= \sum_{k=1}^{N} \left(\sum_{i=1}^{N}\sum_{j=1}^{N} (v_i m_{i,k})(v_j m_{k,j}) K_1(x_i, x_j) \right) \\
&\geq 0 \text{ as the sum of } N \text{ positive terms } (K_1 \text{ is PD})
\end{aligned}$$

Therefore $K_1 \times K_2$ is PD.

Proof of 3.

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j \lambda &= \lambda \sum_{i=1}^{N} v_i \sum_{j=1}^{N} v_j \\
&= \lambda \left(\sum_{i=1}^{N} v_i \right)^2 \\
&\geq 0
\end{aligned}$$

Therefore, $(x_1, x_2) \mapsto \lambda$ is PD and *3* is a corollary of *1.*

In a similar fashion, *4* is a corollary of *2.*

□

Theorem 4.3.2.

Let $f : \mathcal{X} \to \mathbb{R}$.

Then:

$$\begin{array}{rccc}
K: & \mathcal{X}^2 & \to & \mathbb{R} \\
& (x_1, x_2) & \mapsto & f(x_1) f(x_2)
\end{array}$$

is PD.

Proof. K is symmetric. Again, we only need to verify that any Gram matrix is positive

semi-definite. Let $N \in \mathbb{R}$, $(x_1, x_2, \ldots, x_N) \in \mathcal{X}^N$ and $(v_1, v_2, \ldots, v_N) \in \mathbb{R}^N$.

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j K(x_i, x_j) &= \sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j f(x_i) f(x_j) \\
&= \sum_{i=1}^{N} v_i f(x_i) \sum_{j=1}^{N} v_j f(x_j) \\
&= \left(\sum_{i=1}^{N} v_i f(x_i)\right)^2 \\
&\geq 0
\end{aligned}$$

□

The ξRBF kernel K_a is PD as a direct consequence of the two previous results.

Theorem 4.3.3.

K_a is PD.

Proof. By construction, applying Theorem 4.3.1 and Theorem 4.3.2. □

This result entails the existence of a RKHS $\mathcal{H}_a$ for K_a. Thus, the kernel distance d_a in $\mathcal{H}_a$ between two points $(x_1, x_2) \in \mathcal{X}^2$ can be expressed using Theorem 2.2.9 from Chapter 2. By successive transformations, we get:

$$\begin{aligned}
d_a(x_1, x_2)^2 &= K_a(x_1, x_1) + K_a(x_2, x_2) - 2K_a(x_1, x_2) \\
&= \frac{\chi(x_1)^2 + 1}{2} K_{\text{rbf}}(x_1, x_1) + \frac{\chi(x_2)^2 + 1}{2} K_{\text{rbf}}(x_2, x_2) \\
&\quad - 2\frac{\chi(x_1)\chi(x_2) + 1}{2} K_{\text{rbf}}(x_1, x_2) \\
&= \frac{1}{2}\left[(\chi(x_1)^2 + 1) + (\chi(x_2)^2 + 1) - 2(\chi(x_1)\chi(x_2) + 1) K_{\text{rbf}}(x_1, x_2)\right] \\
&= \frac{1}{2}\left[(\chi(x_1)^2 + 1) + (\chi(x_2)^2 + 1) - 2(\chi(x_1)\chi(x_2) + 1)\right. \\
&\quad \left. + 2(\chi(x_1)\chi(x_2) + 1) - 2(\chi(x_1)\chi(x_2) + 1) K_{\text{rbf}}(x_1, x_2)\right] \\
&= \frac{1}{2}\left[\chi(x_1)^2 + \chi(x_2)^2 - 2\chi(x_1)\chi(x_2)\right] \\
&\quad + \frac{1}{2}\left[2(\chi(x_1)\chi(x_2) + 1) - 2(\chi(x_1)\chi(x_2) + 1) K_{\text{rbf}}(x_1, x_2)\right] \\
&= \frac{1}{2}(\chi(x_1) - \chi(x_2))^2 + \frac{1}{2}(\chi(x_1)\chi(x_2) + 1)(2 - 2K_{\text{rbf}}(x_1, x_2)) \\
&= \frac{1}{2}(\chi(x_1) - \chi(x_2))^2
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{1}{2}(\chi(x_1)\chi(x_2)+1)(K_{\text{rbf}}(x_1,x_1)+K_{\text{rbf}}(x_2,x_2)-2K_{\text{rbf}}(x_1,x_2)) \\
&= \frac{1}{2}(\chi(x_1)-\chi(x_2))^2 + \frac{1}{2}(\chi(x_1)\chi(x_2)+1)d_{\text{rbf}}(x_1,x_2)^2
\end{aligned} \tag{4.6}
$$

where $d_{\text{rbf}}(x_1, x_2)$ is the standard RBF kernel distance. Then, by applying a case disjunction, (4.6) becomes:

$$
\begin{aligned}
&d_a(x_1,x_2)^2 \\
&= \begin{cases} d_{\text{rbf}}(x_1,x_2)^2 & \text{if } (x_1,x_2) \in \mathcal{A}^2 \cup \complement\mathcal{A}^2 \\ 2 & \text{if } (x_1,x_2) \in \mathcal{A}\times\complement\mathcal{A} \cup \complement\mathcal{A}\times\mathcal{A} \end{cases} \\
&= \begin{cases} d_{\text{rbf}}(x_1,x_2)^2 & \text{if } (x_1,x_2) \in \mathcal{A}^2 \cup \complement\mathcal{A}^2 \\ (\sup d_{\text{rbf}})^2 & \text{if } (x_1,x_2) \in \mathcal{A}\times\complement\mathcal{A} \cup \complement\mathcal{A}\times\mathcal{A} \end{cases}
\end{aligned} \tag{4.7}
$$

where $\sup d_{\text{rbf}} = \sup_{(x_1,x_2)\in\mathcal{X}^2} d_{\text{rbf}}(x_1,x_2) = \sqrt{2}$ is the upper bound of the RBF kernel distance.

Therefore, the kernel distance associated to K_a increases when the two points x_1 and x_2 are in different sets, increasing the separability of $\mathcal{A}$ and $\complement\mathcal{A}$ in the kernel space.

However, this sudden increase to the upper-bound value of the RBF kernel distance may feel too sharp. To solve this problem, we add a parameter $\mu \in [0,1]$ to control the amount of prior-knowledge incorporated into the ξRBF kernel from none for $\mu = 0$ to the maximum for $\mu = 1$. With this new control parameter, (4.4) becomes:

$$
K_a(x_1,x_2) = (\lambda + \mu\xi(x_1,x_2))K_{\text{rbf}}(x_1,x_2) \tag{4.8}
$$

with $\lambda = 1-\mu \in [0,1]$.

Remark 4.3.4. With $\mu = 0$, (4.8) becomes the standard RBF kernel. The previous expression (4.4) is obtained when $\mu = 1$.

K_a is still PD as a direct consequence of Theorem 4.3.1 and Theorem 4.3.2. Therefore, the notion kernel distance d_a between two points $(x_1, x_2) \in \mathcal{X}^2$ is valid. By successive transformations of its new expression:

$$
d_a(x_1,x_2)^2
$$

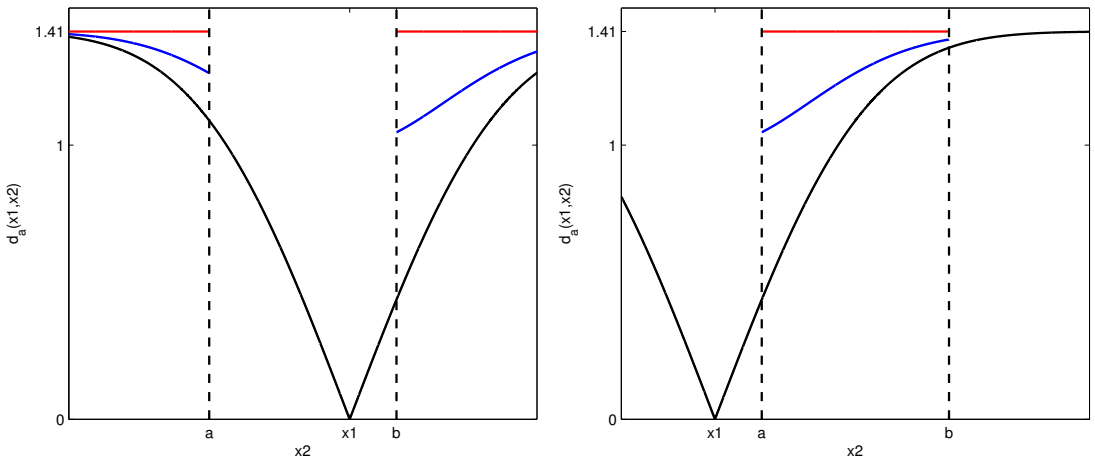
$$
\begin{aligned}
&= K_a(x_1, x_1) + K_a(x_2, x_2) - 2K_a(x_1, x_2) \\
&= (\lambda + \mu\frac{\chi(x_1)^2 + 1}{2})K_{\text{rbf}}(x_1, x_1) + (\lambda + \mu\frac{\chi(x_2)^2 + 1}{2})K_{\text{rbf}}(x_2, x_2) \\
&- 2(\lambda + \mu\frac{\chi(x_1)\chi(x_2) + 1}{2})K_{\text{rbf}}(x_1, x_2) \\
&= \lambda\left[K_{\text{rbf}}(x_1, x_1) + K_{\text{rbf}}(x_2, x_2) - 2K_{\text{rbf}}(x_1, x_2)\right] \\
&+ \frac{\mu}{2}\left[(\chi(x_1)^2 + 1)K_{\text{rbf}}(x_1, x_1) + (\chi(x_2)^2 + 1)K_{\text{rbf}}(x_2, x_2) - 2\frac{\chi(x_1)\chi(x_2) + 1}{2}\right] \\
&= \lambda d_{\text{rbf}}(x_1, x_2)^2 + \frac{\mu}{2}\left[(\chi(x_1)^2 + 1) + (\chi(x_2)^2 + 1) - 2\frac{\chi(x_1)\chi(x_2) + 1}{2}\right] \qquad (4.9)
\end{aligned}
$$

then, by applying the same sequence of transformations as in (4.6):

$$
\begin{aligned}
&d_a(x_1, x_2)^2 \\
&= \lambda d_{\text{rbf}}(x_1, x_2)^2 + \frac{\mu}{2}(\chi(x_1) - \chi(x_2))^2 + \frac{\mu}{2}(\chi(x_1)\chi(x_2) + 1)d_{\text{rbf}}(x_1, x_2)^2 \\
&= \left[\lambda + \frac{\mu}{2}(\chi(x_1)\chi(x_2) + 1)\right] d_{\text{rbf}}(x_1, x_2)^2 + \frac{\mu}{2}(\chi(x_1) - \chi(x_2))^2 \\
&= \begin{cases} (\lambda + \mu)d_{\text{rbf}}(x_1, x_2)^2 & \text{if } (x_1, x_2) \in \mathcal{A}^2 \cup \complement\mathcal{A}^2 \\ \lambda d_{\text{rbf}}(x_1, x_2)^2 + 2\mu & \text{if } (x_1, x_2) \in \mathcal{A} \times \complement\mathcal{A} \cup \complement\mathcal{A} \times \mathcal{A} \end{cases} \\
&= \begin{cases} d_{\text{rbf}}(x_1, x_2)^2 & \text{if } (x_1, x_2) \in \mathcal{A}^2 \cup \complement\mathcal{A}^2 \\ (1 - \mu)d_{\text{rbf}}(x_1, x_2)^2 + \mu(\sup d_{\text{rbf}})^2 & \text{if } (x_1, x_2) \in \mathcal{A} \times \complement\mathcal{A} \cup \complement\mathcal{A} \times \mathcal{A} \end{cases} \qquad (4.10)
\end{aligned}
$$

Figure 4.1 shows plots of the ξRBF kernel distance $d_a(x_1, x_2)$ for $n = 1$, $\mathcal{A} = [a, b]$ and different values of the parameter $\mu \in [0, 1]$. The different possible relative positions of x_1 and x_2 are covered. We can observe that when the two points are in the same set ($\mathcal{A}$ or $\complement\mathcal{A}$), the kernel distance between them is the standard RBF kernel distance. However, when they are in different sets, the kernel distance increases by an amount controllable via the parameter μ: from no increase when $\mu = 0$ to an increase to the maximal RBF kernel distance $\sup d_a = \sqrt{2}$ when $\mu = 1$.

Remark 4.3.5. One may rightfully point out that instead of the expression of ξ given in (4.5), we may use the following simpler and equivalent expression:



(a) case: $x_1 \in \mathcal{A}$ **(b)** case: $x_1 \in \complement\mathcal{A}$

Figure 4.1: ξRBF kernel distance $d_a(x_1, x_2)$ for $n = 1$, $\mathcal{A} = [a, b]$ and different values of the parameter μ. Black plots correspond to $\mu = 0$ *i.e.* the standard RBF kernel, blue plots to $\mu = 0.5$ and red plots to $\mu = 1$.

$$\xi(x_1, x_2) = \begin{cases} 1 & \text{if } (x_1, x_2) \in \mathcal{A}^2 \cup \complement\mathcal{A}^2 \\ 0 & \text{if } (x_1, x_2) \in \mathcal{A} \times \complement\mathcal{A} \cup \complement\mathcal{A} \times \mathcal{A} \end{cases} \tag{4.11}$$

The reason behind this seemingly unnatural choice is that it extends well to the case of fuzzy sets elaborated in Section 4.3.1.2.

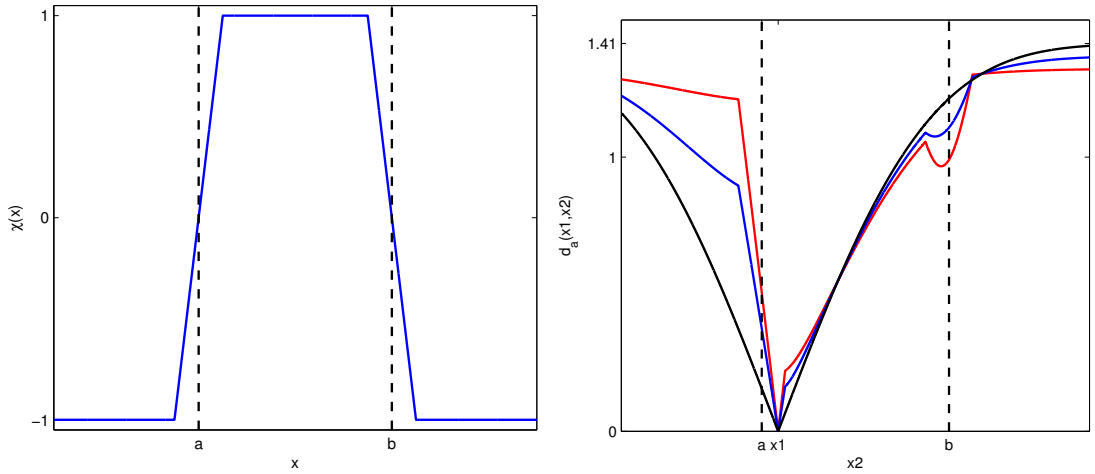
4.3.1.2 Fuzzy unlabeled regions

The above ξRBF kernel can sometimes prove impractical when the boundaries of the unlabeled regions are not precisely known. Instead, the prior-knowledge may correspond to a blur idea of them. Therefore, we propose an extension of the previous method allowing fuzzy set definitions, *i.e.* with a continuous indicator function $\chi : \mathcal{X} \to [-1, 1]$.

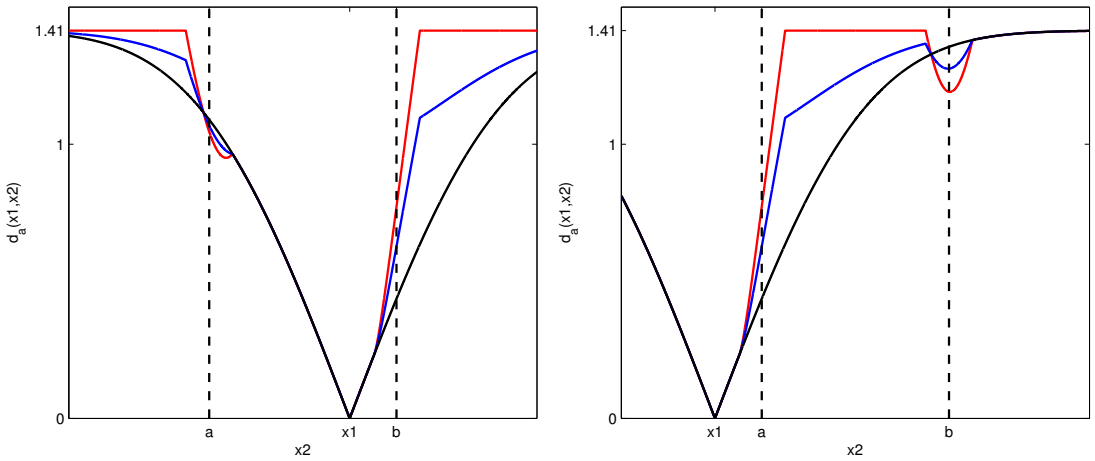
The positive-definiteness of K_a still holds as a consequence of Theorem 4.3.1 and Theorem 4.3.2. The reformulation (4.10) of the kernel distance d_a remains valid as well. Figure 4.2 shows a fuzzified version of the illustration in Figure 4.1 with crisp sets. We can see that the previously discontinuous transitions are now smooth.

4.3.2 Frequency decomposition

Information about the frequency decomposition of the model is sometimes available. The ideal case is a strictly periodic phenomenon, *i.e.* which has a true period P *w.r.t.* a



(a) Indicator function χ **(b)** case: $-1 < \chi(x_1) < 1$



(c) case: $\chi(x_1) = 1$ **(d)** case: $\chi(x_1) = -1$

Figure 4.2: (a) fuzzy indicator function and (b)-(d) corresponding ξRBF kernel distance $d_a(x_1, x_2)$ for $n = 1$. Different values of μ are used: black plots correspond to $\mu = 0$ *i.e.* the standard RBF kernel, blue plots to $\mu = 0.5$ and red plots to $\mu = 1$.

specific feature but such a case does not offer much practical interest from the machine learning standpoint.

In practice, a phenomenon can have a dominant frequency or pseudo-period without being strictly periodic. We propose a type of ξRBF kernel addressing this case in Section 4.3.2.1. In Section 4.3.2.2, we propose an extension to the combination of several dominant frequencies.

We illustrate the use of such kernels with an application to meteorological predictions in Section 5.3 and an experiment using synthetic data in Section 5.4.

4.3.2.1 Pseudo-period

In this section, the decision model is expected to have a pseudo-period of P *w.r.t.* to the j-th component of the feature vector. To address this case, we propose the following ξRBF kernel:

$$K_a(x_1, x_2) = \xi(x_1, x_2) K_{\text{rbf}}(x_1, x_2) \tag{4.12}$$

with $\xi : \mathcal{X}^2 \to [0, 1]$ a function containing the prior-knowledge defined as:

$$\xi(x_1, x_2) = \frac{\cos\left(\frac{2\pi}{P}(x_{1,j} - x_{2,j})\right) + 1}{2} \tag{4.13}$$

where $x_{1,j}$ (resp. $x_{2,j}$) is the j-th component of x_1 (resp. x_2).

As in Section 4.3.1.1, we can introduce a parameter $\mu \in [0, 1]$ controlling the amount of prior-knowledge incorporated into K_a. Thus, (4.12) becomes:

$$K_a(x_1, x_2) = (\lambda + \mu\xi(x_1, x_2)) K_{\text{rbf}}(x_1, x_2) \tag{4.14}$$

with $\lambda = 1 - \mu \in [0, 1]$.

First, we verify that K_a has the properties of a "good" kernel.

Theorem 4.3.6.

K_a is PD.

Proof. By the application of a well-known trigonometric formula, ξ can be expanded in

the following fashion:

$$\xi(x_1, x_2) = \frac{\cos\left(\frac{2\pi}{P}x_{1,j}\right)\cos\left(\frac{2\pi}{P}x_{2,j}\right) + \sin\left(\frac{2\pi}{P}x_{1,j}\right)\sin\left(\frac{2\pi}{P}x_{2,j}\right) + 1}{2} \tag{4.15}$$

Then, Theorem 4.3.1 and Theorem 4.3.2 entail that ξ is PD as a sum of PD kernels. K_a is in turn PD as the product of PD kernels. □

Then, the kernel distance d_a associated to K_a can be expressed applying Theorem 2.2.9 from Chapter 2.

$$\begin{aligned}
&d_a(x_1, x_2)^2 \\
&= K_a(x_1, x_1) + K_a(x_2, x_2) - 2K_a(x_1, x_2) \\
&= (\lambda + \mu\frac{\cos\left(\frac{2\pi}{P}(x_{1,j} - x_{1,j})\right) + 1}{2})K_{\text{rbf}}(x_1, x_1) \\
&+ (\lambda + \mu\frac{\cos\left(\frac{2\pi}{P}(x_{2,j} - x_{2,j})\right) + 1}{2})K_{\text{rbf}}(x_2, x_2) \\
&- 2(\lambda + \mu\frac{\cos\left(\frac{2\pi}{P}(x_{1,j} - x_{2,j})\right) + 1}{2})K_{\text{rbf}}(x_1, x_2) \\
&= (\lambda + \mu)K_{\text{rbf}}(x_1, x_1) + (\lambda + \mu)K_{\text{rbf}}(x_2, x_2) \\
&- 2(\lambda + \mu\frac{\cos\left(\frac{2\pi}{P}(x_{1,j} - x_{2,j})\right) + 1}{2})K_{\text{rbf}}(x_1, x_2) \\
&= \lambda\left[K_{\text{rbf}}(x_1, x_1) + K_{\text{rbf}}(x_2, x_2) - 2K_{\text{rbf}}(x_1, x_2)\right] \\
&+ \mu\left[K_{\text{rbf}}(x_1, x_1) + K_{\text{rbf}}(x_2, x_2) - \left(\cos\left(\frac{2\pi}{P}(x_{1,j} - x_{2,j})\right) + 1\right)K_{\text{rbf}}(x_1, x_2)\right] \\
&= \lambda d_{\text{rbf}}(x_1, x_2)^2 + \mu\left[2 - \left(\cos\left(\frac{2\pi}{P}(x_{1,j} - x_{2,j})\right) + 1\right)K_{\text{rbf}}(x_1, x_2)\right]
\end{aligned} \tag{4.16}$$

where d_{rbf} is the standard RBF kernel distance.

Figure 4.3 shows plots of the kernel distance according to the relative position of x_1 and x_2 for $n = 1$ and different values of the parameter μ. We can observe a pseudo-periodic increase in the ξRBF kernel distance compared to the standard RBF distance ($\mu = 0$) in addition to the exponential increase proper to the RBF kernel. Therefore, objects which are separated by a whole number of pseudo-periods are more strongly related than objects separated by a non-whole number of pseudo-periods. The exponential increase adjustable via the RBF kernel bandwidth parameter γ accounts for the fact that the labels are pseudo-periodic instead of strictly periodic. In this way, objects which at a close distance in $\mathcal{X}$ influence each other more the objects which are far, as

a standard RBF kernel would do. If the labels were strictly periodic, $\gamma = 0$ yielding a infinite-bandwidth kernel would be appropriate. The extent of the modifications can be controlled by tuning μ.

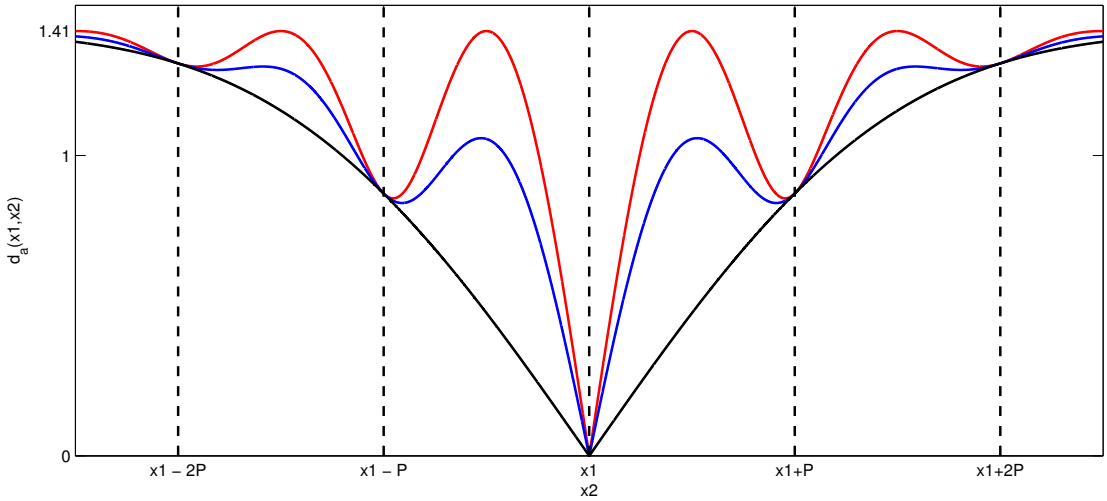


Figure 4.3: ξRBF kernel distance $d_a(x_1, x_2)$ for $n = 1$ and a pseudo-period P. Different values of μ are used: black plots correspond to $\mu = 0$ *i.e.* the standard RBF kernel, blue plots to $\mu = 0.5$ and red plots to $\mu = 1$. Vertical dashed lines are separated by a pseudo-period P.

4.3.2.2 Multiple frequencies

In this section, we propose an extension of the ξRBF kernel presented in Section 4.3.2.1 to the case when more than a single dominant label frequency is known *a priori*. This is for instance the case when multiple cycles of different pseudo-periods combine, *e.g.* a shorter day-and-night cycle ($P = 1$ day) with a longer seasonal cycle ($P = 365.25$ days).

Let $\{f_i\}_{i=1...N_0}$ be the N_0 different frequencies in question and $\{P_i = \frac{1}{f_i}\}_{i=1...N_0}$ the corresponding pseudo-periods. We propose the following extension of the ξRBF kernel (4.12):

$$K_a(x_1, x_2) = \left(\lambda + \mu \prod_{i=1}^{N_0} \xi_i(x_1, x_2)\right) K_{\text{rbf}}(x_1, x_2) \tag{4.17}$$

with $\mu = 1 - \lambda$ a parameter controlling the amount of prior-knowledge and $\{\xi_i\}_{i=1...N_0}$ a family of functions similar to (4.13) defined for each frequency as:

$$\xi_i(x_1, x_2) = \frac{\cos\left(\frac{2\pi}{P_i}(x_{1,j} - x_{2,j})\right) + 1}{2} = \frac{\cos(2\pi f_i(x_{1,j} - x_{2,j})) + 1}{2} \tag{4.18}$$

where $x_{1,j}$ (resp. $x_{2,j}$) is the j-th component of x_1 (resp. x_2).

Once more, K_a is a PD kernel with a valid RKHS.

Theorem 4.3.7.

K_a is PD.

Proof. Similar to the proof of Theorem 4.3.6 □

Following a sequence of transformations similar to (4.16), the associated kernel distance d_a can be expressed as:

$$
\begin{aligned}
&d_a(x_1, x_2)^2 \\
&= K_a(x_1, x_1) + K_a(x_2, x_2) - 2K_a(x_1, x_2) \\
&= (\lambda + \mu \prod_{i=1}^{N_0} \frac{\cos(2\pi f_i(x_{1,j} - x_{1,j})) + 1}{2}) K_{\text{rbf}}(x_1, x_1) \\
&+ (\lambda + \mu \prod_{i=1}^{N_0} \frac{\cos(2\pi f_i(x_{2,j} - x_{2,j})) + 1}{2}) K_{\text{rbf}}(x_2, x_2) \\
&- 2(\lambda + \mu \prod_{i=1}^{N_0} \frac{\cos(2\pi f_i(x_{1,j} - x_{2,j})) + 1}{2})) K_{\text{rbf}}(x_1, x_2) \\
&= (\lambda + \mu) K_{\text{rbf}}(x_1, x_1) + (\lambda + \mu) K_{\text{rbf}}(x_2, x_2) \\
&- 2(\lambda + \mu \prod_{i=1}^{N_0} \frac{\cos(2\pi f_i(x_{1,j} - x_{2,j})) + 1}{2}) K_{\text{rbf}}(x_1, x_2) \\
&= \lambda \left[K_{\text{rbf}}(x_1, x_1) + K_{\text{rbf}}(x_2, x_2) - 2K_{\text{rbf}}(x_1, x_2) \right] \\
&+ \mu \left[K_{\text{rbf}}(x_1, x_1) + K_{\text{rbf}}(x_2, x_2) - 2 \left(\prod_{i=1}^{N_0} \frac{\cos(2\pi f_i(x_{1,j} - x_{2,j})) + 1}{2} \right) K_{\text{rbf}}(x_1, x_2) \right] \\
&= \lambda d_{\text{rbf}}(x_1, x_2)^2 + 2\mu \left[1 - \left(\prod_{i=1}^{N_0} \frac{\cos(2\pi f_i(x_{1,j} - x_{2,j})) + 1}{2} \right) K_{\text{rbf}}(x_1, x_2) \right] \qquad (4.19)
\end{aligned}
$$

where d_{rbf} is the standard RBF kernel distance.

Figure 4.4 shows a plot of d_a for the case $n = 1$, different values of μ and two arbitrary frequencies $f_1 < f_2$ (*i.e.* $P_1 > P_2$). The kernel distance between two objects increases compared to the standard RBF kernel distance ($\mu = 0$). In particular, it is close to d_{rbf} only when x_1 and x_2 are separated by a whole number of both pseudo-periods and significantly larger when the distance separating x_1 and x_2 is not a whole number of pseudo-periods for either one of the pseudo-periods.

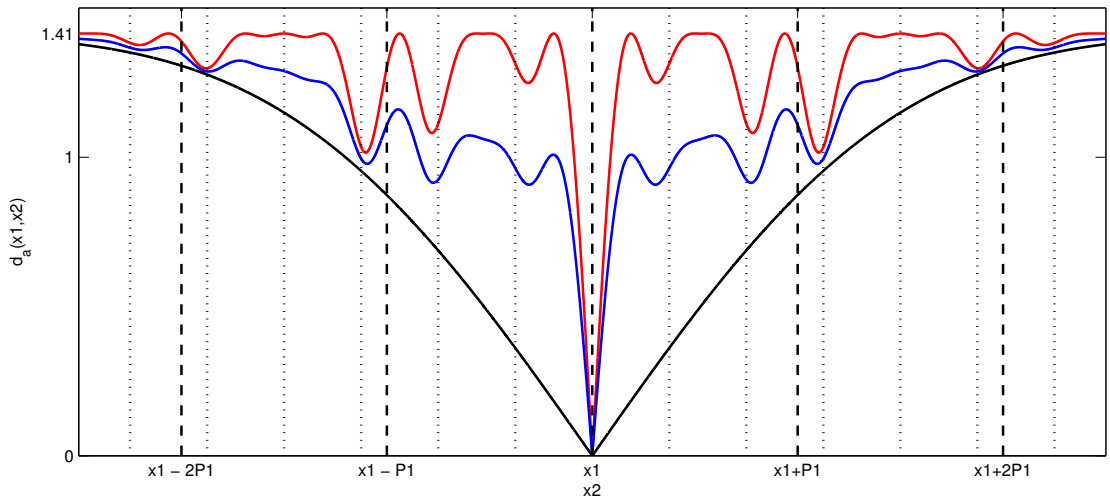


Figure 4.4: ξRBF kernel distance $d_a(x_1, x_2)$ for $n = 1$ and two pseudo-periods $P_1 > P_2$. The interval between dashed lines is equal to P_1 and the interval between dotted lines is equal to P_2. Different values of μ are used: black plots correspond to $\mu = 0$ *i.e.* the standard RBF kernel, blue plots to $\mu = 0.5$ and red plots to $\mu = 1$.

Remark 4.3.8. One may suggest to combine the different frequencies additively instead of multiplicatively, *i.e.* with the following expression for K_a instead of (4.17):

$$K_a(x_1, x_2) = \left(\lambda + \mu \sum_{i=1}^{N_0} \xi_i(x_1, x_2)\right) K_{\text{rbf}}(x_1, x_2) \tag{4.20}$$

This kernel is also PD and the kernel distance would become (we leave the details of the transformations to the reader):

$$\begin{aligned} &d_a(x_1, x_2)^2 \\ &= \lambda d_{\text{rbf}}(x_1, x_2)^2 + 2\mu \left[1 - \left(\sum_{i=1}^{N_0} \frac{\cos(2\pi f_i(x_{1,j} - x_{2,j})) + 1}{2}\right) K_{\text{rbf}}(x_1, x_2)\right] \end{aligned} \tag{4.21}$$

Figure 4.5 plots the multiplicative and the additive versions of this kernel distance for the case $n = 1$ and $\mu = 1$. The deviation from the standard RBF kernel is more important with the multiplicative version. More specifically, with the multiplicative version, the objects need to be separated by a whole number of both pseudo-periods in order to be close from each other in the the feature space, whereas with the additive version, a whole number of either one of the pseudo-period will suffice. The latter is undesirable as it may introduce dependence between data instances that should not be related.

The following example illustrates why you should need a whole number of both

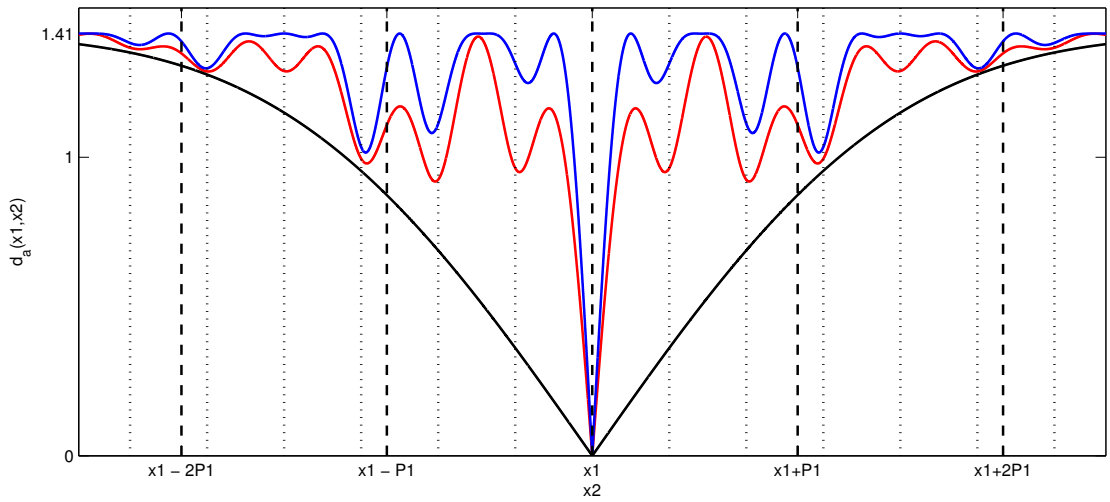


Figure 4.5: Comparison of the ξRBF kernel distance $d_a(x_1, x_2)$ for $n = 1$, $P_1 < P_2$ between the multiplicative version (4.21) and the additive version (4.16) of the ξRBF kernel. The black plot corresponds to the standard RBF (or $\mu = 0$ with either versions), the blue plot to the multiplicative version ($\mu = 1$) and the red plot to the additive version ($\mu = 1$).

pseudo-periods for instances to be closely related. The atmospheric temperature in London follow the cycle of seasons (pseudo-period of 356.25 days) and the diurnal cycle (pseudo-period of 1 day). The temperature recorded on August 1st 2005 at 2:20PM (21°C)[1] is largely different from the temperature on February 1st 2005 at 2:20PM (9°C). The temperature on August 1st 2005 at 2:20PM (21°C) is also largely different from the temperature on August 1st 2005 at 2:20AM (15°C). In comparison, the temperature on August 1st 2005 at 2:20PM (21°C) is fairly close to the temperature on August 1st 2006 at 2:20PM (20°C).

For a more systematic validation, the two versions of the kernel are compared in an empirical study in Section 5.4 which confirms the superiority of the multiplicative framework.

4.4 pRBF kernel

Partially RBF kernels, or pRBF kernels, are tensor products of a standard RBF kernel with another non-RBF kernel.

Often, one or more feature may have explicitly identifiable implications in terms of output labels if taken alone. For instance, a feature may be expected to have a specific correlation pattern with the label, such as "linear" correlation (*e.g.* acceleration

[1] temperatures according to actual records provided by `http://www.wunderground.com`

to force), "quadratic" correlation (*e.g.* speed to friction) or "cubic" correlation (*e.g.* dimensions to weight).

The pRBF kernels, by using more specific kernels only for a determined set of features and by using the RBF kernel for the remaining ones enables to incorporate the specific correlation patterns only with the relevant features while making no particular assumptions for the rest of the features. Under certain conditions specified by Theorem 4.4.6, a pRBF kernel not only incorporates the prior-knowledge into the SVMs but also guarantees that the solutions will have these mathematical properties.

4.4.1 Definition and properties

A pRBF kernel is defined as follows.

Definition 4.4.1. *pRBF kernel*

Let $1 \leq m \leq n-1$. A pRBF kernel over $\mathbb{R}^n$ is a function:

$$K_a = K_{\text{rbf}} \otimes K \tag{4.22}$$

where K_{rbf} is an RBF kernel over $\mathbb{R}^{n-m}$, K is a PD kernel over $\mathbb{R}^m$ and $\otimes$ is the tensor product.

Or equivalently:

$$\begin{array}{rccl} K_a: & \mathbb{R}^n \times \mathbb{R}^n & \to & \mathbb{R} \\ & (x_1, x_2) & \mapsto & K_{\text{rbf}}(x_{1,1}, x_{2,1}) \times K(x_{1,2}, x_{2,2}) \end{array}$$

with $x_1 = (x_{1,1}, x_{1,2}) \in \mathbb{R}^{n-m} \times \mathbb{R}^m$ and $x_2 = (x_{2,1}, x_{2,2}) \in \mathbb{R}^{n-m} \times \mathbb{R}^m$.

Remark 4.4.2. The tensor product used in the definition is the **tensor product of kernel functions** and not the tensor product of the kernel Gram matrices.

Remark 4.4.3. The combination of multiple kernels often referred to as as "multiple kernel learning" has been proposed in several anterior works and mainly linear combinations of different basic kernels [3, 7, 74]. The main idea is to optimize the coefficients of the linear combination during the learning phase. Tensor products have also been used in other works [30, 91], usually to combine data of a heterogeneous nature. Neither

of the approaches are motivated by the incorporation of additional prior-knowledge.

The set of PD kernels is closed under the tensor products of kernels.

Theorem 4.4.4. Tensor product of PD kernels

If $K_1 : \mathcal{X}_1^2 \to \mathbb{R}$ and $K_2 : \mathcal{X}_2^2 \to \mathbb{R}$ are PD kernels, then $K = K_1 \otimes K_2$ is a PD kernel over $\mathcal{X}_1 \times \mathcal{X}_2$.

Proof. We define:

$$\begin{array}{rccc} K_1' : & (\mathcal{X}_1 \times \mathcal{X}_2)^2 & \to & \mathbb{R} \\ & ((x_{1,1}, x_{1,2}), (x_{2,1}, x_{2,2})) & \mapsto & K_1(x_{1,1}, x_{2,1}) \end{array}$$

and:

$$\begin{array}{rccc} K_2' : & (\mathcal{X}_1 \times \mathcal{X}_2)^2 & \to & \mathbb{R} \\ & ((x_{1,1}, x_{1,2}), (x_{2,1}, x_{2,2})) & \mapsto & K_2(x_{1,2}, x_{2,2}) \end{array}$$

Then $K_1 \otimes K_2 = K_1' \times K_2'$ is PD by Theorem 4.3.1. □

Theorem 4.4.5.

A pRBF kernel is a PD kernel.

Proof. Corollary of Theorem 4.4.4. □

Before presenting our main result on pRBF kernels, lets first recall a notation introduced in Chapter 2. Given a PD kernel K over $\mathcal{X}$ and $x \in \mathcal{X}$, $K_x : \mathcal{X} \to \mathbb{R}$ is the function defined as:

$$\forall t \in \mathcal{X},\ K_x(t) = K(x,t) = K(t,x) \tag{4.23}$$

Theorem 4.4.6.

