

Primal and dual model representations in kernel-based learning*

Johan A.K. Suykens, Carlos Alzate

*K.U. Leuven, ESAT-SCD
Kasteelpark Arenberg 10, B-3001 Leuven (Heverlee), Belgium
e-mail: johan.suykens@esat.kuleuven.be; carlos.alzate@esat.kuleuven.be*

and

Kristiaan Pelckmans

*Division of Systems and Control, Department of Information Technology
Uppsala University Box 337, SE-751 05 Uppsala, Sweden
e-mail: kp@it.uu.se*

Abstract: This paper discusses the role of primal and (Lagrange) dual model representations in problems of supervised and unsupervised learning. The specification of the estimation problem is conceived at the primal level as a constrained optimization problem. The constraints relate to the model which is expressed in terms of the feature map. From the conditions for optimality one jointly finds the optimal model representation and the model estimate. At the dual level the model is expressed in terms of a positive definite kernel function, which is characteristic for a support vector machine methodology. It is discussed how least squares support vector machines are playing a central role as core models across problems of regression, classification, principal component analysis, spectral clustering, canonical correlation analysis, dimensionality reduction and data visualization.

Keywords and phrases: Kernel methods, support vector machines, constrained optimization, primal and dual problem, feature map, regression, classification, principal component analysis, spectral clustering, canonical correlation analysis, independence, dimensionality reduction and data visualization, sparseness, robustness.

Received July 2009.

Contents

1	Introduction	149
2	Function estimation in RKHS	151
3	Support vector machine classifier	152
3.1	Primal and dual problem	152
3.2	Positive definite kernel and feature map	153
4	LS-SVM core models	155
4.1	Core models in supervised and unsupervised learning	155

*This paper was accepted by Grace Wahba, Associate Editor for the IMS.

4.2	Sparseness and robustness	158
4.3	Variable selection	161
5	Core models plus additional constraints	161
6	Models for spectral clustering	165
6.1	Weighted kernel PCA for kernel spectral clustering	166
6.2	Multiway kernel spectral clustering with out-of-sample extensions	167
7	Dimensionality reduction and data visualization	171
8	Kernel CCA and ICA	173
8.1	Multivariate kernel CCA	174
8.2	Kernel CCA and independence	175
9	Conclusions	176
	Acknowledgements	177
	References	177

1. Introduction

The use of kernel methods has a long history and tradition in mathematics and statistics with fundamental contributions made by Moore, Aronszajn, Krige, Parzen, Kimeldorf and Wahba, and others [7, 20, 45, 56, 69, 86]. Kernels have been employed in methods of non-parametric statistics, estimation in Reproducing Kernel Hilbert Spaces (RKHS), Gaussian processes and Kriging. A further increasing interest in kernel-based methods has taken place in relation to methods of Support Vector Machines (SVM) [85], which have largely stimulated the research on kernel-based learning in general [21, 23, 41, 64, 69–71, 77, 85]. Especially on problems with a large number of input variables, many successful results in different application areas have been reported, also with the emergence of new technologies that generate high dimensional data such as for microarrays, proteomics, textmining and others.

For black-box modelling applications there has been interest in making use of universal approximators, such as with the use of multilayer perceptrons in the area of neural networks. Due to the often large amount of unknown coefficients, there is a high risk for overfitting the data. However, one can overcome this problem by making use of regularization. One obtains then an effective number of parameters (degrees of freedom) that is much smaller than the number of coefficients. Such regularization mechanisms are also prominently present in methods of support vector machines. A minimization of the regularization term corresponds in the context of classification problems to maximizing the margin. By making use of universal kernels like the Gaussian radial basis function kernel one obtains a flexible class of models. Model selection then typically amounts to the choice of regularization constants and kernel tuning parameters, aiming to achieve a good bias-variance trade-off. In a RKHS interpretation this corresponds to penalizing a norm defined on the unknown function which is restricted to belong to a reproducing kernel Hilbert space. The use of positive definite kernels allows one then to plug-in a large variety of kernel functions

including linear, polynomial, radial basis, splines, wavelets, kernels extracted from graphical models, textmining kernels and others.

Methods of SVMs for classification and regression relate to convex optimization theory [14]. The specification of the estimation problem at the primal level is done by formulating a constrained optimization problem, where the model is expressed in terms of a feature map. The optimal model representation is obtained together with the solution from the conditions for optimality. At the dual level (problem in the Lagrange multipliers) the model is expressed in terms of a positive definite kernel function. For given tuning parameters the problem is convex. Through the choice of an appropriate loss function one obtains a sparse representation in SVMs. A subset of the given training data constitutes the set of support vectors, which follows from solving a convex quadratic programming problem.

Given these attractive properties of SVMs, both conceptually and computationally, how might these be further extended in a systematic and constructive way, beyond problems of classification and regression? At this point Least Squares Support Vector Machines (LS-SVMs) [77] can be considered as *core models* for a wide range of problems in supervised and unsupervised learning and beyond. By making use of the L_2 loss and equality constraints, the conditions for optimality (Karush-Kuhn-Tucker conditions) for LS-SVMs become much simpler than for SVMs. Some key objectives of this approach are to:

- extend support vector machine methodologies to a wide range of problems in supervised and unsupervised learning (regression, classification, principal component analysis, canonical correlation analysis, spectral clustering) and in dynamical systems (identification of different model structures, recurrent networks, optimal control) and others;
- formulate problems in terms of constrained optimization with explicit use of regularization leading to a good generalization performance and to numerically well-conditioned methods;
- achieve primal and dual model representations, relevant for out-of-sample extensions and solving large scale problems;
- consider weighted versions towards statistical robustness and handling general loss functions;
- plug-in different loss functions and positive definite kernels;
- incorporate prior knowledge through additional constraints and conceive hierarchical modelling schemes using convex optimization techniques.

The emphasis of this paper is on illustrating the main concepts and potential of models with primal and dual representations, in particular for LS-SVMs and in connection to other methods. In general this may contribute to achieving an integrative understanding of the subject given its multi-disciplinary nature, being at the intersection of machine learning and neural networks, mathematics and statistics, pattern recognition and signal processing, systems and control, optimization and others. It also leads to a generic framework that can be applied to a large variety of application areas, especially towards high-dimensional problems.

This paper is organized as follows. Section 2 outlines function estimation in RKHS. Section 3 discusses primal and dual problems in support vector machine classifiers. In Section 4 LS-SVM core models are explained for classification, regression and kernel principal component analysis, together with sparseness, robustness and variable selection. Section 5 gives examples on additional constraints to the core models. In Section 6 weighted kernel PCA models for spectral clustering are discussed, including aspects of model selection and sparse representations. Section 7 focuses on dimensionality reduction and data visualization using kernel maps with a reference point. In Section 8 primal and dual problems for kernel canonical correlation analysis are explained, together with its use for independent component analysis.

2. Function estimation in RKHS

Kernel-based function estimation problems are commonly characterized as follows [23, 62, 86]: for a given training data set $\{(x_i, y_i)\}_{i=1}^N$ of N training data with input data $x_i \in \mathbb{R}^d$ and output data $y_i \in \mathbb{R}$, find a function f that minimizes the objective

$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \nu \|f\|_K^2 \tag{2.1}$$

where $L(\cdot, \cdot)$ denotes the chosen loss function and $\|f\|_K$ the norm in the reproducing kernel Hilbert space (RKHS) \mathcal{H} defined by the kernel K . From the beginning, the unknown function f is restricted here to belong to a reproducing kernel Hilbert space. The positive value ν denotes the regularization constant.

For any convex loss function, it can be shown that the solution to (2.1) is of the form

$$f(x) = \sum_{i=1}^N \alpha_i K(x, x_i) \tag{2.2}$$

which is called the *representer theorem*. The model has the reproducing property

$$f(x) = \langle f, K_x \rangle_K \tag{2.3}$$

with $K_x(\cdot) = K(x, \cdot)$.

By plugging-in different loss functions one obtains among the special cases:

$$\begin{aligned} L(y, f(x)) &= (y - f(x))^2: && \text{regularization network} \\ L(y, f(x)) &= |y - f(x)|_\epsilon: && \text{support vector regression} \end{aligned}$$

where $|\cdot|_\epsilon$ denotes the ϵ -insensitive loss function with $\epsilon \geq 0$ (which is defined as $|y - f(x)|_\epsilon$ equals 0 if $|y - f(x)| \leq \epsilon$ and equals $|y - f(x)| - \epsilon$ otherwise), containing a region around the origin of width 2ϵ where the loss function equals zero. This region results into a sparse representation, meaning that many α_i coefficients are zero. For the case $\epsilon = 0$ it corresponds to an L_1 estimator.

The regularization constant ν controls the bias-variance trade-off. Taking ν too small might result in overfitting the data, while ν too large might give a model that is not sufficiently flexible to explain the data. A common and practical approach to set ν is based e.g. on cross-validation or generalized cross-validation [36]. Usually one is interested in estimating a model that minimizes the generalization error

$$E[f] = \int_{\mathcal{X} \times \mathcal{Y}} L(y, f(x)) dP(x, y) \tag{2.4}$$

under the i.i.d. assumption, with random variables $x \in \mathcal{X}, y \in \mathcal{Y}$ drawn from an unknown probability distribution $P(x, y)$ which is assumed to be unknown but fixed. Different upper bounds and lower bounds on the generalization error have been derived, which are expressed in terms of the model complexity (e.g. VC dimension, Rademacher complexity) [22, 23, 69, 78, 85]. It has been shown that the leave-one-out error plays a vital role with respect to stability and generalization [13, 63]. Robust model selection criteria based on the influence function have been investigated in [27].

3. Support vector machine classifier

3.1. Primal and dual problem

While in a functional analysis setting (2.1) a support vector machine solution can be interpreted as plugging in a suitable loss function, SVMs have been originally conceived in a different way within the context of convex optimization theory [19, 85].

For a classifier problem with given training data $\{(x_i, y_i)\}_{i=1}^N$ with input data $x_i \in \mathbb{R}^d$ and class labels $y_i \in \{-1, 1\}$ one estimates the class labels using the model

$$\hat{y} = \text{sign}[w^T \varphi(x) + b]$$

where the feature map $\varphi(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{n_h}$ maps the data from the input space to a high dimensional feature space. The classifier model corresponds to $\hat{y} = \text{sign}[\sum_{j=1}^{n_h} w_j \varphi_j(x) + b]$ with $\varphi(x) = [\varphi_1(x), \varphi_2(x), \dots, \varphi_{n_h}(x)]^T$. This feature map is usually not explicitly defined at the beginning, but implicitly through choosing a positive definite kernel at the dual level.

The training problem for the SVM classifier is formulated as a constrained optimization problem. The primal problem (P) is stated as:

$$\begin{aligned} (P) \quad & \min_{w, b, \xi} \quad \frac{1}{2} w^T w + c \sum_{i=1}^N \xi_i \\ & \text{subject to} \quad y_i [w^T \varphi(x_i) + b] \geq 1 - \xi_i, \quad i = 1, \dots, N \\ & \quad \quad \quad \xi_i \geq 0, \quad i = 1, \dots, N, \end{aligned} \tag{3.1}$$

where the objective function aims at achieving a trade-off between minimization of the regularization term (corresponding to maximization of the margin

$2/\|w\|_2$) and the amount of tolerated misclassifications, controlled by the regularization constant $c > 0$. The model that is expressed in terms of the feature map appears within the N constraints. The slack variables ξ_i are needed to tolerate misclassifications on the training data, in order to avoid that one would overfit and just memorize the data.

Conceiving the problem as a constrained optimization problem is important in order to create a different representation of the model in terms of Lagrange multipliers α_i (dual variables). These are associated with the first set of constraints in (3.1). One constructs the Lagrangian for the problem and characterizes the saddle point. The solution is given then by the convex quadratic programming problem (dual problem (D))

$$\begin{aligned}
 (D) \quad & \max_{\alpha} \quad -\frac{1}{2} \sum_{i,j=1}^N y_i y_j K(x_i, x_j) \alpha_i \alpha_j + \sum_{j=1}^N \alpha_j \\
 & \text{subject to} \quad \sum_{i=1}^N \alpha_i y_i = 0 \\
 & \quad \quad \quad 0 \leq \alpha_i \leq c, \quad i = 1, \dots, N
 \end{aligned} \tag{3.2}$$

where a positive definite kernel $K(\cdot, \cdot)$ is used satisfying

$$K(x, z) = \varphi(x)^T \varphi(z) = \sum_{j=1}^{n_h} \varphi_j(x) \varphi_j(z) \tag{3.3}$$

for any pair of points $x, z \in \mathbb{R}^d$ (which is often called the *kernel trick*). From the conditions for optimality it further follows that $w = \sum_{i=1}^N \alpha_i y_i \varphi(x_i)$ such that one obtains the following dual representation of the model

$$\hat{y} = \text{sign} \left[\sum_{i \in \mathcal{S}_{SV}} \alpha_i y_i K(x, x_i) + b \right] \tag{3.4}$$

where \mathcal{S}_{SV} denotes the set of support vectors (which is a subset of the training data set) corresponding to the non-zero α_i values. This set automatically follows from solving the convex quadratic programming problem (3.2). Note that the size of the kernel matrix in the quadratic programming problem grows with the number of training data N . On the other hand it is independent of the dimension d of the input space. For large data sets often chunking and decomposition methods are applied.

3.2. Positive definite kernel and feature map

In (3.3) the choice of a positive definite kernel guarantees the existence of a feature map. Only inner products on the feature map are appearing in the derivations, which are replaced then by the positive definite kernel. In fact one can read to equation in two possible ways: from left to right or from right to left.

From left to right in (3.3) one fixes the choice of a positive definite kernel. This guarantees then the existence of an underlying feature map. In this case one does not need to know an explicit expression for the feature map. From right to left in (3.3) one may also explicitly define a feature map and correspondingly obtain the kernel from $K(x, z) := \varphi(x)^T \varphi(z)$.

Some basic choices of commonly used kernels are:

$$\begin{aligned} K(x, x_i) &= x_i^T x \quad (\text{linear kernel}) \\ K(x, x_i) &= (x_i^T x + \tau)^{d_p}, \tau \geq 0 \quad (\text{polynomial kernel of degree } d_p) \\ K(x, x_i) &= \exp(-\|x - x_i\|_2^2 / \sigma^2) \quad (\text{Gaussian radial basis function kernel}). \end{aligned} \tag{3.5}$$

These kernels need a careful model selection for the tuning parameters σ, τ, c . In the case of the linear and polynomial kernel the feature map is finite dimensional. For a Gaussian kernel it is infinite dimensional¹.