Let E a vector field over $\mathbb{R}$, K be a PD kernel over $\mathbb{R}^m$ such that $\{K_x | x \in \mathbb{R}^m\} \subset E$, $m < n$, $\mathcal{S} = \{x_1, \dots x_N\} \in (\mathbb{R}^n)^N$, $\Omega : \mathbb{R} \to \mathbb{R}$ strictly increasing, $\lambda > 0$ and $\Lambda : \mathbb{R}^N \to \mathbb{R}$.

Let:

$$\begin{array}{rccc} K_a : & (\mathbb{R}^{n-m} \times \mathbb{R}^m)^2 & \to & \mathbb{R} \\ & ((x_{1,1}, x_{2,1}), (x_{1,2}, x_{2,2})) & \mapsto & K_{rbf}(x_{1,1}, x_{2,1}) K(x_{1,2}, x_{2,2}) \end{array}$$

be a pRBF kernel over $\mathbb{R}^n$ *with* $\mathcal{H}_a$ *its RKHS.*

If $\hat{f} : \mathbb{R}^{n-m} \times \mathbb{R}^m \to \mathbb{R}$ *is a solution of the optimization problem:*

$$\underset{f \in \mathcal{H}_a}{\operatorname{argmin}} \Lambda(f(x_1), \ldots, f(x_N)) + \lambda \Omega(\|f\|_{\mathcal{H}_a}) \tag{4.24}$$

then $\forall x' \in \mathbb{R}^{n-m},\ \hat{f}_{x'} \in E$ *where:*

$$\begin{array}{rccc} \hat{f}_{x'} : & \mathbb{R}^m & \to & \mathbb{R} \\ & x & \mapsto & \hat{f}(x', x) \end{array} \tag{4.25}$$

Theorem 4.4.6 has a rather complicated formulation but its implications are simple to understand. All SVMs fit the formulation of the optimization problem (4.24). Therefore, in plain words, Theorem 4.4.6 implies that the properties of the non-RBF portion of the kernel pRBF kernel will be inherited by the labeling model. A graphical illustration of Theorem 4.4.6 is later given in Figure 4.6.

Proof. The optimization problem (4.24) satisfies the hypothesis of the representer theorem (Theorem 2.2.23). Therefore there exist $(\alpha_1, \ldots, \alpha_N) \in \mathbb{R}^N$ such that:

$$\hat{f} = \sum_{i=1}^{N} \alpha_i K_{a x_i} = \sum_{i=1}^{N} \alpha_i K_{\text{rbf} x_i} \otimes K_{x_i} \tag{4.26}$$

Then, for $x' \in \mathbb{R}^{n-m}$:

$$\hat{f}_{x'} = \sum_{i=1}^{N} \alpha_i K_{\text{rbf}}(x_i, x') K_{x_i} \tag{4.27}$$

Since $\alpha_i K_{\text{rbf}}(x_i, x') \in \mathbb{R}$ and $K_{x_i} \in E$, (4.27) is a linear combination of terms belonging to E. E being a real vector space, this completes the proof. □

Remark 4.4.7. The reader may raise the question why a direct sum $K_{\text{rbf}} \oplus K$ is not used instead of the tensor product $K_{\text{rbf}} \otimes K$ in Definition 4.4.1. There are at least two reasons for this choice.

The first reason in theoretical. With a direct sum, Theorem 4.4.6 is not valid anymore (although it would work with affine spaces instead of vector spaces). In particular, results in relation with the common types of prior knowledge presented in Section 4.4.2 would

not be valid anymore.

The second reason is practical. Using a direct sum creates the question of the relative weights attributed to the RBF and non-RBF parts of the kernel, *i.e.* $K_a = \lambda K_{\text{rbf}} \oplus (1-\lambda)K$ which introduces an additional learning parameter making the use of pRBF kernel much less practical.

4.4.2 Polynomial and monomial correlation

In this section, we investigate the use of monomials and polynomials in order to incorporate specific prior-knowledge into pRBF kernels. Practical cases corresponding to this situation are not rare as described in the introduction of Section 4.4 or as shown in the example based on real biological data in Section 5.5.

First, lets introduce a few notations.

Definition 4.4.8. *Real polynomial functions*

Let $n \in \mathbb{N}$ and $N \in \mathbb{N}$.

- $\mathbb{R}_n[x] = \{\sum_{i=0}^{n} p_i x^i | \forall i,\ p_i \in \mathbb{R}\}$ is the set of *polynomial functions in x of degree at most n with coefficients in $\mathbb{R}$.*
- $\mathbb{R}[x] = \bigcup_{i=0}^{\infty} \mathbb{R}_i[x]$ is the set of *polynomial functions in x with coefficients in $\mathbb{R}$.*
- $\mathbb{R}_n[x_1, \ldots, x_N] = \{\sum_{i_1+\ldots+i_N \leq n} p_{i_1,\ldots,i_N} \prod_{k=1}^{N} x_k^{i_k} | \forall i_1 + \ldots + i_N \leq n,\ p_{i_1,\ldots,i_N} \in \mathbb{R}\}$ is the set of *multivariate polynomial functions in $x_1, \ldots, x_N$ of degree at most n with coefficients in $\mathbb{R}$.*
- $\mathbb{R}[x_1, \ldots, x_N] = \bigcup_{i=0}^{\infty} \mathbb{R}_i[x_1, \ldots, x_N]$ is the set of *multivariate polynomial functions in $x_1, \ldots, x_N$ with coefficients in $\mathbb{R}$.*

Remark 4.4.9. We make an abuse of notations by using the polynomial expressions to designate the corresponding polynomial functions.

The above structures are vector spaces over $\mathbb{R}$. Therefore, Theorem 4.4.6 is applicable when the non-RBF portion of the kernel is a univariate or multivariate polynomial.

However, most of the commonly available prior-knowledge on feature-label correlation patterns translate well into relations involving simple monomials rather than more complex polynomials.

For instance, knowing that the label is linearly (*e.g.* surface to price of a property in real-estate), quadratically (*e.g.* speed to energy in physics) or cubically (*e.g.* radius to volume in geometry) correlated with a specific feature x_{i_0} requires the model $\hat{f}$ to be a univariate monomial of corresponding degree *w.r.t.* to x_{i_0}.

Multivariate monomials are also sufficient for more elaborate correlations involving several features (*e.g.* weight is the product of density and volume). Hence, pRBF kernels should mainly be used with monomial expressions rather than polynomial expressions.

Definition 4.4.10. *Real monomial functions*

Let $n \in \mathbb{N}$ and $N \in \mathbb{N}$.

- $m\mathbb{R}_n[x] = \{p_i x^i | i \in [\![0, n]\!] \wedge p_i \in \mathbb{R}\}$ is the set of *monomial functions in x of degree at most n with coefficients in $\mathbb{R}$.*
- $m\mathbb{R}[x] = \bigcup_{i=0}^{\infty} m\mathbb{R}_i[x]$ is the set of *monomial functions in x with coefficients in $\mathbb{R}$.*
- $m\mathbb{R}_n[x_1, \ldots, x_N] = \{p_{i_1,\ldots,i_N} \prod_{k=1}^{N} x_k^{i_k} | i_1 + \ldots + i_N \leq n \wedge p_{i_1,\ldots,i_N} \in \mathbb{R}\}$ is the set of *multivariate monomial functions in $x_1, \ldots, x_N$ of degree at most n with coefficients in $\mathbb{R}$.*
- $m\mathbb{R}[x_1, \ldots, x_N] = \bigcup_{i=0}^{\infty} m\mathbb{R}_i[x_1, \ldots, x_N]$ is the set of *multivariate monomial functions in $x_1, \ldots, x_N$ with coefficients in $\mathbb{R}$.*

Unfortunately, those structures are **not** vector spaces over $\mathbb{R}$. On one hand, $m\mathbb{R}_n[x]$ and $m\mathbb{R}_n[x_1, \ldots, x_N]$ are not vector spaces since they do not contain 0 (the neutral element of the addition). On the other hand, $m\mathbb{R}[x]$ and $m\mathbb{R}[x_1, \ldots, x_N]$ are not vector spaces since they contain 1 and x but not $1+x$. As a consequence, Theorem 4.4.6 cannot be applied to these structures.

Fortunately, this problem can be circumvented in the following fashion.

Definition 4.4.11. *Real monomial functions of degree exactly n*

Let n, $n_1, \ldots, n_N$ and N be elements of $\mathbb{N}$.

- $e\mathbb{R}_n[x] = \{px^n | p \in \mathbb{R}^*\}$ is the set of *monomial functions in x of degree exactly n with coefficients in $\mathbb{R}$.*

- $e\mathbb{R}_{n_1,\ldots,n_N}[x_1,\ldots,x_N] = \{p\prod_{k=1}^{N} x_k^{n_k} | p \in \mathbb{R}^*\}$ is the set of *multivariate monomial functions in* $x_1,\ldots,x_N$ *of respective partial-degrees exactly* $n_1,\ldots,n_N$ *with coefficients in* $\mathbb{R}$.

Note that for $n \geq 0$, $e\mathbb{R}_n[x]$ and $e\mathbb{R}_n[x_1,\ldots,x_N]$ do not contain 0 and are therefore not vector spaces yet. This can be solved by simply adding 0 to the respective structures as in the following rather trivial theorem.

Theorem 4.4.12.

Let n, $n_1,\ldots,n_N$ *and* N *be elements of* $\mathbb{N}$.

$e\mathbb{R}_n[x] \cup \{0\}$, *and* $e\mathbb{R}_{n_1,\ldots,n_N}[x_1,\ldots,x_N] \cup \{0\}$ *are vector spaces over* $\mathbb{R}$.

Proof. $e\mathbb{R}_n[x] \cup \{0\} \subset \mathbb{R}_n[x]$ and $\mathbb{R}_n[x]$ is a vector space over $\mathbb{R}$. It is therefore sufficient to prove that $e\mathbb{R}_n[x] \cup \{0\}$ is a vector subspace of $\mathbb{R}_n[x]$, *i.e.* that it is non-empty and stable by linear combination.

$0 \in e\mathbb{R}_n[x] \cup \{0\}$, thus $e\mathbb{R}_n[x] \cup \{0\}$ is not empty. Let $\lambda \in \mathbb{R}$ and $(P,Q) \in (e\mathbb{R}_n[x] \cup \{0\})^2$ *i.e.* $P = px^n$ with $p \in \mathbb{R}$ and $Q = qx^n$ with $q \in \mathbb{R}$.

- $\lambda.P = \lambda.px^n = (\lambda p)x^n$ with $(\lambda p) \in \mathbb{R}$, therefore $\lambda.P \in e\mathbb{R}_n[x] \cup \{0\}$.
- $P + Q = px^n + qx^n = (p+q)x^n$ with $(p+q) \in \mathbb{R}$, therefore $P + Q \in e\mathbb{R}_n[x] \cup \{0\}$.

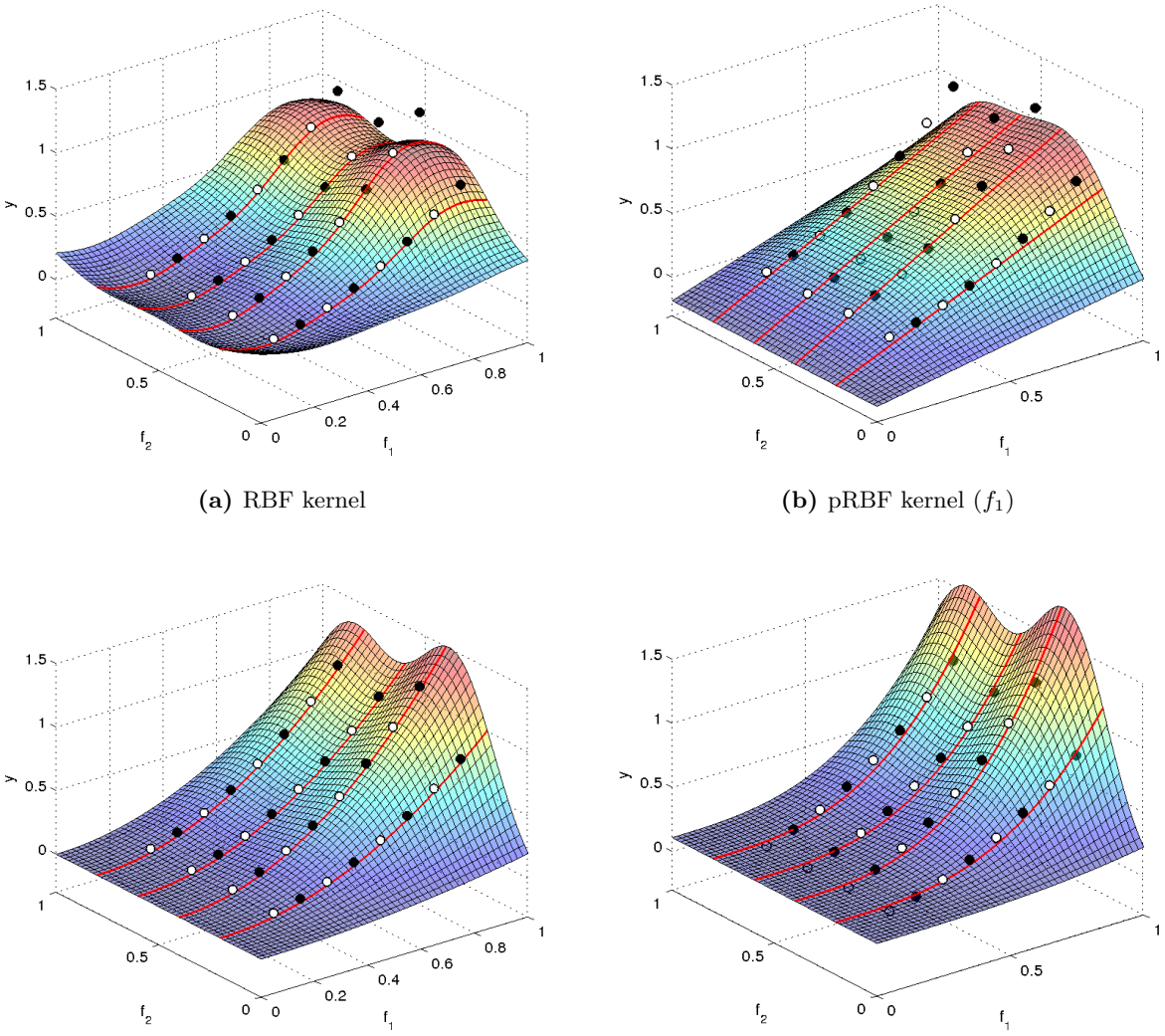
Therefore $e\mathbb{R}_n[x] \cup \{0\}$ is closed *w.r.t.* the monomial sum and the scalar multiplication and $e\mathbb{R}_n[x] \cup \{0\}$ is a vector space as a vector subspace of $\mathbb{R}_n[x]$.

The proof for $e\mathbb{R}_{n_1,\ldots,n_N}[x_1,\ldots,x_N] \cup \{0\}$ can be done in a similar fashion. □

The consequence of Theorem 4.4.6 and Theorem 4.4.12 is that if the non-RBF portion of a pRBF kernel is a univariate or multivariate monomial *w.r.t.* to certain features, then the resulting labeling model $\hat{f}$ is also a monomial of the same degree *w.r.t.* the same features (including the degenerate case its coefficient is equal to 0).

Figure 4.6 proposes a graphical illustration of regression with the ϵ-SVR+pRBF combination. The feature space is 2-dimensional with features f_1 and f_2. In this example, the label has a quadratic correlation *w.r.t.* f_1. When the standard RBF kernel is used (Figure 4.6a), the resulting decision model fits the training data (white dots) but not the test data (black dots). Using a pRBF kernel with monomials in f_1 (Figure 4.6b

to Figure 4.6d) causes the decision model to have the properties predicted by Theorem 4.4.6 as shown by the level curves *w.r.t.* f_2. Most importantly the pRBF kernel using the monomial f_1^2, *i.e.* making the correct assumption about the model, can label all the test data correctly including the data out of the range of the training data. Such a generalizability of the model outside of the range of the training data is usually not expected from SVMs.



(a) RBF kernel

(b) pRBF kernel (f_1)

(c) pRBF kernel (f_1^2 – correct assumption)

(d) pRBF kernel (f_1^3)

Figure 4.6: Examples of regression with the ϵ-SVR+pRBF combination. The data is 3-dimensional with 2 features (f_1, f_2) and 1 output label y. For f_2 fixed, y is proportional to f_1^2, *i.e.* the correlation between f_1 and y is quadratic. The training data points are indicated with white dots and the test data points with black dots. The red curves drawn on the decision surface are level curves *w.r.t.* f_2. Each graph corresponds to a different monomial expression: f_1 for (b), f_1^2 for (c) and f_1^3 for (d). (a) corresponds to the standard RBF kernel.

Remark 4.4.13. The framework can be extended to non-integer exponents in order to

take into account other types of correlations such as roots (with $n = \frac{1}{2}$ for square roots, $n = \frac{1}{3}$ cube roots...). Corresponding precautions must then be taken regarding the domains of definition of features.

4.4.3 Monotonic correlation

pRBF kernels can also deal with monotonicity *w.r.t.* specific features, a weaker and more common form of prior-knowledge.

For $n \in \mathbb{N} - 2\mathbb{N}$ (*i.e.* n odd), the set $e\mathbb{R}_n[x]$ of univariate monomials of degree exactly n presented in Definition 4.4.11 only contains strictly monotonic functions. Therefore $e\mathbb{R}_n[x] \cup \{0\}$ contains only monotonic functions for $n \in \mathbb{N} - 2\mathbb{N}$. In a similar way, multivariate monomials are also monotonic *w.r.t.* the variables for which the partial degree is odd (*e.g.* the degree of x_2 in $P = x_1^2 x_2^3$ is $3 \in \mathbb{N} - 2\mathbb{N}$, hence P is monotonic *w.r.t.* x_2).

Without any additional knowledge, it is therefore reasonable to use monomials of degree 1 (*i.e.* linear) for the features *w.r.t.* which we want the labeling model to be monotonic.

4.5 gRBF kernel

The gRBF kernel, standing for "generalized RBF kernel", is a generalization of the standard RBF kernel from $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ to $\mathfrak{P}(\mathbb{R}^n) \times \mathfrak{P}(\mathbb{R}^n) \to \mathbb{R}$, *i.e.* from points of the feature space to sets of the feature space. The gRBF kernel treats data and prior-knowledge without distinction.

The gRBF kernel can be used to incorporate prior-knowledge about labeled regions of the feature space, *i.e.* make hypothesis about the labels of specific regions of the feature space. A labeled set can be interpreted as an average label value over a region and can be used to compensate for missing data.

Visual examples are given throughout the section to illustrate the different steps and notions involved in the utilization of the gRBF kernel. An example of application of the gRBF kernel on real-life data is proposed in Section 5.6.

Section 4.5.1 provides a formal definition for the gRBF kernel. Section 4.5.2 describes how to create a single training set from labeled data points and prior-knowledge while

dealing with eventual conflicts. Section 4.5.3 presents the new technical challenges associated to the gRBF kernel and how to deal with them. Finally, Section 4.5.4 summarizes the workflow associated to the use of the gRBF kernel.

4.5.1 Definitions

Formally, the gRBF kernel is obtained by replacing the usual Euclidean distance between elements of $\mathbb{R}^n$ in the expression of standard RBF kernel with a distance between sets of $\mathbb{R}^n$.

Definition 4.5.1. *Set distance*

The distance $d(\mathcal{A}, \mathcal{B})$ between the sets $\mathcal{A} \in \mathfrak{P}(\mathbb{R}^n)$ and $\mathcal{B} \in \mathfrak{P}(\mathbb{R}^n)$ is defined as:

$$d(\mathcal{A}, \mathcal{B}) = \begin{cases} \inf_{a \in \mathcal{A} \wedge b \in \mathcal{B}} \|a - b\|_2 & \text{if } \mathcal{A} \neq \emptyset \text{ and } \mathcal{B} \neq \emptyset \\ \infty & \text{otherwise} \end{cases} \tag{4.28}$$

Note that the set distance is a well-defined notion. Indeed, if $\mathcal{A} \neq \emptyset$ and $\mathcal{B} \neq \emptyset$, then $\{\|a - b\|_2 | a \in \mathcal{A} \wedge b \in \mathcal{B}\}$ is a non-empty subset of $\mathbb{R}$ with 0 as a lower bound, and therefore has a unique infimum.

Remark 4.5.2. The set distance is **not** a metric. In particular, it does not satisfy the triangular inequality. For instance, with $\mathcal{X} = \mathbb{R}$: $d(\{1\}, \{4\}) = 3$, $d(\{1\}, [2, 3]) = 1$, and $d([2, 3], \{4\}) = 1$. Therefore $d(\{1\}, \{4\}) > d(\{1\}, [2, 3]) + d([2, 3], \{4\})$, which contradicts the triangular inequality.

Definition 4.5.3. *gRBF kernel*

The gRBF kernel with parameter $\gamma > 0$ is the function:

$$\begin{array}{rccl} K_{\text{grbf}}: & \mathfrak{P}(\mathbb{R}^n)^2 & \to & \mathbb{R} \\ & (\mathcal{A}, \mathcal{B}) & \mapsto & \exp(-\gamma d(\mathcal{A}, \mathcal{B})^2) \end{array}$$

4.5.2 Dataset creation

The gRBF kernel deals with data points and prior-knowledge together as elements of $\mathfrak{P}(\mathbb{R}^n)$ without a particular distinction. This section describes the creation of the dataset

from the two heterogeneous types of input.

First, Section 4.5.2.1 illustrates how commonly available prior-knowledge can lead to the creation of labeled sets. Then, Section 4.5.2.2 describes how the usual data points and the labeled sets originating from the prior-knowledge are combined together into a single dataset. The contradictions occurring between data points and prior-knowledge can sometimes produce adverse effects. Section 4.5.2.3 proposes a way to deal with such conflicts during the creation of the dataset.

4.5.2.1 Using labeled sets as prior-knowledge

A labeled region is a pair $(\mathcal{X}_0, y_0)$ where $\mathcal{X}_0 \in \mathfrak{P}(X)$ and $y_0 \in \mathbb{R}$. Therefore, defining a labeled region requires 2 types of information: a subset $\mathcal{X}_0$ of $\mathcal{X}$ and a label value y_0. The region $\mathcal{X}_0$ of the feature space is typically derived from prior-knowledge about bounds and ranges on specific features. The label y_0 can be viewed an average label value for the data points within this regions. In this regard, labeled regions correspond to a more elaborate type of prior-knowledge than the unlabeled regions presented in Section 4.3.1 which do not contain any hypothesis on the label space. Defining labeled sets is particularly useful in order to improve the quality of the decision model over regions where data is scare or entirely missing.

The most common way of obtaining labeled regions is via external advice from an expert. For instance, in a simplistic computer vision example using morphological features to distinguish apples from bananas, a botanist might provide the information that an object having a total length $l \geq 20$cm is systematically in the banana-class (with label $+1$) and never in the apple-class (with label -1). This results in a labeled set $(\mathcal{O}_1, +1)$ where $\mathcal{O}_1$ is the half-space for which $l \geq 20$cm. In another regression example involving the prediction of daily rainfall in the Indian city of Bhopal, past monthly records indicate that virtually no rainfall is expected from January to April. This suggests the construction of the labeled set $(\mathcal{O}_2, 0)$ where $\mathcal{O}_2$ is the set of dates for which the value of the "month" feature is either "January", "February", "March" or "April".

The gRBF kernel enables training from prior-knowledge only without any training data points. Indeed, the gRBF kernels treat data points and labeled regions without distinction, therefore, prior-knowledge constitutes valid training data. Unlike the ξRBF

kernels with unlabeled regions from Section 4.3.1 which need at least a training data point for every class, gRBF kernels can be used with labeled sets alone. Figure 4.7 provides a visual illustration of a binary classification and a scalar regression performed without training data points.

Practical examples of gRBF kernels using different types of labeled regions are available in Section 5.6.

4.5.2.2 Combining data and prior-knowledge

The next task consists in creating a single dataset by merging the following two heterogeneous types of input:

- the usual labeled training data set $\mathcal{S}_d = (x_i, y_i)_{i=1,\dots,N_d} \in (\mathbb{R}^n \times \mathbb{R})^{N_d}$ of N_d input-output pairs;
- a set $\mathcal{S}_k = (\mathcal{X}_i, y'_i)_{i=1,\dots,N_k} \in (\mathfrak{P}(\mathbb{R}^n) \times \mathbb{R})^{N_k}$ of N_k labeled regions corresponding to problem-specific prior-knowledge.

Typically, $N_k < N_d$ but this is not required.

$\mathcal{S}_d$ can trivially be transformed so that the whole training data has values in $\mathfrak{P}(\mathbb{R}^n) \times \mathbb{R}$ by taking singletons of the feature vectors:

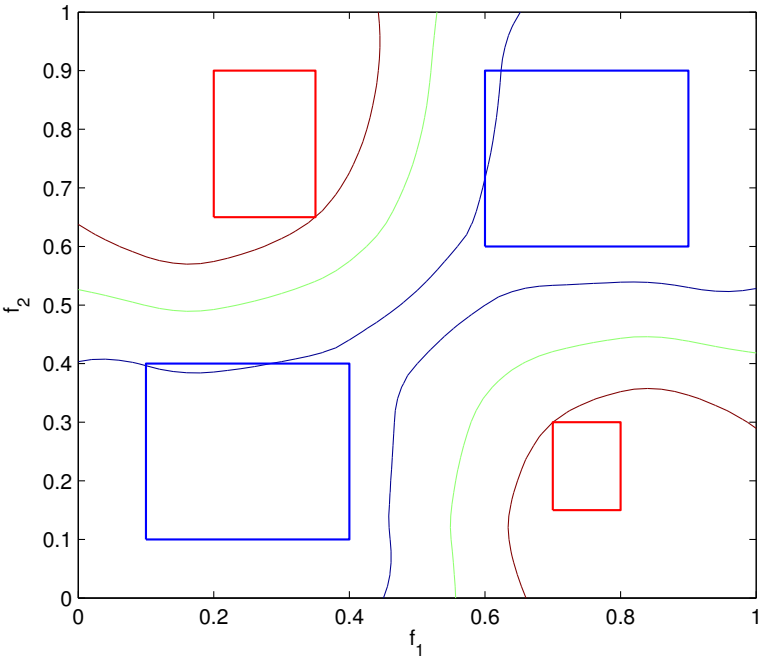
$$\tilde{\mathcal{S}}_d = (\{x_i\}, y_i) \tag{4.29}$$

The homogeneous dataset $\tilde{\mathcal{S}}_d \cup \mathcal{S}_k$ can then be used to train an SVM+gRBF combination in a similar way an SVM+RBF combination would use labeled data points.

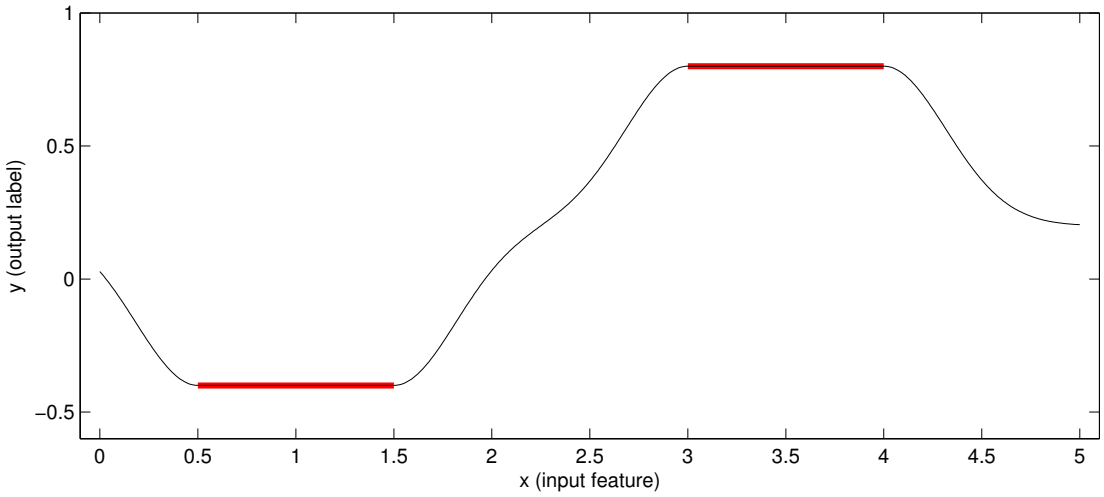
Remark 4.5.4. If $\mathcal{S}_k = \emptyset$, the gRFB kernel is equivalent to the standard RBF kernel.

Figure 4.8 is a visual example of binary classification with the C-SVM+gRBF combination. The labeled regions produce the intended effect on the decision boundary. Figure 4.8d is an example of conflict that can occur between the data and the prior-knowledge: a data point from the "red" class conflicts with a labeled region from the the "blue" class. In this particular case, the SVM finds a reasonable decision boundary which classifies the data point correctly and still takes the labeled region into account.

Figure 4.9 is a visual example of regressions using the ϵ-SVM+RBF combination. Different values of the kernel bandwidth parameter γ have been tested. We can see

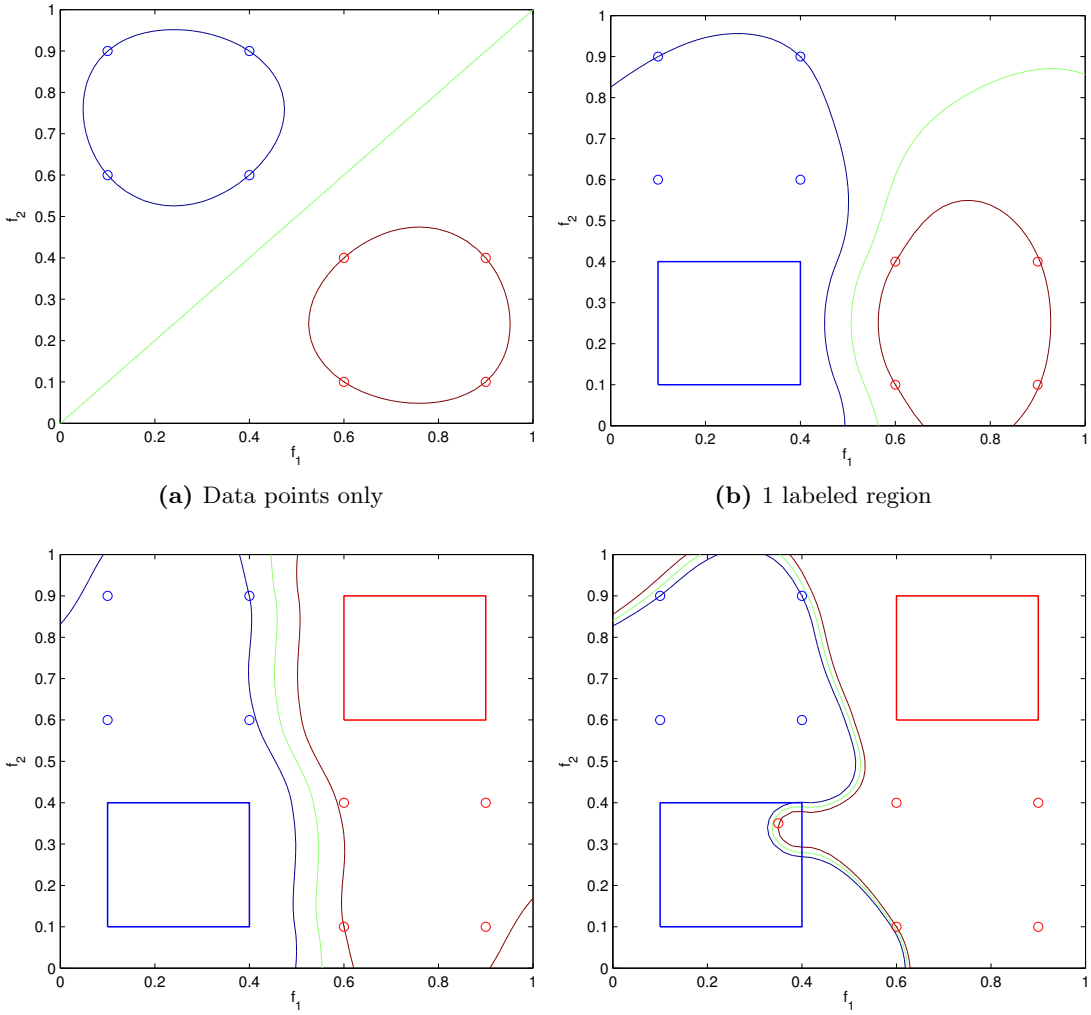


(a) C-SVM+gRBF, no training data



(b) ϵ-SVR+gRBF, no training data

Figure 4.7: Decision models obtained from labeled regions alone without training data. (a) is a binary classification problem with 2 features f_1 and f_2. The red and blue boxes indicate the labeled regions belonging to different classes. The green line indicates the decision boundary and the red and blue lines the SVM margin. (b) is a regression problem with a single feature x. The red segments represent the labeled regions.



(a) Data points only

(b) 1 labeled region

(c) 2 labeled regions

(d) 2 labeled regions with conflict

Figure 4.8: Example of binary classification with the C-SVM+gRBF combination on 2-dimensional data. Training data from the 2 classes are represented with red and blue circles. Labeled regions are represented with red and blue rectangles according to their label. The decision boundary is represented in green and the margin by the 2 adjacent red and blue curves.

that the data points have a local influence whereas the labeled regions have a more spread-out influence (this is particularly obvious with large values of γ). Figure 4.9d contains a conflict between data and prior-knowledge. Unlike for the previous example in Figure 4.8d, we can see that the decision function has a very erratic behavior which requires fixing.

Remark 4.5.5. Erratic behaviors such as in Figure 4.9d are caused by the conjugation of 2 different factors: conflicts between data and prior-knowledge (treated in Section 4.5.2.3), and the fact that gRBF kernels are non-PD (treated in Section 4.5.3.1) causing the optimization process to stop at a local optimum. Dealing with just a single one of the causes usually solves the problem as shown in the respective sections.

4.5.2.3 Resolution of conflicts

In this section, we propose a way to solve conflicts between data and prior-knowledge. Conflicts occur when there is $i_1 \in [\![1, N_d]\!]$ and $i_2 \in [\![1, N_k]\!]$ such that $x_{i_1} \in \mathcal{X}_{i_2}$ with $y_{i_1} \neq y'_{i_2}$. Then, the data point x_{i_1} is in contradiction with the labeled region $\mathcal{X}_{i_2}$ from the prior-knowledge. As seen on Figure 4.9d, conflicts may cause the decision function to behave strangely.

The proposed solution involves a transformation of the labeled regions of $\mathcal{S}_k$ in order to "avoid" the data samples in $\mathcal{S}_d$ by "drilling holes" into them. The objective of the KE-RBF framework is to use prior-knowledge in order to compensate for insufficient data rather than for incorrect data. Therefore, it is a reasonable approach to modify the prior-knowledge which is general and more approximative than the data carrying specific and therefore more precise information.

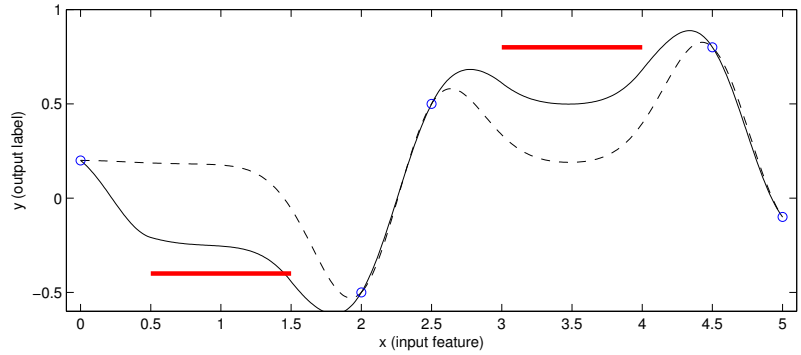
The "holes" created in the labeled sets are topological open balls.

Definition 4.5.6. *Open ball in* $\mathbb{R}^n$

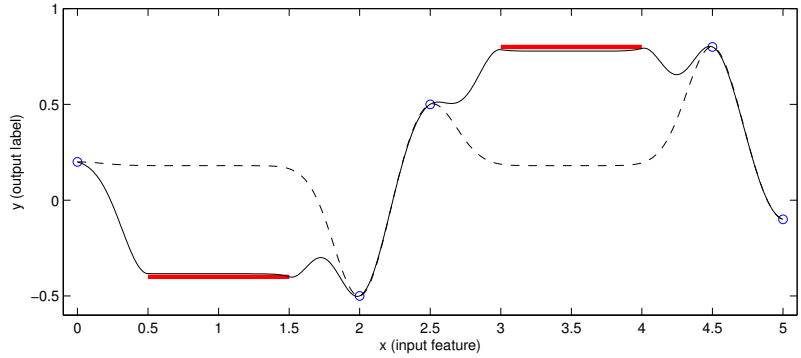
Let $x_0 \in \mathbb{R}^n$ and $\rho > 0$. The *open ball* with center x_0 and radius ρ is the set defined as:

$$B(x_0, \rho) = \{x \in \mathbb{R}^n | \|x - x_0\|_2 < \rho\} \tag{4.30}$$

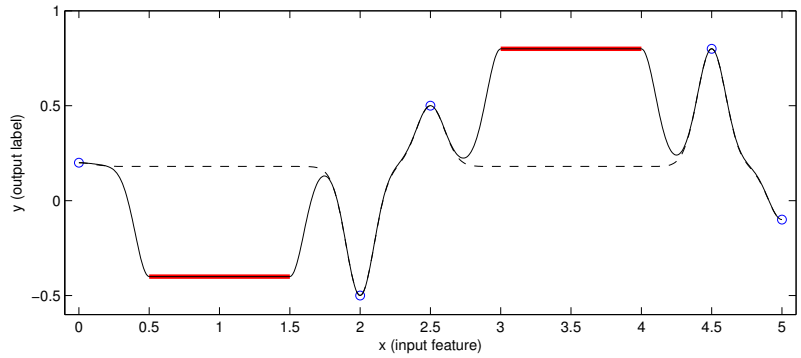
We denote with $\mathcal{B}_\rho = \bigcup_{i=1}^{N_d} B(x_i, \rho)$ the set of all the open balls with radius ρ centered on every training data point. The idea is to remove $\mathcal{B}_\rho$ from every labeled region in $\mathcal{S}_k$.



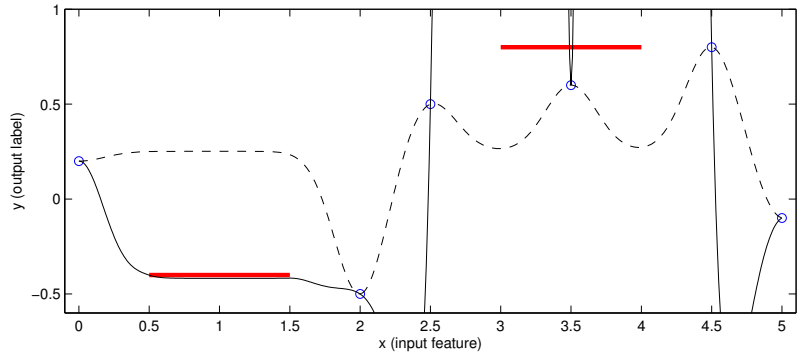
(a) $\gamma = 5$



(b) $\gamma = 15$



(c) $\gamma = 50$



(d) $\gamma = 15$ with conflict

Figure 4.9: Example of 1-dimensional regression with the ϵ-SVR+gRBF combination (continuous line). Training data are represented with blue circles. Labeled regions are represented thick red lines. The regression obtained with the standard RBF kernel without labeled regions is given as a reference (dashed line).

Therefore, we get a modified set of labeled regions:

$$\tilde{\mathcal{S}_k} = (\mathcal{X}_i - \mathcal{B}_\rho, y'_i)_{i=1,\ldots,N_k} \tag{4.31}$$

The full training set containing data and knowledge is $\mathcal{S} = \tilde{\mathcal{S}_t} \cup \tilde{\mathcal{S}_k}$. The kernel Gram matrix is then computed from the training set $\mathcal{S}$ like for any standard kernel over $\mathbb{R}^n$.

Figure 4.10 shows how choosing an adequate value for ρ solves the problem caused by conflicts. When ρ becomes larger, the erratic behavior of the decision model is attenuated and becomes consistent with the data and the prior-knowledge. An empirical study on the ρ parameter is available in the example of application in Section 5.6.

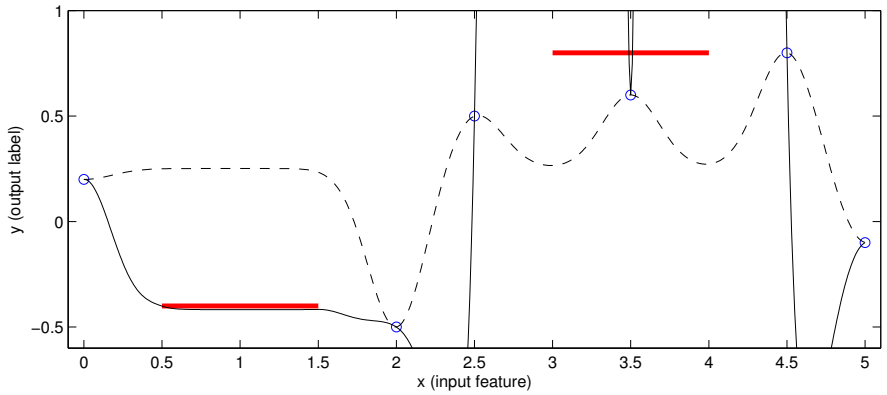
ρ is a new learning parameter which implications might not be transparent. We propose an alternative approach for setting ρ a priori, without resorting to computationally intensive methods such as a grid-search during the learning phase. This is achieved by specifying the maximal collinearity allowed between a labeled data sample and a labeled region. The value of the gRBF kernel product varies between 0 for orthogonal (*i.e.* unrelated) objects and 1 for perfectly collinear objects (*i.e.* similar objects). Therefore, if we want the collinearity between a labeled data sample x and a labeled region $\mathcal{X}$ to be limited to a fraction $0 < p \leq 1$ of the maximal value:

$$\begin{aligned} K_{\text{grbf}}(\mathcal{X}, \{x\}) \leq p &\iff \exp(-\gamma d(\mathcal{X}, \{x\})^2) \leq p \\ &\iff -\gamma d(\mathcal{X}, \{x\})^2 \leq \ln(p) \\ &\iff \gamma d(\mathcal{X}, \{x\})^2 \geq \ln(\frac{1}{p}) \\ &\iff d(\mathcal{X}, \{x\})^2 \geq \frac{1}{\gamma} \ln(\frac{1}{p}) \end{aligned} \tag{4.32}$$

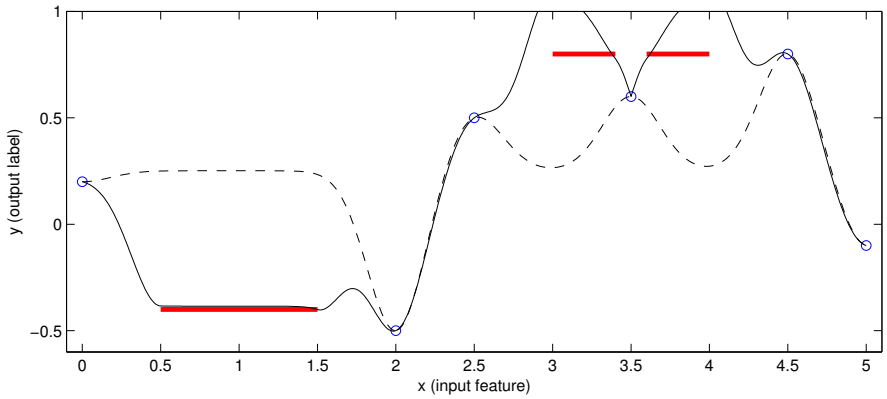
And since $d(\mathcal{X}, \{x\}) \geq \rho$, it is sufficient to take:

$$\rho^2 = \frac{1}{\gamma} \ln(\frac{1}{p}) \iff \rho = \sqrt{\frac{1}{\gamma} \ln(\frac{1}{p})} \tag{4.33}$$

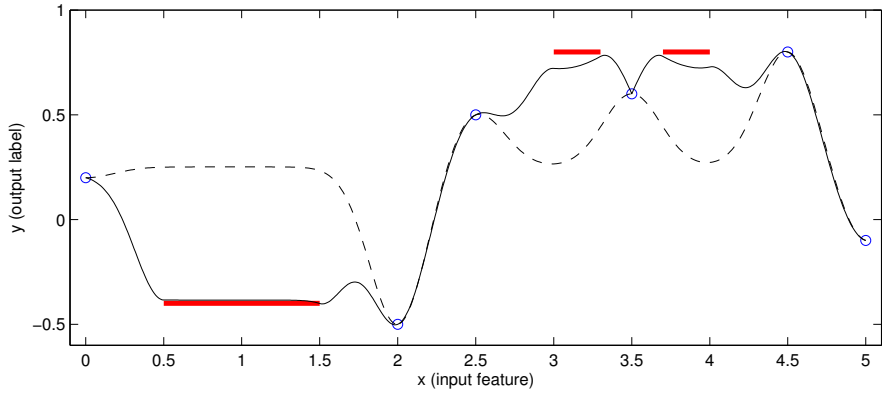
Therefore, with this method, the value of ρ depends the value of the kernel bandwidth parameter γ. Table 4.2 gives reference values for ρ according to the p chosen.



(a) $\rho = 0$



(b) $\rho = 0.1$



(c) $\rho = 0.2$

Figure 4.10: Effects of ρ on the labeled regions and the decision model ($\gamma = 15$ for all the models). Conventions are the same as for Figure 4.9.

p	ρ
0	∞
0.01	$\frac{2.1460}{\sqrt{\gamma}}$
0.1	$\frac{1.5174}{\sqrt{\gamma}}$
0.2	$\frac{1.2686}{\sqrt{\gamma}}$
0.3	$\frac{1.0973}{\sqrt{\gamma}}$
0.4	$\frac{0.9572}{\sqrt{\gamma}}$
0.5	$\frac{0.8326}{\sqrt{\gamma}}$
1	0

Table 4.2: Values for ρ corresponding to different values of p.

4.5.3 Computational challenges

gRBF kernels bring a number of new challenges of computational order for which solutions must be proposed. First, gRBF kernels are not PD kernels causing SVM solvers to return local optima instead of global ones (Section 4.5.3.1). Moreover, computing the set distance is not a trivial problem (Section 4.5.3.2). Finally, the computational complexity of computing a Gram matrix is higher with the gRBF kernel (Section 4.5.3.3).

4.5.3.1 Non-positive kernels: a spectral approach

gRBF kernels are not PD kernels as shown in the following minimal example.

Example 4.5.7. Let $n = 1$, $\gamma = 1$ and $\rho = 0$ (*i.e.* we ignore conflicts between data and prior-knowledge). The gRBF kernel Gram matrix for the sets $\mathcal{X}_1 = \{-1\}$, $\mathcal{X}_2 = \{1\}$ and $\mathcal{X}_3 = [-0.5, 0.5]$ is:

$$M = \begin{pmatrix} 1 & e^{-4} & e^{-0.25} \\ e^{-4} & 1 & e^{-0.25} \\ e^{-0.25} & e^{-0.25} & 1 \end{pmatrix} \tag{4.34}$$

The eigenvalues of the matrix are roots of the characteristic polynomial in λ:

$$\det(M - \lambda I) = \begin{vmatrix} 1-\lambda & e^{-4} & e^{-0.25} \\ e^{-4} & 1-\lambda & e^{-0.25} \\ e^{-0.25} & e^{-0.25} & 1-\lambda \end{vmatrix} \tag{4.35}$$

which are approximately $\lambda_1 = 2.1106$, $\lambda_2 = 0.9817$ and $\lambda_3 = -0.0923$. We notice that

$\lambda_3 < 0$ and therefore a gRBF kernel is not a PD kernel.

Since $\lambda_1\lambda_3 < 0$ (*i.e.* it has eigenvalues of opposite signs), a gRBF kernel is an indefinite kernel.

Non-positive kernels pose two different issues. The first one is a problem of computational order since the resulting optimization problem is not convex anymore. The second one is more theoretical. Non-positive kernels do not entail the existence of a RKHS. Therefore, essential results such as the Moore-Aronszajn theorem or the representer theorem cannot be used to justify the statistical soundness of SVMs as done in Chapter 2 with PD kernels.

Nevertheless, the use of non-positive kernels with SVMs is increasingly popular and various solutions have been proposed to overcome the first issue. The simplest solution is to passively deal with the problem and to solve the non-convex problem with the standard SVM solvers. Sometimes, this can work well as in [27, 93] or in the example in Figure 4.8. However, the SVM solver will return a local optimum which is not guaranteed to be a global one. Therefore, the quality of the solution may be very unstable as in Figure 4.9 and this solution is not recommended.

Solutions actively dealing with this problem have also been proposed. New types of SVMs or solvers in order to deal with non-positive kernels have been proposed [41, 54]. However, those solutions are not strictly kernel-based approaches and give up on the use of standard SVMs.

Other solutions working on direct transformations of the kernel Gram matrix are more in-line with our purpose. In particular, there are different ways of turning the kernel Gram matrix into a positive semi-definite matrix using the eigenvalue decomposition of the original matrix. Wu et al. [94] propose an empirical study of those methods.

Being symmetrical, a kernel Gram matrix K admits the following eigenvalue decomposition:

$$K = U\mathrm{diag}(\lambda_1, \ldots, \lambda_N)U^T \tag{4.36}$$

where N is the size of the input data, U is an orthogonal matrix and $\mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ is the diagonal matrix of the eigenvalues $\lambda_1, \ldots, \lambda_N$ some of which may be negative.

Wu et al. [94] found 2 methods to work particularly well: *flipping* and *shifting*.

Flipping consists in taking the opposite of negative eigenvalues. Accordingly, the "flipped" kernel Gram matrix is:

$$\text{flip}(K) = U\text{diag}(|\lambda_1|, \ldots, |\lambda_N|)U^T \tag{4.37}$$

Shifting consists in adding $\eta > 0$ to each of the eigenvalues in order to make them positive. Usually, the minimal value for η is chosen, *i.e.* $\eta = -\min_{i=1,\ldots,N} \lambda_i$. Therefore, the "shifted" kernel Gram matrix is:

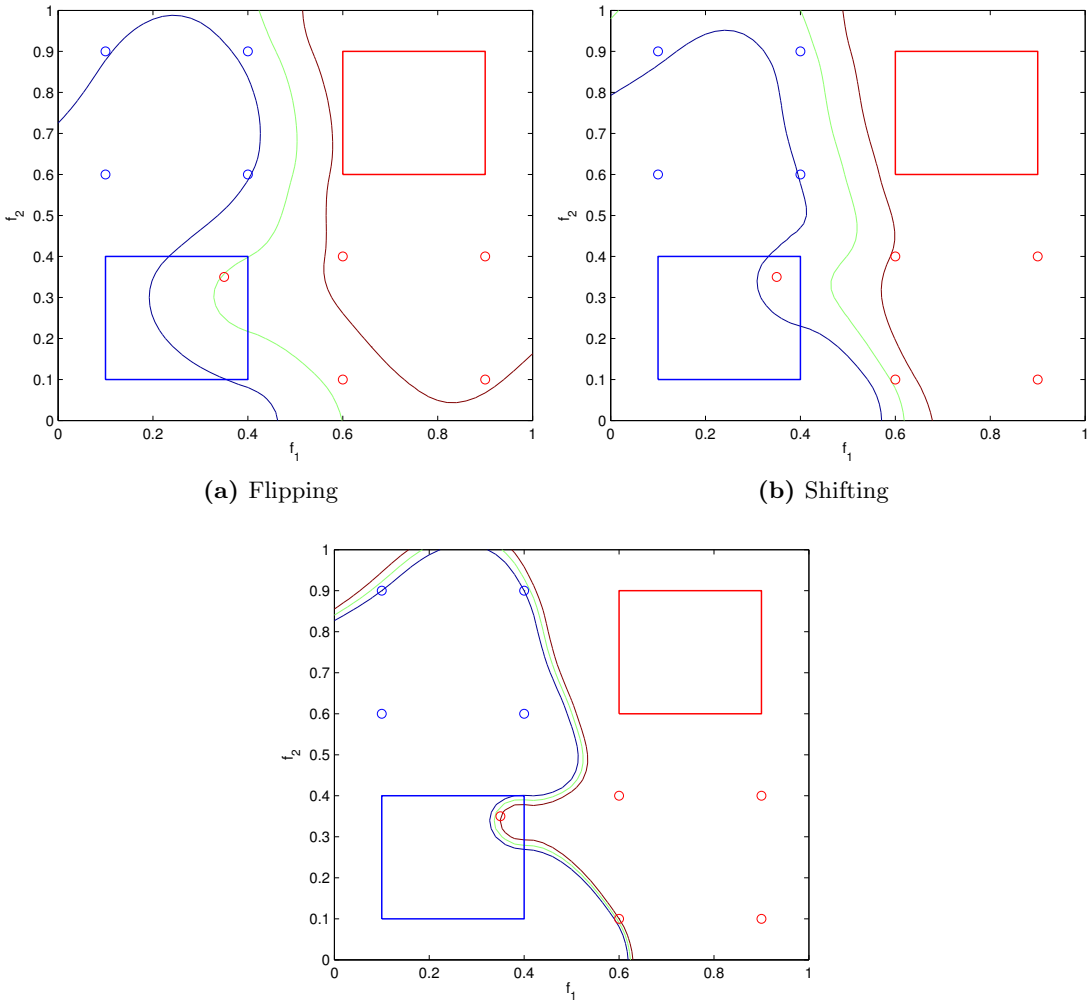
$$\text{shift}(K) = U\text{diag}(\lambda_1 + \eta, \ldots, \lambda_N + \eta)U^T \tag{4.38}$$

with $\eta = -\min_{i=1,\ldots,N} \lambda_i$.

Figure 4.11 shows the effects of applying flipping and shifting on the classification example used in Figure 4.8d. We can see that both methods have the effect of smoothing out the decision boundary. In this case, the results of flipping is clearly more desirable than shifting. Figure 4.12 does the same for the regression example used in Figure 4.9d (though with a lower value of γ). Again, the decision model is smoother after transformation of the matrix and flipping appears to perform better than shifting. An empirical comparison of flipping and shifting, also suggesting the superiority of flipping, can be found in Section 5.6.3.

As pointed out by Wu et al. [94], flipped and shifted kernels have decreased generalization capabilities (*i.e.* they become less good at labeling data not seen in the training set) due to the transformation applying to the training data only. If the unlabeled data is available at training time, applying flipping or shifting on the kernel Gram matrix containing the full data (labeled and unlabeled) may improve generalizability at the expense of additional time required for computing and transforming the full matrix. A precise estimation of the additional cost as-well-as a method for keeping it minimal is proposed in Section 4.5.3.3.

An empirical study available in Section 5.6.4 shows that applying the transformation on the full data can indeed improve the results. However, the gains are rather marginal and may not be worth the overhead in computing time when speed is a critical aspect.

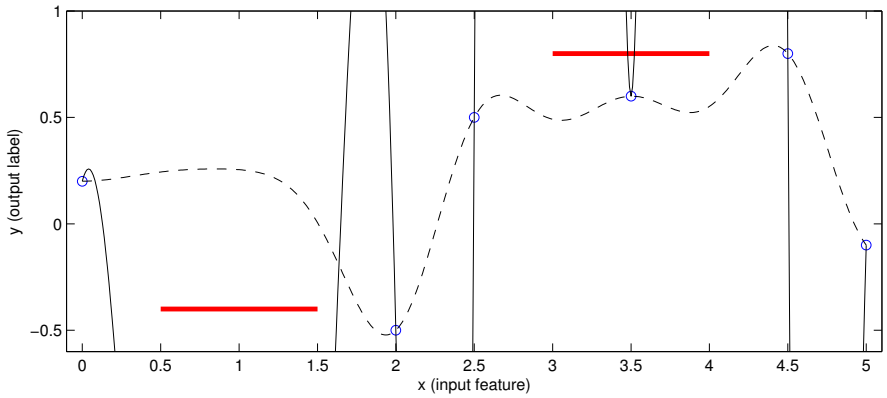


(a) Flipping

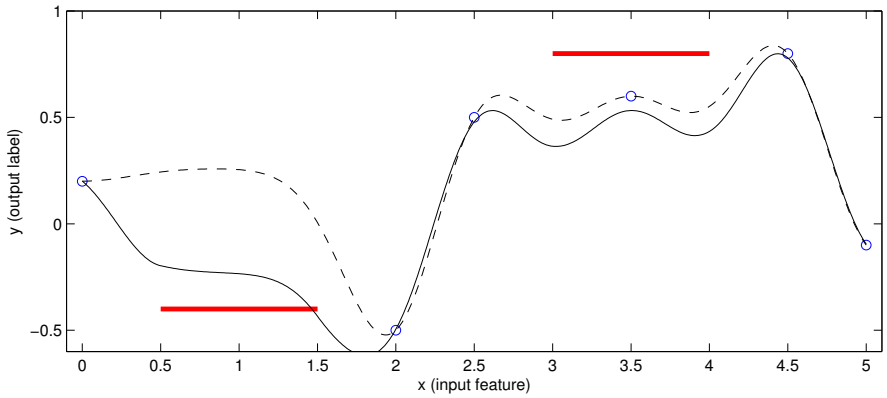
(b) Shifting

(c) No transformation

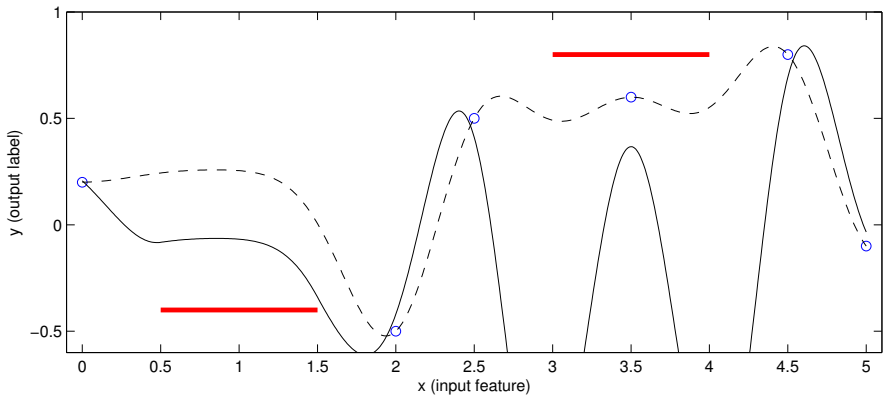
Figure 4.11: Shifting and flipping applied to the example from Figure 4.8d.



(a) No transformation



(b) Flipping



(c) Shifting

Figure 4.12: Shifting and flipping applied to the example from Figure 4.8d ($\gamma = 5$ and $\rho = 0$).

The second theoretical issue related to the use of indefinite kernels can be ignored in practice since it does not prevent the effective use of non-positive kernels. Therefore, it is more of a philosophical question than a practical hurdle. An element of answer may be that the theory of Reproducing Kernel Krein Spaces (RKKS) for non-positive kernels has results similar to the Moore-Aronszajn and representer theorems. [54] can be consulted for more details.

4.5.3.2 Computation of the set distance

Another computational challenge associated with the gRBF kernel is that there is no generic way of computing the set distance $d(\mathcal{A}, \mathcal{B})$ for arbitrary sets $\mathcal{A}$ and $\mathcal{B}$. Whether the set distance can be computed and how quickly it can be computed depends on the analytical expression of the sets. Therefore, it is necessary to restrict the labeled regions obtained from prior-knowledge to types of sets for which the set distance is easily computable.

In order of increasing computational complexity, we consider *balls*, *orthotopes* (better known as "hyperrectangles") and *convex polytopes*. Considering the way the prior-knowledge is usually obtained from ranges on the features, orthotopes are a good compromise between flexibility and computational complexity.

For each type of sets, 2 types of distances need to be computed: *set-to-set* distances for non-singleton sets corresponding to distance between 2 labeled regions and *set-to-singleton* distances corresponding to the distance between a labeled set and a data point.

Balls Open balls $B(x, r)$ are the topological structures introduced in Definition 4.5.6. They are characterized by their center x and their radius $r > 0$. The distance between two balls $\mathcal{B}_1 = B(x_1, r_1)$ and $\mathcal{B}_2 = B(x_2, r_2)$ is:

$$d(\mathcal{B}_1, \mathcal{B}_2) = \max(\|x_1 - x_2\|_2 - r_1 - r_2, 0) \tag{4.39}$$

The distance between a ball $\mathcal{B}_1 = B(x_1, r_1)$ and a singleton $\{x_2\}$ is:

$$d(\mathcal{B}_1, \{x_2\}) = \max(\|x_1 - x_2\|_2 - r_1, 0) \tag{4.40}$$

Remark 4.5.8. Choosing open balls or closed balls makes no difference.

Therefore, the set distance between balls and singletons is as quick to compute as the standard Euclidean distance between points. However, balls are of a limited practical use as they do not correspond to the way the prior-knowledge is commonly defined.

Orthotopes Orthotopes are a generalization of rectangles from 2 dimensions to n dimensions. They are fully characterized by $2n$ bounds: one lower bound l_i and one upper bound u_i for every dimension $i \in [\![1, n]\!]$.

Definition 4.5.9. *Orthotope*

Let $(l_i)_{i=1,\dots,n} \in \mathbb{R}^n$ and $(u_i)_{i=1,\dots,n} \in \mathbb{R}^n$ be such that $\forall i,\ l_i \leq u_i$. The orthotope of $\mathbb{R}^n$ with lower bounds $(l_i)_{i=1,\dots,n} \in \mathbb{R}^n$ and upper bounds $(u_i)_{i=1,\dots,n} \in \mathbb{R}^n$ denoted $R((l_i, u_i)_{i=1,\dots,n})$ is defined as:

$$R((l_i, u_i)_{i=1,\dots,n}) = \{(x_1, \dots, x_n) \in \mathbb{R}^n | \forall i,\ l_i \leq x_i \leq u_i\} \tag{4.41}$$

The distance between two othotopes $\mathcal{O}_1 = R((l_{i,1}, u_{i,1})_{i=1,\dots,n})$ and $\mathcal{O}_2 = R((l_{i,2}, u_{i,2})_{i=1,\dots,n})$ is given by:

$$d(\mathcal{O}_1, \mathcal{O}_2) = \sqrt{\sum_{i=1}^{n} \max(0, l_{i,2} - u_{i,1}, l_{i,1} - u_{i,2})^2} \tag{4.42}$$

The distance between an orthotope $\mathcal{O} = R((l_i, u_i)_{i=1,\dots,n})$ and a singleton $\{(x_1, \dots, x_n)\}$ is given by:

$$d(\mathcal{O}, \{(x_1, \dots, x_n)\}) = \sqrt{\sum_{i=1}^{n} \max(0, l_i - x_i, x_i - u_i)^2} \tag{4.43}$$

Therefore, the distance between orthotopes and singletons can be computed in $O(n)$-time where n is the dimension of the feature space, which can be considered constant time, which is the same order as for the Euclidean distance.

In addition, orthotopes are much more flexible than balls and correspond better to the way prior-knowledge is available through explicit bounds and ranges of the features.

Remark 4.5.10. Definition 4.5.9 defines the bounded orthotopes. The unbounded orthotopes reaching until $+\infty$ or $-\infty$ in one or more directions can also be considered by extending the domain of bounds to $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$. The set distances (4.42) and

(4.43) are still valid provided we pose $\infty - \infty = 0$.

Convex polytopes Convex polytopes can be viewed as an extension of othotopes for which bounding hyperplanes do not need to be perpendicular to the axes. They can be constructed by intersecting half-spaces.

Definition 4.5.11. *Half-space* Let $a \in \mathbb{R}^n$ with $a \neq 0$ and $b \in \mathbb{R}$. The half-space of $\mathbb{R}^n$ parametrized by (a, b) denoted $H(a, b)$ is defined as:

$$H(a, b) = \{x \in \mathbb{R}^n | a \cdot x \leq b\} \tag{4.44}$$

Definition 4.5.12. *Convex polytope (non-empty)*

A convex polytope is the non-empty intersection of an arbitrary number of half-spaces.

Let $\mathcal{P}_1 = \bigcap_{i=1}^{N_1} H(a_{i,1}, b_{i,1})$ and $\mathcal{P}_2 = \bigcap_{i=1}^{N_2} H(a_{i,2}, b_{i,2})$ be two convex polytope. Let $\hat{x_1}$ and $\hat{x_2}$ be solutions of the quadratic program:

$$\begin{aligned} \underset{(x_1,x_2)\in(\mathbb{R}^n)^2}{\text{minimize}} \quad & \|x_1 - x_2\|_2^2 \\ \text{subject to} \quad & a_{i,1} \cdot x_1 \leq b_{i,1}, \quad i_1 = 1, \ldots, N_1 \\ & a_{i,2} \cdot x_2 \leq b_{i,2}, \quad i_2 = 1, \ldots, N_2 \end{aligned} \tag{4.45}$$

By definition, the set distance between the polytopes is $d(\mathcal{P}_1, \mathcal{P}_2) = \|\hat{x_1} - \hat{x_2}\|_2$.

In a similar fashion, the distance between the convex polytope $\mathcal{P}_1 = \bigcap_{i=1}^{N_1} H(a_{i,1}, b_{i,1})$ and the singleton $\{x_2\}$ can be computed by solving the quadratic program:

$$\begin{aligned} \underset{x_1\in\mathbb{R}^n}{\text{minimize}} \quad & \|x_1 - x_2\|_2^2 \\ \text{subject to} \quad & a_{i,1} \cdot x_1 \leq b_{i,1}, \quad i_1 = 1, \ldots, N_1 \end{aligned} \tag{4.46}$$

Then, the corresponding set distance is $d(\mathcal{P}_1, \{x_2\}) = \|\hat{x_1} - x_2\|_2$.

(4.45) and (4.46) are convex optimization problems for which a global optimum can be efficiently computed. Therefore, computing the set distance between convex polytopes requires solving a convex quadratic program which is much more costly than for orthotopes. Since the additional expressiveness of convex polytopes compared to

orthotopes is difficult to exploit, orthotopes are expected to be the best choice in most practical situations.

4.5.3.3 Managing the computational complexity

Using gRBF kernels is more costly than using the standard RBF kernel. Additional cost may be incurred in the following steps:

1. computing the region-to-region kernel products ($\frac{N_k(N_k+1)}{2}$ products) and the region-to-sample kernel products ($N_k N_d$ products);
2. flipping or shifting the kernel Gram matrix.

Step *1* has a potentially high additional cost due to the undetermined cost associated to the computation of the set distance. However, this cost can be maintained low (comparable to the cost of computing the Euclidean distance in the standard RBF kernel) by restricting oneself to specific types of sets such as orthotopes as seen in Section 4.5.3.2.

Finding the eigenvalues of the kernel Gram matrix involves finding the roots of a degree $N_d + N_k$ polynomial, for which no effective exact method exists. The most efficient numerical methods have orders of complexity of $O((N_d+N_k)^3)$ [94]. Fast matrix multiplication with the Coppersmith-Winograd algorithm can be done in $O((N_d+N_k)^\omega)$ operations with $\omega \leq 2.376$ [13]. Therefore, step *2* can be done within $O((N_d + N_k)^3)$ operations.

In most practical cases $N_k << N_d$ is a reasonable assumption. Therefore, the additional cost due to steps *1* and *2* should be limited. However, this is not true when the full kernel matrix containing the test data as well is processed as suggested in Section 4.5.3.1. In this case, the cost of step *2* becomes $O((N_d + N_k + N_t)^3)$ where N_t is the amount of unlabeled "test" data. This can be problematic if $N_t >> N_d + N_t$, a very realistic possibility.

One may try to reduce this cost by splitting the test data in several batches treated successively. The cost would then become:

$$\begin{aligned} &O(k(N_d + N_k + \frac{N_t}{k})^3) \\ &= O((N_d + N_k)^3 k + 3(N_d + N_k)^2 N_t + 3(N_d + N_k) N_t^2 k^{-1} + N_t^3 k^{-2}) \end{aligned} \tag{4.47}$$

where k is the amount of batches. Let $g(k) = (N_d + N_k)^3 k + 3(N_d + N_k)^2 N_t + 3(N_d + N_k) N_t^2 k^{-1} + N_t^3 k^{-2}$.

$$\frac{\partial g}{\partial k}(k) = (N_d + N_k)^3 - 3(N_d + N_k) N_t^2 k^{-2} - 2N_t^3 k^{-3}) \tag{4.48}$$

Therefore:

$$\begin{aligned} & \frac{\partial g}{\partial k}(k) = 0 \wedge k \neq 0 \\ \iff & k^3 \frac{\partial g}{\partial k}(k) = 0 \wedge k \neq 0 \\ \iff & ((N_d + N_k)k)^3 - 3N_t^2((N_d + N_k)k) - 2N_t^3 = 0 \wedge k \neq 0 \end{aligned} \tag{4.49}$$

This degree 3 polynomial equation in $(N_d + N_k)k$ can be solved using Cadrano's method. The discriminant is:

$$\Delta = (-2N_t^3)^2 + \frac{4}{27}(-3N_t^2)^3 = 4N_t^6 + \frac{4}{27}(-27N_t^6) = 0 \tag{4.50}$$

Therefore, the equation in $(N_d + N_k)k$ has 2 distinct real solutions:

$$\begin{cases} (N_d + N_k)k_1 = \frac{3(-2N_t^3)}{-3N_t^2} = 2N_t \\ (N_d + N_k)k_2 = \frac{-3(-2N_t^3)}{2(-3N_t^2)} = -N_t \end{cases} \tag{4.51}$$

among which only one is positive:

$$k_1 = \frac{2N_t}{N_d + N_k} \tag{4.52}$$

Example 4.5.13. For example, if there are $N_d = 100$ labeled training data, $N_k = 2$ labeled sets and $N_t = 1000$ unlabeled data, $k_1 = 19.6078$. Therefore, the unlabeled data should be split in about 20 batches.

An empirical study in Section 5.6.4 suggests that the overall improvement brought by processing the full data matrix is relatively minimal, and therefore might not be worth the potentially huge additional cost in time.

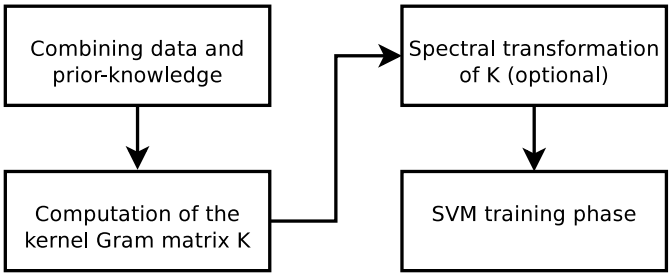


Figure 4.13: General workflow diagram involving the gRBF kernel.

4.5.4 Workflow diagram

The general workflow involving the gRBF kernel can be summarized as follows:

1. Combination of labeled data points and labeled regions into a single training set. Labeled regions may need to be adjusted using the parameter ρ in order to avoid conflicts with the data (optional).

2. Computation of the kernel Gram matrix K from the training set. Test data may also be included in order to improve generalization (optional).

3. Spectral transformation of K by flipping or shifting (optional but strongly recommended).

4. Training of any standard SVM using K.

A graphical representation of the workflow is available in Figure 4.13.

4.6 Discussion: complementary role of prior-knowledge and data

The possibility to take into account global properties of the class distribution is a fundamentally lacking aspect of the SVM+RBF combination. By nature, the SVM relies on the local characteristics of the data (the support vectors) in order to define the decision model. Adding or removing any amounts of points outside of the margin does not affect the decision boundary. As a matter of fact, methods (such as combination of SVM with discriminant analysis in [31]) have proposed to specifically address this issue.

In contrast, the prior-knowledge incorporated into KE-RBF kernels has a global influence (affecting the whole feature space) or semi-global influence (affecting large

regions of the feature space). Unlabeled and labeled regions incorporated using gRBF and ξRBF kernels induce semi-global effects over areas exceeding these regions. A priori correlations introduced by pRBF kernels have a global influence spreading across the entire feature space: the monomial and polynomial properties which are inherited by the decision model (see Theorem 4.4.6) are global properties.

Overall, KE-RBF kernels provide an effective way to incorporate prior-knowledge with global or semi-global influence which is complementary to the training points providing a local influence.

Chapter 5

Empirical Evaluation of KE-RBF Kernel Framework

5.1 Introduction

In this Chapter, we provide a detailed performance evaluation for the KE-RBF kernel framework presented in Chapter 4.

5.1.1 Objectives

The objectives of this validation are multiple. First, we prove that the different KE-RBF kernel designs work as intended: they lead to significant performance improvements when used with adequate prior-knowledge in place of the standard RBF kernel.

Next, with the variety of applications on multiple domains of application proposed in this chapter, we show that the framework is easily usable in practice and that opportunities for the KE-RBF kernels in real-world applications are numerous.

Finally, we show that KE-RBF kernels are able to overperform standard kernels with much smaller or strongly biased training sets, thereby contributing to significantly broaden the field of application of SVMs.

5.1.2 Outline

Five different and independent applications are proposed in this performance evaluation. They are the following:

1. An application to the diagnosis of breast cancer from cytological images using expert medical advice in the form of unlabeled sets with ξRBF kernels in Section 5.2.

2. An application to the prediction of meteorological data with prior-knowledge on pseudo-periodicity using ξRBF kernels in Section 5.3.

3. The last application of ξRBF kernels in Section 5.4 involves signal reconstruction using the combination of multiple frequencies. The choice of a multiplicative design over an additive design for the combination of frequencies is also validated here.

4. Section 5.5 is an application of pRBF kernels to the prediction of zootomical[1] data on a population of abalones using a priori correlations between features and labels.

5. The last application on meteorological data in Section 5.6 uses the gRBF kernel and different types of labeled regions as prior-knowledge.

All the applications use real-life data available from public sources, with the exception of Section 5.4 which involves synthetic data.

5.2 Diagnosis of breast cancer from fine needle aspiration biopsy micrographs using expert medical advice

The following binary classification problem using real-life data is an example of application of ξRBF kernels incorporating the unlabeled regions presented in Section 4.3.1. It consists in the diagnosis of breast cancer from the aspect of breast cell nuclei from biopsy micrographs. This application uses the "Wisconsin Breast Cancer" dataset publicly available at the UCI Machine Learning Repository[2].

Section 5.2.1 presents the data, prior-knowledge and classifiers used in this application. A first batch of experiments presented in Section 5.2.2 studies the effects of incorporating unlabeled regions according to the size of the training sample. A second batch presented in Section 5.2.3 compares the use of crisp sets and fuzzy sets.

[1] "Zootomy" is the study of animal anatomy.

[2] `http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)`

5.2.1 Data, prior-knowledge and learning algorithm

The dataset was constructed from micrographs of breast Fine Needle Aspiration (FNA) biopsies performed on healthy subjects and breast cancer patients. A breast FNA biopsy is a standard diagnostic procedure for breast cancer. As the name suggests, it involves the extraction of cells by aspiration with a needle. A micrograph from an FNA biopsy typically consists in a few cells on a clear and uniform background. Cell nuclei are extracted using an Active Contour (AC) method, a relatively simple task compared to other image modalities such as excisional biopsies where a whole mass of tissue is removed (see Chapter 4 for an application to excisional biopsies). Figure 5.1 shows an example of breast FNA biopsy micrograph with some cell nuclei extraction results.

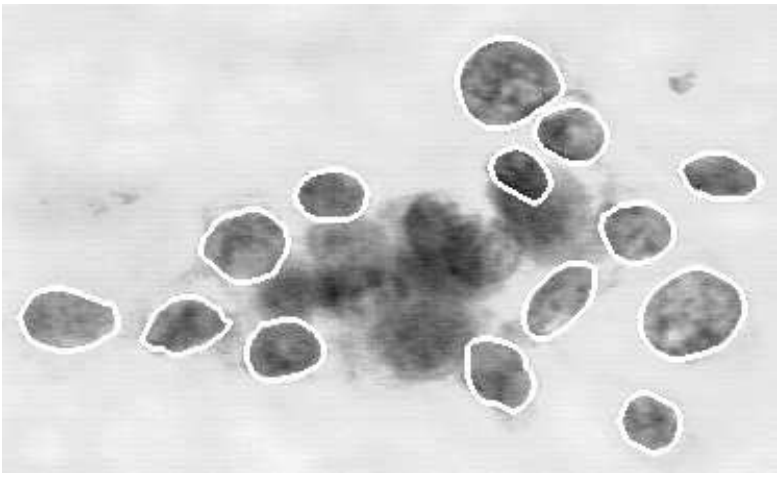


Figure 5.1: Sample breast FNA micrograph from [75]. Extracted nuclei are delineated in white.

The database itself is a collection of input-output pairs with the input being a real-valued vector containing morphological features calculated from the contours of the extracted nuclei and the output being a Boolean value indicating the occurrence of breast cancer. It contains 569 data instances, 357 corresponding to benign cases (non cancer) and 212 to malignant cases (cancer). We make use of two specific features: the mean texture and the mean smoothness of the cell nuclei. Both features are normalized in $[-1, 1]$. Full details on the database are available in [75].

The unlabeled set $\mathcal{A}$ used as prior-knowledge is obtained from expert medical knowledge about cell morphology. The diagnosis of breast cancer from cytological images is based upon the study of Nuclear Atypia (NA), *i.e.* any feature uncharacteristic of normal cell nuclei. Nuclei with homogeneous interiors and smooth contours are considered

normal nuclei. Accordingly, we translate this expert knowledge into an unlabeled set of the feature space: if both normalized features are smaller than -0.5, then nuclei are typical. This translates into the following unlabelled set $\mathcal{A} = [-\infty, -0.5]^2$. Note that we cannot a priori label $\mathcal{A}$ or $\complement\mathcal{A}$ as benign or malignant since the presence of NA alone is not a valid characterization of breast cancer. Indeed, nuclei can be atypical due to other reasons that cancers and some rare cancers show seemingly normal nuclei in early stages.

The C-SVM described in Chapter 2 and the ξRBF kernel presented in Section 4.3.1 are used. The C and γ parameters are adjusted every time by performing a grid search combined with a 2-folds cross-validation. Numerical results correspond to average misclassification rates over 100 training-testing cycles during which the training data is randomly selected.

5.2.2 Effects of prior-knowledge with different sizes of training set

The first batch of experiments uses the ξRBF kernel described in Section 4.3.1.1 incorporating the above prior-knowledge as a crisp set. Training sets are created by randomly choosing N instances. The models are tested on the $569 - N$ remaining instances.

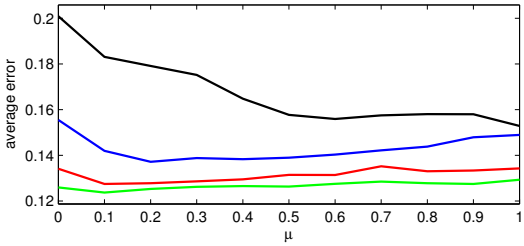
Figure 5.2 shows average results over 100 random selections for different sizes N of the training sets and different values of the parameter $\mu \in [0, 1]$ controlling the amount of prior-knowledge into the kernel. Overall, the ξRBF kernel outperforms the original RBF kernel ($\mu = 0$), specially when the training set is small: the best rate of improvement over the RBF kernel is 23.89% and is achieved when $N = 8$ and $\mu = 1$. This rate decreases when N becomes larger and the adapted kernel is about on a par with the RBF kernel when $N = 64$. Moreover, we can notice that the optimal μ (in bold in the tables) decreases when N increases: $\mu = 1$ for $N = 8$, $\mu = 0.2$ for $N = 16$ and $\mu = 0.1$ for $N = 36$ or $N = 64$. This confirms the intuitive idea that the prior-knowledge is more important when the training set is small and becomes less useful as more training data is available. In general, $\mu = 0.5$ seems to be a good default value for the parameter μ.