The kernel trick has also been further used on its own in order to generate kernel versions of existing algorithms. For example with respect to cluster algorithms, instead of computing the Euclidean distance $\|x - z\|_2$ in the input space between data points x and z , one can create a distance measure by considering the distance in the feature space as

$$\|\varphi(x) - \varphi(z)\|_2^2 = K(x, x) + K(z, z) - 2K(x, z)$$

and use a suitable kernel function then for the given data type. In this context it is also interesting to see that considering the angle θ_{xz} between two vectors x and z in the input space with $\cos \theta_{xz} = x^T z / (\|x\|_2 \|z\|_2)$ becomes a normalized kernel function $\tilde{K}(\cdot, \cdot)$ when considering this in the feature space:

$$\cos \theta_{\varphi(x), \varphi(z)} = \frac{\varphi(x)^T \varphi(z)}{\|\varphi(x)\|_2 \|\varphi(z)\|_2} = \frac{K(x, z)}{\sqrt{K(x, x)} \sqrt{K(z, z)}} = \tilde{K}(x, z). \tag{3.6}$$

Though such a straightforward application of the kernel trick looks attractive at first sight, it might be dangerous. When employing e.g. a Gaussian kernel the model might easily become too flexible (which is often revealed in numerical ill-conditionings). There is a need then to additionally introduce regularization in the scheme in order to avoid overfitting and to achieve a good generalization performance. To avoid such problems often ad hoc regularization schemes are applied afterwards. A more principled approach is taken with methods of least squares support vector machines: regularization terms are considered from the beginning in the primal formulations. These models are also easier to extend to a wider class of problems in supervised and unsupervised learning than standard SVMs.

¹Since the unknown w in the primal has the same dimensionality as the feature map an infinite dimensional Hilbert space setting has to be employed then. However, the Lagrangian approach can also be extended to infinite dimensional problems, see e.g. [47]. An alternative is to treat this infinite dimensional case within a finite dimensional setting by considering a very large but finite value n_h which leads then to an approximate version of the true Gaussian kernel (this difference is small given that the series (3.3) converges for $n_h \rightarrow \infty$). An additional property of well-conditioning is required then (which can be achieved e.g. by the regularization mechanism) to ensure that this small perturbation to the true feature map and the kernel also has a small influence on the overall solution.

4. LS-SVM core models

4.1. Core models in supervised and unsupervised learning

In least squares support vector machines one works with equality constraints instead of inequality constraints and an L_2 loss function. Advantages are that

- characterizing the conditions for optimality becomes simpler. The core models are easier extendable with additional constraints;
- it becomes possible to extend support vector methodology to a wide range of problems in supervised and unsupervised learning and beyond;
- it captures the simple essence while still providing high performant models (often also with easier software implementations);
- it leads to numerically reliable schemes and problems for which issues like conditioning are better understood.

These points will be further illustrated in the sequel of this paper.

Classification

The LS-SVM classifier training is formulated as follows [74]

$$\begin{aligned} \min_{w,b,e_i} \quad & \frac{1}{2}w^T w + \gamma \frac{1}{2} \sum_{i=1}^N e_i^2 \\ \text{subject to} \quad & y_i[w^T \varphi(x_i) + b] = 1 - e_i, \quad i = 1, \dots, N. \end{aligned} \tag{4.1}$$

Instead of considering the value 1 within the constraints as a threshold value, it is taken here as a target value. This implicitly corresponds to a regression on the class labels ± 1 , from which the link between this method and kernel Fisher discriminant analysis can be understood.

From the Lagrangian

$$\mathcal{L}(w, b, e; \alpha) = \frac{1}{2}w^T w + \gamma \frac{1}{2} \sum_{i=1}^N e_i^2 - \sum_{i=1}^N \alpha_i \{y_i[w^T \varphi(x_i) + b] - 1 + e_i\}$$

with Lagrange multipliers α_i , one takes the conditions for optimality which are given by

$$\begin{cases} \partial \mathcal{L} / \partial w = 0 & \rightarrow w = \sum_i \alpha_i y_i \varphi(x_i) \\ \partial \mathcal{L} / \partial b = 0 & \rightarrow \sum_i \alpha_i y_i = 0 \\ \partial \mathcal{L} / \partial e_i = 0 & \rightarrow \gamma e_i = \alpha_i, \quad i = 1, \dots, N \\ \partial \mathcal{L} / \partial \alpha_i = 0 & \rightarrow y_i [w^T \varphi(x_i) + b] = 1 - e_i, \quad i = 1, \dots, N. \end{cases} \tag{4.2}$$

Eliminating w, e and writing the solution in α, b gives the square linear system

$$\left[\begin{array}{c|c} 0 & y^T \\ \hline y & \Omega^{(y)} + I/\gamma \end{array} \right] \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_N \end{bmatrix} \tag{4.3}$$

where $\Omega_{ij}^{(y)} = y_i y_j \varphi(x_i)^T \varphi(x_j) = y_i y_j K(x_i, x_j)$ and column vectors $y = [y_1; \dots; y_N] = [y_1 \dots y_N]^T$, $1_N = [1; \dots; 1]$ and I the identity matrix. For the LS-SVM classifier model \mathcal{M} evaluated at any point $x_* \in \mathbb{R}^d$, the primal (P) and dual (D) model representations and corresponding prediction \hat{y}_* are given by

$$\begin{array}{ccc}
 & (P) : \hat{y}_* = \text{sign}[w^T \varphi(x_*) + b] & \\
 \nearrow & \updownarrow & \\
 \mathcal{M} & & \\
 \searrow & (D) : \hat{y}_* = \text{sign}[\sum_{i=1}^N \alpha_i y_i K(x_*, x_i) + b]. &
 \end{array} \tag{4.4}$$

The proximal support vector machine classifier [33] has been related to the LS-SVM. A main difference is that the former regularizes the bias (intercept) term b as well.

Regression

In a similar way one can perform a ridge regression in the feature space [67, 77] with additional bias term b

$$\begin{aligned}
 \min_{w, b, e_i} \quad & \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^N e_i^2 \\
 \text{subject to} \quad & y_i = w^T \varphi(x_i) + b + e_i, \quad i = 1, \dots, N
 \end{aligned} \tag{4.5}$$

which gives as dual problem

$$\left[\begin{array}{c|c} 0 & 1_N^T \\ \hline 1_N & \Omega + I/\gamma \end{array} \right] \left[\begin{array}{c} b \\ \alpha \end{array} \right] = \left[\begin{array}{c} 0 \\ y \end{array} \right] \tag{4.6}$$

where $\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$. The corresponding primal and dual model representations are

$$\begin{array}{ccc}
 & (P) : \hat{y}_* = w^T \varphi(x_*) + b & \\
 \nearrow & \updownarrow & \\
 \mathcal{M} & & \\
 \searrow & (D) : \hat{y}_* = \sum_i \alpha_i K(x_*, x_i) + b. &
 \end{array} \tag{4.7}$$

Kernel principal component analysis

Kernel principal component analysis as proposed in [68] can be obtained as the dual problem to the following LS-SVM formulation [79]:

$$\begin{aligned}
 \min_{w, b, e_i} \quad & -\frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^N e_i^2 \\
 \text{subject to} \quad & e_i = w^T \varphi(x_i) + b, \quad i = 1, \dots, N.
 \end{aligned} \tag{4.8}$$

The problem in the Lagrange multipliers α_i related to the constraints is then given by

$$\Omega^{(c)}\alpha = \lambda\alpha \quad \text{with} \quad \lambda = 1/\gamma \tag{4.9}$$

where $\Omega_{ij}^{(c)} = (\varphi(x_i) - \hat{\mu}_\varphi)^T(\varphi(x_j) - \hat{\mu}_\varphi)$ denote the elements of the centered kernel matrix and $\hat{\mu}_\varphi = (1/N) \sum_{i=1}^N \varphi(x_i)$. The centering of the kernel matrix is automatically obtained from the conditions for optimality by taking a bias term b in the model.

The interpretation between (4.8) and kernel PCA is as follows:

1. *Pool of candidate components:*

Equation (4.8) characterizes the pool of all candidate components. Note that all eigenvectors which are the solution to (4.9) lead to a value zero for the objective function $-\frac{1}{2}w^T w + \gamma \frac{1}{2} \sum_{i=1}^N e_i^2$. The corresponding possible choices of the regularization constant γ follow from the eigenvalues $\lambda = 1/\gamma$ of the different possible solutions.

2. *Relevant components:*

For the kernel PCA problem one is interested in the components that are maximizing the variance. The component corresponding to λ_{\max} results in maximizing the second term in the objective $\gamma \frac{1}{2} \sum_{i=1}^N e_i^2$.

The primal and dual model representations are given by

$$\begin{array}{ccc} & (P) : \hat{e}_* = w^T \varphi(x_*) + b & \\ \mathcal{M} \nearrow & & \updownarrow \\ & (D) : \hat{e}_* = \sum_i \alpha_i K(x_*, x_i) + b. & \end{array} \tag{4.10}$$

By means of this underlying model it is also clear how out-of-sample extensions can be made. When making eigenvalue decompositions on data matrices directly, as commonly done in the literature, the out-of-sample extension aspects are mostly unclear. The model with out-of-sample extensions also enables to evaluate on validation data.

Solving in primal or dual?

In case the feature map is finite dimensional and explicitly known one has the choice between solving the primal or the dual problem (for the Gaussian kernel on the other hand one can only solve the dual). Consider e.g. the case of a linear parametric regression model $\hat{y} = w^T x + b$ with $w \in \mathbb{R}^d$. The dual representation of the linear model is $\hat{y} = \sum_{i=1}^N \alpha_i x_i^T x + b$ with $\alpha \in \mathbb{R}^N$. One distinguishes between the following cases then:

- Case d small, N large: solving the primal problem in $w \in \mathbb{R}^d$ is more convenient.
- Case d large, N small: solving the dual problem in $\alpha \in \mathbb{R}^N$ is more convenient.

Therefore within a setting of primal-dual model representations one can tailor the approach towards the given data problem, while preserving the global picture with parametric interpretations in the primal and kernel representations in the dual. This view is further exploited in fixed-size kernel models with estimation in the primal based on general positive definite kernels. In the next subsection this topic is further addressed.

4.2. Sparseness and robustness

Though the use of least squares and equality constraints simplifies the formulations, it also has the drawback that in general no sparse model representation is obtained and the estimator is non-robust. However, different methods have been developed to overcome these problems.

Sparseness: Fixed-size kernel models

Reduction and pruning techniques have been used to achieve the sparse representation in a second stage, which have been successfully applied [77]. A different approach which makes use of the primal-dual setting are *fixed-size* techniques. The fixed-size method has the following main characteristics:

1. For a given positive definite kernel, estimate an approximate finite dimensional feature map based on a small subset of the training data;
2. Define a selection criterion for obtaining the subset;
3. Estimate the model in the primal leading to a model of the form

$$\hat{y} = \sum_{i \in \mathcal{S}} \beta_i K(x, x_i) + b \quad (4.11)$$

where \mathcal{S} is a subset of the training data set.

For fixed-size LS-SVMs proposed in [77] step 1 is based on the Nyström approximation as proposed in the context of Gaussian processes [88]. A direct consequence of step 2 is that a sparse representation is obtained with a number of support vectors and dimensionality of the feature space equal to the size of the subset. In step 3, instead of taking the subset at random, the support vectors are chosen such that the sum of the elements of the kernel matrix is optimized, which characterizes the quadratic Renyi entropy. In this way the support vectors act as prototypes for the underlying input data distribution, as in vector quantization methods. While vector quantization approaches have been studied in the past for placing the centers of radial basis function networks, the fixed-size techniques are applicable to a broader class of positive definite kernels. It is also based on the existing connection between kernel PCA and density estimation [35].

Optimized versions of fixed-size LS-SVMs are currently applicable to large data sets with a million of data points for training and tuning on a personal computer [26]. Successful applications in electricity load forecasting have been reported in [31].

Robust regression: Weighting

For linear parametric models it is well-known that outliers may breakdown the quality of the estimated model when using a least squares estimator. A key element at this point is that the derivative of the loss function is not bounded, which on the other hand is the case when using e.g. the Huber loss function. When using an L_2 loss function with kernel-based models the situation is not as bad as in the linear parametric case. When using a bounded kernel such as the Gaussian kernel the model quality will rather be locally instead of globally destroyed. Nevertheless, it might be important to further improve the estimates. A procedure proposed for use in LS-SVM regression [76] is to first estimate with an L_2 loss function and then further apply (an) additional weighting step(s) by weighted least squares:

$$\begin{aligned} \min_{w,b,e_i} \quad & \frac{1}{2}w^T w + \gamma \frac{1}{2} \sum_{i=1}^N v_i e_i^2 \\ \text{subject to} \quad & y_i = w^T \varphi(x_i) + b + e_i, \quad i = 1, \dots, N. \end{aligned} \quad (4.12)$$

The influence of outlier points is down-weighted in this scheme by associating a small weight v_i to them in view of robust statistics [42, 65]. The choice of the weights is based on the distribution of the residuals e_i from the first (unweighted) estimation step. In [28] fast convergence and robust estimation by means of using a logistic weighting and bounded kernels has been reported. Further theoretical studies on robust model selection and conditions on the weighting, loss function and kernel have been made in [27, 28].

In general, with respect to the choice of the loss function, one has two possible ways to proceed: top-down or bottom-up. Either one chooses the loss function in a top-down fashion and, in case of a convex loss function $L(\cdot)$, in

$$\begin{aligned} \min_{w,b,e_i} \quad & \frac{1}{2}w^T w + \gamma \frac{1}{2} \sum_{i=1}^N L(e_i) \\ \text{subject to} \quad & y_i = w^T \varphi(x_i) + b + e_i, \quad i = 1, \dots, N \end{aligned} \quad (4.13)$$

one applies a convex optimization procedure to compute the unique solution to the problem. Otherwise, in a bottom-up way as described above in (4.12), one starts from the simpler LS-SVM model and further improves the estimates by re-weighting. In both approaches least squares plays a central role. In the top-down procedure, when using e.g. an interior point algorithm for convex optimization (and other methods as [61]) the reduced Karush-Kuhn-Tucker system to be solved at one iteration step has the same structure as one single LS-SVM. In the bottom-up approach few re-weighting steps are needed, in practice often one additional step is satisfactory [25]. Besides producing robust estimates the bottom-up approach might also give a computational advantage. The applied weighting will implicitly correspond to a modified loss function (with bounded derivative). Also in sparse recovery problems the importance of iteratively re-weighted least squares has been stressed [24].

Kernel component analysis: Robustness and sparseness

Also in unsupervised learning, the issue of robustness has been studied. New methods of Kernel Component Analysis (KCA) [2] have been studied, with robust and sparse modifications to kernel PCA. This is done by starting from the LS-SVM formulation to kernel PCA and plugging-in different loss functions $L(\cdot)$:

$$\begin{aligned} \min_{w,b,e_i} \quad & -\frac{1}{2}w^T w + \gamma \sum_{i=1}^N L(e_i) \\ \text{subject to} \quad & e_i = w^T \varphi(x_i) + b, \quad i = 1, \dots, N. \end{aligned} \quad (4.14)$$

Kernel component analysis has been studied for the Huber loss function and weighted least squares. In the weighted least squares approach one takes $L(e_i) = \frac{1}{2}v_i e_i^2$. The components are computed then in different stages, where in stage k the new component is made orthogonal with respect to the $k - 1$ previous components. The knowledge of the previous components is incorporated by additional orthogonality constraints to the primal problem. It leads to solving a sequence of generalized eigenvalue problems [2]. When employing a Huber loss with epsilon-insensitive zone, sparse and robust KCA models are obtained, which is illustrated in Figure 1. In contrast with regression problems, the different components have different degrees of sparseness in this case [2].