	$\mu = 0$	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.3$	$\mu = 0.4$	$\mu = 0.5$	$\mu = 0.6$	$\mu = 0.7$	$\mu = 0.8$	$\mu = 0.9$	$\mu = 1$
$N = 8$	0.2009	0.1831	0.1792	0.1752	0.1648	0.1577	0.1559	0.1575	0.1581	0.1580	**0.1529**
$N = 16$	0.1555	0.1420	**0.1372**	0.1388	0.1384	0.1390	0.1404	0.1422	0.1438	0.1479	0.1490
$N = 32$	0.1342	**0.1275**	0.1278	0.1287	0.1295	0.1315	0.1314	0.1353	0.1331	0.1334	0.1343
$N = 64$	0.1260	**0.1237**	0.1253	0.1263	0.1266	0.1263	0.1276	0.1285	0.1278	0.1275	0.1294

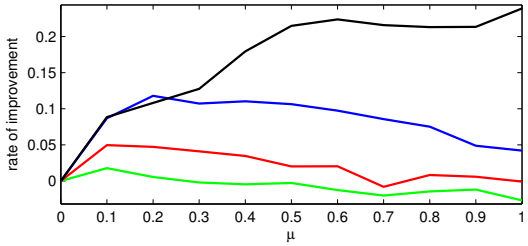
(a) Average misclassification rates

	$\mu = 0$	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.3$	$\mu = 0.4$	$\mu = 0.5$	$\mu = 0.6$	$\mu = 0.7$	$\mu = 0.8$	$\mu = 0.9$	$\mu = 1$
$N = 8$	0	0.0885	0.1081	0.1276	0.1794	0.2148	0.2238	0.2159	0.2131	0.2134	**0.2389**
$N = 16$	0	0.0869	**0.1179**	0.1072	0.1103	0.1063	0.0974	0.0856	0.0751	0.0487	0.0421
$N = 32$	0	**0.0497**	0.0472	0.0409	0.0346	0.0201	0.0203	-0.0082	0.0082	0.0058	-0.0010
$N = 64$	0	**0.0178**	0.0055	-0.0022	-0.0047	-0.0028	-0.0126	-0.0203	-0.0145	-0.0119	-0.0269

(b) Average improvement rates



(c) Graphical representation of (a)



(d) Graphical representation of (b)

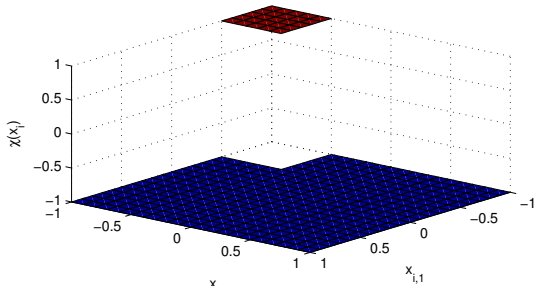
Figure 5.2: Average results with a crisp unlabeled set for different sizes N of training set and values of μ. (a) and (c) correspond to misclassification rates. (b) and (d) correspond to improvement rates over the standard RBF kernel (*i.e.* $\mu = 0$). For (c) and (d), the color convention is: black for $N = 8$, blue for $N = 16$, red for $N = 32$ and green for $N = 64$.

5.2.3 Crisp sets versus fuzzy sets

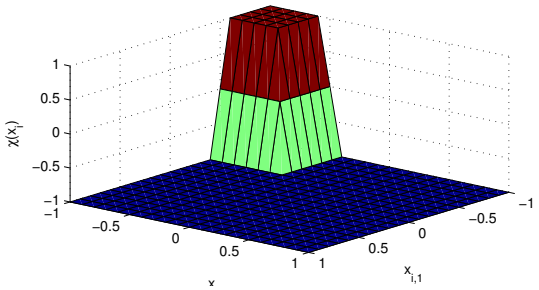
A second batch of experiments was performed in a similar setting with fuzzified versions of the indicator function. Instead of a discontinuous transition from $\chi(x) = -1$ when $x \notin \mathcal{A}$ to $\chi(x) = 1$ when $x \in \mathcal{A}$, the transition is made linear with a slope α as shown in Figure 5.3.

Figure 5.4 shows average results over 100 random selections for different values of $\mu \in [0, 1]$ and α. All the means are computed for the same 100 randomly selected training sets. The training sample size is fixed to $N = 8$, a small size which proved to favor the adapted kernel in the previous batch. It appears that the fuzzified version also performs well, with the ξRBF kernel clearly improving the results obtained with the standard RBF kernel. This improvement is however generally less when the slope is more gentle (specially $\alpha = 2.5$), which can be justified by the fact that the prior-knowledge is more approximate.

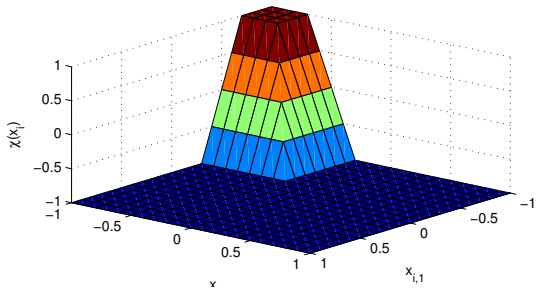
In conclusion of this application, the prior-knowledge corresponding to unlabeled sets can substantially reduce the required amount of training data by improving the classification results by a large margin when training set size is small. This improve-



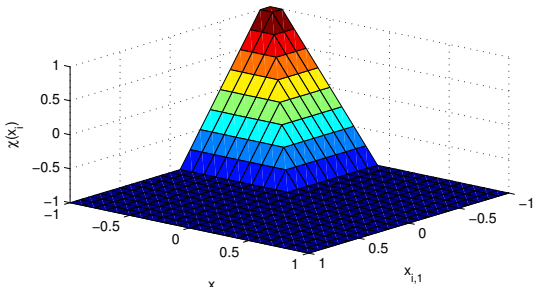
(a) $\alpha = \infty$ (crisp indicator function)



(b) $\alpha = 10$



(c) $\alpha = 5$



(d) $\alpha = 2.5$

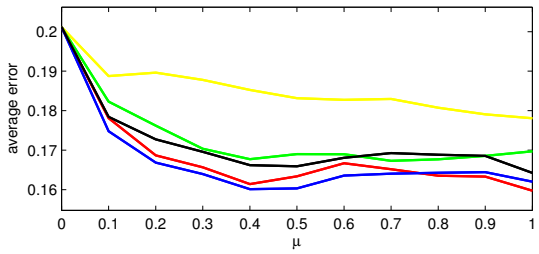
Figure 5.3: Indicator functions with different values of α.

	$\mu = 0$	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.3$	$\mu = 0.4$	$\mu = 0.5$	$\mu = 0.6$	$\mu = 0.7$	$\mu = 0.8$	$\mu = 0.9$	$\mu = 1$
$\alpha = \infty$	0.2012	0.1784	0.1727	0.1696	0.1662	0.1659	0.1681	0.1693	0.1688	0.1686	**0.1642**
$\alpha = 20$	0.2012	0.1748	0.1668	0.1640	**0.1601**	0.1603	0.1636	0.1640	0.1643	0.1644	0.1620
$\alpha = 10$	0.2012	0.1781	0.1687	0.1657	0.1614	0.1634	0.1667	0.1652	0.1635	0.1633	**0.1598**
$\alpha = 5$	0.2012	0.1823	0.1762	0.1704	0.1677	0.1690	0.1690	**0.1673**	0.1677	0.1686	0.1697
$\alpha = 2.5$	0.2012	0.1888	0.1896	0.1878	0.1853	0.1832	0.1827	0.1830	0.1807	0.1791	**0.1781**

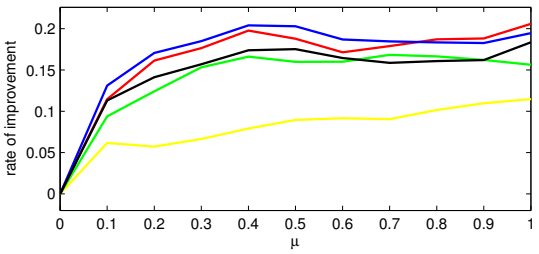
(a) Average misclassification rates

	$\mu = 0$	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.3$	$\mu = 0.4$	$\mu = 0.5$	$\mu = 0.6$	$\mu = 0.7$	$\mu = 0.8$	$\mu = 0.9$	$\mu = 1$
$\alpha = \infty$	0	0.1131	0.1413	0.1570	0.1738	0.1752	0.1645	0.1586	0.1607	0.1621	**0.1836**
$\alpha = 20$	0	0.1311	0.1707	0.1850	**0.2041**	0.2030	0.1869	0.1846	0.1834	0.1826	0.1946
$\alpha = 10$	0	0.1147	0.1614	0.1765	0.1977	0.1878	0.1714	0.1789	0.1870	0.1881	**0.2059**
$\alpha = 5$	0	0.0938	0.1241	0.1532	0.1662	0.1598	0.1600	**0.1684**	0.1665	0.1621	0.1563
$\alpha = 2.5$	0	0.0618	0.0573	0.0665	0.0791	0.0896	0.0916	0.0905	0.1015	0.1098	**0.1148**

(b) Average improvement rates



(c) Graphical representation of (a)



(d) Graphical representation of (b)

Figure 5.4: Average results for $N = 8$ and different values of μ and α. (a) and (c) correspond to misclassification rates. (b) and (d) correspond to improvement rates over the standard RBF kernel (*i.e.* $\mu = 0$). For (c) and (d), the color convention is: black for $\alpha = \infty$ (crisp indicator function), blue for $\alpha = 20$, red for $\alpha = 10$ green for $\alpha = 5$ and yellow for $\alpha = 2.5$.

ment is less significant when more training data are available which suggests that the additional data play a role similar to the prior-knowledge in compensating for the lack of training data.

5.3 Prediction of meteorological data using pseudo-periodicity

The following application based upon real-life meteorological data is an example of the use of prior-knowledge related to pseudo-periodicity using the ξRBF kernel as presented in Section 4.3.2.1.

5.3.1 Data, prior-knowledge and learning algorithm

This application is based upon publicly available meteorological data from the UK Climate Projections database[3]. It is a scalar regression problem using the monthly average temperatures measured from January 1914 to December 2006 at the geographic point with coordinates: easting 337500 - northing 1032500. A training set of N values from these $93 \times 12 = 1104$ monthly averages is used to predict values of the remaining ones. The only feature is the corresponding date.

Although some variations are usually observed from one year to another, average temperatures follow the cycle of seasons. Accordingly, the prior-knowledge is a pseudo-periodicity of 1 year incorporated into the advice function in a fashion described in Section 4.3.2.1.

The ϵ-SVR described in Chapter 2 was used with $\epsilon = 0.1$. Results are compared in terms of average absolute error. The procedure followed is similar to the one used for the application in Section 5.2, including the grid search combined with a 2-folds cross validation to set C and γ.

5.3.2 Empirical results

Figure 5.5 shows the average results over 50 randomly selected training sets for different values of the training set size N and the parameter μ. The overall improvement compared to the standard RBF kernel is very significant, reaching 62.06% for $N = 100$ and $\mu = 1$. As for the previous applications, the rate of improvement is less when the

[3] `http://www.metoffice.gov.uk/climatechange/science/monitoring/ukcp09/`

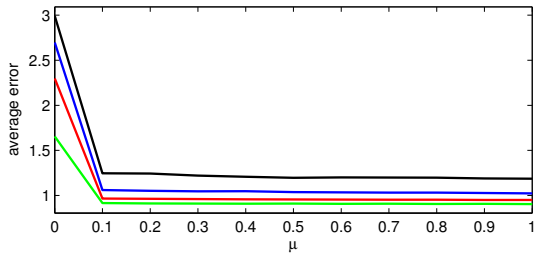
training set becomes larger. The incorporation of prior-knowledge radically improves the results even when $\mu = 0.1$, and larger values of μ only yield marginal additional improvements. Best rates of improvements are obtained with large values of μ ($\mu = 1$ for $N = 50, 100, 400$ and $\mu = 0.9$ for $N = 200$).

	$\mu = 0$	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.3$	$\mu = 0.4$	$\mu = 0.5$	$\mu = 0.6$	$\mu = 0.7$	$\mu = 0.8$	$\mu = 0.9$	$\mu = 1$
$N = 50$	2.9915	1.2456	1.2432	1.2201	1.2072	1.1961	1.1999	1.1982	1.1972	1.1881	**1.1865**
$N = 100$	2.6978	1.0597	1.0510	1.0457	1.0473	1.0378	1.0339	1.0314	1.0308	1.0271	**1.0236**
$N = 200$	2.2980	0.9659	0.9631	0.9594	0.9577	0.9552	0.9545	0.9532	0.9528	**0.9500**	0.9503
$N = 400$	1.6554	0.9155	0.9110	0.9093	0.9092	0.9107	0.9076	0.9079	0.9059	0.9067	**0.9049**

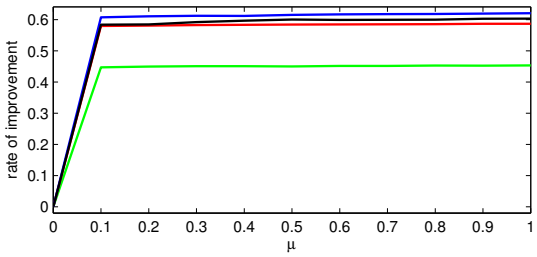
(a) Average error

	$\mu = 0$	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.3$	$\mu = 0.4$	$\mu = 0.5$	$\mu = 0.6$	$\mu = 0.7$	$\mu = 0.8$	$\mu = 0.9$	$\mu = 1$
$N = 50$	0	0.5836	0.5844	0.5921	0.5964	0.6002	0.5989	0.5995	0.5998	0.6028	**0.6034**
$N = 100$	0	0.6072	0.6104	0.6124	0.6118	0.6153	0.6168	0.6177	0.6179	0.6193	**0.6206**
$N = 200$	0	0.5797	0.5809	0.5825	0.5832	0.5844	0.5846	0.5852	0.5854	**0.5866**	0.5865
$N = 400$	0	0.4470	0.4497	0.4507	0.4508	0.4499	0.4517	0.4516	0.4527	0.4523	**0.4534**

(b) Average improvement rates



(c) Graphical representation of (a)



(d) Graphical representation of (b)

Figure 5.5: Average results for different values of N and μ. (a) and (c) correspond to mean errors. (b) and (d) correspond to improvement rates over the standard RBF kernel (*i.e.* $\mu = 0$). For (c) and (d), the color convention is: black for $N = 50$, blue for $N = 100$, red for $N = 200$ and green for $N = 400$.

The results also show that the amount of required training data can be significantly reduced by the use of the ξRBF kernel. Indeed, the average error obtained with the ξRBF kernel ($\mu = 1$) and only $N = 50$ training data points is lower that the average error obtained with the standard RBF kernel ($\mu = 0$) and $N = 400$ training data points. This corresponds to an almost 90% cut in data requirement.

In conclusion, incorporating pseudo-periodicity as prior-knowledge using the ξRBF kernel presented in Section 4.3.2 can effectively improve learning performance. The improvements are particularly significant with small datasets which allows a significant reduction in training data requirements.

5.4 Reconstruction of signal using information on its frequency decomposition

Following the case of a single frequency in Section 5.3, we now study the incorporation of multiple dominant frequencies with the ξRBF kernel described in Section 4.3.2.2.

This application consists in the reconstruction of a noisy signal with 2 dominant frequencies using the ϵ-SVR. The signal is artificially generated according to a procedure described in Section 5.4.1. The different kernels compared in this study are presented in Section 5.4.2 and include ξRBF kernels with just one or both of the frequencies as prior-knowledge. Empirical results are presented in Sections 5.4.3, 5.4.4 and 5.4.5.

5.4.1 Mixture of harmonics with additive white Gaussian noise

The data for this study is artificially generated by sampling the following 1-dimensional signal:

$$f(t) = a_1 \sin\left(\frac{2\pi}{p_1}t\right) + a_2 \sin\left(\frac{2\pi}{p_2}t\right) + \text{awgn}_{\sigma_n}(t) \tag{5.1}$$

It is a sum of 2 periodic signals with respective periods p_1 and p_2, and some average white Gaussian noise with standard deviation σ_n. For this whole study, $p_1 = 7$ and $p_2 = 3$ (note: $p_1 \wedge p_2 = 1$).

The data is sampled randomly and uniformly from the interval $I = [1, 100]$. A training set $\mathcal{S}_N$ of size N is constructed by taking N points $(x_i)_{i=1,\ldots,N}$ *i.i.d.* according to the uniform distribution over I from which the set of N input-output pairs $\mathcal{S}_N = (x_i, f(x_i))_{i=1,\ldots,N}$ is obtained.

Given a training set $\mathcal{S}_N = (x_i, f(x_i)_{i=1,\ldots,N})$ and a test set $\mathcal{S}_M = (x'_i, f(x'_i)_{i=1,\ldots,M})$ constructed following the above procedure, the task consists in creating a labeling model $\hat{f} : I \to \mathbb{R}$ using the training set $\mathcal{S}_N$ in order to provide the least absolute error on the labeling of $\mathcal{S}_M$, *i.e.* minimizing $\frac{1}{M}\sum_{i=1}^{M} |f(x'_i) - \hat{f}(x'_i)|$.

The learning machine used for this task is the ϵ-SVR described in Chapter 2 (with $\epsilon = 0.1$). Results are compared in terms of average absolute error. The C and γ parameters are adjusted every tIme by performing a grid search combined with a 5-

folds cross-validation. The size of each randomly sampled test set is $M = 100$. Each numerical result is an average value over 100 random iterations.

5.4.2 Candidate kernels

The ξRBF kernel K_2 which is the central focus of this study incorporates the 2 periods p_1 and p_2 as prior-knowledge. Its expression which follows equation (4.17) is:

$$K_2(x_1, x_2) = \xi_1(x_1, x_2)\xi_2(x_1, x_2)K_{\text{rbf}}(x_1, x_2) \tag{5.2}$$

with

$$\xi_1(x_1, x_2) = \frac{\cos\left(\frac{2\pi}{p1}(x_1 - x_2) + 1\right)}{2} \tag{5.3}$$

and

$$\xi_2(x_1, x_2) = \frac{\cos\left(\frac{2\pi}{p2}(x_1 - x_2) + 1\right)}{2} \tag{5.4}$$

During this study, all ξRBF kernels are used with $\mu = 1$, a reasonable default choice according to the previous empirical study in Section 5.3. For comparison, we also use the additive version K_2' from Equation (4.20) predicted to perform less good than the multiplicative version K_2 (see discussion in Section 4.3.2.2):

$$K_2'(x_1, x_2) = \left(\xi_1(x_1, x_2) + \xi_2(x_1, x_2)\right) K_{\text{rbf}}(x_1, x_2) \tag{5.5}$$

The ξRBF kernel K_1 from Equation (4.12) incorporating a single period p_1 is also used in this comparative study. This kernel has already been studied in details in Section 5.3. Its expression is:

$$K_1(x_1, x_2) = \xi_1(x_1, x_2)K_{\text{rbf}}(x_1, x_2) \tag{5.6}$$

5.4.3 Kernels versus size of the training set

Figure 5.6 shows a comparison of the results obtained with the different ξRBF kernels (K_2,K_2' and K_1) and the standard RBF kernel. For this batch of experiments, the 2 periodic components have the same amplitude $a_1 = a_2 = 1$ and a small amount of white noise is introduced $\sigma_n = 0.05$.

K_2 is the ξRBF kernel giving the best results by far. It systematically performs better than the standard RBF kernel by a large margin. At most, the average error is reduced by $76,16\%$ compared to the RBF kernel for a training set size of $N = 60$. K_1 is notably better than the RBF only for very small training sets ($N \leq 10$). Otherwise it fares comparably to the RBF kernel but systematically less good than K_2. This confirms that the multiplicative framework for combining multiple frequencies is effective. As expected, the additive version K_2' of the kernel provides results systematically worse that K_2. They can be clearly bad even compared to the RBF kernel (141.59% worse that the RBF kernel for $N = 150$). Therefore, the additive framework should be discarded in favour of the multiplicative framework.

In general, K_2 performs better than the RBF kernel with 4 times less data. Indeed, the results with K_2 and $N = 5$ (resp. $N = 10$, $N = 20$) training samples are better than the results with the RBF kernel and $N = 20$ (resp. $N = 40$, $N = 80$) training samples.

5.4.4 Kernels versus amplitude of the dominant frequencies

Figure 5.7 are the results for a second batch of experiments. It studies cases when the amplitudes of the 2 periodic components are different. The ratio $\frac{a_2}{a_1}$ takes different values ranging from 0 to 1. The size of training data is set to $N = 50$. As for the previous batch, $a_1 = 1$ and $\sigma_n = 0.05$.

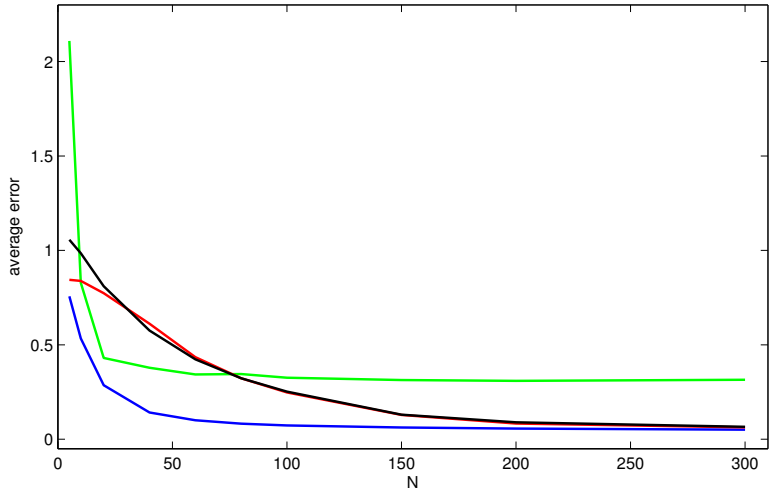
K_2 performs very stably regardless of the balance between a_1 and a_2 with an average absolute error oscillating between 0.1040 and 0.1170. K_1 performs better than K_2 only when $a_2 = 0$. It performs less and less good when the second frequency becomes more dominant. Therefore, the framework for combining multiple frequencies in a ξRBF kernel is preferable to the framework incorporating a single frequency even if a frequency largely dominates the others.

	K_2	K_2'	K_1	K_{rbf}
$N = 5$	**0.7566**	2.1091	0.8440	1.0561
$N = 10$	**0.5336**	0.8285	0.8381	0.9848
$N = 20$	**0.2862**	0.4304	0.7731	0.8102
$N = 40$	**0.1414**	0.3782	0.6117	0.5752
$N = 60$	**0.1008**	0.3430	0.4350	0.4230
$N = 80$	**0.0828**	0.3453	0.3227	0.3230
$N = 100$	**0.0732**	0.3260	0.2479	0.2518
$N = 150$	**0.0621**	0.3136	0.1289	0.1298
$N = 200$	**0.0562**	0.3089	0.0836	0.0895
$N = 300$	**0.0510**	0.3153	0.0631	0.0658

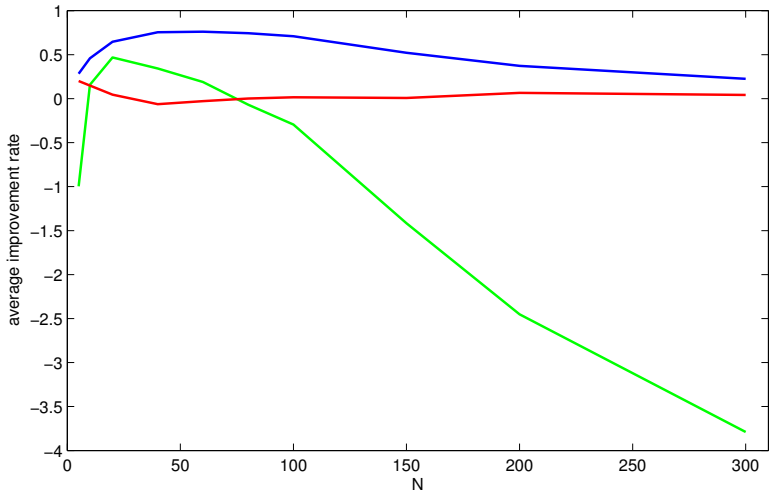
(a) Average error

	K_2	K_2'	K_1
$N = 5$	**0.2836**	-0.9971	0.2008
$N = 10$	**0.4581**	0.1587	0.1490
$N = 20$	**0.6467**	0.4688	0.0458
$N = 40$	**0.7542**	0.3425	-0.0634
$N = 60$	**0.7616**	0.1891	-0.0284
$N = 80$	**0.7435**	-0.0692	0.0010
$N = 100$	**0.7092**	-0.2948	0.0154
$N = 150$	**0.5214**	-1.4159	0.0071
$N = 200$	**0.3721**	-2.4522	0.0662
$N = 300$	**0.2260**	-3.7890	0.0416

(b) Average improvement rate



(c) Average error



(d) Average improvement rate

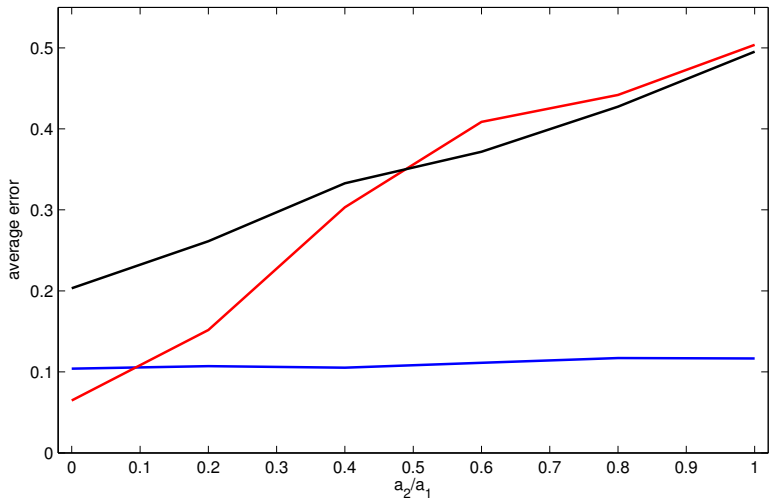
Figure 5.6: Average results over 100 experiments using the ξRBF kernels K_1, K_2 and K_2', and the standard RBF kernel K_{rbf} for different values of N. For all the results, $a_1 = a_2 = 1$ and $\sigma_n = 0.05$. (a) and (c) correspond to mean errors. (b) and (d) correspond to improvement rates over the standard RBF kernel. For (c) and (d), the color convention is: blue for K_2, green for K_2', red for K_1 and black for K_{rbf}.

	K_2	K_1	K_{rbf}
$a2 = a1$	**0.1165**	0.5037	0.4953
$a2 = 0.8a1$	**0.1170**	0.4419	0.4274
$a2 = 0.6a1$	**0.1112**	0.4086	0.3717
$a2 = 0.4a1$	**0.1051**	0.3032	0.3328
$a2 = 0.2a1$	**0.1071**	0.1518	0.2613
$a2 = 0$	0.1040	**0.0646**	0.2033

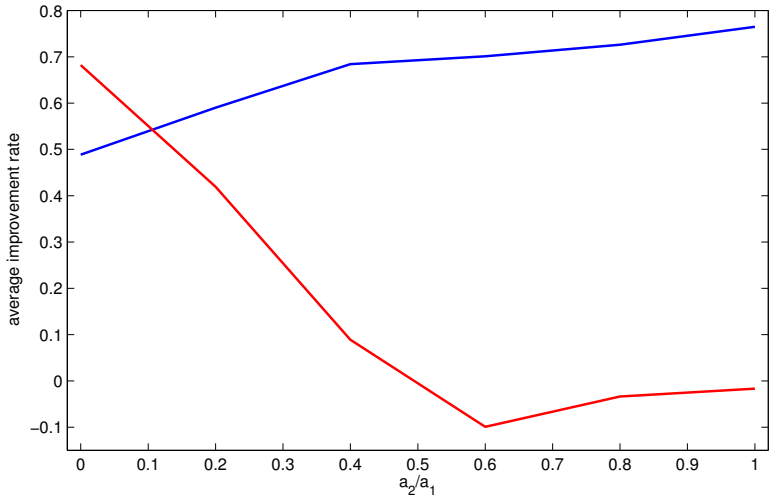
(a) Average error

	K_2	K_1
$a2 = a1$	**0.7648**	-0.0168
$a2 = 0.8a1$	**0.7262**	-0.0339
$a2 = 0.6a1$	**0.7010**	-0.0993
$a2 = 0.4a1$	**0.6841**	0.0887
$a2 = 0.2a1$	**0.5901**	0.4193
$a2 = 0$	0.4886	**0.6820**

(b) Average improvement rate



(c) Average error



(d) Average improvement rate

Figure 5.7: Average results over 100 experiments using the ξRBF kernels K_1 and K_2, and the standard RBF kernel K_{rbf} for different values of $\frac{a2}{a1} \in [0, 1]$. For all the results, $N = 50$, $a_1 = 1$ and $\sigma_n = 0.05$. (a) and (c) correspond to mean errors. (b) and (d) correspond to improvement rates over the standard RBF kernel. For (c) and (d), the color convention is: blue for K_2, red for K_1 and black for K_{rbf}.

5.4.5 Kernels versus noise

A third batch studies the effects of noise. In this batch $N = 50$ and $a_1 = a_2 = 1$ and different noise-to-signal ratios $\frac{\sigma}{a_1+a_2}$ ranging from 0 to 1 are studied. The results are available in Figure 5.8.

Unsurprisingly, all results become worse when the amount of noise is increased. The ξRBF kernels perform comparably to the RBF kernel when the noise dominates the signal (for K_2, the improvement rate is at most 13.91% when $\frac{\sigma_n}{a1+a2} \geq 0.5$, *i.e.* $\sigma_n \geq a_1$ and $\sigma_n \geq a_1$). Note that the jagged aspect of the curves for high σ_n is explained by the increased variance in the results due to noise.

In conclusion, ξRBF kernels incorporating several frequencies are a clear improvement over ξRBF kernels with a single frequency when such prior-knowledge is available. This is the case even when one of the frequencies largely dominates the others. The study also confirms that the nature of the combination should be multiplicative (as in Equation (4.17)) rather than additive (as in Equation (4.20)).

5.5 Prediction of zootomical data on a population of abalones using a priori correlations between features and labels

In this section, we show the application of pRBF kernels presented in Section 4.4 on real-life zoological data. The application consists in the prediction of the unit weight of abalones (marine gastropod molluscs) from their morphological features. The dataset publicly available from the UCI Machine Learning Repository[4] contains data for 4177 abalones.

The morphological parameters are: the length of the abalone, *i.e.* the longest shell measurement, in centimetres (feature f_1); the width of the abalone, perpendicular to the length, in centimetres (feature f_2); the height of the abalone, with the meat inside, in centimetres (feature f_3); and the amount of rings visible on the shell (feature f_4). Therefore, a single instance consists in a quintuple (f_1, f_2, f_3, f_4, y) with the 4 morphological features of the abalone f_1, f_2, f_3 and f_4, and the total weight of the abalone

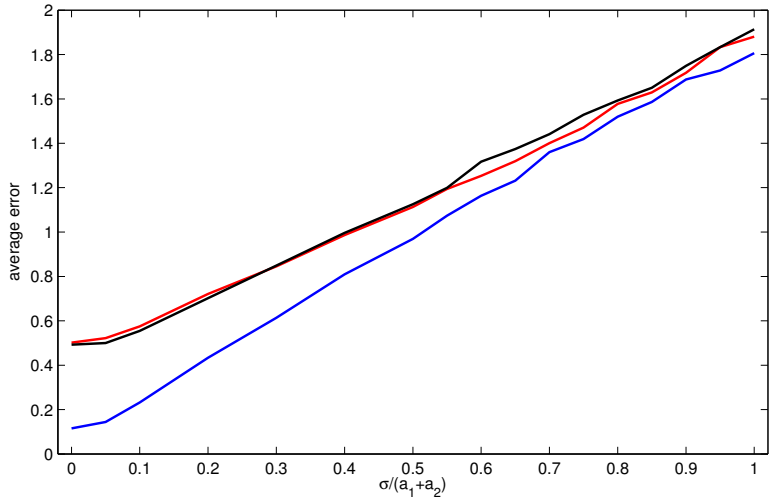
[4] `http://archive.ics.uci.edu/ml/datasets/Abalone`

	K_2	K_1	K_{rbf}
$\sigma_n = 0$	**0.1153**	0.5015	0.4927
$\sigma_n = 0.05(a_1 + a_2)$	**0.1440**	0.5222	0.5000
$\sigma_n = 0.1(a_1 + a_2)$	**0.2320**	0.5749	0.5546
$\sigma_n = 0.2(a_1 + a_2)$	**0.4336**	0.7209	0.7018
$\sigma_n = 0.3(a_1 + a_2)$	**0.6133**	0.8454	0.8486
$\sigma_n = 0.4(a_1 + a_2)$	**0.8093**	0.9864	0.9961
$\sigma_n = 0.5(a_1 + a_2)$	**0.9691**	1.1134	1.1256
$\sigma_n = 0.6(a_1 + a_2)$	**1.1633**	1.2534	1.3169
$\sigma_n = 0.7(a_1 + a_2)$	**1.3603**	1.4008	1.4412
$\sigma_n = 0.8(a_1 + a_2)$	**1.5201**	1.5776	1.5929
$\sigma_n = 0.9(a_1 + a_2)$	**1.6874**	1.7176	1.7491
$\sigma_n = a_1 + a_2$	**1.8062**	1.8804	1.9136

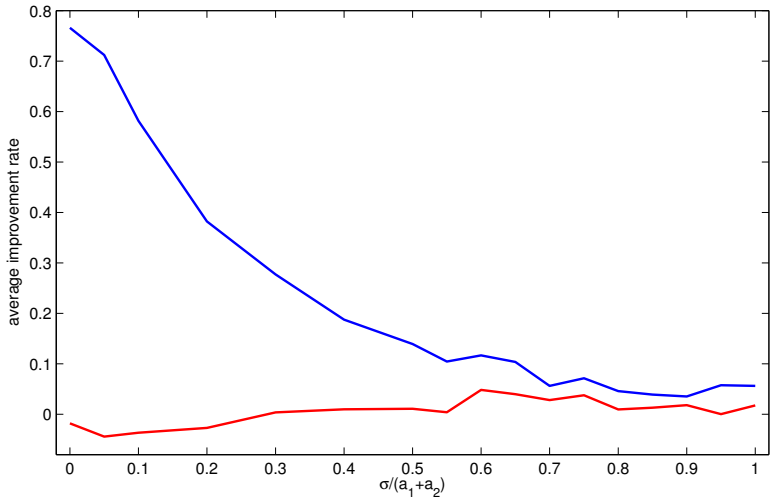
(a) Average error

	K_2	K_1
$\sigma_n = 0$	**0.7660**	-0.0180
$\sigma_n = 0.05(a_1 + a_2)$	**0.7120**	-0.0444
$\sigma_n = 0.1(a_1 + a_2)$	**0.5817**	-0.0366
$\sigma_n = 0.2(a_1 + a_2)$	**0.3822**	-0.0272
$\sigma_n = 0.3(a_1 + a_2)$	**0.2772**	0.0038
$\sigma_n = 0.4(a_1 + a_2)$	**0.1875**	0.0097
$\sigma_n = 0.5(a_1 + a_2)$	**0.1391**	0.0109
$\sigma_n = 0.6(a_1 + a_2)$	**0.1167**	0.0483
$\sigma_n = 0.7(a_1 + a_2)$	**0.0561**	0.0281
$\sigma_n = 0.8(a_1 + a_2)$	**0.0457**	0.0096
$\sigma_n = 0.9(a_1 + a_2)$	**0.0353**	0.0180
$\sigma_n = a_1 + a_2$	**0.0561**	0.0173

(b) Average improvement rate



(c) Average error



(d) Average improvement rate

Figure 5.8: Average results over 100 experiments using the ξRBF kernels K_1 and K_2, and the standard RBF kernel K_{rbf} for different values of the noise-to-signal ratio $\frac{\sigma_n}{a1+a2} \in [0, 1]$. For all the results, and $N = 50$, $a_1 = a_2 = 1$. Conventions are the same as for Figure 5.7.

y.

In Section 5.5.1, we present the correlation patterns between features and labels wich can be expected a priori and show that they are validated by the actual data distribution. The empirical results for a random, unbiased selection of the training data are presented in Section 5.5.2 and in Section 5.5.3 for a biased selection of the training data.

5.5.1 Feature-label correlation patterns

The prior-knowledge for this problem corresponds to simple geometrical intuition which suggests that the weight y should be cubical correlated to the length f_1, the width f_2 or the height f_3.

Figure 5.9 represents the weight y of the 4177 abalones plotted against a few monomial combinations of the parameters. The monotonic increase of the weight w *w.r.t.* the length f_1 is clearly visible on Figure 5.9a. Figure 5.9b shows that the relationship is in fact cubical, confirmed by the linear correlation between f_1^3 and y. In addition, w is monotonically increasing *w.r.t.* $f_1 f_2$ (Figure 5.9c) and the relationship between $f_1 f_2 f_3$ and w is linear (Figure 5.9d). Therefore, the above assumption are qualitatively confirmed by the plots. This justifies the use of the pRBF with monomials as the non-RBF portion, in particular monomials of degree 3 in f_1, f_2 and f_3.

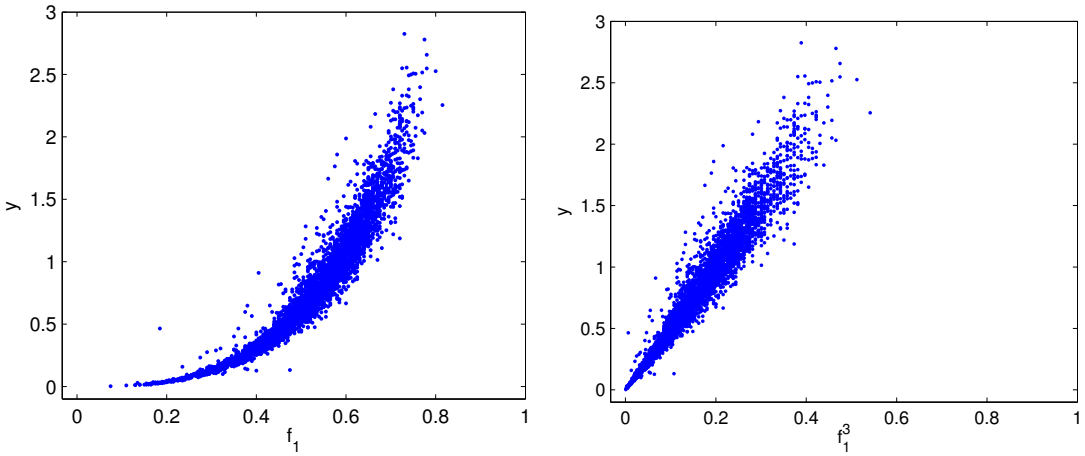
Accordingly, this batch of experiments uses the pRBF kernel described in Section 4.4 incorporating the above prior-knowledge as monomials in f_1, f_2 and f_3. For instance, if we choose the monomial $f_1 f_2$, the expression of the pRBF kernel product between the feature vectors $x_a = (f_{a,1}, f_{a,2}, f_{a,3}, f_{a,4})$ and $x_b = (f_{b,1}, f_{b,2}, f_{b,3}, f_{b,4})$ is:

$$K(x_a, x_b) = \exp\left[-\gamma\left((f_{a,3} - f_{b,3})^2 + (f_{a,4} - f_{b,4})^2\right)\right] \times f_{a,1} f_{a,2} \times f_{b,1} f_{b,2} \tag{5.7}$$

where $\gamma > 0$ is the RBF kernel bandwidth parameter.

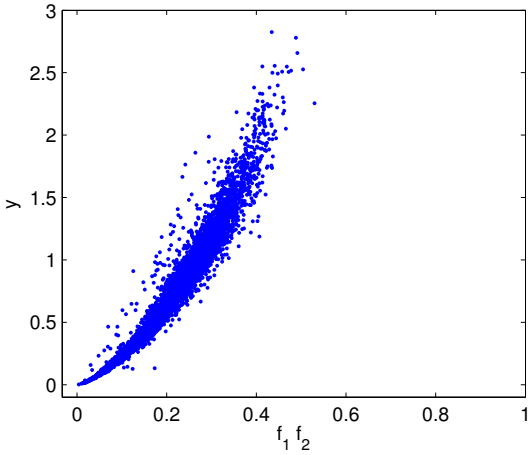
5.5.2 Learning with few data

The type of SVM used was the ϵ-SVR with $\epsilon = 0.1$. Results are compared in terms of average absolute error. Training sets are created by randomly choosing N instances.