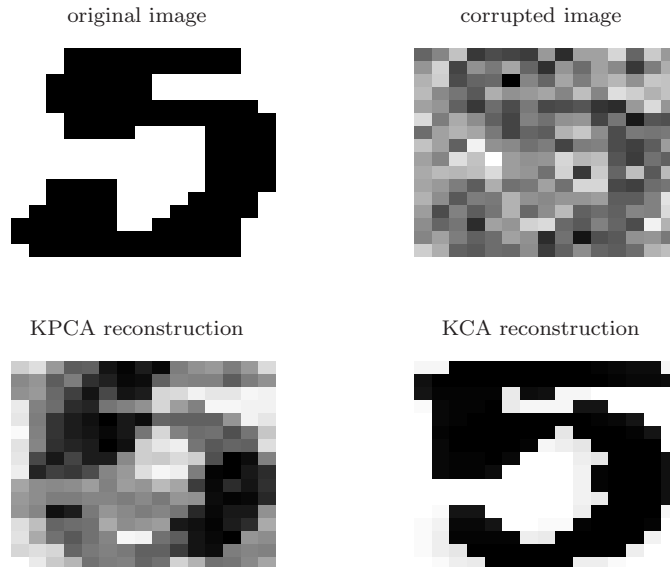


FIG 1. Robust denoising using Kernel Component Analysis: (Top-left) Original digit; (Top-right) digit corrupted by outliers and Gaussian noise; (Bottom-left) KPCA reconstruction result; (Bottom-right) KCA reconstruction result using a Huber loss function with epsilon-insensitive zone. The algorithms were trained on 300 images from the CI “multiple features” dataset. The number of components used was 32.

4.3. Variable selection

With the use of Gaussian kernels a common method for classification and regression is automatic relevance determination [50] where instead of one single kernel parameter σ a diagonal weighting matrix W is considered by taking $K(x, z) = \exp(-(x-z)^T W (x-z))$ where the elements of W are positive. Using Bayesian inference these elements are inferred at a higher level of inference [77]. For the linear kernel one can take in a similar way $K(x, z) = x^T W z$. Note that leaving out the j -th variable from the model corresponds to setting $W_{jj} = 0$. One typically searches then for a subset of variables that minimizes e.g. the leave-one-out error or cross-validation error, which results into solving a combinatorial optimization problem. Common heuristics that are used as alternative to the latter are forward or backward subset selection methods. In kernel-based modelling, the process of variable selection is usually time-consuming. Computationally efficient techniques based on low-rank updates for fast variable selection have therefore been proposed in [55]. Currently these can be applied for the case of linear and polynomial kernels. An overview of filtering, wrapper and embedded methods for variable selection in support vector machine classifiers with applications in chemometrics has been reported in [49].

A number of methods for variable selection are based on convex optimization. While in a linear parametric setting one has a large flexibility in taking different loss functions and regularization terms, in SVM and LS-SVM formulations the 2-norm based regularization term $w^T w$ is crucial in order to generate the kernel-based representation in the dual from the conditions for optimality. The choice of other norms on w (like e.g. L_1 regularization or LASSO [81]) within the SVM or LS-SVM primal problem is only possible for an explicitly given expression of a finite dimensional feature map and provided one directly solves the primal problem. In this case the problem reduces to estimating the parameters of a parameterized model for a fixed set of basis functions. In the linear SVM case one therefore has more flexibility in using different norms for achieving sparse representations for variable selection (see e.g. [15]). A conceptually different approach of defining LS-SVM substrates has been proposed as a more general alternative [59]. It conceives different hierarchical levels where at the basic level the LS-SVM substrate is taken with an additive (instead of a multiplicative) regularization trade-off. The variable selection problem is then defined at a higher hierarchical level e.g. with the use of L_1 regularization. Computationally, the different hierarchical levels are fused into solving a convex optimization problem.

5. Core models plus additional constraints

The optimization setting enables to add different regularization terms and constraints in a systematic way. After conceiving and formulating the problem in the primal, from the conditions for optimality one obtains the optimal kernel based model representation and the final model estimate (Figure 2).

Some examples are given here to illustrate this.

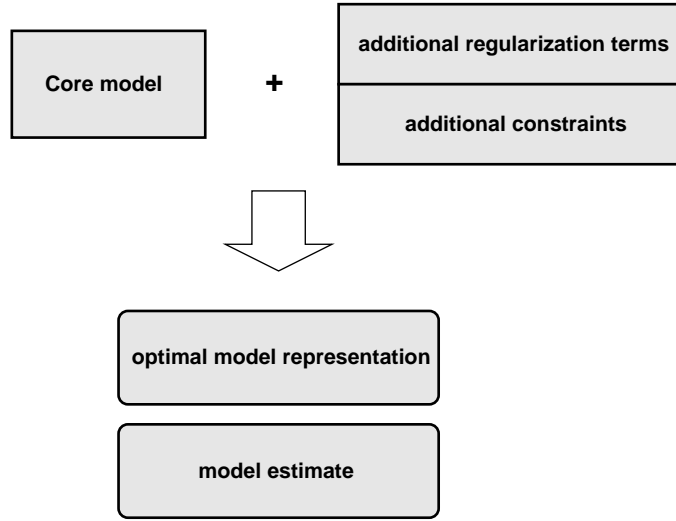


FIG 2. Schematic illustration of conceiving the primal problem as a core model that is systematically extendable with additional constraints and regularization terms. The optimal model representation and model estimates follow from the conditions for optimality.

Multi-class problems

Multi-class problems have been approached using LS-SVMs by including additional sets of constraints in the primal formulation. For a given training data set $\{(x_i, y_i)\}_{i=1}^N$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}^{n_y}$ where n_y denotes the number of output variables with $y_i = [y_i^{(1)}; \dots; y_i^{(n_y)}]$, one formulates

$$\begin{aligned}
 \min_{w^{(j)}, b^{(j)}, e_i^{(j)}} & \quad \frac{1}{2} \sum_{j=1}^{n_y} w^{(j)T} w^{(j)} + \frac{1}{2} \sum_{j=1}^{n_y} \gamma_j \sum_{i=1}^N e_i^{(j)2} \\
 \text{subject to} & \quad y_i^{(1)} [w^{(1)T} \varphi^{(1)}(x_i) + b^{(1)}] = 1 - e_i^{(1)}, \quad i = 1, \dots, N, \\
 & \quad y_i^{(2)} [w^{(2)T} \varphi^{(2)}(x_i) + b^{(2)}] = 1 - e_i^{(2)}, \quad i = 1, \dots, N, \\
 & \quad \vdots \\
 & \quad y_i^{(n_y)} [w^{(n_y)T} \varphi^{(n_y)}(x_i) + b^{(n_y)}] = 1 - e_i^{(n_y)}, \quad i = 1, \dots, N.
 \end{aligned} \tag{5.1}$$

The classifier is based in this case on a number of n_y estimated output values $\hat{y}_*^{(j)} = \text{sign}[w^{(j)T} \varphi^{(j)}(x_*) + b^{(j)}]$ for $j = 1, \dots, n_y$. For each part one may consider a different feature map $\varphi^{(j)}(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{n_{h_j}}$. The number n_y depends then on the chosen coding/decoding scheme (e.g. one versus one, one versus all, minimal output coding) [34, 75, 77, 84]. Also the regression case with multiple outputs has been handled in a similar way [77].

Monotonicity constraints

If one has the prior knowledge that the estimated values \hat{y}_i (where $y_i = \hat{y}_i + e_i$) have to satisfy the ordering $\hat{y}_1 \leq \hat{y}_2 \leq \dots \leq \hat{y}_N$, one can add these constraints pointwise in a pairwise way at the primal level:

$$\begin{aligned} \min_{w,b,e} \quad & \frac{1}{2}w^T w + \gamma \frac{1}{2} \sum_{i=1}^N e_i^2 \\ \text{subject to} \quad & y_i = w^T \varphi(x_i) + b + e_i, \quad i = 1, \dots, N \\ & w^T \varphi(x_i) \leq w^T \varphi(x_{i+1}), \quad i = 1, \dots, N - 1. \end{aligned} \tag{5.2}$$

In [57] this has been discussed in the context of estimating cumulative distribution functions, for the case (5.2) and the case of monotone Chebyshev kernel regression. Related work of knowledge incorporation has also been addressed in [51].

Structure detection

L_1 regularization is a common tool in parametric methods to achieve a sparse solution vector (e.g. LASSO estimator, compressed sensing). In [58] an L_1 regularization mechanism has been used on top of an LS-SVM core model:

$$\begin{aligned} \min_{w^{(p)}, e, t_p} \quad & \frac{1}{2} \sum_{p=1}^P w^{(p)T} w^{(p)} + \gamma \frac{1}{2} \sum_{i=1}^N e_i^2 + \mu \sum_{p=1}^P t_p \\ \text{subject to} \quad & y_i = \sum_{p=1}^P w^{(p)T} \varphi^{(p)}(x_i^{(p)}) + e_i, \quad i = 1, \dots, N \\ & -t_p \leq w^{(p)T} \varphi^{(p)}(x_i^{(p)}) \leq t_p, \quad i = 1, \dots, N; p = 1, \dots, P. \end{aligned} \tag{5.3}$$

In this case an additive or componentwise model is used where each of the components $p = 1, \dots, P$ is equipped with a feature map $\varphi^{(p)}$. At the dual level this leads to a sum of kernel functions (as in an additive model [40, 86]). The structure detection has been done then by inspecting how the solution changes when varying the regularization constant μ . This is illustrated on a synthetic example created from the motorcycle data set in Figure 3. In this method, components which persist with a large non-zero t_p value for a wide range of μ values, are considered to be relevant. Instead of the scheme (5.3) with multiplicative regularization trade-off, an alternative scheme with additive regularization trade-off has been investigated in [58, 60].

Semi-supervised learning

In [48] a semi-supervised learning model has been formulated by adding constraints that specify whether a point has been labeled or not:

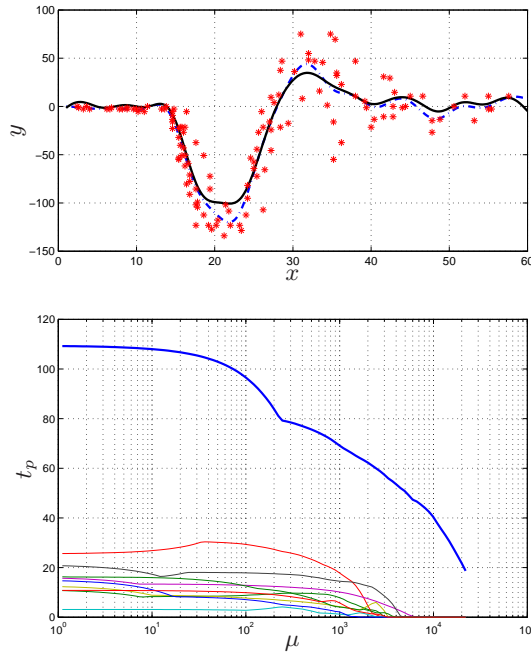


FIG 3. Illustration of structure detection by (5.3) on a synthetic example based on the univariate motorcycle data set (9 irrelevant input variables have been artificially added with random and spurious components): (Top) comparison between basic LS-SVM regression (solid line) and structure detection (dashdot line) (for $\mu = 3000$); (Bottom) Optimal values t_p for each of the $p = 1, \dots, P$ components as a function of the regularization constant μ .

$$\begin{aligned}
 \min_{w,b,e,\hat{y}} \quad & \frac{1}{2}w^T w + \gamma \frac{1}{2} \sum_{i=1}^N e_i^2 + \eta \frac{1}{2} \sum_{i,j=1}^N v_{ij} (\hat{y}_i - \hat{y}_j)^2 \\
 \text{subject to} \quad & \hat{y}_i = w^T \varphi(x_i) + b, \quad i = 1, \dots, N \\
 & \hat{y}_i = \nu_i y_i - e_i, \nu_i \in \{0, 1\}, \quad i = 1, \dots, N
 \end{aligned} \tag{5.4}$$

with $\nu_i = 1$ for a labeled point and $\nu_i = 0$ for an unlabeled point. Related but different formulations for semi-supervised learning in RKHS have been discussed in [10, 16] and for graph-based learning in [82].

Colored noise models

In [30, 31] auto-correlated errors have been modelled by adding constraints to the core model by

$$\begin{aligned}
 \min_{w,b,r,e} \quad & \frac{1}{2}w^T w + \gamma \frac{1}{2} \sum_{i=1}^N r_i^2 \\
 \text{subject to} \quad & y_i = w^T \varphi(x_i) + b + e_i, \quad i = 1, \dots, N \\
 & e_i = \rho e_{i-1} + r_i, \quad i = 2, \dots, N.
 \end{aligned} \tag{5.5}$$

At the dual level this results into an equivalent kernel which depends then on ρ as an additional tuning parameter.

Structured nonlinear models

An example towards estimating structured nonlinear dynamical systems is found in the identification of Hammerstein systems. These systems consist of the interconnection of a static nonlinear function applied to the input variable followed by a linear dynamical system model. This also relate to problems in independent component analysis [12]. Kernel-based modelling based on primal and dual representations of these systems has been addressed in [37, 38]. Suppose a single input single output variable system with a linear ARX part and a nonlinear function $g(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ as a model for the Hammerstein system: $\hat{y}_k = \sum_{i=1}^{n_y} a_i y_{k-i} + \sum_{j=1}^{n_u} \beta_j g(u_{k-j})$ with the representation $g(u_{k-j}) = w^T \varphi(u_{k-j}) + b_0$. The estimation of a_i, β_j, w, b_0 would lead then to a non-convex problem with many local minima solutions. This can be rephrased as a convex problem by making use of overparametrization as

$$\begin{aligned} \min_{w_j, a, b, e_k} \quad & \frac{1}{2} \sum_{j=1}^{n_u} w_j^T w_j + \gamma \frac{1}{2} \sum_{k=d+1}^{d+N} e_k^2 \\ \text{subject to} \quad & y_k = \sum_{i=1}^{n_y} a_i y_{k-i} + \sum_{j=1}^{n_u} w_j^T \varphi(u_{k-j}) + b + e_k, \quad \forall k \\ & \sum_{k=1}^N w_j^T \varphi(u_k) = 0, \quad \forall j = 1, \dots, n_u. \end{aligned} \tag{5.6}$$

The solution to the original problem can then be obtained by projecting the solution onto the Hammerstein model class using a singular value decomposition [37, 38]. Further extensions have been made to Wiener-Hammerstein systems [32].