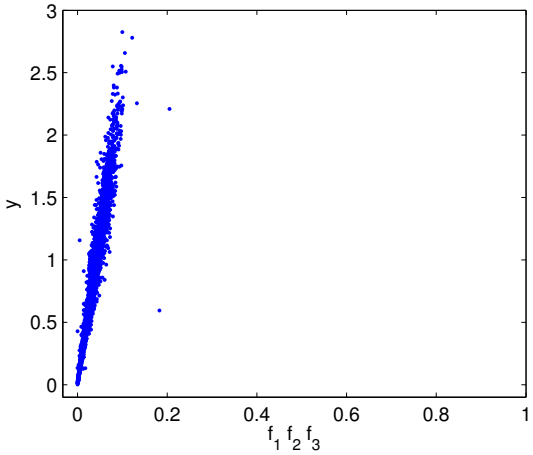


(a) y against f_1.

(b) y against f_1^3.



(c) y against $f_1 f_2$.



(d) y against $f_1 f_2 f_3$.

Figure 5.9: Weight of the abalones (output label y) against several monomial combinations of length (feature f_1), diameter (feature f_2) and height (feature f_3). The linear and polynomial relationships are clearly visible.

The C and γ parameters are adjusted every time by performing a grid search (values yielding the best average results in 5-folds cross-validation are chosen).

Figure 5.10 shows a comparison of the results obtained with different pRBF kernels and the standard RBF kernel. Each numerical result is an average value over 100 random iterations. The monomials used for the pRBF kernels were f_1, f_1^2, f_1^3, $f_1 f_2$ and $f_1 f_2 f_3$.

Every pRBF kernel systematically improves the results of the standard RBF kernel, with the exception of the pRBF kernel with monomial f_1 for which the rate of improvement is between -6.02% and 9.18%. The best results are obtained with the degree 3 monomials f_1^3 (rate of improvement between 15.19% and 41.45%) and $f_1 f_2 f_3$ (rate of improvement between 12.92% and 36.75%). The order of the monomials from worse to best is: first the degree 1 monomial f_1 which is the worse by far, then the degree 2 monomials f_1^2 and $f_1 f_2$, and finally the degree 3 monomials $f_1 f_2 f_3$ and f_1^3.

The above order is consistent with the prior-knowledge available on the problem. While a degree of 1 or 2 capture the monotonicity of the relationship between output label and input features, only the degree 3 monomials are a faithful representation of the cubic relationship between dimensions and weight. The fact that degree 2 monomials perform better than degree 1 monomials is also expected since a quadratic relationship is a better approximation of a cubic relationship than a linear relationship. Overall, this is a confirmation that the most faithfully the pRBF kernel incorporates the prior-knowledge, the better are the results.

The impact in terms of the required amount of training data is significant. On this example, the required amount of training data is divided by more than 4 thanks to the use of the pRBF kernel with proper prior-knowledge. Indeed, the pRBF kernel associated to the monomial f_1^3 with $N = 10$ training samples (average absolute error of 14.74%) performs better than the standard RBF kernel with $N = 40$ training samples (average absolute error of 15.45%).

5.5.3 Learning with biased data

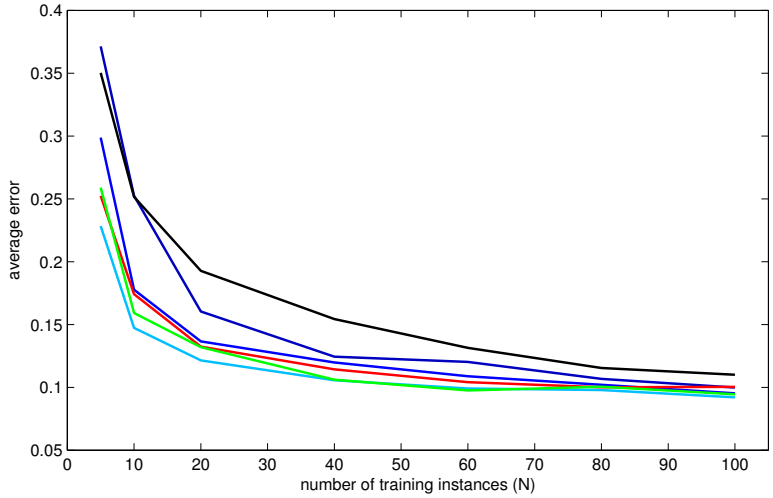
Another batch of similar experiments were conducted after a biased selection of the data instead of the uniformly distributed random selection of Section 5.5.2. The training sets are constituted by only selecting infant (sexually immature) abalones which are on average smaller in size than adult abalones. Infant and adult abalones are used

	f_1	f_1^2	f_1^3	$f_1 f_2$	$f_1 f_2 f_3$	1
$N = 5$	0.3713	0.2988	**0.2284**	0.2524	0.2589	0.3502
$N = 10$	0.2524	0.1776	**0.1474**	0.1742	0.1591	0.2516
$N = 20$	0.1604	0.1366	**0.1215**	0.1325	0.1319	0.1927
$N = 40$	0.1244	0.1198	**0.1056**	0.1144	0.1060	0.1543
$N = 60$	0.1203	0.1088	0.0991	0.1041	**0.0975**	0.1314
$N = 80$	0.1068	0.1021	**0.0979**	0.1001	0.1005	0.1154
$N = 100$	0.0999	0.0953	**0.0920**	0.1004	0.0945	0.1100

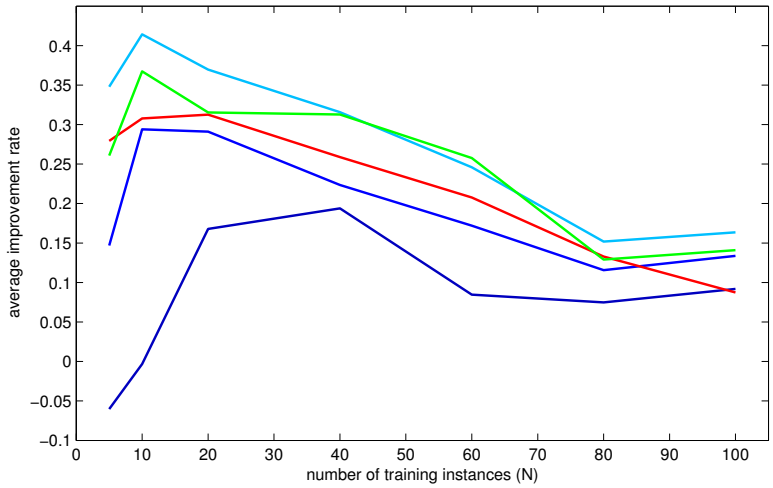
(a) Average error

	f_1	f_1^2	f_1^3	$f_1 f_2$	$f_1 f_2 f_3$
$N = 5$	-0.0602	0.1469	**0.3480**	0.2794	0.2607
$N = 10$	-0.0033	0.2939	**0.4143**	0.3077	0.3675
$N = 20$	0.1678	0.2911	**0.3697**	0.3126	0.3154
$N = 40$	0.1939	0.2235	**0.3157**	0.2589	0.3128
$N = 60$	0.0846	0.1720	0.2459	0.2077	**0.2577**
$N = 80$	0.0748	0.1156	**0.1519**	0.1329	0.1292
$N = 100$	0.0918	0.1337	**0.1635**	0.0873	0.1411

(b) Average improvement rate



(c) Average error



(d) Average improvement rate

Figure 5.10: Average results over 100 randomly selected training sets using the pRBF kernel for different values of N and different monomial expressions. (a) and (c) correspond to mean errors. (b) and (d) correspond to improvement rates over the standard RBF kernel (i.e. when the monomial expression is 1). For (c) and (d), the color convention according to the monomial expression used is: black for 1 (standard RBF kernel), dark blue for f_1, blue for f_1^2, light blue for f_1^3, red for $f_1 f_2$ and green for $f_1 f_2 f_3$.

indiscriminately for testing. In practice, this could for instance happen if the abalones used for the training data set where artificially cultivated and could not be given enough time to reach maturity.

Figure 5.11 presents the numerical results obtained with this second batch of experiments. Again, the pRBF kernels substantially improve the results obtained with the stand RBF kernel with the degree 3 monomials offering the best improvements (except for the smallest training set size $N = 5$ for which $f_1 f_2$ performed the best). The best rate of improvement is 35.78% obtained with the monomial $f_1 f_2 f_3$ for $N = 80$.

A notable difference with the case of the unbiased training sets is that improvement rates remain consistently high even when the training set becomes larger (up to 33.73% for $N = 100$). This shows that the pRBF kernel with prior-knowledge allows for accurate predictions even outside of the range of the training data which is usually impossible for the standard RBF kernel, thus confirming the observations made in Section 4.4 Figure 4.6.

As a matter of fact, the best result obtained for $N = 100$ with the pRBF kernel on biased training sets (an average error of 0.1082) is almost on a par with the best result obtained with the pRBF kernel on unbiased training sets (0.0920) whereas the best result obtained with the standard RBF kernel on biased training sets (0.1633) remains considerably worse than its counterpart on unbiased training sets (0.1100).

5.6 Prediction of daily meteorological data using monthly, seasonal and yearly statistics

This study is an application of the gRBF kernel presented in Section 4.5 to the prediction of daily meteorological data using prior-knowledge in the form of monthly, seasonal and yearly averages.

Data, prior-knowledge and learning algorithm are presented in Section 5.6.1. The impact of labeled sets in the presence of a variable amount of data is studied in Section 5.6.2. An empirical comparison between switching and shifting is proposed in Section 5.6.3. Another empirical comparison between applying the spectral transformation to the whole dataset or to the training data alone is proposed in Section 5.6.4.

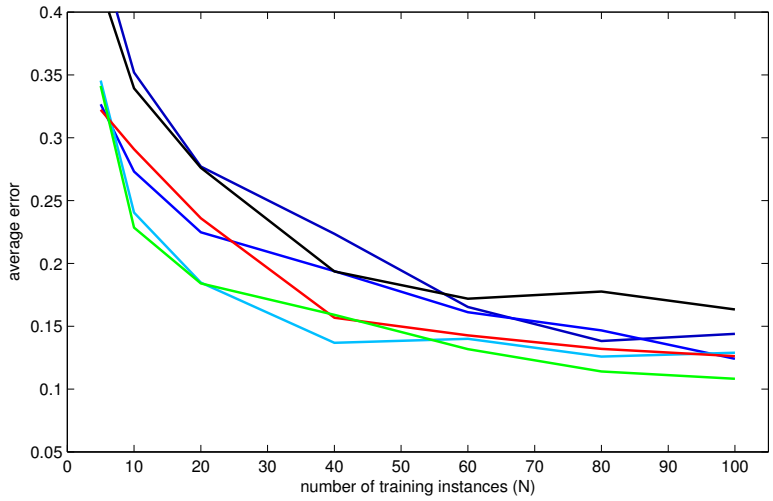
In addition, due to the sometimes narrow gap between the performance curves and

	f_1	f_1^2	f_1^3	$f_1 f_2$	$f_1 f_2 f_3$	1
$N = 5$	0.4448	0.3266	0.3454	**0.3223**	0.3412	0.4197
$N = 10$	0.3519	0.2731	0.2404	0.2909	**0.2284**	0.3393
$N = 20$	0.2770	0.2247	0.1847	0.2359	**0.1840**	0.2761
$N = 40$	0.2236	0.1938	**0.1368**	0.1567	0.1590	0.1936
$N = 60$	0.1653	0.1611	0.1400	0.1427	**0.1318**	0.1718
$N = 80$	0.1382	0.1467	0.1258	0.1320	**0.1140**	0.1775
$N = 100$	0.1439	0.1240	0.1289	0.1262	**0.1082**	0.1633

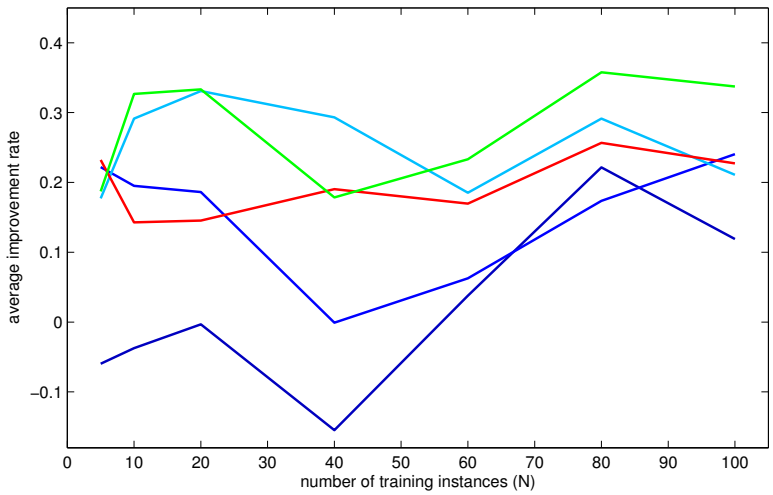
(a) Average error

	f_1	f_1^2	f_1^3	$f_1 f_2$	$f_1 f_2 f_3$
$N = 5$	-0.0598	0.2219	0.1771	**0.2323**	0.1871
$N = 10$	-0.0374	0.1950	0.2915	0.1427	**0.3268**
$N = 20$	-0.0033	0.1861	0.3309	0.1454	**0.3333**
$N = 40$	-0.1548	-0.0009	**0.2932**	0.1904	0.1785
$N = 60$	0.0379	0.0626	0.1853	0.1696	**0.2331**
$N = 80$	0.2215	0.1736	0.2913	0.2567	**0.3578**
$N = 100$	0.1189	0.2405	0.2108	0.2272	**0.3373**

(b) Average improvement rate



(c) Average error



(d) Average improvement rate

Figure 5.11: Average results over 100 training sets selected from infants abalones. Conventions and notations are similar as for Figure 5.10.

their apparent instability, a statistical validation of the relevance of the measurements is presented in Section 5.6.5.

5.6.1 Data, prior-knowledge and learning algorithm

The data consists in daily average temperature measurements on a square grid of 100 locations in the UK over a period of 10 years from 1960 to 1969 included (hence 3653 days due to the presence of 3 bissextile years over the period). The 100 locations are given by their geographical coordinates in the easting-northing system. The database contains a total of $100 \times 3653 = 365300$ data instances. Each data instance is an input-output tuple (f_1, f_2, f_3, y) where f_1 (the date given in number of days elapsed from 01/01/1960), f_2 (the easting coordinate) and f_3 (the northing coordinate) are the input features, and y (the temperature in degrees Celsius) is the output label. Features f_2 and f_3 corresponding to geographical coordinates have been normalized to fit in a range from 0 to 10. The original data is publicly available from the UK Climate Projections database[5] upon request.

The task consists in predicting the daily temperature y from the 3 features f_1, f_2 and f_3. A training set of size N randomly sampled from the database is used to create a prediction model which is evaluated on a randomly sampled test set (disjoint from the training set). The results are compared in terms of average absolute error.

The prior-knowledge available for this experiment consists in monthly (120 instances), seasonal (40 instances) and yearly (10 instances) average values of the temperature over the whole area. Preserving the notation for orthotopes introduced in Section 4.5.3.2, each average value y over a period $[d_a, d_b]$ where d_a is the day from which the period starts and d_b the day at which the period ends translates into an orthotope:

$$\mathcal{O} = R(d_a, d_b, -\infty, +\infty, -\infty, +\infty) \tag{5.8}$$

and then, into an input-output pair $(\mathcal{O}, y)$ used as training data in the gRBF kernel.

The learning algorithm used in this study is the standard ϵ-SVR (with $\epsilon = 0.1$). The C and γ parameters are adjusted with a grid search combined with a 5-folds cross validation. In the absence of explicit mentions, flipping as described in Section 4.5.3.1 is

[5] `http://www.metoffice.gov.uk/climatechange/science/monitoring/ukcp09/download/daily/time_series.html`

applied to the kernel matrix containing training and test data. Indeed, flipping performs better than shifting as shown in Section 5.6.3 and applying the transformation on the whole data improves generalizability as shown in Section 5.6.4.

Every numerical result in this study is an average over 100 training-testing cycles with a random selection of the training and testing data. The size of every test set is always 100.

5.6.2 Impact of labeled regions

In this section, we study the use of different labeled sets as prior-knowledge. First, we compare results obtained when using sets corresponding to monthly, seasonal or yearly averages (or no labeled sets at all). Next, we investigate the effects of using different values for the parameter ρ (see Section 4.5.2.2 for a detailed explanation about the parameter ρ).

Figure 5.12 shows the numerical results obtained with different sizes of training set N and different labeled sets corresponding to monthly, seasonal, yearly averages or no labeled sets at all, which is equivalent to using the standard RBF kernel. In this batch, $p = 1$ (*i.e.* $\rho = 0$), therefore labeled sets are not modified according to interferences with the training data.

Best results are obtained with labeled sets corresponding to monthly averages (improvement of $48,63\%$ for $N = 5$ compared to the RBF kernel), followed by seasonal averages (improvement of $29,61\%$ for $N = 5$). The use of yearly averages yields results comparable to the standard RBF kernel which is understandable since temperatures follow a yearly cycle (thus a yearly average does not capture any variations). These results are coherent with the fact that monthly averages contain more information than seasonal averages which in turn contain more information than yearly averages.

The greatest improvement rates are obtained with small training sets. For larger training sets ($N \geq 300$) the results are fairly similar regardless of the label sets (improvement rates compared to the RBF kernel vary in a narrow range between -1.10% and 2.82%). This illustrates that general prior-knowledge about average values becomes less necessary as more specific data is available.

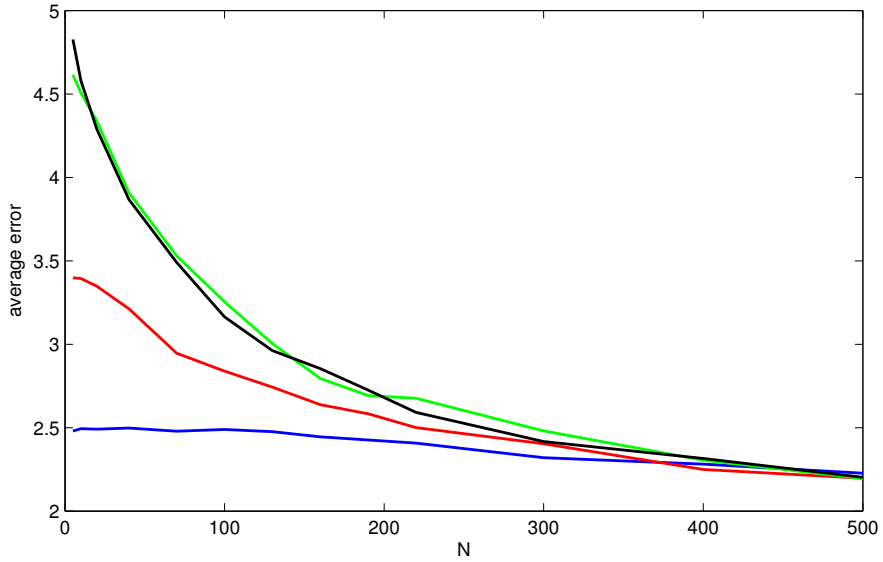
The improvements still hold if we count the labeled sets as additional training data ($N + 120$ for monthly averages and $N + 40$ for seasonal averages). However, this com-

	monthly	seasonal	yearly	none
$N = 5$	**2.4795**	3.3977	4.6156	4.8270
$N = 10$	**2.4937**	3.3943	4.5070	4.5808
$N = 20$	**2.4915**	3.3470	4.3360	4.2884
$N = 40$	**2.4979**	3.2130	3.9079	3.8684
$N = 70$	**2.4787**	2.9456	3.5300	3.4897
$N = 100$	**2.4893**	2.8388	3.2543	3.1639
$N = 130$	**2.4764**	2.7432	3.0043	2.9617
$N = 160$	**2.4452**	2.6375	2.7943	2.8538
$N = 190$	**2.4262**	2.5830	2.6925	2.7247
$N = 220$	**2.4077**	2.5002	2.6759	2.5916
$N = 300$	**2.3202**	2.4033	2.4807	2.4165
$N = 400$	2.2815	**2.2499**	2.3050	2.3151
$N = 500$	2.2271	2.1974	**2.1946**	2.2030

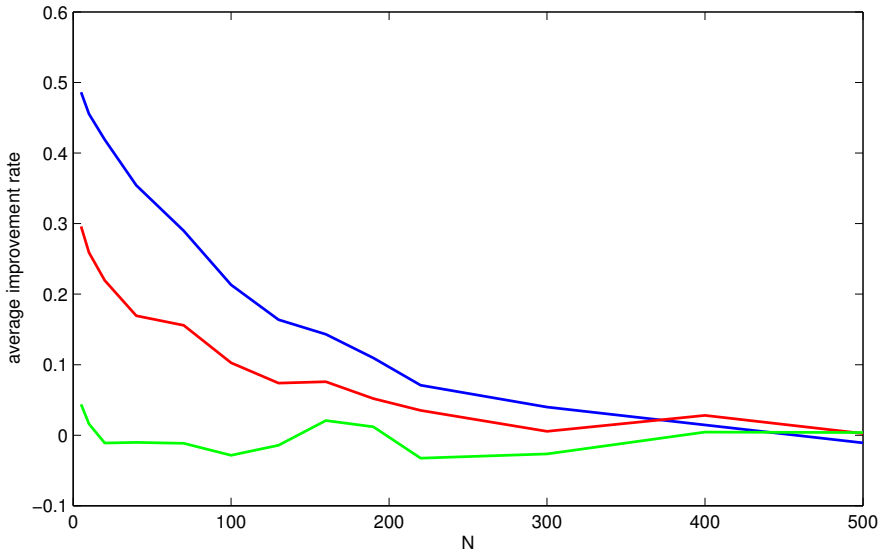
(a) Average error

	monthly	seasonal	yearly
$N = 5$	**0.4863**	0.2961	0.0438
$N = 10$	**0.4556**	0.2590	0.0161
$N = 20$	**0.4190**	0.2195	-0.0111
$N = 40$	**0.3543**	0.1694	-0.0102
$N = 70$	**0.2897**	0.1559	-0.0115
$N = 100$	**0.2132**	0.1028	-0.0286
$N = 130$	**0.1638**	0.0738	-0.0144
$N = 160$	**0.1432**	0.0758	0.0209
$N = 190$	**0.1096**	0.0520	0.0118
$N = 220$	**0.0709**	0.0352	-0.0325
$N = 300$	**0.0398**	0.0055	-0.0266
$N = 400$	0.0145	**0.0282**	0.0043
$N = 500$	-0.0110	0.0025	**0.0038**

(b) Average improvement rate



(c) Average error



(d) Average improvement rate

Figure 5.12: Average results for different labeled sets and sizes of the training set N. For all the results, $p = 1$ (*i.e.* $\rho = 0$). (a) and (c) correspond to mean errors. (b) and (d) correspond to improvement rates over the standard RBF kernel. For (c) and (d), the color convention is: blue for monthly average sets, red for seasonal average sets, green for yearly average sets and black for none (standard RBF kernel).

parison is fictitious since in practice labeled set and ordinary training data are not interchangeable as labeled sets come from prior-knowledge and not training data.

The required amount of training data is greatly reduced by the use of labeled sets. For instance, the standard RBF kernel needs 300 training samples in order to beat the gRBF kernel with 5 training samples and monthly average sets, or 100 samples to beat the gRBF kernel with 20 training samples and seasonal average sets.

The second batch of experiments studies the impact of the parameter $\rho \geq 0$ over classification results. As described in more details in Section 4.5.2.2, we propose to deal with contradictions between training data and labeled sets by modifying the labeled sets according to the training data. This is done by subtracting from the labeled sets open balls of radius ρ centered around the training data. Since the level of interaction between data and labeled sets depends on the kernel parameter γ, it is desirable to control ρ indirectly through another parameter $p = \exp(-\gamma\rho^2) \in]0, 1]$ quantifying the maximal interaction between training data and labeled sets (see Section 4.5.2.2 for more details).

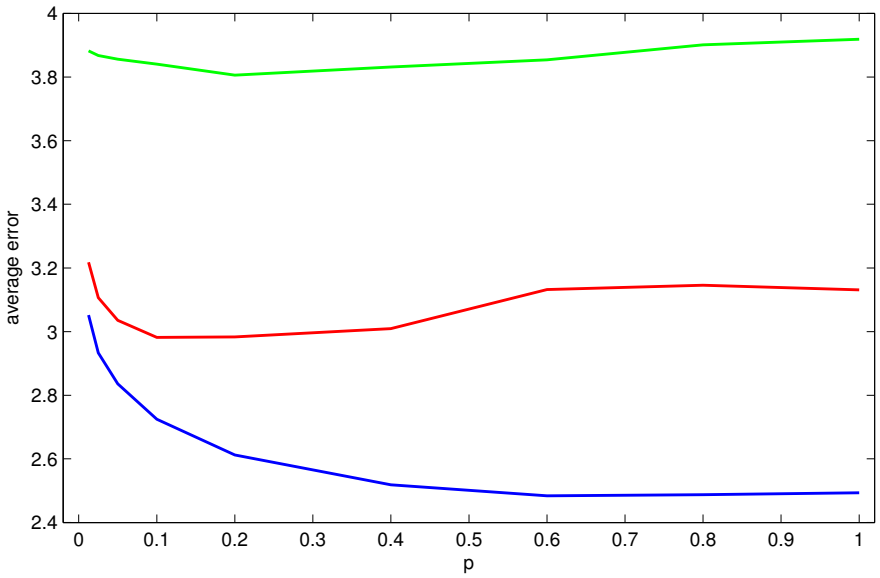
Figure 5.13 shows the average results obtained with different values of p (hence different values of ρ). The size of training sets is fixed ($N = 40$). With monthly averages, large values of p (higher than 0.6) work best, corresponding to small modifications for the labeled sets. With seasonal averages, smaller values of p (between 0.1 and 0.4) work best, corresponding to larger modifications for the labeled sets. This is consistent with the fact that monthly averages are a more faithful approximation of the daily temperatures than seasonal data.

A smaller p has the effect to reduce the labeled sets. Therefore, when p gets close to 0, the gRBF kernels degenerates into standard RBF kernels which explains the degradation of the results observed with very small p (except for the gRBF kernel with yearly averages which already perform on a par with the RBF kernel). This also implies that any potential negative impact associated to a bad choice of the parameter p is bounded by the performance of the RBF kernel.

In conclusion, this study has confirmed that adequate labeled sets can significantly improve the performance of the standard RBF kernel. The parameter p (related to ρ)

	monthly	seasonal	yearly
$p = 1$	2.4935	3.1312	3.9186
$p = 0.8$	2.4876	3.1459	3.9008
$p = 0.6$	**2.4840**	3.1319	3.8541
$p = 0.4$	2.5187	3.0094	3.8314
$p = 0.2$	2.6121	2.9833	**3.8059**
$p = 0.1$	2.7248	**2.9818**	3.8403
$p = 0.05$	2.8361	3.0352	3.8560
$p = 0.025$	2.9330	3.1065	3.8673
$p = 0.0125$	3.0519	3.2179	3.8818

(a) Average error



(b) Average error

Figure 5.13: Average results for different values of p and labeled sets. For all the results, $N = 40$. The color convention is: blue for monthly average sets, red for seasonal average sets and green for yearly average sets.

can also help getting the better results. It should be set to a high value (closer to 1) if the labeled sets are an accurate description of the data and lower (closer to 0) if they are a fuzzy description. Otherwise, we do not expect a critical degradation of the results from choosing a bad parameter p.

5.6.3 Shifting versus flipping

In general, gRBF kernels are not PD kernels. In Section 4.5.3.1, two different spectral methods applied to the kernel matrix have been proposed to solve the problem: flipping and shifting. The next batch of experiments provides an empirical comparison of the two methods.

The results of this comparative study are given in Figure 5.14. p was set to $p = 1$ and only monthly average sets were used. The interpretation of the results is very straightforward: flipping performs consistently and significantly better than shifting. Shifting even yields worse results than the standard RBF kernel (which is PD and requires no spectral transformation) when $N \geq 300$.

5.6.4 Improving generalizability

Applying the spectral transformation (shifting or flipping) on the training data alone poses a problem with respect to the generalizability to test data on which the transformation was not performed. In this last batch, we compare flipping (the better of the two methods according to Section 5.6.3) the training data only to flipping the whole data set including training and test data.

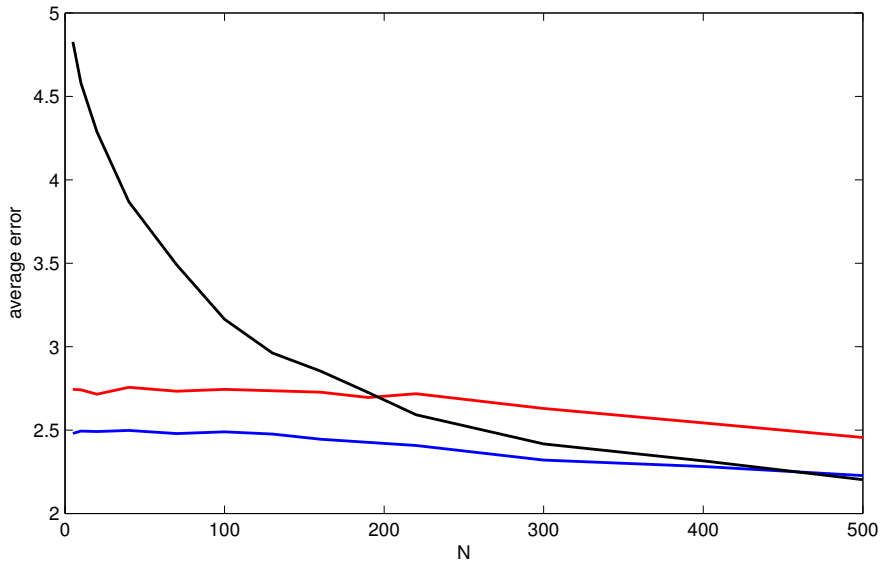
Figure 5.15 recapitulating the results from this last batch show that applying the transformation on the whole data does not have a significant impact when N is small. The improvement becomes more obvious when the training data set becomes larger. In particular, we observe that using the gRBF kernel without applying the transformation to test data ends up giving worse results than the RBF kernel when lots of training data are used ($N > 300$).

	flipping	shifting
$N = 5$	**2.4795**	2.7438
$N = 10$	**2.4937**	2.7415
$N = 20$	**2.4915**	2.7150
$N = 40$	**2.4979**	2.7561
$N = 70$	**2.4787**	2.7324
$N = 100$	**2.4893**	2.7435
$N = 130$	**2.4764**	2.7352
$N = 160$	**2.4452**	2.7268
$N = 190$	**2.4262**	2.6947
$N = 220$	**2.4077**	2.7173
$N = 300$	**2.3202**	2.6291
$N = 400$	**2.2815**	2.5431
$N = 500$	**2.2271**	2.4549

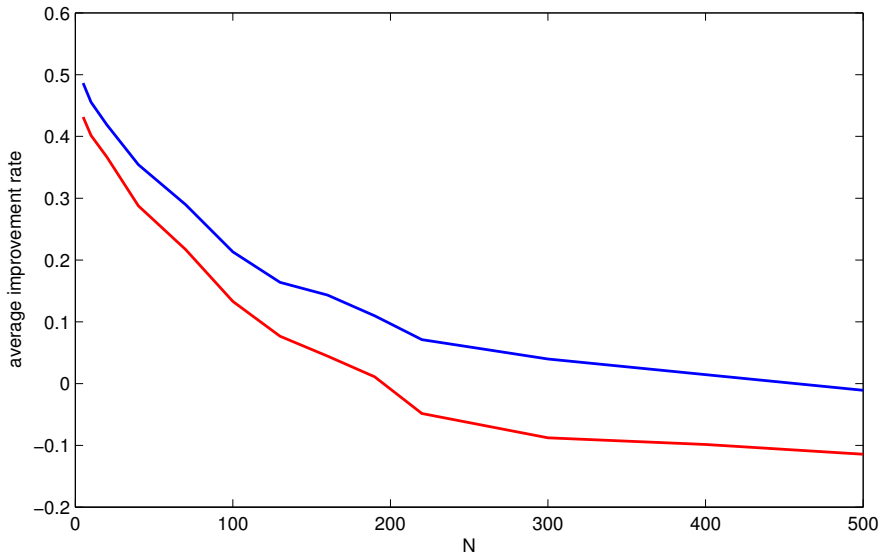
(a) Average error

	flipping	shifting
$N = 5$	**0.4863**	0.4316
$N = 10$	**0.4556**	0.4015
$N = 20$	**0.4190**	0.3669
$N = 40$	**0.3543**	0.2875
$N = 70$	**0.2897**	0.2170
$N = 100$	**0.2132**	0.1329
$N = 130$	**0.1638**	0.0765
$N = 160$	**0.1432**	0.0445
$N = 190$	**0.1096**	0.0110
$N = 220$	**0.0709**	-0.0485
$N = 300$	**0.0398**	-0.0880
$N = 400$	**0.0145**	-0.0985
$N = 500$	**-0.0110**	-0.1144

(b) Average improvement rate



(c) Average error



(d) Average improvement rate

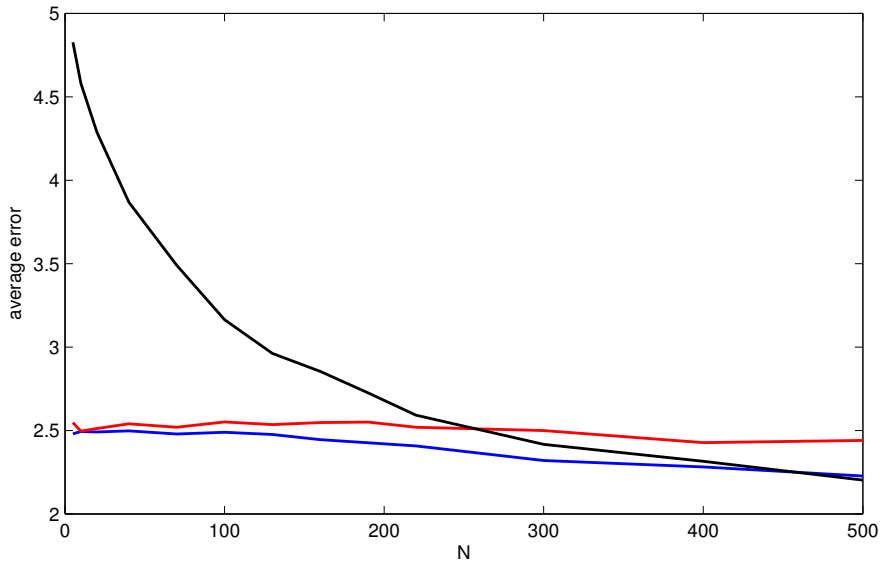
Figure 5.14: Average results for different N and spectral transformation methods. For all the results, labeled sets corresponding to monthly averages are used and $p = 1$ (*i.e.* $\rho = 0$). (a) and (c) correspond to mean errors. (b) and (d) correspond to improvement rates over the standard RBF kernel without labeled sets (values from Figure 5.12). The color convention is: blue for flipping, red for shifting, and black for the standard RBF.

	training	training+test
$N = 5$	**2.4795**	2.5476
$N = 10$	**2.4937**	2.4954
$N = 20$	**2.4915**	2.5118
$N = 40$	**2.4979**	2.5397
$N = 70$	2.4787	2.5194
$N = 100$	**2.4893**	2.5512
$N = 130$	**2.4764**	2.5350
$N = 160$	**2.4452**	2.5470
$N = 190$	**2.4262**	2.5505
$N = 220$	**2.4077**	2.5197
$N = 300$	**2.3202**	2.4997
$N = 400$	**2.2815**	2.4274
$N = 500$	**2.2271**	2.4400

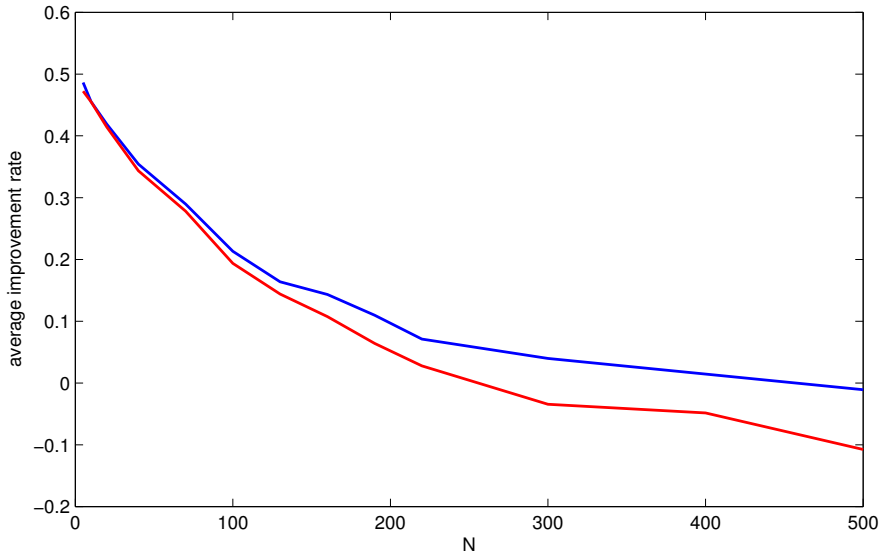
(a) Average error

	training	training+test
$N = 5$	**0.4863**	0.4722
$N = 10$	**0.4556**	0.4552
$N = 20$	**0.4190**	0.4143
$N = 40$	**0.3543**	0.3435
$N = 70$	**0.2897**	0.2780
$N = 100$	**0.2132**	0.1937
$N = 130$	**0.1638**	0.1441
$N = 160$	**0.1432**	0.1075
$N = 190$	**0.1096**	0.0640
$N = 220$	**0.0709**	0.0277
$N = 300$	**0.0398**	-0.0344
$N = 400$	**0.0145**	-0.0485
$N = 500$	**-0.0110**	-0.1076

(b) Average improvement rate



(c) Average error



(d) Average improvement rate

Figure 5.15: Comparison of average results for different N between applying flipping to the training data alone or to the whole data including test data. For all the results, labeled sets corresponding to monthly averages are used and $p = 1$ (*i.e.* $\rho = 0$). (a) and (c) correspond to mean errors. (b) and (d) correspond to improvement rates over the standard RBF kernel (values from Figure 5.12). The color convention is: blue for training+testing, red for training only and black for the standard RBF.

5.6.5 Statistical relevance of the measurements

In this section, we estimate the reliability of the numerical results presented in this study. Indeed, numerical results from this study and corresponding plotted curves may seem very close and unstable. Thus, one may legitimately question the validity the numerical results. To clarify the issue, we compute intervals of confidence for the data using results from the probability theory.

Every individual measurement of the average absolute error (*i.e.* for a single training-testing cycle) has a measured standard deviation of $\sigma_1 = 0.2$ or less. Therefore, an average result over 100 independent iterations has a standard deviation of:

$$\sigma_{100} = \sqrt{\frac{1}{100}}\sigma_1 = \frac{1}{10}\sigma_1 = 0.02 \tag{5.9}$$

Chebyshev's inequality states that if a random variable X has a mean μ and a standard deviation σ, then for any $k > 0$:

$$\mathbb{P}(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2} \tag{5.10}$$

Applied to our averages over 100 iterations X_{100} with mean μ_{100} and standard deviation $\sigma_{100} = 0.02$, Equation (5.10) becomes:

$$\mathbb{P}(|X - \mu_{100}| \geq k \times 0.02) \leq \frac{1}{k^2} \tag{5.11}$$

With $k = 5$, we get that the chance that a measurement is off by more that 0.1 is lower than 4%. 0.1 being approximately the order of magnitude of the space between two adjacent curves in this study, this ensures that the vast majority of the measures are significant.