6. Models for spectral clustering

Spectral clustering algorithms have been formulated as relaxations to graph partitioning problems [17, 54, 72]. For the case of finding two clusters \mathcal{A}, \mathcal{B} in a graph \mathcal{G} , one considers the minimal cut problem

$$\min_{\xi_i \in \{-1, +1\}} \frac{1}{2} \sum_{i,j=1}^N a_{ij} (\xi_i - \xi_j)^2 \tag{6.1}$$

with cluster membership indicator $\xi_i = 1$ if $i \in \mathcal{A}$, $\xi_i = -1$ if $i \in \mathcal{B}$. The values a_{ij} of the affinity matrix characterize the links between nodes i and j where $i, j = 1, \dots, N$. Relaxing $\xi \in \{-1, +1\}^N$ into $\xi^T \xi = 1$ leads then to solving an eigenvalue problem for the given graph Laplacian matrix. The

clustering information is contained in the eigenvectors of the Laplacian matrix $L = D - A$ derived from the data with degree matrix D and $A = [a_{ij}]$. This type of unsupervised learning method is often considered as a pre-processing step when mapping the original data to an eigenspace where the clusters become more evident. An additional clustering step (e.g. by k -means) is needed then to obtain the final grouping from the eigenvectors. The focus of this Section is now to describe models of kernel spectral clustering based on a weighted version of kernel PCA [4].

6.1. Weighted kernel PCA for kernel spectral clustering

Starting from the LS-SVM formulation to kernel PCA without bias term and by introducing weighting factors $v_i \in \mathbb{R}^+, i = 1, \dots, N$ one has the primal problem

$$\begin{aligned} \min_{w,e} \quad & -\frac{1}{2}w^T w + \gamma \frac{1}{2} \sum_{i=1}^N v_i e_i^2 \\ \text{subject to} \quad & e_i = w^T \varphi(x_i), \quad i = 1, \dots, N. \end{aligned} \tag{6.2}$$

The dual problem in the Lagrange multipliers is given by the following non-symmetric eigenvalue problem:

$$V\Omega\alpha = \lambda\alpha, \quad \lambda = 1/\gamma \tag{6.3}$$

where $V = \text{diag}([v_1, \dots, v_N])$ is the user-defined weighting matrix. The kernel matrix is playing the role here of the affinity matrix in (6.1). If V is chosen to be the inverse degree matrix of the graph $D^{-1} = \text{diag}([1/d_1, \dots, 1/d_N])$ with $d_i = \sum_{j=1}^N \Omega_{ij}, i = 1, \dots, N$ and $\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$, then the dual problem of the weighted kernel PCA formulation becomes the random walks algorithm for spectral clustering [52], which is also related to the normalized cut problem [72]. The non-symmetric eigenvalue problem (6.3) corresponds then to the generalized eigenvalue problem

$$\Omega\alpha = \lambda D\alpha \tag{6.4}$$

with $\lambda_1 = 1 \geq \lambda_2 \geq \dots \geq \lambda_N$.

Like in the kernel PCA case, the pool of candidate components is obtained from specifying the primal (6.2). The relevant components are the ones that minimize the normalized cut. The estimated cluster indicators $\hat{\xi}_i$ are obtained then by binarizing $\alpha^{(2)}$, which is the eigenvector corresponding to the second largest eigenvalue of $D^{-1}\Omega$:

$$\hat{\xi}_i = \text{sign}(\alpha_i^{(2)} - \theta), i = 1, \dots, N$$

where θ is a suitable threshold. This interpretation of spectral clustering as a special case of weighted kernel PCA also allows extending the cluster indicators to unseen data (out-of-sample extension) by means of projections onto the eigenvectors.

6.2. Multiway kernel spectral clustering with out-of-sample extensions

Primal and dual problem

The weighted kernel PCA formulation can be further extended to more than two clusters. This is achieved by introducing additional score variables and equality constraints. Bias terms are added for obtaining optimal centering [4]:

$$\begin{aligned}
 \min_{w^{(l)}, e^{(l)}, b_l} \quad & -\frac{1}{2} \sum_{l=1}^{k-1} w^{(l)T} w^{(l)} + \frac{1}{2N} \sum_{l=1}^{k-1} \gamma_l e^{(l)T} D^{-1} e^{(l)} \\
 \text{subject to} \quad & e^{(1)} = \Phi_{N \times n_h} w^{(1)} + b_1 1_N \\
 & e^{(2)} = \Phi_{N \times n_h} w^{(2)} + b_2 1_N \\
 & \vdots \\
 & e^{(k-1)} = \Phi_{N \times n_h} w^{(k-1)} + b_{k-1} 1_N
 \end{aligned} \tag{6.5}$$

where k denotes the number of clusters, $e^{(l)} = [e_1^{(l)}; \dots; e_N^{(l)}]$ the score variables, $\Phi_{N \times n_h} = [\varphi(x_1)^T; \dots; \varphi(x_N)^T] \in \mathbb{R}^{N \times n_h}$ is the feature maps matrix evaluated on the training data, b_l the bias terms where $l = 1, \dots, k - 1$. The equality constraints of the primal problem (6.5) represent a set of $k - 1$ binary cluster decisions from $\text{sign}(e^{(l)})$. These binary cluster indicators can be interpreted as possible codewords. The final cluster membership is assigned by comparing the binary cluster indicators with the k codewords of the codebook and selecting the codeword which minimizes the Hamming distance. Note that a related coding/decoding approach has been taken for multi-class classification problems (see Section 5), but in a supervised instead of unsupervised learning context.

The dual problem is given by the following eigenvalue problem

$$D^{-1} M_D \Omega \alpha^{(l)} = \lambda_l \alpha^{(l)}, \quad l = 1, \dots, k - 1, \tag{6.6}$$

where $M_D = I_N - (1_N 1_N^T D^{-1} / 1_N^T D^{-1} 1_N)$ and $\lambda_l = N / \gamma_l, l = 1, \dots, k - 1$ ordered as $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$. The primal and dual model representations evaluated at a point x_* are given by

$$\begin{array}{ccc}
 & (P) : \quad \text{sign}[\hat{e}_*^{(l)}] = \text{sign}[w^{(l)T} \varphi(x_*) + b_l], \quad l = 1, \dots, k - 1 & \\
 \mathcal{M} & \nearrow & \updownarrow \\
 & (D) : \quad \text{sign}[\hat{e}_*^{(l)}] = \text{sign}[\sum_j \alpha_j^{(l)} K(x_*, x_j) + b_l], \quad l = 1, \dots, k - 1. & \\
 & \searrow & \\
 & & (6.7)
 \end{array}$$

In classical spectral clustering, the extension of the clustering results to new points (out-of-sample points) relies on approximations such as the Nyström method. A main advantage of using a weighted kernel PCA model for spectral clustering lies in extending the clustering results to out-of-sample points

naturally via projections onto the eigenvectors, without having to rely on approximations. For evaluation at a new point x_* , the cluster indicators can be obtained by binarizing the score variables $\text{sign}[\hat{e}_*^{(l)}]$, $l = 1, \dots, k - 1$ which represents the $(k - 1)$ -dimensional codeword of x_* . It is decoded by assigning x_* to the cluster that minimizes the Hamming distance with respect to the codewords in the codebook.

As demonstrated in [5] additional prior knowledge of must-link and/or cannot-link clusters can be incorporated into the primal (6.5) by adding constraints.

Model selection

The out-of-sample extension also allows model selection in a learning framework with training, validation and test parts. When the clusters present in the data are well-separated, the leading eigenvectors of the dual problem (6.6) are piecewise constant for an appropriate choice of the kernel parameter. This property means that the clusters are represented as single points in the eigenspace and hence easy to cluster using e.g. k -means. However, this structural property only holds for the eigenvectors which are representing the training data. In the case of out-of-sample data, the clusters can become represented as lines in the score variables space.