With $k = \sqrt{2}$, we get that the chance that a measurement is off by more that ≈ 0.03 is lower than 50% which ensures that more than half of the measures can be considered as precise.

Chapter 6

Application: Automatic Grading of Invasive Breast Carcinoma from Histopathological Images

6.1 Introduction

In this chapter, we propose a complete system for Breast Cancer Grading (BCG) from Haematoxylin-Eosin (H&E) stained surgical biopsies. It specifically addresses the grading of Nuclear Atypia (NA), a central component of most BCG procedures. This work also provides an example of application of the KE-RBF framework to complex, real-word situations.

A short introduction to BCG from H&E stained biopsies is first given in Section 6.2, and the challenges related to computer-aided BCG are presented with a review of the state-of-the-art in Section 6.3.

Our BCG system can be decomposed into 3 independent components answering to specific challenges: a robust detection and extraction of cell nuclei with an approach combining a wide range of information including color, texture, scale and geometry (Section 6.4); a local frame-level grading of NA using the gRBF kernel to combine annotated medical data and formalized medical knowledge (Section 6.5); and an efficient strategy based on dynamic sampling and computational geometry tools to explore large images for the grading of entire biopsy slides within a clinically acceptable timeframe

(Section 6.6).

The BCG system is a component of the Cognitive Microscope (MICO) project[1]. MICO is an ongoing initiative funded by Agence Nationale de la Recherche (a French institution tasked with funding scientific research) and involving academic research laboratories[2,3], industrial partners[4,5,6] and pathologists from a university hospital[7]. Therefore, strong emphasis is put on the validity of the approach from a medical standpoint and its viability in a real clinical environment.

Accordingly, an empirical evaluation on clinical data provided and annotated by experienced anatomopathologists from the Pitié-Salpêtrière University Hospital in Paris is available for each component of the system. The H&E stained breast cancer slides from the dataset where digitized using an APERIO ScanScope © slide scanner and annotated using the TRIBVN ICS-framework © virtual slide browser.

To our best knowledge, our system is the first proposing a complete, full-slide approach to BCG. It is scheduled for actual clinical deployment with the whole MICO platform 2012 for validation purposes in fall.

6.2 Breast cancer grading from H&E stained surgical biopsies

Breast cancer accounts for one quarter of all cancers among the female population causing nearly half a million deaths every year [20]. Fortunately, with early enough detection, it is also one of the cancers with the highest rate of recovery. Therefore, early and accurate diagnosis of breast cancer stands as a strong medical requirement.

In recent years, histopathology which is the microscopic analysis of biological tissues became the gold standard for the diagnosis and prognosis of breast cancer. BCG is a codified protocol attributing a numerical grade according to the degree of advancement (*i.e.* malignancy) or the cancer, and is performed routinely in clinical practice [21]. The

[1] `http://ipal.cnrs.fr/project/mico`
[2] Image and Pervasive Access Lab (IPAL), Université Joseph Fourier, Grenoble, France
[3] Laboratoire d'Informatique de Paris 6 (LIP6), Université Pierre et Marie Curie, Paris, France
[4] Thales Communications & Security, France
[5] AGFA-HealthCare, Belgium
[6] TRIBVN, France
[7] Groupement Hospitalier Universitaire de la Pitié-Salpêtrière (GHU-PS), Université Pierre et Marie Curie, Paris, France

state-of-the-art BCG procedures require H&E stained slides obtained from a surgical breast biopsy. BCG from surgical breast biopsies plays a particularly important role due to the prognostic value of the grading, largely influencing decisions for the follow-up treatment of the patient.

The most common type of breast cancer is the breast carcinoma (cancer of the epithelial cells). Up to 75% of diagnosed breast cancers are invasive ductal carcinomas [63]. Accordingly, this study is restricted to the grading of invasive ductal carcinoma. The different types of breast cancer follow different BCG procedures.

In this introductory section on BCG, we first present the general workflow of the preparation of a H&E stained breast histopathology slide in Section 6.2.1. Next, the standard BCG procedures are presented in Section 6.2.2 with an emphasis on the grading of NA, a central component of BCG procedures which is the focus of our study.

6.2.1 Slide preparation workflow

The different steps for the preparation of an H&E stained surgical breast biopsy slide starting from the surgically extracted tumor are illustrated on the workflow diagram in Figure 6.1.

Precision in the process is of paramount importance in order to get a stable quality of result: slight changes in conditions such as the thickness of the layer or the time spent in the staining solutions can significantly alter the results. Even with the greatest precautions, some instability in the final quality of the image is unavoidable in daily clinical practice and needs to be dealt with, which constitutes a challenge as presented in Section 6.3.1.1.

Note that the digitization of the slide, although available in medical research, is still uncommon in today's clinical practice which is reliant on traditional optical microscopes.

6.2.2 BCG procedures for invasive ductal carcinoma

Several BCG systems with a recognized diagnostic and prognostic value can be used for invasive ductal carcinoma [79]. A BCG system is a template used to attribute a numerical score to different criteria. Several BCG systems exist for the grading of invasive ductal carcinoma [79]. Although specifics (such as the interpretation of the

Figure 6.1: Slide preparation workflow diagram. Photographs reproduced with permission from *Service d'Anatomopathologie, Groupement Hospitalier Pitié-Salpetrière, Paris, France.*

numerical scales used for the scores) can vary from a grading system to another, most popular grading systems are based on the following 3 criteria illustrated on Fig 6.2.

Nuclear atypia (NA) - Cell nuclei in malignant tumors often develop morphological irregularities. Accordingly, the study of the abnormal appearance of cell nuclei is a central aspect of BCG systems. The morphology of cell nuclei is scrutinized for any sign uncharacteristic of normal, non-cancerous cells. The more atypical the nuclei, the higher the score.

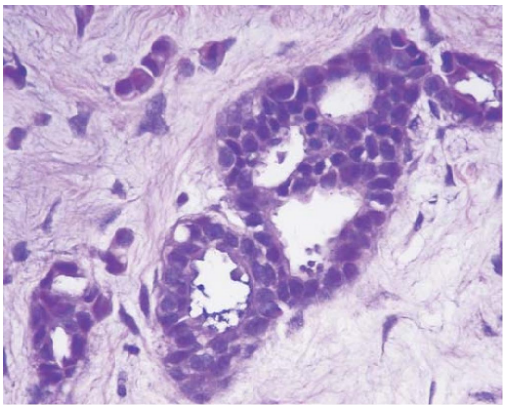
Structure of the tumor - In the earlier stages of the cancer, the tumor will usually proliferate creating gland-like patterns. This structure is progressively lost as the cancer reaches more advanced stages. Therefore, on a surgical biopsy preserving the original structure of the tissues, a score can be given according to how well differentiated a tumor is. A well differentiated tumor is given a low score whereas a poorly differentiated tumor is presumed more malignant and given a high score.

Mitotic count - The frequency of mitosis (dividing cells) is a sign of the speed at which a tumor is spreading. A low mitotic count reflects a slowly developing cancer whereas a high mitotic count indicates an aggressively spreading tumor.

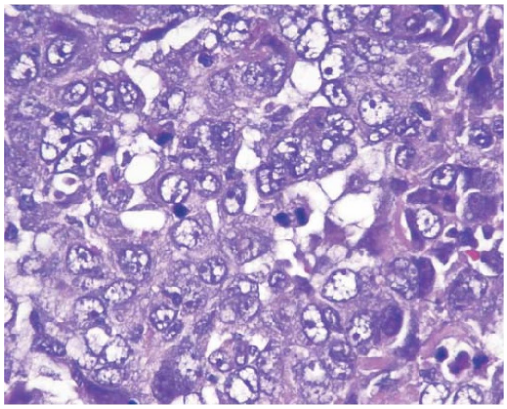
A BCG system called the "Nottingham" system [22] is well-known for being widely used in North America. It gives a score from 1 (least malignant) to 3 (most malignant) to 3 criteria: "nuclear pleomorhpism" (a particular subtype of NA), "tubular formations" (another name for glandular structures) and "mitotic count".

This present study is restricted to the assessment of NA. Unlike the other criteria which require a surgical biopsy preserving the structure of the tissues, the assessment of NA can be performed on any type of biopsy such as fine needle aspiration biopsies. Accordingly, it is a central aspect of most BCG studies.

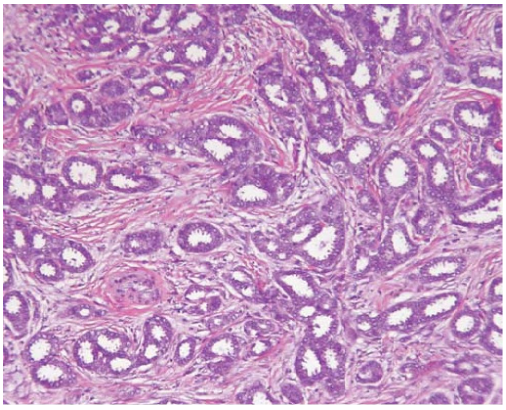
The study of NA is based on morphological features related to the size, shape and interior of the nuclei. Therefore, automatic tools able to reliably detect and extract cells from histopathological images are a strong requirement from computer aided BCG systems. More specific details on the assessment of NA are given in Section 6.5.



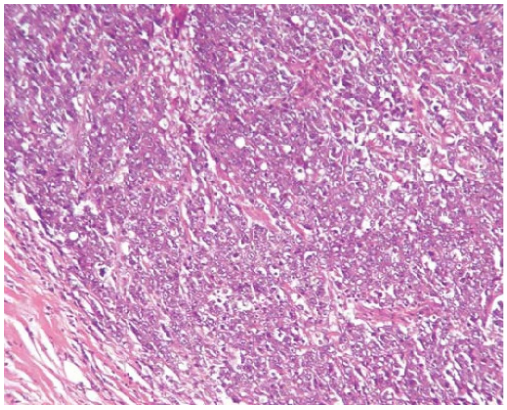
(a) A benign tumor with small and regular nuclei.



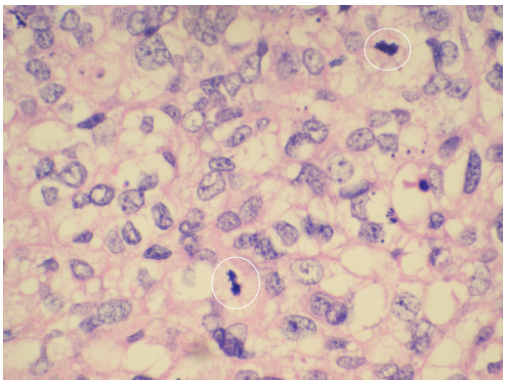
(b) A malignant cancer showing large and irregular nuclei.



(c) A well differentiated tumor shows glandular formations.



(d) A poorly differentiated tumor in more homogeneous.



(e) A few mitotic nuclei circled in white.

Figure 6.2: Main scoring criteria of BCG systems. (a)-(b) low and high nuclear atypia, (c)-(d) structured and amorphous tumor and (c) examples of mitosis.

6.3 Computer-aided BCG systems

The current clinical practice for BCG is still reliant on observations with an optical microscope. As proved by Dune and Going [18], the grading of NA is a tedious and time consuming task which outcome is highly inconsistent even for well trained specialists. Therefore, the practice would largely benefit from techniques susceptible to improve the stability of the diagnosis.

Meanwhile, the recent developments in digital histopathology have lead to the relative maturity of virtual slide technologies: full slides digitized using slide scanners can be viewed and annotated using virtual slide browsers such as the TRIBVN ICS-framework©[8]. Such new technologies can be used to partially or fully automate the process with the main benefit of improving the robustness of the grading.

In Section 6.3.1, we discuss the specific technical challenges related to the grading of NA from H&E stained surgical biopsies. In Section 6.3.2, we give a review of the current state-of-the-art regarding this task and modality.

6.3.1 Technical challenges

Three major challenges proper to the task and image modality can be identified: a computer vision challenge due to the complexity of the images making the extraction of the cell nuclei difficult, a machine learning challenge due to the scarcity of the medical data available, and a computational challenge due to the very large size of the full slide images.

6.3.1.1 Complexity of the images

H&E stained surgical breast cancer slides present particularly steep challenges compared with other types of biopsies mainly due to the great diversity of the situations encountered. High-magnification H&E breast cancer micrographs are given in Figure 6.3 to illustrate this diversity (note that the micrographs used for actual grading have a wider field).

In particular, we can point out: the heterogeneity of the nuclei and the background, the uneven and low object-background contrast (see Figure 6.3a), and the frequent

[8] website: `http://www.tribvn.com`

overlaps between the nuclei (see Figure 6.3b).

Moreover, breast ductal carcinoma are recognized for being a very heterogeneous group with regard to pathological features [63]. Therefore, the morphology of nuclei can drastically change according to the histological grade (*i.e.* malignancy of the cancer): nuclei from lower grade tumors (Figure 6.3c and Figure 6.3e) are typically much smaller, rounder and homogeneous compared to higher grade tumors (Figure 6.3d and Figure 6.3f) which can be very irregular.

Finally, the differences in slide preparation techniques and staining methods between hospitals can result in significant visual differences including color and texture as visible between Figure 6.3c and Figure 6.3d from the National University Hospital (NUH) in Singapore, and Figure 6.3e and Figure 6.3f from the Pitié-Salpêtrière University Hospital (PSL) in Paris. Accordingly, robust algorithms able to deal with the overlaps and the high variability in the images are necessary.

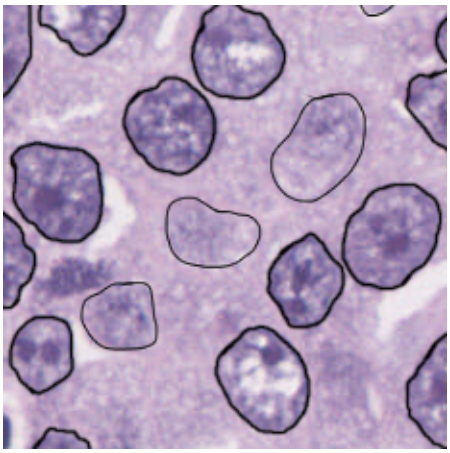
6.3.1.2 Scarcity of medical data

The current clinical practice involves traditional optical microscopes. The pathologist browses the entire slide at different resolutions and chooses a few frames for the grading, following an unrecorded procedure. The entire procedure results in a BCG report only indicating the numerical scores of the tumor. All additional information such as the specific frames chosen or the specific observations leading to the final grading is lost. This is unlike other image modalities such as mammograms (x-rays) or sonograms (ultrasounds) which can easily be annotated.

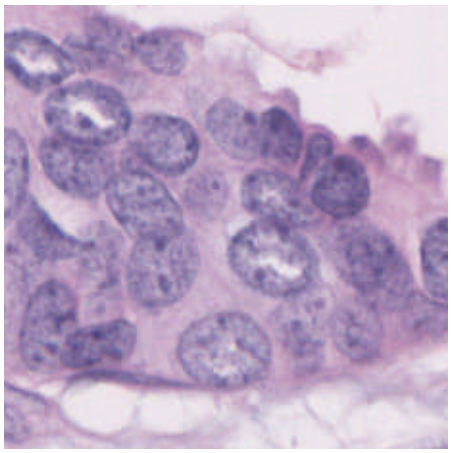
As a consequence, annotated breast cancer slides which can be used for machine learning are difficult to obtain. Considering the complexity of the BCG task, constituting a database covering a comprehensive set of possible cases is impractical if not infeasible. Instead, most of the knowledge used for grading needs to be formalized from the expertise of the pathologist rather than statistically extracted from an exhaustive database of cases with standard supervised learning methods.

6.3.1.3 Very large images

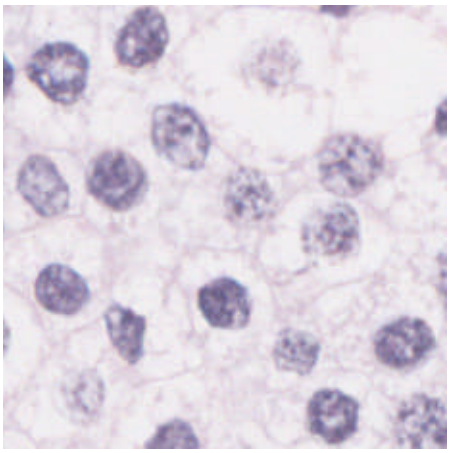
A typical breast cancer slide represents a very large amount of data. As illustrated on Figure 6.4, the area of the neoplasm (tumor) on a slide is usually much larger than a



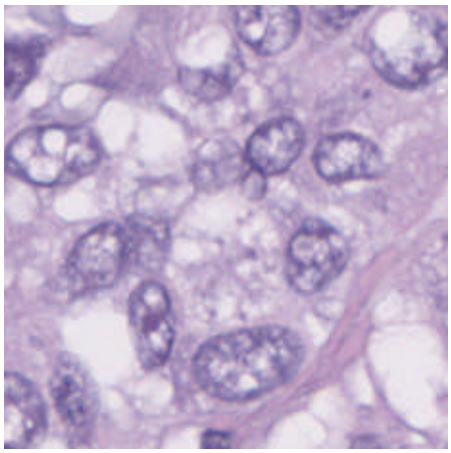
(a) Manually outlined nuclei.



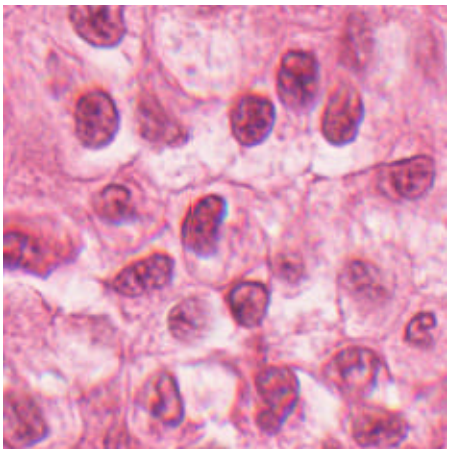
(b) Touching and overlapping nuclei.



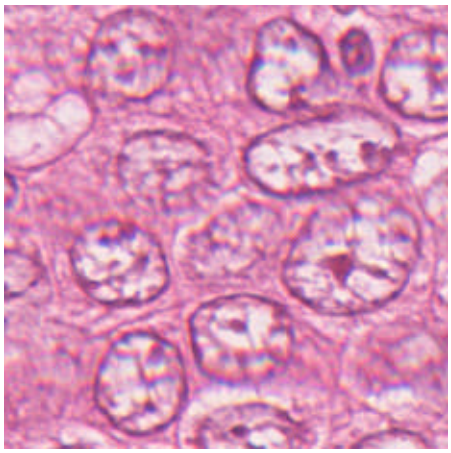
(c) NUH hospital, low grade.



(d) NUH hospital, high grade.



(e) PSL hospital, low grade.



(f) PSL hospital, high grade.

Figure 6.3: High magnification H&E breast micrographs corresponding to $57.75\mu m \times 57.75\mu m$ windows covering approx. $1/25^{th}$ of a frame typically used for grading. (a) Nuclei have heterogeneous interiors and uneven object-background contrast. Some nuclei with particularly poor object-background contrast (thinner outline) are easily missed. (b) The visual identification of nuclear boundaries is challenging due to frequent overlaps between nuclei. (c-f) The aspect of nuclei can largely change according to the grade of the cancer or subtle differences in slide preparation techniques.

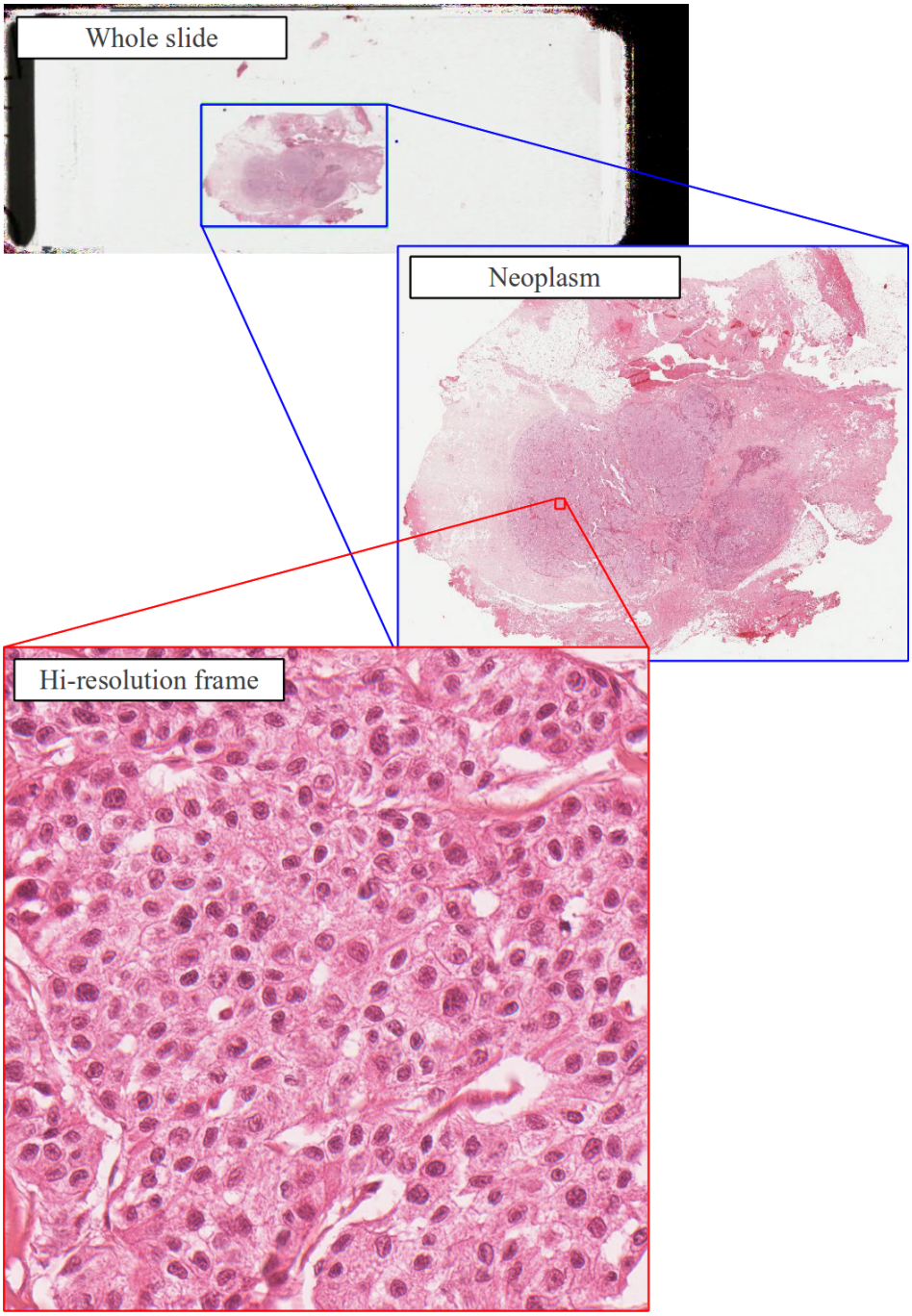


Figure 6.4: Whole slide, neoplasm and $256\mu m \times 256\mu m$ high-resolution frame typically used for the grading of NA.

high-magnification frame typically used for the grading of NA. Although specific figures will vary according to slides, tumors larger than $1cm^2$ are common, which approximately corresponds to $40 \times 40 = 1600$ frames.

The assessment of NA must be based on the region showing the highest grade of NA in the tumor. An exhaustive analysis of the entire tumor in order to find the highest grade frames is impractical due to time constraints. Therefore, a slide exploration method able to quickly and reliably find the highest grading frames must be implemented.

6.3.2 State-of-the-art review

The problem of computer-aided breast cancer diagnosis has already been the focus of several works. For reference, a broad overview is available in Subramaniam et al. [77]. A majority of the previous work deals with other modalities than histopathological images such as x-ray mammograms.

A comparatively smaller amount of methods is related to the diagnosis of breast cancer from histopahological images. Gurcan et al. [26] have compiled a more recent review specific to histopathology (though not limited to breast cancer). However, the largest part deals with Fine Needle Aspiration (FNA) biopsies, a less challenging type of biopsy which consists in well-separated cell nuclei over a well-contrasted background on a much smaller image. A small amount of cells is extracted with a needle and deposited on a clean glass slide. With FNA biopsies, the objective is not to perform a precise grading with a prognostic value but rather to detect the presence of cancerous cells.

Among the methods dealing with FNA biopsies we can note the early work from Schnorrenberg et al. [64, 65] based on receptive fields for the detection of nuclei and a neural network to classify the individual nuclei as cancerous or non cancerous, the method from Street [76] segmenting nuclei with edge detection techniques and an ellipsoidal approximation by generalized Hough transform, and the system from Estévez et al. [19] using the texture of nuclei and fuzzy-finite state machines to classify the individual nuclei.

Methods for the extraction of cell nuclei were also proposed on a number of other modalities. This includes the work by Yang et al. [97] on time-lapse fluorescence image sequences in which nuclei are bright objects on a dark background, so they can be

easily extracted from background by thresholding. Yang et al. [96] also proposed a method based on Active Contour (AC) models to accurately delineate lymphocytes on blood smears which present a clear image background so cell boundaries can be easily identified.

The relevant previous work on H&E stained breast biopsy images presented below can be divided into the following categories according to their main focus: methods dealing with the detection of cell nuclei, methods also addressing the problem of their accurate extraction (delineation of their boundaries) and methods focused on providing a diagnosis of the pathology.

6.3.2.1 Detection of nuclei

A number of methods are aimed at the detection of cell nuclei from H&E stained cancer biopsies which is a relatively easier problem than their precise extraction. Most of these works are based on adaptive thresholding on the RGB image. A system able to label several histological and cytological microstructures in high resolution frames of H&E stained breast cancer slides, including different types of cell nuclei was proposed by Petushi et al. [56, 57]. The method uses Otsu thresholding and morphological operations. Sertel et al. [70] also proposed a method able to detect nuclei of centroblast cells (large malignant cells) on H&E stained histology images of follicular lymphoma. The color band having the highest contrast is selected and a locally adaptive thresholding is performed.

6.3.2.2 Extraction of nuclei

Previous works aimed at accurately delineating nuclei on H&E stained biopsies are usually based on image gradient. Ali and Madabhushi [1] proposed an AC-based extraction method using a watershed segmentation for the initialization. A computationally efficient method has been proposed by Dalle et al. [10] using local polar transforms of the gradient field of the original image. Recently, Kulikova et al. [35] proposed a stochastic method based on a Marked Point Process (MPP) with AC models and object shape priors.

6.3.2.3 Diagnosis of breast cancer

A number of previous works, which are not BCG systems per se, are able to differentiate between normal tissue and cancerous tissue from a single high-magnification frame. Doyle et al. [16] used geometrical features from the spatial distribution of the nuclei, and Wang and Wan [90] used geometrical features and SVMs with asymmetrical margins.

Oger et al. [52] proposed a rare type of application focusing on the analysis of the whole slide at low magnification. Low resolution analysis of the whole slide is necessary in order to spot the relevant tumoral tissues from other tissues. The system is able to distinguish regions corresponding to invasive ductal carcinoma, invasive lobular carcinoma, colloid carcinoma and fibroadenoma.

So far, Dalle et al. [9, 10] proposed the only method presented as a grading solution. It claims to perform BCG on a single frame following the Nottingham system. Nuclear pleomorphism (a subtype of NA in the jargon of the Nottingham system) is graded by classifying each of the nuclei as low, medium or high grade. Unfortunately, it reflects a number of misunderstandings from the medical standpoint: for instance, it considers a frame-based problem whereas BCG is a slide-based procedure and is based on a medically incorrect interpretation of the notion of nuclear pleomorphism.

6.3.2.4 Discussion and identification of gaps

First, the previous "diagnostic" applications able to label a single frame as cancer or non-cancer do not have real clinical relevance for grading purposes. Without denying the interest of such work from the computer vision standpoint, the clinical significance of performing BCG is not to diagnose if the tissue is tumoral (which is already established since the biopsies are obtained from surgically extracted tumors), but rather to grade the severity of the cancer for prognostic purposes.

Moreover, the previous works do not consider the problem posed by the analysis of very large images. They consider a frame-based problem whereas actual BCG is a slide-based problem. The only slide-based method by Oger et al. [52], which is not a grading system, deals with the whole slide at low-magnification and does not provide a solution for processing the entire slide at high-magnification.

To our best knowledge, none of the previous methods on the detection and extraction

of nuclei was proven to perform well with H&E stained images representing high-grade (malignant) cancers and examples of good results are only available for images presenting low histological grades and isolated nuclei. This is a great limitation for clinical applications which require good results with all histological grades including the more challenging high grades.

In our opinion, the reliance on the image color intensity and gradient field alone as in the previous methods is not sufficient to deal with the complexity of the H&E stained breast surgical biopsy images and in particular the irregularity of the high grade images as detailed in Section 6.3.1.1.

This provides a motivation to our approach detailed in Section 6.4 consisting in incorporating additional, higher-level information such as texture, scale and geometry with a machine learning framework. The resulting image modality has characteristics stable enough to allow for an accurate extraction of the nuclei robust to variations in histological grades or other conditions affecting the aspect of the images.

A thorough empirical comparison available in [35] of state-of-the-art methods on clinical data validated by pathologists suggests their MPP-based approach gives the best overall performances for detection and extraction by a good margin. Accordingly, the final extraction of nuclei from the new image modality is performed using an MPP-based method as described in Section 6.4.1.4.

6.4 Extraction of cell nuclei

In the current and following sections, we present our complete solution for the automatic grading of NA from H&E stained surgical breast cancer slides. Our system can be decomposed into 3 independent components: the detection and extraction of cell nuclei (Section 6.4), the local grading of NA on individual high-magnification frames using annotated medical data and formalized medical knowledge (Section 6.5), and the grading of full slides (Section 6.6).

As pointed out in Section 6.3.1.1, H&E stained surgical biopsies present a particularly steep computer vision challenge. A number of methods have already been proposed for the automatic detection and extraction of nuclei from histopathological images and are reviewed in Section 6.3.2. Several methods are able to reliably detect isolated nuclei

or accurately extract them from comparatively less challenging images such as FNA biopsies which present a clear background or biopsies with low histological grades which present regular nuclei. However, to our best knowledge, no method is yet able to accurately and reliably extract the nuclei from images covering a wide range of histological grades. Therefore, previous methods lack the robustness required for clinical applications.

In this section, we propose a robust method for the extraction of nuclei from H&E stained surgical breast cancer slides. Our approach consists in substituting the original H&E image with a new image modality created using a wide variety of information from the original image including: color, texture, scale and geometry. The new image modality is a grayscale map where the value of each pixel is a probability estimate (between 0 and 1) indicating whether or not the pixel belongs to a nuclei. A fully detailed description of the method is available in Section 6.4.1. Regardless of the histological grade, the resulting modality presents stable characteristics including a strong object-background contrast, and homogeneous nuclei and background, greatly facilitating the subsequent extraction of the nuclei.

The actual extraction is performed from the new image modality using a method based on MPP, a methodology for the extraction of multiple, arbitrarily-shaped objects from images using shape priors [34]. The MPP-based method used in this paper is able to deal with overlapping objects through the use of shape priors.

A validation proposed in Section 6.4.2 on real clinical data provided and annotated by pathologists from different cases of breast cancer representing a wide range of histological grades shows that our method greatly improves the the detection of the nuclei and the accuracy of their extraction.

6.4.1 Method

Our method involves the creation of a grayscale map incorporating color, texture, scale and geometrical information from which the nuclei are extracted using an MPP-based approach. The process can be divided into 4 successive steps:

1. First, the haematoxylin and the eosin from the H&E stain are separate by applying a color deconvolution to the original H&E image (Section 6.4.1.1).

2. Then, a first probability map is computed from local features based on color, texture and scale. The probability estimates associated to each pixels are obtained by using SVM classification and rescaling the output (Section 6.4.1.2).

3. A second probability map is then computed using similar methods from the previous local features and new geometrical features. The geometrical features are computed using the first map. The addition of geometrical information allows a significant intra-nuclear and background noise reduction (Section 6.4.1.3).

4. Finally, the extraction of nuclei is performed from the second map using an MPP-based method described in Section 6.4.1.4.

The different steps are summarized on the workflow diagram in Figure 6.5.

6.4.1.1 H&E color deconvolution

First, a color deconvolution as described in [60] is applied in order to separate the haematoxylin and the eosin from the original H&E stain. Mathematically, it can be summarized as a change of basis from the original RGB basis $B_{\text{RGB}} = I_3$ (the 3-by-3 identity matrix) to a new basis of normal vectors $B_{\text{HE}} = (\vec{h}, \vec{e}, \vec{r})$. $\vec{h}$ (resp. $\vec{e}$) is a vector of 3 elements corresponding to the average color of haematoxilin (resp. eosin) stains in the RGB system and $\vec{r}$ is a complementary color such that:

$$\vec{h} \otimes \vec{h} + \vec{e} \otimes \vec{e} + \vec{r} \otimes \vec{r} = \vec{1} \tag{6.1}$$

where $\otimes$ designates the component-wise product of vectors. In practice, if (6.1) yields negative components for $\vec{r}$, we take 0 instead.

The specific values of $\vec{h}$ and $\vec{e}$ depend on several factors such as the specific solutions used for staining, the thickness of the cut or the microscope/slide scanner used for the acquisition of the image. For an optimal quality of results, our values are calibrated using slides stained with only one of the colors but otherwise prepared and digitized by the pathologists in the same conditions as the H&E slides.

As illustrated on Figure 6.6, the deconvolution uses colors from the the monochromatic sample slides to separate the haematoxylin and eosin from the original H&E image. The channel corresponding to the complementary color $\vec{r}$ contains only residual

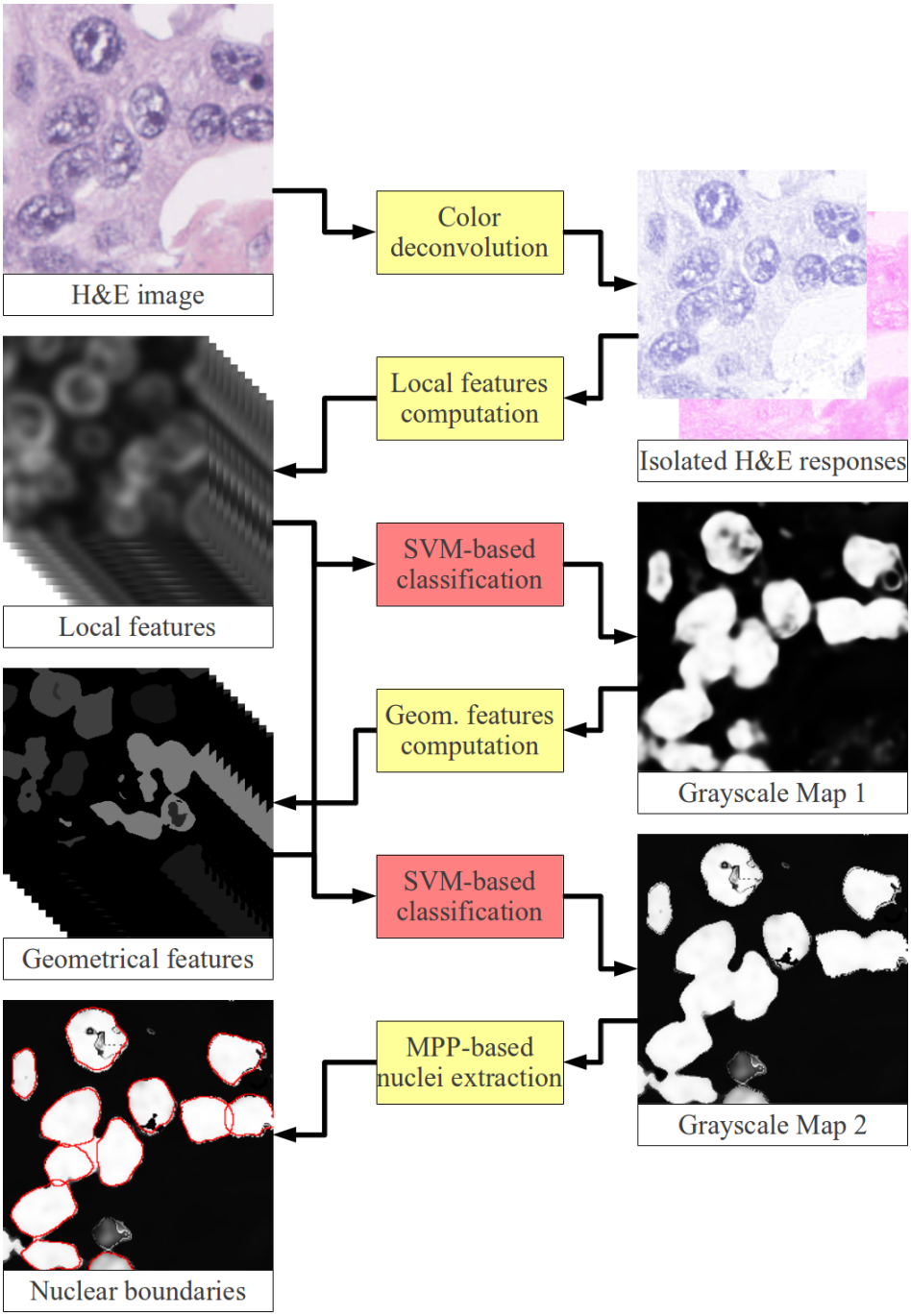
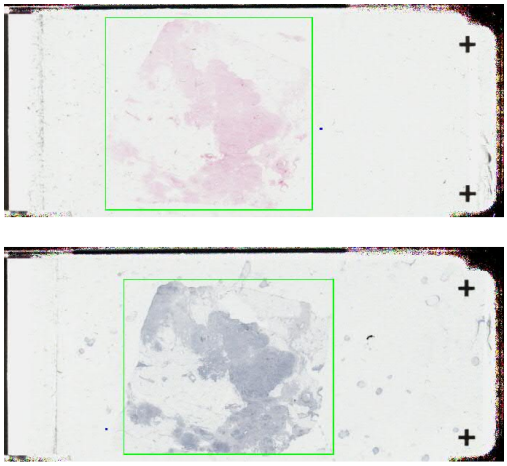
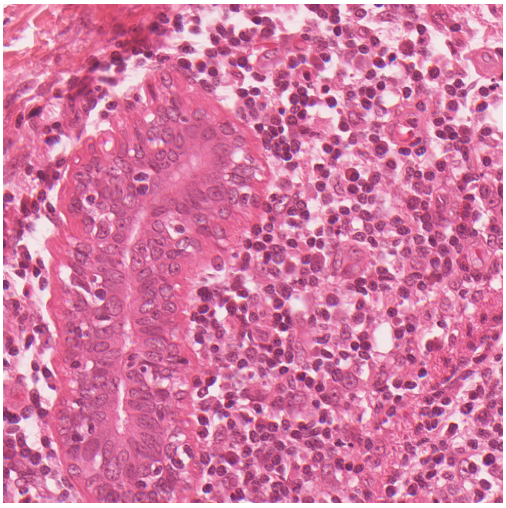


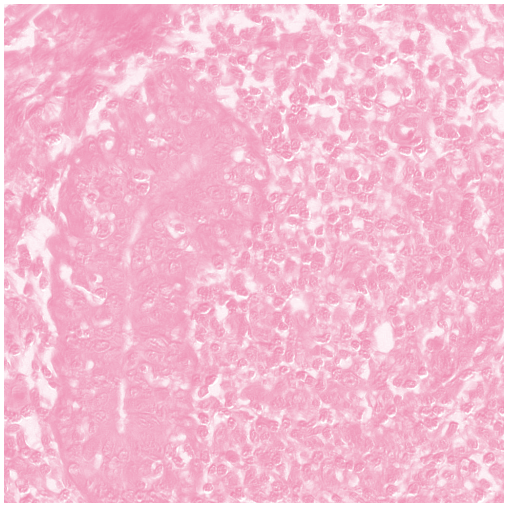
Figure 6.5: Workflow diagram for the extraction of nuclei from H&E stained histopathological images.



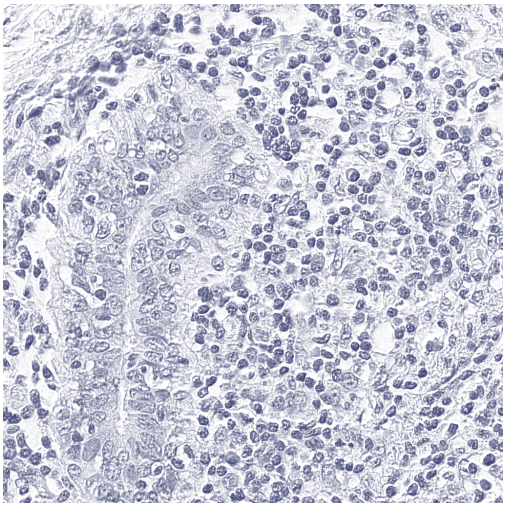
(a) Monochromatic eosin (top) and haematoxilin (bottom) slides for calibration.



(b) H&E stained frame.



(c) Isolated eosin response mostly revealing stroma.

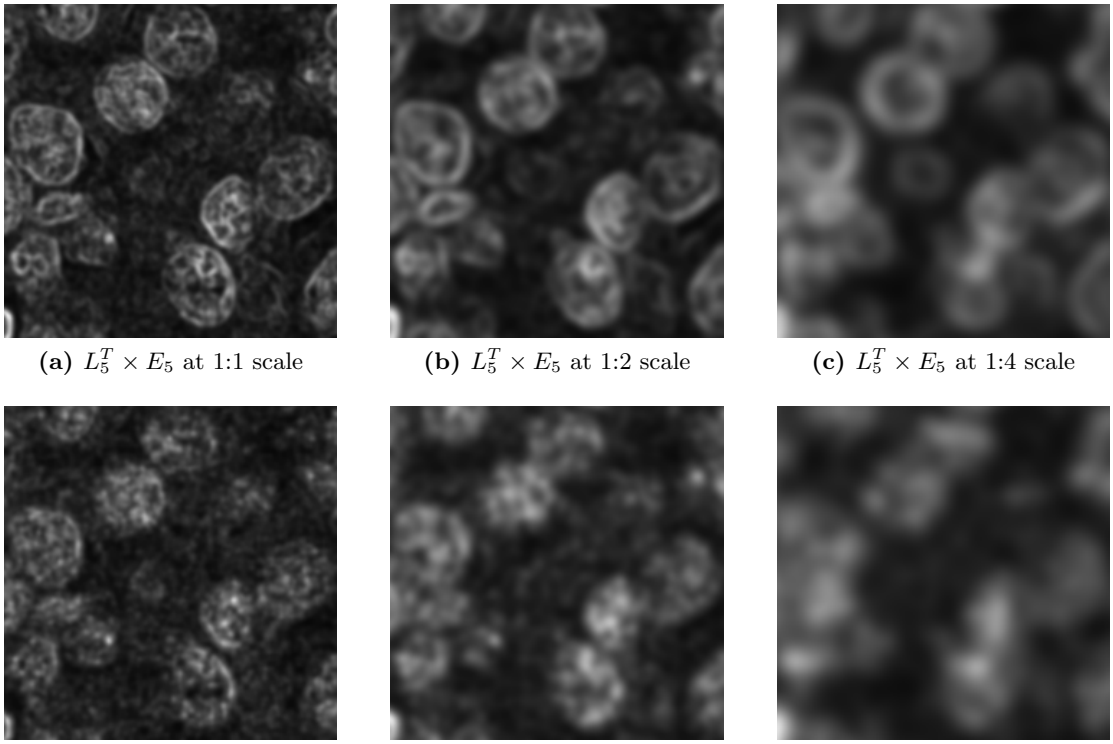


(d) Isolated haematoxilin response mostly revealing nuclei.

Figure 6.6: Color deconvolution applied to an H&E stained $256\mu m \times 256\mu m$ frame typically used for grading.

noise and is discarded.

6.4.1.2 Map from local features



(a) $L_5^T \times E_5$ at 1:1 scale **(b)** $L_5^T \times E_5$ at 1:2 scale **(c)** $L_5^T \times E_5$ at 1:4 scale

(d) $L_5^T \times W_5$ at 1:1 scale **(e)** $L_5^T \times W_5$ at 1:2 scale **(f)** $L_5^T \times W_5$ at 1:4 scale

Figure 6.7: Example of local texture feature corresponding to two different kernels at different scales on a high-magnification portion of the haematoxylin image.

During this step, the images obtained from the color deconvolution are used to compute a total of 120 local features (60 from the eosin image and 60 from the haematoxylin image) for every pixel using texture information at different scales. Then, a probability estimate is computed for each pixel based on SVM classification and rescaling of the output.

The local features are based on Laws' texture measures [39] which are the response to a set of 5-by-5 convolution kernels. The 5-by-5 kernels are generated from 5 different

1-by-5 base kernels:

$$\begin{aligned} L_5 &= (1, 4, 6, 4, 1) \\ E_5 &= (-1, -2, 0, 2, 1) \\ W_5 &= (-1, 2, 0, -2, 1) \\ S_5 &= (-1, 0, 2, 0, -1) \\ R_5 &= (1, -4, 6, -4, 1) \end{aligned} \tag{6.2}$$

A total of 25 different 5-by-5 kernels are computed by taking the product of every vertical 5-by-1 kernel with every horizontal 1-by-5 one. The 5-by-5 kernels are applied at every pixel to extract 25 features which are then combined into 15 rotationally invariant features after normalizing by the output of the $L_5^T \times L_5$ kernel and smoothing with a Gaussian kernel of standard deviation $\sigma = 1.5$ pixels.

The same process is repeated at 4 different scales using low-pass filtering with Lanczos filters [17]. In practice, local texture features are computed at 1:1, 1:2, 1:4 and 1:8 scales for every pixel after resampling the 5-by-5 convolution kernel into 10-by-10, 20-by-20 and 40-by-40 convolution kernels with the following 2-dimensional filter:

$$L(x, y) = l(x)l(y) \tag{6.3}$$

with:

$$l(x) = \begin{cases} \frac{3\sin(\pi x)\sin(\pi x/a)}{\pi^2 x^2} & \text{if } x \in [-3, 3] \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

An illustration of the result for 2 specific feature at different scales is given in Figure 6.7.

For every pixel represented by its feature vector $\vec{x}$, its probability $p_{\mathrm{n}}(\vec{x})$ of belonging to a cell nuclei is obtained in 2 steps. First, the class of the pixel is predicted using SVM classification, then the output of the SVM is rescaled into a probability estimate belonging to $[0, 1]$ using a softmax transform.

We use the C-SVM with the RBF kernel K_{rbf}. The resulting labeling model $f(\vec{x}) = \sum_{i=1}^{N} \alpha_i K_{\mathrm{rbf}}(\vec{x}, \vec{x_i}) + b$ is an affine combination of kernel sections. The training sets are

created by selecting pixels from images where the nuclei have been manually delineated by pathologists.

Following a method detailed in [61], the output $f(\vec{x}) \in \mathbb{R}$ is rescaled into a probability estimate $p_{\mathrm{n}}(\vec{x}) \in [0, 1]$ using a softmax transform:

$$p_{\mathrm{n}}(\vec{x}) = \frac{1}{1 + \exp \frac{f(\vec{x})}{\sigma_f}} \tag{6.5}$$

A normalization by σ_f which is the variance of f over the entire image is necessary since the values of f can be more-or-less spread out over the data.

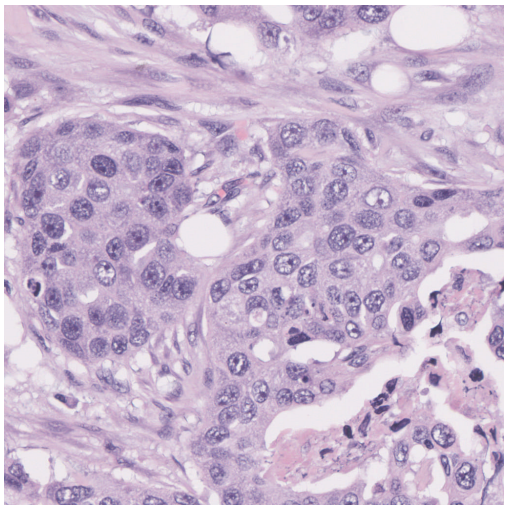
As shown in Figure 6.8c, the resulting probability map exhibits strong contrast with objects clearly distinguishable from the background. Moreover, nuclei and background appear significantly more homogeneous than in the original image.

6.4.1.3 Incorporating geometrical information

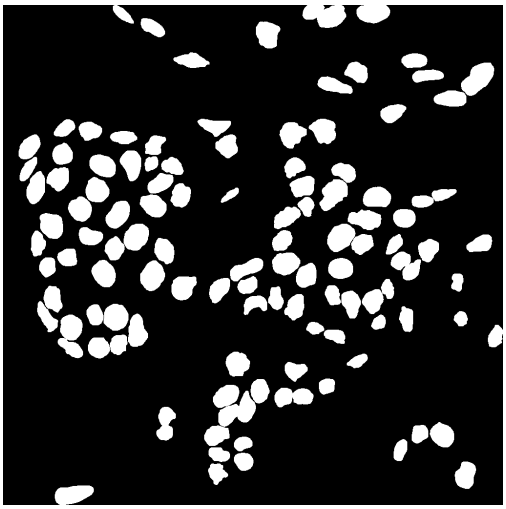
A significant amount of intra-nuclear and background noise is still present in the probability map obtained with local features alone. In order to mitigate this issue, we propose to compute a new map incorporating information about the geometry of the objects in the image.

The geometrical information is derived from Connected Components (CC) obtained by applying global thresholding to the initial map obtained in Section 6.4.1.2. CCs are computed for a set of threshold values $\{t_m = 0.5 + 0.05m | m \in [\![-5, 5]\!]\}$. The CCs for a given threshold value form a partition of the image, therefore, every pixel from the original image is associated to 11 CCs it belongs to (one per threshold value). Subsequently, 12 features are computed from each CC which results in a total of 132 geometrical features for each pixel.

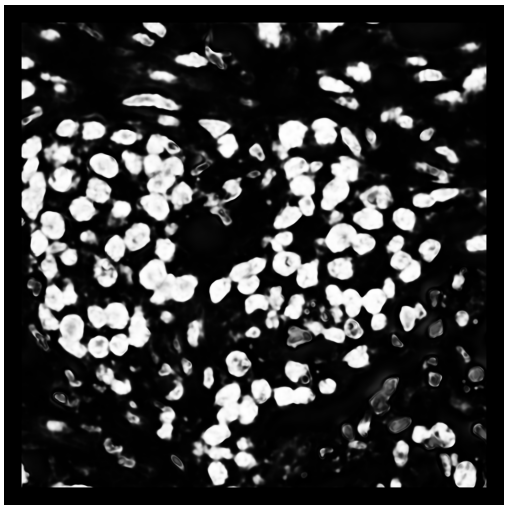
The first 6 features associated to a CC are: the mean and variance of the pixel intensity on the first probability map, the area, the perimeter, the roundness ρ (zeroth-order regularity) and elasticity λ (first-order regularity) of the exterior boundary. ρ and λ are computed following a method suggested in [32] using a representation of the boundary as a 2π-periodic closed curve $\vec{\gamma} : \mathbb{R} \to \mathbb{R}^2$ parametrized such as the speed



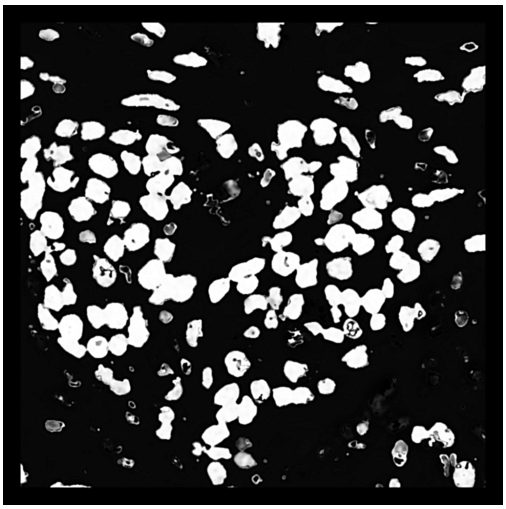
(a) Original H&E image.



(b) Binary mask from a manual delineation of most nuclei.



(c) Probability map from local features.



(d) Probability map from local and geometrical features.

Figure 6.8: Examples of probability maps over a $256\mu m \times 256\mu m$ frame.

along the curve is constant:

$$\forall s,\ \|\frac{\partial \vec{\gamma}}{\partial s}(s)\| = c \tag{6.6}$$

Subsequently,

$$\theta(s) = \left(\vec{u}, \widehat{\frac{\partial \vec{\gamma}}{\partial s}(s)}\right) \tag{6.7}$$

is defined as the angle between a fixed reference vector $\vec{u}$ and the tangent to the curve. Then, the elasticity ϵ can be defined as:

$$\lambda(\gamma) = \int_0^{2\pi} \left(\frac{\partial \theta}{\partial s}(s)\right)^2 ds \tag{6.8}$$

and the roundness ρ as:

$$\rho(\gamma) = \int_0^{2\pi} |\theta(s) - s|\, ds \tag{6.9}$$

Note that $\theta(s) = s$ corresponds to a perfect circle. The remaining 6 features are the same 6 features for the CC wrapping around this CC.

The 132 new geometrical features are added to the previous 120 local features from Section 6.4.1.2 and a second probability map is computed following a similar procedure (SVM classification and rescaling of the output). Figure 6.8d is the resulting probability map after incorporation of the geometrical information. Compared to the first map (Figure 6.8c), we can see that the background and intra nuclear noise levels are further reduced and that the result is visually closer to the manual extraction on Figure 6.8b.

6.4.1.4 MPP with shape priors for nuclei extraction

The actual extraction of cell nuclei is performed from the probability maps incorporating geometrical information. Stochastic MPPs are a well known methodology for the extraction of multiple objects from images. They were first applied for the extraction of objects of simple geometrical shapes from remote sensing images [55] and were subsequently extended to potentially arbitrarily-shaped objects [34]. A recent comparative study [35] showed that applied to the extraction of nuclei from H&E stained biopsy

images, they offer better results than other existing state-of-the-art methods.

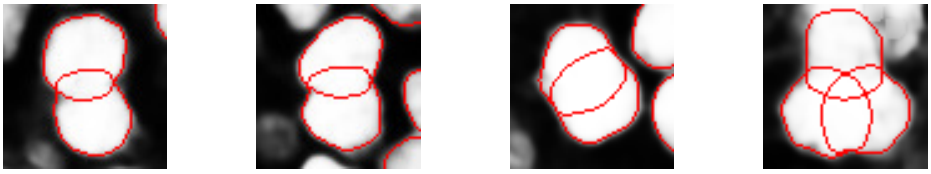
The method uses AC models incorporating shape priors to extract the objects from the image. However, unlike the active contour based methods presented in section 6.3.2 which require a prior detection of the objects, the MPP framework constructs the objects using a methodology known as "high order AC" which does not require the location or the number of objects to be known in advance. The optimal configuration of objects in the image is obtained by sampling from the Gibbs probability distribution using a Markov chain, which consists of a discrete-time multiple birth-and-death process following a logarithmic simulated annealing schedule to minimize the overall configuration energy. The discrete process converges to a continuous-time process reaching a global optimum as detailed in [14]. Full technical details on the method are available from [35].

The energy $E(\gamma)$ associated to a nucleus boundary γ is a weighted sum of an image term $E_i(\gamma)$ and a shape term $E_s(\gamma)$. The latter is itself the weighted sum of a smoothing term $E_{sm}(\gamma)$ and a shape prior term $E_{sp}(\gamma)$. The shape prior term:

$$E_{sp}(\gamma) = \frac{1}{2\pi} \sum_{k \in \mathbb{Z}} f_k \left| \int_{[0,2\pi]} \exp(-ikt) \delta r(t) dt \right|^2 \tag{6.10}$$

allows or restricts the perturbations $\delta r(t)$ of the boundary from a circle at a specific frequency k by tuning the coefficient $f_k \geq 0$.

In particular, the shape prior information allows to properly extract overlapping nuclei according to their expected shape without arbitrarily discarding the overlapping parts as shown in Figure 6.9.



(a) **(b)** **(c)** **(d)**

Figure 6.9: Overlapping nuclei extracted using shape priors.

6.4.2 Empirical study

6.4.2.1 Data

The data used for validation corresponds to slides from 5 breast cancer patients graded by the pathologist following the Nottingham system and covering a wide range of histological grades including the lowest (TF1-MC1-NP1) and the highest (TF3-MC3-NP3) possible grades. The gradings were independently performed by 2 experienced pathologists and found to be concordant.

From each slide, a $256\mu m \times 256\mu m$ frame at a resolution of $0.25\mu m$/pixel was selected in the tumoral region which typically corresponds to a region observed through an optical microscope at a $40\times$ magnification during grading. A total of 862 cell nuclei were identified and manually delineated by the pathologist in the 5 frames.

The manual annotations are used both to create training sets for the SVMs and evaluate the methods. It is important to note that although performed by an expert pathologist, the manual delineation is inherently subjective due to the ambiguity of the images and the relative imprecision of work done manually. In particular, some nuclei are left out and the delineation must sometimes rely on guessing, specially when overlaps are present. Therefore, the work should rather be considered as a bona fide annotation effort from an expert pathologist rather than an unquestionable ground truth, which is not possible to obtain.

The validation was performed using a leave-one-out scheme with each frame successively used for validation and the remaining 4 used for training from which 100 intra-nuclear pixels and 100 background pixels are randomly selected to constitute the training sets for the SVMs.

6.4.2.2 Evaluation metrics

The methods are first assessed for the detection of the nuclei and subsequently for the accuracy of the extraction of the detected nuclei. From this point on, a nuclei extracted by the method will be referred to as a "candidate" and a manually delineated nuclei as a "reference".

First, the best 1-to-1 mapping between the candidates and the references is found. Here, the best mapping is defined as the one maximizing the total overlapping area

between candidates and references. This assignment problem can be solved in $O(n^3)$ using the "Hungarian" method [37] where n is the amount of objects. Let p be the number of pairs established (i.e. the number of well-detected nuclei), r be the number of reference nuclei and c be the number of candidates.

The quality of the detection is evaluated by measuring the precision and the recall rate of the detection. The precision score, defined by $\text{prec} = \frac{p}{c}$, measures the proportion of true positives among all the cells detected by the algorithm. The recall score defined by $\text{rec} = \frac{p}{r}$ measure the proportion of actual positives with are correctly recognized by the algorithm.

The accuracy of the extraction is evaluated for every pair in the mapping with its Jaccard index. For ever candidate-reference pair (A_i, B_i), the Jaccard index is defined as: $J_i = \frac{|A_i \cap B_i|}{|A_i \cup B_i|}$. The score ranges from 0 (no overlapping) to 1 (perfect correspondence). A global extraction score for the N pairs is computed by taking the arithmetic mean of the individual Jaccard indices: $\text{acc} = \frac{1}{N} \sum_{i=1}^{N} J_i$.

6.4.2.3 Results and discussion

In this section, we compare the detection and extraction performances of the MPP-based algorithm applied to 3 different image modalities: the luminosity of the original H&E image (as most of the existing methods presented in Sec. 6.3.2), the first map using local information only and the second map incorporating the geometrical information. The modality-dependent parameters of the method are tuned to reach a comparable sensitivity on the different modalities.

	n	prec	rec	acc
luminosity	646	0.627	0.470	0.403
first map	623	0.828	0.599	0.690
second map	641	0.832	0.618	0.686

Table 6.1: Numerical results for the detection and extraction of nuclei. prec, rec and acc are compared for an extraction using the luminosity of the original H&E image, the first map using local features only and the second map incorporating the geometrical features. n is the amount of candidate nuclei detected by the method.

Table 6.1 summarizes the numerical results for the detection and the extraction of nuclei. Note that it is unrealistic to expect figures close to 100% due to the subjectivity inherent to the manual annotations, as discussed in Sec. 6.4.2.1.

First, we notice that the amount n of detected nuclei is relatively stable in the 623-646 range implying that the sensitivity of the MPP-based extraction method is calibrated equivalently for the 3 modalities. The first probability map increases the precision rate of the detection by more than 20 percentage points and the recall rate by nearly 12 points. The second probability map with additional geometrical information further improves the precision by 0.4 points and the recall rate by 1.9 points. The accuracy of the extraction is also greatly improved by the use of the probability maps (nearly 30 points).

Figure 6.10 provides a visual illustration of the improvements achieved by the use of the new modality, with and without geometrical information, on a portion of high-grade cancer frame.

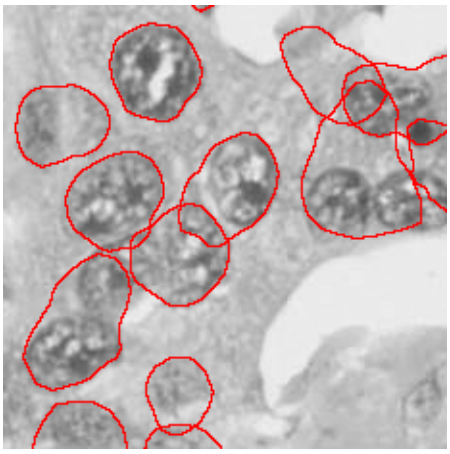
In conclusion, by integrating a wide variety of information including color, texture, scale and geometry into a unified framework, our method succeeds in greatly improving the detection and extraction of nuclei from histopathological images. In particular, our method produces a new, stable image modality which provides the robustness to deal adequately with very irregular, high-grade cancers.

6.5 Grading of nuclear atypia

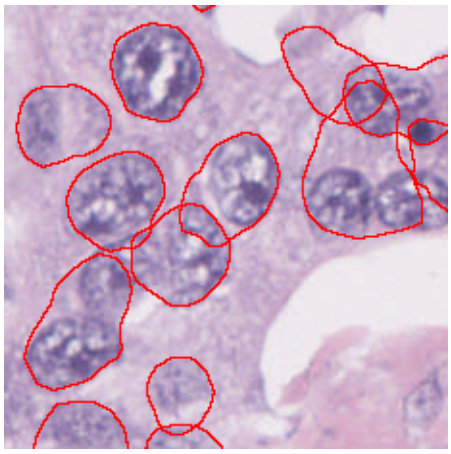
The grading of NAs consists in giving a numerical grade to individual high-magnification frames according to the severity of the NAs observed on the cell nuclei. The numerical grade corresponds to a judgement on the overall situation of the NAs and is attributed by the pathologist without providing additional details.

Nevertheless, the concept of NA covers several specific aspects of the morphology of the nuclei. The following is an attempt made under the supervision of expert pathologists at formalizing the different aspects covered by the notion of NA.

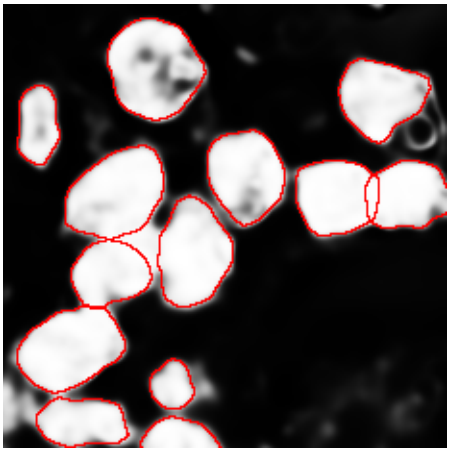
Macrokaryosis – It designates the presence of nuclei larger than their normal size. Nuclei from normal epithelial cells have a stable and small size, whereas cancerous nuclei have an increased nuclear size. This is due to the fact that normal nuclei have a fixed amount of chromosomes whereas cancerous nuclei may have more chromosomes. As a practical rule of thumb used by the pathologists, non cancerous



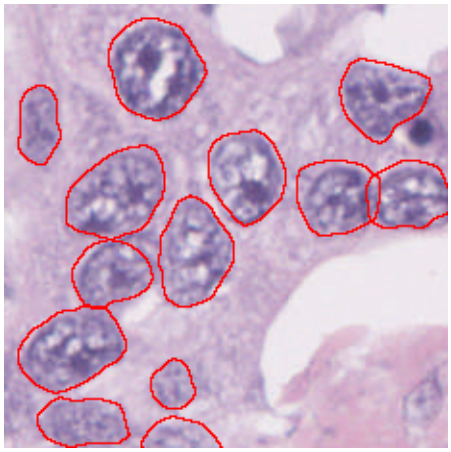
(a) From luminosity.



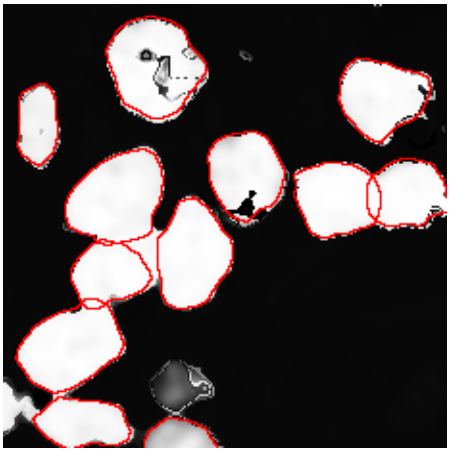
(b) From luminosity, transposed to H&E image.



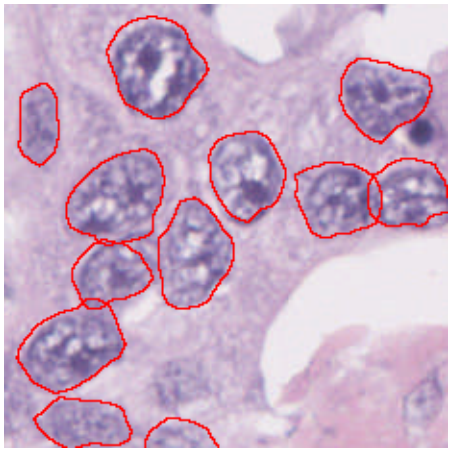
(c) From first map.



(d) From first map, transposed to H&E image.



(e) From second map.



(f) From second map, transposed to H&E image.

Figure 6.10: Side-by-side examples of extracted nuclei in a small $57.75\mu m \times 57.75\mu m$ window showing high-grade cancer using the different modalities, and transposed back to the original H&E image.

nuclei are approximately 2.5 times larger than the nuclei from inflammatory cells. Cells with nuclei more than 3 times this normal size can be considered exceptionally large.

Nuclear pleomorphism – It designates the presence of differences between the sizes and shapes of nuclei. Macrokaryosis does not occur evenly for all the nuclei in the tumor. Therefore, malignant nuclei will usually show size and shape variations within a same frame.

Homogeneity of the chromatin – Normal chromatin called "euchromatin" is homogeneous in appearance whereas pathological chromatin called "heterochromatin" forms small clusters. Therefore, the heterogeneity of the chromatin is a sign of malignancy.

Amount and size of nucleoli – Nucleoli are structures found within the nuclei of active cells. Epithelial cells from a normal, non lactating breast have a low activity and should seldom have any nucleoli. In contrast, cells from aggressively spreading cancers have more numerous and larger nucleoli.

Thickness of the nuclear membrane – The presence of heterochromatin on the nuclear membrane of cancerous cells causes it to become thicker.

According to the pathologists, macrokaryosis is the single most informative subtype of NA. However, many BCG systems such as the Nottingham system put the focus on the nuclear pleomorphism which is an indirect consequence of macrokaryosis. From our understanding, this in not due to medical reasons but rather to the constraints imposed by standard optical microscopes. Indeed, the precise size of objects is difficult to evaluate on an optical microscope, whereas objects can easily be compared side-by-side. This also explains why the stable size of inflammatory nuclei is used as a reference by the pathologists.

6.5.1 Method

As detailed in Section 6.3.1.2, labeled medical data which can be used as training data for the problem is hard to obtain. In particular, it is difficult to construct a full training set covering the different possible cases of NA in an exhaustive fashion. Therefore, we

choose to perform the actual grading using the ϵ-SVR together with the gRBF kernel described in Section 4.5.

The feature model computed from the extracted nuclei is presented in Section 6.5.1.1 and the labeled knowledge sets are presented in Section 6.5.1.2.

6.5.1.1 Feature model

For each frame, a set of 21 features is computed from the nuclei extracted using the method described in Section 6.4.

First, 5 values are computed for every individual nucleus including: its area α, the roundness ρ and elasticity λ of the contour (see Section 6.4.1.3), and the mean h_μ and standard deviation h_σ of the intensity of haematoxilin found inside. Then, the frame-based features are computed by taking the mean, variance, minimum and maximum of the above values. The total amount n of nuclei in the frame is also added, which makes a total of 21 features for each frame.

The full set of features covers the different aspects of the definition of NA. The concept of macrokaryosis is captured by the average and maximal values of α. Moreover, the concept of nuclear pleomorphism is well represented by the standard deviation of α, and by the features computed from ρ and λ. Finally, although we are unable to explicitly detect the nucleoli or the nuclear membrane, the last 3 concepts have an impact in terms of texture of the nuclei which is captured by the features computed from h_μ and h_σ.

6.5.1.2 Knowledge sets

A total of 3 labeled knowledge sets were constructed by interpreting the medical knowledge previously formalized with the help of the pathologists. All of them can be represented as unbounded orthotopes which is important for computational reasons (see Section 4.5.3.2).

On one hand, the definition of macrokaryosis implies that nuclei not exceeding 2.5 times the size of nuclei from inflammatory cells can be considered as normal. We deduce from actual measurements performed on the virtual slides that this corresponds to an area of $30\mu m^2$. Following this observation, we can construct the first labeled set $(\mathcal{X}_1, v_m)$ where v_m is the minimal score used by the pathologist on the grading scale and $\mathcal{X}_1$ is the half-space for which the mean value of α is smaller than $30\mu m^2$.

On the other hand, the definition also implies that nuclei larger than 3 times this size are highly abnormal. We can construct the second labeled set $(\mathcal{X}_2, v_M)$ where v_M is the maximal score used by the pathologist on the grading scale and $\mathcal{X}_2$ is the half-space for which the mean value of α is greater than $90\mu m^2$.

Finally, cancerous tissue are characterized by a proliferation of cancerous cells. Therefore, frames presenting a small amount on nuclei are usually not cancerous. This leads to the definition of the last labeled set $(\mathcal{X}_3, v_m)$ where $\mathcal{X}_3$ is the half-space for which the value of n is smaller than 5.

6.5.2 Empirical study

6.5.2.1 Data

The dataset contains 221 frames at a resolution of 1024×1024 pixels covering an area of $256\mu m \times 256\mu m$. Each of the frame was given a grade from the pathologist on a scale going from 0 (least severe) to 100 (most severe). A fine scale was chosen to avoid the adverse effects from an artificial discretization. The most extreme values used by the pathologist from the scale where $v_m = 40$ and $v_M = 90$.

In order to study the relevance of the precision of the scores, a subset of 30 images were graded twice by the same pathologist in the same conditions. The pathologist achieved a standard deviation of $\sigma_0 = 7.97$ in terms of absolute difference of the scores. Subsequently, differences between scores lower than σ_0 can therefore be considered irrelevant. This figure will constitute our point of reference in order to appreciate the quality of the results.

6.5.2.2 Results and discussion

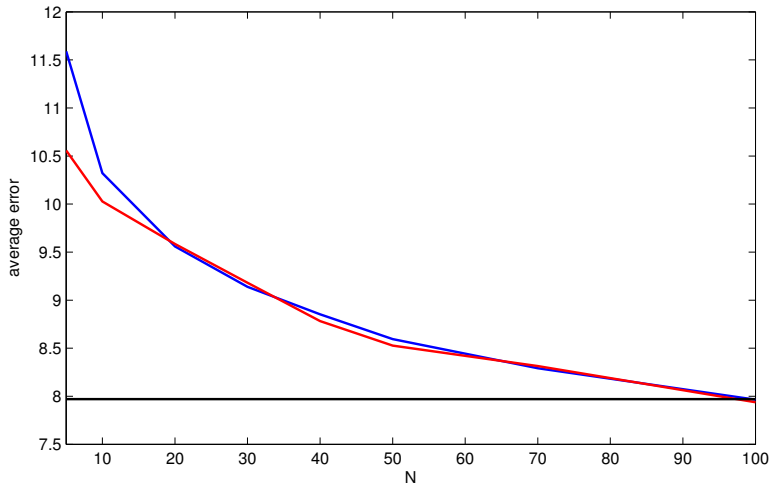
Each numerical result presented in this section corresponds to the average absolute error for 100 training-testing cycles. For each cycle, N sample frames where used for training and the remaining $N - 221$ where used for testing. The ϵ-SVR with the gRBF kernel were used. The learning parameters C and γ where tuned by grid search (best 5-folds cross validation results). Flipping is applied for the entire dataset including the test data (see Section 4.5.3.3). No active measure was taken to deal with the conflicts between labeled data and knowledge sets, thus $\rho = 0$ (see Section 4.5.2.3).

Figure 6.11 presents the results obtained with the gRBF kernel and the standard RBF kernel for different values of N. The results show that the incorporation of prior-knowledge improves the quality of results specially for small training sets ($N < 20$). Unfortunately, the average error quickly reaches σ_0 when N increases, which prevents further comparison between the methods. Further comparisons would require annotated frames with more stable gradings which are not available at this point in time.

For $N \approx 100$, results are very close to the threshold $\sigma_0 = 7.97$ which proves that it is possible for the automatic grading of NA to perform as-well-as the pathologist.

	RBF	gRBF
$N = 5$	11.5887	**10.5576**
$N = 10$	10.3221	**10.0268**
$N = 20$	**9.5590**	9.5840
$N = 30$	**9.1389**	9.1810
$N = 40$	**8.8525**	8.7820
$N = 50$	8.5950	**8.5274**
$N = 70$	**8.2921**	8.3140
$N = 100$	7.9647	**7.9373**

(a) Numerical results



(b) Graphical representation

Figure 6.11: Average error rates over 100 random iterations. The blue line corresponds to the gRBF kernel and the red line to the RBF kernel. The threshold value σ_0 is indicated by the black line.

6.6 Exploration of very large images

The grade corresponding to the entire slide should be computed from the most malignant frames. Although the grading of NA is possible for a single high-magnification frame using the method presented in Section 6.5, a single biopsy virtual slide is a Very Large

Image (VLI) commonly comprising several thousands of high magnification frames, making an exhaustive analysis of all of them not feasible (see Section 6.3.1.3). Therefore, a method able to efficiently find the highest grading regions of the slide is necessary.

In this section, we propose an efficient, generic strategy to explore large images. Our system combines a specific measure of local relevance together with a generic dynamic sampling method based on computational geometry. Applied to our BCG problem, it is able to provide both an accurate and time efficient solution for the grading of full biopsy slides.

The generic algorithm is described in Section 6.6.1. Then, we propose an empirical comparison of random sampling versus our guided sampling algorithm in Section 6.6.2.

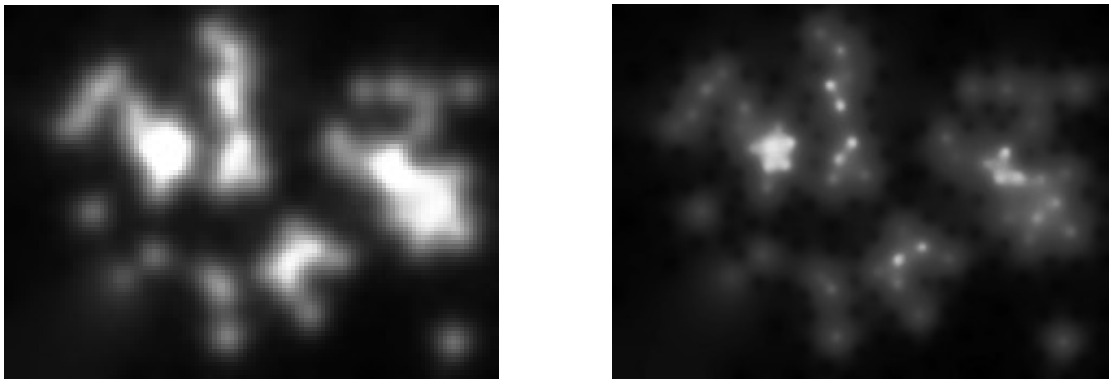
6.6.1 Method

Let I be a VLI split into a large number of square frames $x \in I$. For every frame x, a specific measure of local relevance $S(x)$ referred to as "score" can be computed. The goal of our algorithm (referred as EX-grad) is to efficiently locate the frames in I having the largest relevance score $S(x)$. In our application, the local score is the frame-based NA grade.

The steps of this VLI exploration method are the following. First, a dynamic sampling method is used to identify a subset of the most relevant frames (with high $S(x)$). The objective is to save computational effort by progressively discarding regions showing uniformly low scores and focus the analysis around high-scoring regions. Then, the scores from the sampled subsets are used to interpolate a local score for each of the remaining frames in the VLI. Finally, the highest scoring areas can be precisely identified and extracted from the map of the interpolated score values.

6.6.1.1 Local assessment

Ideally, the local relevance score $S(x)$ should be a semantic information specific to the context of the application such as the local NA grade $S_{NA}(x)$ in our application. Alternatively, when such an information is not available, it can be a low-level feature characterizing the amount of information available such as the compression rate S_{CR} of the image. Maps obtained with the two different score functions on the same biopsy slide are shown on Figure 6.12. The high level of similarity between the two maps



(a) S_{CP} map **(b)** S_{NA} map

Figure 6.12: Maps of (a) the low-level S_{CR} score and (b) the high-level S_{NA} score for the same biopsy slide.

indicates that the low-level S_{CR} can be used as an alternative to S_{NA} when such high-level information is not available.

6.6.1.2 Dynamic sampling

The frame sampling procedure is a dynamic and incremental scheme based on computational geometry tools. At each iteration, given E the set of frames already sampled in the VLI I, we construct the Voronoi diagram of the centroids of the frames in E denoted as Vor_E. Vor_E is a collection of Voronoi cells $\{\nu_x | x \in E\}$, defined as:

$$\nu_x = \{p \in I | \forall y \in I - \{x\},\ dist(p, x) \leq dist(p, y)\} \tag{6.11}$$

The set of Voronoi vertices, referred to as V_E, is the set of the vertices of the planar graph representation of Vor_E. Voronoi vertices share the propriety to be locally the farthest position from their nearest neighbor in E, therefore from already sampled frames.

This geometric construction is aimed at approximating the score S within a whole Voronoi cell by the score of the frame at its center which results in a nearest neighbor approximation. Accordingly, the most undetermined areas are at the intersection of multiple cells, i.e. frames containing a vertex from V_E. We select our next sample x out of V_E following two criteria:

1. *At least one of its neighboring cells has a high score.* Practically, we check that the score $MaxScore(x)$ of its highest scoring neighbor in E is higher that $p \times \max_E$ where $\max_E$ is the currently observed maximal score among E and $p \in [0, 1]$ is a

preset parameter defining the selectivity of the algorithm. This condition controls the convergence of the algorithm towards areas with high scores.

2. *The distance between the new sample and its neighbors is not too short.* In practice, we want $dist(x, E) \geq d$ where $d \in [0, \infty[$ is a parameter determining the fineness of sampling. This condition prevents oversampling.

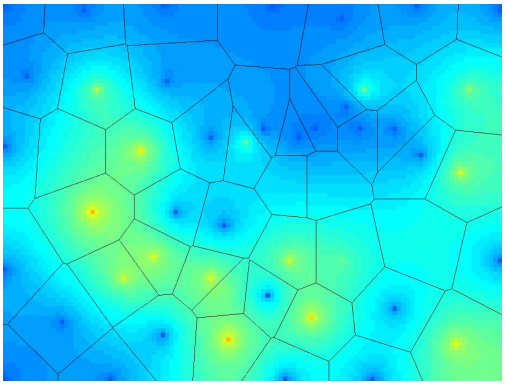
The pseudo-code for one iteration of the sampling algorithm is given in Algorithm 1.

Algorithm 1: One iteration of the dynamic sampling algorithm

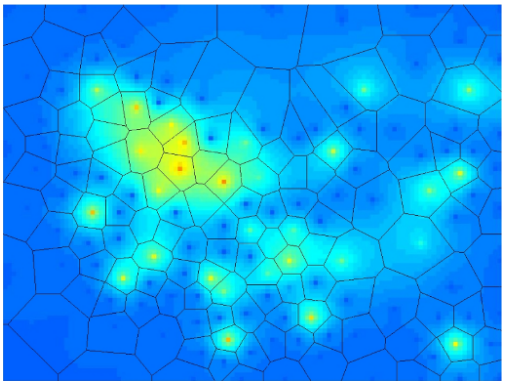
```
Input: current samples E, Vor_E, p, d, max_E
Output: updated values of E, Vor_E, max_E
begin
    compute V_E
    sort V_E according to decreasing distance to E
    for every x ∈ V_E do
        if dist(x, E) ≥ d then
            if MaxScore(x) ≥ p × max_E then
                E = E ∪ {x}
                update Vor_E
                max_E = max(S(x), max_E)
                break loop
            end
        else
            break loop
        end
    end
end
```

To avoid re-computing entire Voronoi diagrams at the addition of every new sample, the new Voronoi diagram is obtained by updating the previous one. Ohya et al. [53] have proposed an algorithm for incremental Voronoi diagram construction with an average time-complexity of $O(n)$ where n is the amount of generators. Sugihara and Iri [78] have later proposed a numerically robust version of it. In the case of the NA grade S_{NA}, it ensures that the cost of selecting all the necessary samples remains negligible compared to the cost of grading a frame. The sampling phase is initialized with three arbitrarily selected frames. Choosing centroids of connected components based on low resolution gray scale analysis has proved to work fast and well. The iterative sampling algorithm is run until depletion of candidate samples. In practice, the parameters d and p are adapted during the whole process by successively taking lower values of d and higher values of p every time samples are depleted. The rationale behind this is to adapt the density of sampling to the score of the regions: regions with homogeneously low scores are

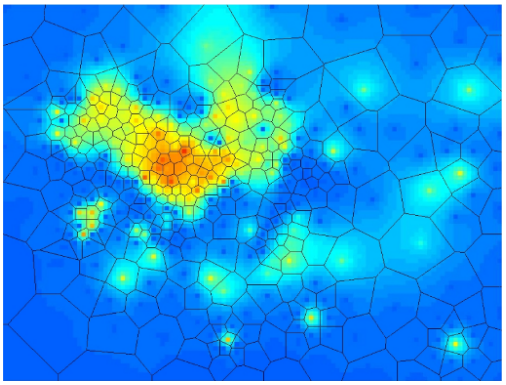
assumed to be less interesting and therefore to require less exploration than regions with higher or more heterogeneously distributed scores. Figure 6.13 illustrates the evolution of sampling over a biopsy slide. It shows that the algorithm is indiscriminate at first and becomes progressively more selective towards regions with high scores.



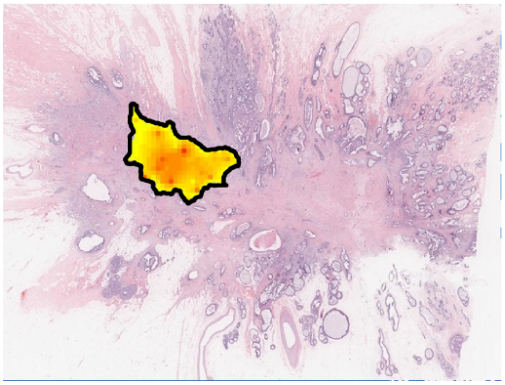
(a) After 50 samples: the whole VLI is being explored. No area seems favored.



(b) After 150 samples: the algorithm converges towards a high grade area.



(c) After 400 samples: the sampling is very dense around this area and remains sparse in others.



(d) The highest grading area superimposed over a low magnification image of the VLI

Figure 6.13: Dynamic sampling method applied to a histopathological VLI of size $59,000 \times 44,000$ pixels. The S_{NA} score has been used. The incrementally constructed Voronoi diagrams are shown in black. Each cell contains a single sample at its center. The maps resulting from the interpolation are shown in colors. Hot colors represent higher grades.

6.6.1.3 Map interpolation

Finally, a full map of the scores over the whole VLI I is interpolated from the scores of the sampled frames. The map is expected to describe accurately the regions with a high local relevance score. In this study, two different interpolation paradigms have been considered to produce the global map from the samples: a nearest neighbors framework where all the frames contained in a Voronoi cell have the same score, and a model based on spring mechanics where every frame is linked to its four neighbors by virtual springs

of length zero and equal stiffness. The map show in color on Figure 6.13 correspond to the spring-based interpolation method.

6.6.2 Experiments and discussion

The method is evaluated for the grading of NA as our local relevance score. The test set consists of 4 H&E stained biopsy slides containing a total of $20,696$ frames graded with the method presented in Section 6.5. The typical size of a VLI is approximately $50,000 \times 50,000$ pixels.

Performances are measured for the retrieval of the set Rel_f of frames having a score of at least $0.8 \times$ max where max is the global maximum score in the slide. Ret_f refers to the set of frames retrieved by EX-grad for having an interpolated score of at least $0.8 \times$ max. The precision, recall and F-measure (harmonic mean) of the retrieval are defined as:

$$prec = \frac{|Ret_f \cap Rel_f|}{|Ret_f|} \quad rec = \frac{|Ret_f \cap Rel_f|}{|Rel_f|} \quad F = 2 \times \frac{prec \times rec}{prec + rec} \tag{6.12}$$

Results are compared to random uniform sampling of the same amount of frames followed by similar interpolation methods. Figures for random sampling are average values over 100 trials. Comprehensive empirical results corresponding to the 4 cases of breast cancer can be found in Table 6.2.

case no.	no. of frames	no. of samples	EX-grad						Random sampling					
			Nearest neighbor approx.			Spring based approx.			Nearest neighbor approx.			Spring based approx.		
			prec.	rec.	F-meas.	prec.	rec.	F-meas.	prec.	rec.	F-meas.	prec.	rec.	F-meas.
case 1	3648	159 (4%)	1.000	0.650	0.788	1.000	0.650	0.788	0.104	0.148	0.122	0.548	0.040	0.075
case 2	5880	102 (2%)	1.000	0.800	0.889	1.000	0.800	0.889	0.007	0.082	0.013	0.120	0.024	0.040
case 3	2544	527 (21%)	1.000	0.286	0.444	1.000	0.286	0.444	0.216	0.209	0.212	0.740	0.196	0.310
case 4	8624	164 (2%)	1.000	0.318	0.482	1.000	0.318	0.482	0.045	0.076	0.057	0.540	0.019	0.036

Table 6.2: Experimental results for the dynamic sampling of frames.

As shown on Figure 6.14, the nearest neighbor method tends to have a better recall rate whereas the spring based method has much higher precision. Both interpolation methods eventually converge towards the same results. F-measures are roughly similar at any sampling rate. Nevertheless, given that the recall rate remains at acceptable levels, it is advisable to opt for the more sophisticated spring based approximation since perfect precision is more critical for an accurate diagnosis than better recall.

All results show the excellent overall performances of our algorithm. Our method

has always achieved absolute precision, with as little as 2% of the frames analyzed in half of the cases. Recall rates span from 32% to 80% with an average value above 50% which allows the retrieval of enough high NA frames to grade the slide. The effectiveness of the dynamic sampling algorithm has been proved by the dramatically lower performances at similar sampling levels with random sampling (followed by any interpolation method).

In conclusion, our method has proved its ability to accurately find and measure the highest levels of NA in a biopsy slide within an acceptable time frame as well as to provide a useful, reliable visualization map for the end-user.

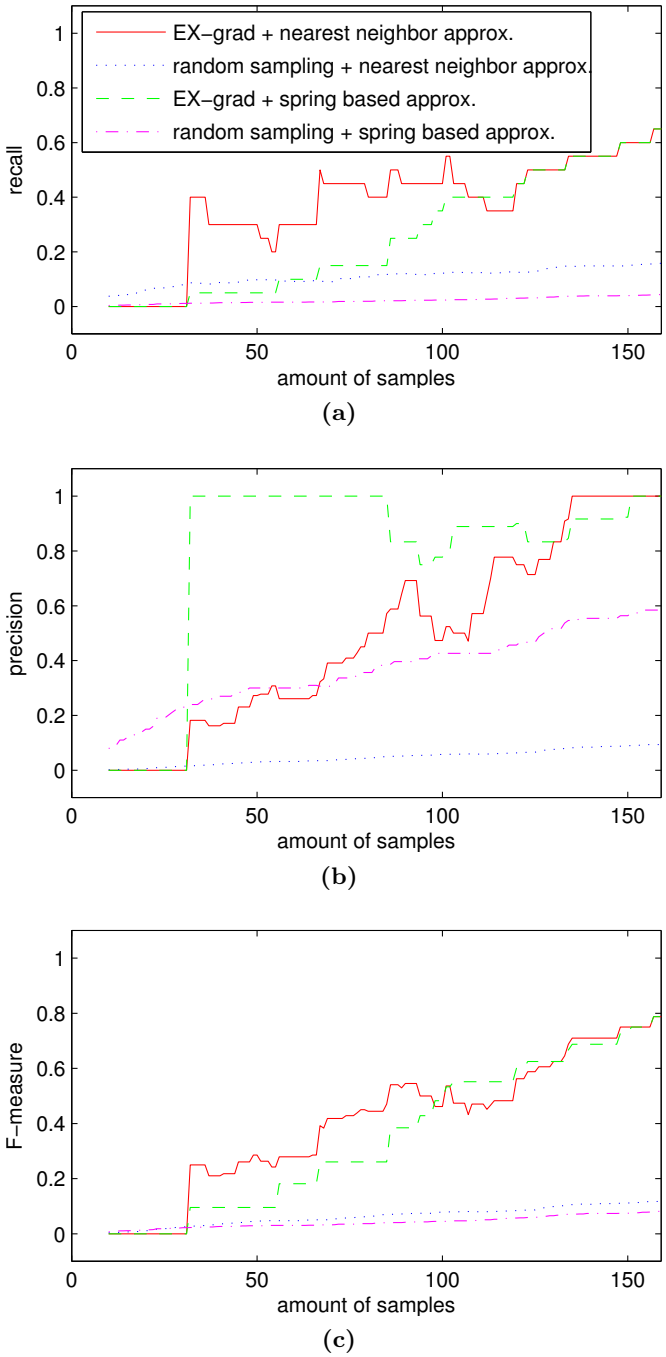


Figure 6.14: Detailed results for *case 1* showing differences between the two interpolation methods at lower levels of sampling.

Chapter 7

Conclusion

In this thesis, we proposed the KE-RBF kernel framework, a set of kernel methods for the incorporation of various types of problem-specific prior-knowledge into SVMs.

First, we gave a statistical introduction to SVMs emphasizing on the importance of kernels in Chapter 2. Then, we put up a structured and critical review of the state-of-the-art on the incorporation of prior-knowledge into SVMs in Chapter 3 and proposed the KE-RBF framework, our original contribution to the problem based on 3 families of kernels (ξRBF, pRBF and gRBF) in Chapter 4. A thorough empirical validation of the framework basing on a wide variety of fields of application was proposed in Chapter 5. Finally, we proposed a valorization of our work in a computer-aided BCG application done in close collaboration with pathologists from the MICO project and scheduled for real clinical deployment in Chapter 6.

7.1 Summary of the contributions

The various contributions of this thesis can be summarized in the following fashion.

First, SVMs where introduced in a didactic tutorial as an implementation of a sound statistical risk minimization strategy known as the structural risk minimization principle. In particular, we justified the importance of using kernels inducing an adequate hypothesis space for the resolution of the problem.

Then, we showed that the KE-RBF framework proposed in this thesis provides prac-

tical and effective tools for the incorporation of a variety of commonly available prior-knowledge into SVMs.

Their systematic evaluation on five different applications using publicly available real-world data (and synthetic data in a lesser extent) from very diversified fields of application showed that KE-RBF kernels are effective and easy to use in practice. We showed that they can lead to significant performance improvements when used with adequate prior-knowledge, and are able to overperform the standard RBF kernel with training sets up to ten times smaller in some cases.

The improvements were particularly pronounced with very small or strongly biased training sets. This remarkable reduction in training data requirements enabled by the KE-RBF kernels, both quantitatively and qualitatively, opens new perspectives for SVMs significantly broadening their usual field of application.

Finally, we proposed a valorization of our contribution through an application to BCG able to satisfy the actual operational requirements of the pathologists. This application demonstrates how the KE-RBF framework can work as one of the numerous components or a complex, real-life engineering project an proves the operational readiness of the framework.

7.2 Future works

Future developments to the work carried out in this thesis can be considered from several perspectives: theoretical, computational and applicational.

In this thesis, we showed that the KE-RBF framework is able to incorporate a wide variety of prior-knowledge into SVMs. However, the different types of prior-knowledge where considered successively and independently from each other. The question of how heterogeneous types of prior-knowledge could be concurrently considered was not answered in the scope of this work.

By itself, the ξRBF kernel is able to deal with different types of prior-knowledge and one should be able to compose them by multiplication of the corresponding knowledge functions (in a fashion similar to the way multiple frequencies were composed in

Sect 4.3.2.2). Technically, the pRBF kernel ($K_{\text{rbf}} \otimes K$) and the ξRBF kernel (ξK_{rbf}) can also be used simultaneously ($\xi(K_{\text{rbf}} \otimes K)$) but there is no theoretical guarantee that the originally good properties of the pRBF kernel (preservation of the correlation patterns) or the ξRBF kernel (appropriate modification the kernel distance) will be preserved. The case of the gRBF kernel which extends the domain of the data seems even more complex to deal with.

Accordingly, an interesting theoretical development to the work would be to study the *simultaneous incorporation of heterogeneous types of prior-knowledge* in a systematic fashion. Overall, it appears that the KE-RBF framework would benefit from a unification effort.

In this thesis, the prior-knowledge was considered as a complement for or as an alternative to annotated training data, in order to improve the overall quality of the results. Another theoretical extension to the work would be to use the prior-knowledge for a different purpose, in a *validation role*.

Indeed, a number of critical systems are not aiming for the best possible average performances, but rather for the prevention of failures. For instance, the pathologist engages his legal responsibility when he performs a diagnostic. Therefore, it is impossible for him to blindly trust an automatic system such as our BCG platform no matter how good are the results on average if there are no guarantees on the result.

Usually, statistical learning from data does not provide such guarantees. Therefore, an interesting problem would be related to the use of the prior-knowledge in order to enforce properties on the labeling model, in a similar fashion to what was done with theorem 4.4.6.

This thesis was mainly focused on the theoretical validity of the methods and their empirical performance evaluation. In comparison, computational issues such as *online, incremental learning* with KE-RBF kernels were not considered in the scope of this work. As a matter of fact, an online version for another optimization-based method for the incorporation of prior-knowledge into SVMs, known as the KBSVM, was recently proposed by Kunapuli et al. [36]. Therefore, more work could be conducted on aspects which do not directly relate to the validity of the methods but rather to their computational efficiency.

The application to BCG developed during this thesis in the context of the MICO[1] project has a planned extension with the FlexMIm project starting from September 2012 and funded for a 3 years term by the Fond Unitaire Interministériel (France). It has a structure comparable to the MICO project involving academic partners[2,3], industrial partners[4,5] and pathologists[6]. FlexMIm is an assistive framework for histopathology and cytopathology with a focus on collaborative issues such as the sharing of data, knowledge and technical tools between different medical specialities and locations.

Unlike the MICO project, the platform addresses the different fields for histopathology not restricted to the study of breast cancer. This introduces new interesting questions such as *domain adaptation* for problems with training data and prior-knowledge. FlexMIm is also scheduled for a larger scale deployment in 27 medical units and has a much stronger emphasis on operational issues. Therefore, *knowledge modeling* by end-users (medical doctors) who do not have specialized knowledge in the machine learning field becomes and central issue.

[1] `http://ipal.i2r.a-star.edu.sg/project/mico`
[2] Université Pierre et Marie Curie, Paris, France
[3] Université Paris Descartes, Paris, France
[4] Orange, France
[5] TRIBVN, France
[6] Assistance Publique – Hôpitaux de Paris, France

Appendix A

Further developments on PD kernels and their RKHS

In theorem 2.2.20, we proved that a PD kernel is a reproducing kernel. The reciprocal of theorem 2.2.20 is also true:

Theorem A.0.1. A reproducing kernel is a PD kernel

Let $K : \mathcal{X}^2 \to \mathbb{R}$ be a reproducing kernel. Then, K is a PD kernel.

Proof. In accordance with definition 2.2.1, we must prove that K is symmetric and positive definite.

K is symmetric because for any $(x, y) \in \mathcal{X}^2$:

$$\begin{aligned} K(x,y) &= \langle K_x, K_y \rangle_{\mathcal{H}} \text{ by the reproducing property of } K \\ &= \langle K_y, K_x \rangle_{\mathcal{H}} \text{ by symmetry of the inner product} \\ &= K(y,x) \text{ by the reproducing property of } K \end{aligned}$$

K is positive definite because for $N \in \mathbb{N}$, $(x_1, x_2, \ldots, x_N) \in \mathcal{X}^N$, $(v_1, v_2, \ldots, v_N) \in$

$\mathbb{R}^N$:

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j K(x_i, x_j) &= \sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j \langle K_{x_i}, K_{x_j}\rangle_{\mathcal{H}} \text{ by the reproducing property of } K \\
&= \langle \sum_{i=1}^{N} v_i K_{x_i}, \sum_{j=1}^{N} v_j K_{x_j}\rangle_{\mathcal{H}} \text{ by bilinearity of the inner product} \\
&= \| \sum_{i=1}^{N} v_i K_{x_i} \|_{\mathcal{H}}^2 \\
&\geq 0
\end{aligned}$$

□

PD kernels and reproducing kernels are therefore two different ways of characterizing the same objects.

Theorem A.0.2. Characterization of reproducing kernels

Let $K : \mathcal{X}^2 \to \mathbb{R}$. The two following properties are equivalent:

1. *K is a PD kernel*

2. *K is a reproducing kernel*

Proof. Direct consequence of theorems 2.2.20 and A.0.1. □

Remark A.0.3. RKHS also have a simple characterization: a vector subspace $\mathcal{H}$ of $\mathbb{R}^{\mathcal{X}}$ is a RKHS if and only if the function evaluating of a function $f \in \mathcal{H}$ to a point $x \in \mathcal{X}$ is continuous of every x.

So far, we have always been referring to "a" RKHS associated to a reproducing kernel. In fact, every reproducing kernel defines a unique RKHS.

Theorem A.0.4. RKHS of a reproducing kernel: uniqueness

A function $K : \mathcal{X}^2 \to \mathbb{R}$ is the reproducing kernel of at most one RKHS.

Proof. Lets assume $(\mathcal{H}, \langle ., . \rangle_{\mathcal{H}})$ is a RKHS associated to the reproducing kernel K. The proof is done in two phases:

1. The unicity of $\mathcal{H}$

2. The unicity of $\langle ., . \rangle_{\mathcal{H}}$

By definition, $\mathcal{H}$ contains $\mathcal{H}_K = span_{\mathbb{R}}\{K_x\}_{x\in\mathcal{X}}$. The goal is to prove that $\mathcal{H} = \mathcal{H}_K$.

$\mathcal{H}$ is a Hilbert space. Therefore for any subset $A \subset \mathcal{H}$, $\mathcal{H} = A \oplus A^{\perp}$ where $\oplus$ represents the direct sum and $^{\perp}$ designates the set of elements orthogonal to a set. In particular, since $\mathcal{H}_K \subset \mathcal{H}$, then $\mathcal{H} = \mathcal{H}_K \oplus \mathcal{H}_K^{\perp}$.

Now, we prove that $\mathcal{H}_K^{\perp} = \{0\}$. Let $f \in \mathcal{H}_K^{\perp}$. For any $x \in \mathcal{X}$, the reproducing property gives us:

$$\begin{aligned} f(x) &= \langle f, K_x\rangle_{\mathcal{H}} \\ &= 0 \text{ since } K_x \perp f \end{aligned}$$

Thus, $\forall x \in \mathcal{X},\ f(x) = 0$ *i.e.* $f = 0$.

Therefore, we get the uniqueness of $\mathcal{H}$:

$$\begin{aligned} \mathcal{H} &= \mathcal{H}_K \oplus \mathcal{H}_K^{\perp} \\ &= \mathcal{H}_K \oplus \{0\} \\ &= \mathcal{H}_K \end{aligned}$$

We now prove the uniqueness of the inner product. For any two element of $\mathcal{H}_K$:

$$\begin{aligned} &\langle \sum_{i=1}^{N} \alpha_i K_{x_i}, \sum_{j=1}^{M} \beta_j K_{y_j}\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} \alpha_i\beta_j \langle K_{x_i}, K_{y_j}\rangle_{\mathcal{H}} \text{ by bilinearity of the inner product} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} \alpha_i\beta_j K_{x_i}(y_j) \text{ by the reproducing property of } K \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} \alpha_i\beta_j K(x_i, y_j) \end{aligned}$$

which uniquely defines the inner product. □

Based on theorems A.0.2 and A.0.4, it is therefore legitimate to refer to "the" RHKS of a PD/reproducing kernel.

Remark A.0.5. The contrary of theorem A.0.4 is also true: a given RKHS admits a single PD/reproducing kernel.

In addition, the proof of theorem A.0.4 yields an explicit form for the RKHS, similar to the one introduced in theorem 2.2.20.

Theorem A.0.6. RKHS of a reproducing kernel: explicit form

The unique RKHS associated to a reproducing kernel K is the Hilbert space $(\mathcal{H}_K, \langle ., . \rangle_{\mathcal{H}_K})$ such that:

- *$\mathcal{H}_K$ is the real vector space generated (spanned) by the functions $\{K_x | x \in \mathcal{X}\}$.*
- $\langle \sum_{i=1}^{N} \alpha_i K_{x_i}, \sum_{j=1}^{M} \beta_j K_{y_j} \rangle_{\mathcal{H}_K} = \sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_i \beta_j K(x_i, y_j)$

Proof. Corollary of the proof of theorem A.0.4. □

Appendix B

Geometrical construction of the SVC

This appendix provides a sketchy outlook on how an equivalent formulation for the SVC can be obtained from geometrical considerations alone.

Remark B.0.7. The naming of notions such as the "margin" or the "slack" variables come from this geometrical interpretation of the SVCs.

Hard-margin SVC The particular case of (2.57) with the linear kernel and without slack variables ($\forall i,\ \xi_i = 0$) referred to as the *hard-margin* SVC is often presented as the most basic type of SVC.

The optimization problem corresponding to the hard-margin SVC is:

$$\begin{aligned} &\underset{w\in\mathbb{R}^n,\, b\in\mathbb{R}}{\text{minimize}} && \|w\|^2 \\ &\text{subject to} && y_i(\langle w, x_i\rangle + b) \geq 1, \quad i = 1, \ldots, N \end{aligned} \tag{B.1}$$

The problem is equivalent to finding a hyperplane (perpendicular to w) separating points from each of the classes such as the distance $\frac{1}{\|w\|^2}$ between the hyperplane and the nearest sample point is maximized.

Problem (B.1) is therefore equivalent to maximizing the width $\frac{2}{\|w\|^2}$ of a "margin" around the decision surface which is clear of any training sample.

Soft-margin SVC The main issue of the hard-margin version is that it requires the classes to be linearly separable in order to admit a solution. The introduction of "slack" into the problem through the use of the slack variables ξ_i ensures that the problem is always solvable.

This version of the hard-margin SVC with relaxed constraints is known as the *soft-margin* SVC. Its primal formulation is:

$$\begin{aligned} \underset{w\in\mathbb{R}^n,\, b\in\mathbb{R}}{\text{minimize}} \quad & \sum_{i=1}^{N} \xi_i + \lambda \|w\|^2 \\ \text{subject to} \quad & y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N \\ & \xi_i \geq 0, \qquad\qquad\qquad\quad i = 1, \ldots, N \end{aligned} \tag{B.2}$$

The tolerance to misclassification is controlled by adjusting the parameter $\lambda > 0$, a high value of λ allowing for more slack.

Nonlinear case Finally, the nonlinear formulation (2.57) directly obtained by derivation from the SRM principle can be presented as an extension of the soft-margin linear SVC to nonlinear classification using the kernel trick.

Bibliography

[1] S. Ali and A. Madabhushi. Active contour for overlap resolution using watershed based initialization (ACOReW): Applications to histopathology. In *Proc. International Symposium on Biomedical Imaging: Nano to Macro*, 2011.

[2] A. Basavanhally, S. Doyle, and A. Madabhushi. Predicting classifier performance with a small training set: Applications to computer-aided diagnosis and prognosis. In *Proc. International Symposium on Biomedical Imaging: Nano to Macro*, 2010.

[3] O. Bousquet and D. J. L. Herrmann. On the complexity of learning the kernel matrix. In *Proc. Neural Information Processing Systems*, 2003.

[4] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[5] P.-H. Chen, C.-J. Lin, and B. Schölkopf. A tutorial on nu-support vector machines: Research articles. *Applied Stochastic Models in Business and Industry*, 21:111–136, 2005.

[6] C. Cortes and V. N. Vapnik. Support-vector networks. *Machine Learning*, 20: 273–297, 1995.

[7] K. Crammer, J. Keshet, and Y. Singer. Kernel design using boosting. In *Proc. Neural Information Processing Systems*, 2002.

[8] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press, 2000.

[9] J.-R. Dalle, W. K. Leow, D. Racoceanu, A. E. Tutac, and T. C. Putti. Automatic breast cancer grading of histopathological images. In *Proc. Engineering in Medicine and Biology Society*, 2008.

[10] J.-R. Dalle, H. Li, C.-H. Huang, W. K. Leow, D. Racoceanu, and T. C. Putti. Nuclear pleomorphism scoring by selective cell nuclei detection. In *Proc. Workshop on Applications of Computer Vision*, 2009.

[11] D. Decoste and M. C. Burl. Distortion-invariant recognition via jittered queries. In *Proc. Conference on Computer Vision and Pattern Recognition*, 2000.

[12] D. Decoste and B. Schölkopf. Training invariant support vector machines. *Machine Learning*, 46:161–190, 2002.

[13] J. Demmel, I. Dumitriu, and O. Holtz. Fast linear algebra is stable. *Numerische Mathematik*, 108:59–91, 2007.

[14] X. Descombes, R. Minlos, and E. Zhizhina. Object extractionusing a stochastic birth-and-death dynamics in continuum. *Mathematical Imaging and Vision*, 33: 347–359, 2009.

[15] J. Diederich and N. Barakat. Knowledge initialisation for support vector machines. In *Proc. Conference on Neuro-Computing and Evolving Intelligence*, 2004.

[16] S. Doyle, S. Agner, A. Madabhushi, M. Feldman, and J. Tomaszewski. Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features. In *Proc. International Symposium on Biomedical Imaging: Nano to Macro*, 2008.

[17] E. C. Duchon. Lanczos filtering in one and dimentsions. *Applied Meteorology*, 18: 1016–1022, 1979.

[18] B. Dunne and J. J. Going. Scoring nuclear pleomorphism in breast cancer. *Histopathology*, 39:259–265, 2001.

[19] J. Estévez, S. Alayón, L. Moreno, R. Aguilar, and J. Sigut. Cytological breast fine needle aspirate images analysis with a genetic fuzzy finite state machine. In *Proc. Symposium on Computer-Based Medical Systems*, 2002.

[20] A. Fabbri, M. L. Carcangiu, and A. Carbone. Histological classification of breast cancer. In *Breast Cancer*. Springer Berlin Heidelberg, 2008.

[21] F. Fan and P. A. Thomas. *Tumors of the breast*, chapter 11, pages 75–81. Springer New York, 2007.

[22] S. Frkovic-Grazio and M. Bracko. Long term prognostic value of nottingham histological grade and its components in early (pT1N0M0) breast carcinoma. *Clinical Pathology*, 55:88–92, 2002.

[23] G. M. Fung, O. L. Mangasarian, and J. W. Shavlik. Knowledge-based support vector machine classifiers. In *Porc. Neural Information Processing Systems*, 2002.

[24] G. M. Fung, O. L. Mangasarian, and J. W. Shavlik. Knowledge-based nonlinear kernel classifiers. In *Proc. Conference on Learning Theory*, 2003.

[25] T. Graepel and R. Herbrich. Invariant pattern recognition by semidefinite programming machines. In *Proc. Advances in Neural Information Processing Systems*, 2003.

[26] M. N. Gurcan, L. E. Boucheron, A. Can, A. Madabhushi, N. M. Rajpoot, and B. Yener. Histopathological image analysis: A review. *Reviews in Biomedical Engineering*, 2:147–171, 2009.

[27] B. Haasdonk. Feature space interpretation of SVMs with indefinite kernels. *Pattern Analysis and Machine Intelligence*, 27:482–492, 2005.

[28] B. Haasdonk and D. Keysers. Tangent distance kernels for support vector machines. In *Proc. International Conference on Pattern Recognition*, 2002.

[29] B. Haasdonk, A. Vossen, and H. Burkhardt. Invariance in kernel methods by haar-integration kernels. In *Lecture Notes in Computer Science*. Springer, 2005.

[30] D. R. Hardoon and J. Shawe-Taylor. Decomposing the tensor kernel support vector machine for neuroscience data with structured labels. *Machine Learning*, 1:29–46, 2010.

[31] N. M. Khan, R. Ksantini, I. S. Ahmad, and B. Boufama. A novel SVM+NDA model for classification with an application to face recognition. *Pattern Recognition*, 45: 66–79, 2012.

[32] E. Klassen, A. Srivastava, W. Mio, and S. H. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *Pattern Analysis and Machine Intelligence*, 26: 372–383, 2004.

[33] R. Kondor and T. Jebara. A kernel between sets of vectors. In *Proc. International Conference on Machine Learning*, 2003.

[34] M. S. Kulikova, I. H. Jermyn, X. Descombes, E. Zhizhina, and J. Zerubia. A marked point process model with strong prior shape information for extraction of multiple, arbitrarily-shaped objects. In *Proc. Conference on Signal-Image Technology and Internet-Based Systems*, 2009.

[35] M. S. Kulikova, A. Veillard, L. Roux, and D. Racoceanu. Nuclei extraction from histopathological images using a marked point process approach. In *Proc. SPIE Medical Imaging*, 2012.

[36] G. Kunapuli, K. P. Bennett, A. Shabbeer, R. Maclin, and J. W. Shavlik. Online knowledge-based support vector machines. In *Proc. European Conference on Machine Learning*, 2010.

[37] H. W. Kunth. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[38] F. Lauer and G. Bloch. Incorporating prior knowledge in support vector machnies for classification: a review. *Neurocomputing*, 71:1578–1594, 2008.

[39] K. Laws. *Textured Image Segmentation.* PhD thesis, University of Southern California, 1980.

[40] Q. V. Le and A. J. Smola. Simpler knowledge-based support vector machines. In *Proc. International Conference on Machine Learning*, 2006.

[41] R. Luss and A. Aspremont. Support vector machine classification with indefinite kernels. In *Proc. Neural Information Processing Systems*, 2007.

[42] R. Maclin, J. Shavlik, T. Walker, and L. Torrey. A simple and effective method for incorporating advice into kernel methods. In *Proc. Association for the Advancement of Artificial Intelligence*, 2006.

[43] R. Maclin, E. W. Wild, J. Shavlik, L. Torrey, and T. Walker. Refining rules incorporated into knowledge-based support vector learners via successive linear programming. In *Proc. Association for the Advancement of Artificial Intelligence*, 2007.

[44] O. L. Mangasarian. Generalized support vector machines. In *Advances in Large Margin Classifiers*. MIT Press, 1998.

[45] O. L. Mangasarian. Knowledge-based linear programming. *SIAM Journal on Optimization*, 15:375–382, 2004.

[46] O. L. Mangasarian and E. W. Wild. Nonlinear knowledge in kernel approximation. *Neural Networks*, 18:300–306, 2007.

[47] O. L. Mangasarian and E. W. Wild. Nonlinear knowledge-based classification. *Neural Networks*, 10:1826–1832, 2008.

[48] O. L. Mangasarian and E. W. Wild. Nonlinear knowledge in kernel machines. In *Proc. Centre de Recherches Mathématiques*, 2008.

[49] O. L. Mangasarian, J. Shavlik, and E. W. Wild. Knowledge-based kernel approximation. *Machine Learning Research*, 5:1127–1141, 2004.

[50] O. L. Mangasarian, E. W. Wild, and G. Fung. Proximal knowledge-based classification. *Statistical Analysis and Data Mining*, 1:215–222, 2009.

[51] P. Niyogi, F. Girosi, and T. Poggio. Incorporating prior information in machine learning by creating virtual examples. *Proceedings of the IEEE*, 86:2196–2209, 1998.

[52] M. Oger, P. Belhomme, J. Klossa, J. J. Michels, and A. Elmoataz. Automated region of interest retrieval and classification using spectral analysis. In *Proc. European Congress on Telepathology and International Congress on Virtual Microscopy*, 2008.

[53] T. Ohya, M. Miri, and K. Murota. Improvements of the incremental method for the Voronoi diagram with computational comparison of various algorithms. *Operational Research Society of Japan*, 27:306–336, 1984.

[54] C. S. Ong, X. Marie, S. Canu, and A. J. Smola. Learning with non-positive kernels. In *Proc. International Converence on Machine Learning*, 2004.

[55] G. Perrin, X. Descombes, and J. Zerubia. A marked point process model for tree crown extraction in plantation. In *Proc. International Conference on Image Processing*, 2005.

[56] S. Petushi, C. Katsinis, C. Coward, F. Garcia, and A. Tozeren. Automated identification of microstructures on histology slides. In *Proc. International Symposium on Biomedical Imaging: Nano to Macro*, 2004.

[57] S. Petushi, F. U. Garcia, M. Haber, C. Katsinis, and A. Tozeren. Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer. *BMC Medical Imaging*, 6(14):1–11, 2006.

[58] T. Poggio and T. Vetter. Recognition and structure from one 2D model view: Observations on prototypes, object classes and symmetries. Technical Report 1347, Massachusetts Institute of Technology, 1992.

[59] A. Pozdnoukhov and S. Bengio. Tangent vector kernels for invariant image classification with SVMs. In *Proc. International Conference on Pattern Recognition*, 2004.

[60] A. C. Ruifrok and D. A. Johnston. Quantification of histochemical staining by color deconvolution. *Analytical and Quantitative Cytology and Histology*, 23:291–299, 2001.

[61] S. Rüping. A simple method for estimating conditional probabilities for SVMs. In *Proc. Lernen - Wissensentdeckung - Adaptivität*, 2004.

[62] C. Salperwyck and V. Lemaire. Learning with few examples: An empirical study on leading classifiers. In *Proc. International Joint Conference on Neural Networks*, 2011.

[63] S. J. Schnitt and L. C. Collins. *Biopsy Interpretation of the Breast*. Lippincot-Williams-Wilkins, 2008.

[64] F. Schnorrenberg. Comparison of manual and computer-aided breast cancer biopsy grading. In *Proc. Engineering in Medicine and Biology Society*, 1996.

[65] F. Schnorrenberg, C.S. Pattichis, K. Kyriacou, and C.N. Schizas. Detection of cell nuclei in breast biopsies using receptive fields. In *Proc. Engineering in Medicine and Biology Society*, 1994.

[66] B. Schölkopf, C. Burges, and V. N. Vapnik. Incorporating invariances in support vector learning machines. In *Proc. International Conference on Artificial Neural Networks*, 1996.

[67] B. Schölkopf, P. Simard, V. N. Vapnik, and A. J. Smola. Prior knowledge in support vector kernels. In *Proc. Neural Information Processing Systems*. The MIT Press, 1998.

[68] B. Schölkopf, A. J. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, July 1998.

[69] H. Schulz-Mirbach. Constructing invariant features by averaging techniques. In *Proc. International Pattern Recognition Conference on Computer Vision and Image Processing*, 1994.

[70] O. Sertel, G. Lozanski, A. Shana'ah, and M. N. Gurcan. Computer-aided detection of centroblasts for follicular lymphoma grading using adaptive likelihood-based cell segmentation. *Biomedical Engineering*, 57:2613–2616, 2010.

[71] J. Shawe-Taylor and N. Cristianini. Margin distribution and soft margin. In *Advances in Large Margin Classifiers*. The MIT Press, 2000.

[72] P. K. Shivaswamy and T. Jebara. Permutation invariant SVMs. In *Proc. International Conference on Machine Learning*, 2006.

[73] P. Y. Simard, Y. A. Le Cun, J. S. Denker, and B. Victorri. Transformation invariance in pattern recognition, tangent distance and tangent propagation. In *Lecture Notes in Computer Science*. Springer, 1998.

[74] S. Sonnenburg, G. Rätsch, and C. Schäfer. A general and efficient multiple kernel learning algorithm. In *Neural Information Processing Systems*, 2006.

[75] N. Street, W. H. Wolberg, and O. L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Proc. International Symposium on Electronic Imaging: Science and Technology*, 1993.

[76] W. N. Street. Xcyt: A system for remote cytological diagnosis and prognosis of breast cancer. In *Artificial Intelligence Techniques in Breast Cancer Diagnosis And Prognosis*. World L. C. Scientific Publishing, 2000.

[77] E. Subramaniam, K. L. Tan, M. Y. Mashor, and N. Ashidi Mat Isa. Breast cancer diagnosis systems: A review. *The Computer, the Internet and Management*, 14: 24–35, 2006.

[78] K. Sugihara and M. Iri. Construction of the Voronoi diagram for 'one million' generators in single-precision arithmetic. *Proceedings of the IEEE*, 80:1471–1484, 1992. ISSN 0018-9219.

[79] F. A. Tavassoli and P. Devilee, editors. *World Health Organization Classification of Tumour. Tumours of the Breast and Female Genial Organs*. IARC Press, 2003.

[80] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[81] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[82] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[83] V. N. Vapnik and A. J. Chervonenkis. *Teoriya Raspoznavaniya Obrazov: Statisticheskie Problemy Obucheniya*. Nauka, 1974.

[84] A. Veillard, D. Racoceanu, and S. Bressan. Incorporating prior-knowledge in support vector machines by kernel adaptation. In *Proc. International Conference on Tools with Artificial Intelligence*, 2011.

[85] K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the sensitivity of support vector machines. In *Proc. International Joint Conference on Artificial Intelligence*, 1999.

[86] J. P. Vert, H. Saigo, and T. Akutsu. Local alignment kernels for biological sequences. In *Kernel Methods in Computational Biology*. MIT Press, 2004.

[87] G. Wang. Incorporating prior knowledge in support vector machines: Retrospect and prospect. In *Proc. International Conference on Networked Computing and Advanced Information Management*, 2008.

[88] L. Wang, P. Xue, and K. L. Chan. Incorporating prior knowledge into SVM for image retrieval. In *Proc. International Conference on Pattern Recognition*, 2004.

[89] L. Wang, Y. Gao, K. L. Chan, P. Xue, and W. Y. Yau. Retrieval with knowledge-driven kernel design: an approach to improving svm-based cbir with relevance feedback. In *Proc. International Conference on Computer Vision*, 2005.

[90] Y. Wang and F. Wan. Breast cancer diagnosis via support vector machines. In *Proc. Chinese Control Conference*, pages 1853–1856, 2006.

[91] J. Weston, B. Schölkopf, and O. Bousquet. Joint kernel maps. In *Proc. International Conference on Artificial Neural Networks: computational Intelligence and Bioinspired Systems*, 2005.

[92] L. Wolf, A. Shashua, and D. Geman. Learning over sets using kernel principal angles. *Machine Learning Research*, 4:913–931, 2003.

[93] A. Woznica, A. Kalousis, and M. Hilario. Distances and (indefinite) kernels for sets of objects. In *Proc. International Conference on Data Mining*, 2006.

[94] G. Wu, E. Y. Chang, and Z. Zhang. An analysis of trans formation on non-positive semidefinite similarity matrix for kernel machines. In *Proc. International Conference on Machine Learning*, 2005.

[95] X. Wu and R. Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *Proc. International Conference on Knowledge Discovery and Data Mining*, 2004.

[96] L. Yang, P. Meer, and D. J. Foran. Unsupervised segmentation based on robust estimation and color active contour models. *Information Technology in Biomedicine*, 9:475–486, 2005.

[97] X. Yang, H. Li, and X. Zhou. Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and Kalman filter in time-lapse microscopy. *Circuits and Systems: Regular Papers*, 53:2405–2414, 2006.

[98] J. Yuan, K. Wang, T. Yu, and X. Liu. Incorporating fuzzy prior knowledge into relevance vector machine regression. In *Proc. International Joint Conference on Neural Networks*, 2008.