The Balanced Line Fit (BLF) criterion proposed in [4] can be used to obtain the number of clusters k and the kernel parameters such that the projections are as collinear as possible together with balanced clusters. This can be evaluated on a validation set or cross-validation can be applied to it. The BLF value ranges between zero and one, taking its maximal value when the projections are perfectly collinear and zero when the projections are spherically distributed. Figure 4 shows a model selection experiment with the BLF on a toy data set. The score variables and the corresponding clustering results are shown for two different RBF kernel parameters. The BLF is optimal for $\sigma^2 = 0.16$ in this example which leads to correctly detecting the three clusters.

Sparse kernel models

For large scale problems, the cost of storing the matrix $D^{-1}M_D\Omega$ and computing its eigendecomposition can be prohibitive. Sparse kernel models using the incomplete Cholesky decomposition have been studied in [3]. This sparse model aims at approximating the full eigenvectors by solving a smaller eigenvalue problem. The incomplete Cholesky decomposition also gives a set of *pivots* which can serve as support vectors to approximate the expansions for the estimation of the cluster indicators. The kernel matrix can be decomposed as $\Omega \approx GG^T$ where $G \in \mathbb{R}^{N \times R}$ is the lower triangular incomplete Cholesky factor. R denotes the number of pivots which is controlled via a user-defined error threshold and $R \ll N$. Instead of solving (6.6), the following approximation is taken then

$$U^T D^{-1} M_D U \Lambda^2 \rho^{(l)} = \lambda_l \rho^{(l)}, \quad l = 1, \dots, k - 1 \quad (6.8)$$

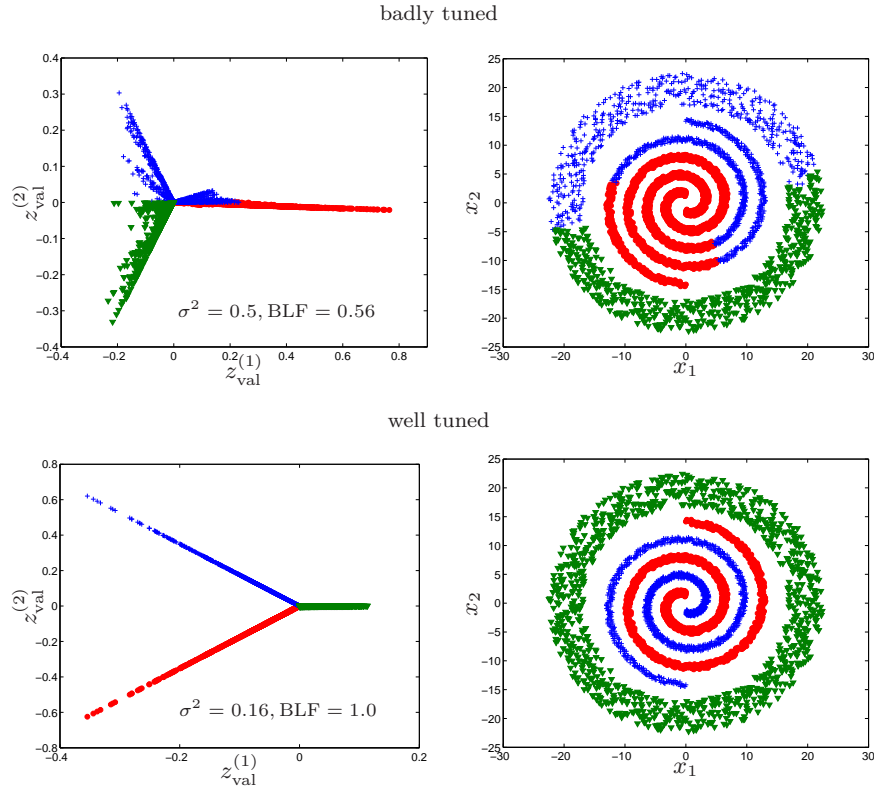


FIG 4. Model selection in multiway kernel spectral clustering based on weighted kernel PCA: (Right) Weighted kernel PCA for spectral clustering illustrated on a toy data set; (Left) Application of the balanced line fit (BLF) criterion for model selection using a Gaussian kernel. (Top) Score variables on validation data with $\sigma^2 = 0.5$ and corresponding clustering result; (Bottom) optimal value $\sigma^2 = 0.16$ corresponding to a clear line structure in the model selection.

with $\rho^{(l)} = U^T \alpha^{(l)}$. $U \in \mathbb{R}^{N \times R}$ is the matrix of left singular vectors of G and $\Lambda \in \mathbb{R}^{R \times R}$ the diagonal matrix of singular values. Note that (6.8) involves the eigendecomposition of a $R \times R$ matrix which can be much smaller than the full $N \times N$ matrix in (6.6). The cluster indicators can also be expressed in terms of the pivots by using a reduced set method and solving a linear system of size $R \times R$ [3]. Figure 5 shows the method on an image segmentation application. The image is from the Berkeley image dataset <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench>. The total number of pixels is 154,401 from which only 175 compose the support pixel set.

Highly sparse representations

Highly sparse kernel models for spectral clustering have been discussed in [6]. In this case, the *pivots* are selected by choosing specific points from the line struc-

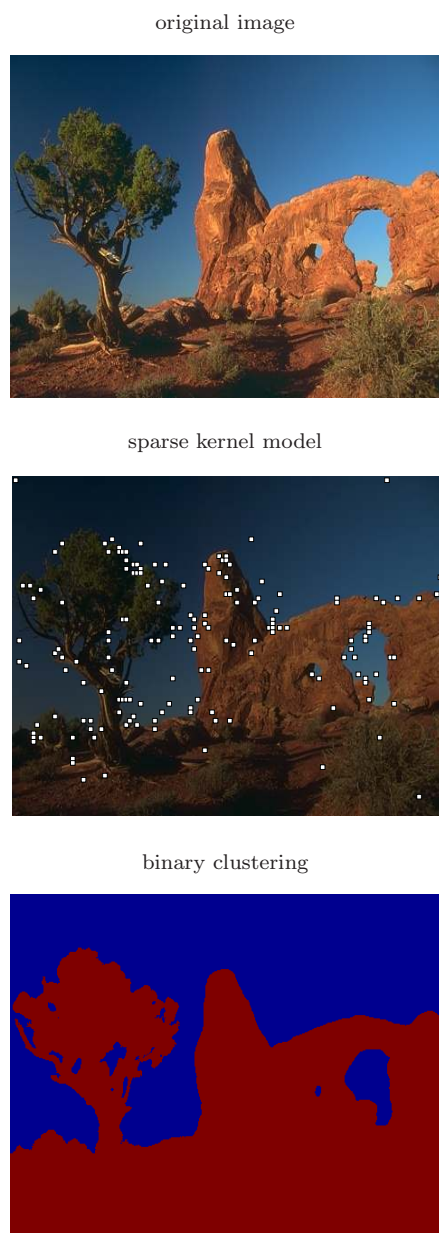


FIG 5. Sparse kernel model for spectral clustering: (Top) Original 321×481 pixels image for a total of 154,401 pixels; (Center) 175 Support pixels found by the sparse kernel model using the incomplete Cholesky decomposition with error tolerance $\eta = 0.8$ and a χ^2 -kernel with $\chi^2 = 2.5 \times 10^{-3}$; (Bottom) Segment-label image indicating the clustering results with $k = 2$ found using the BLF on validation data.

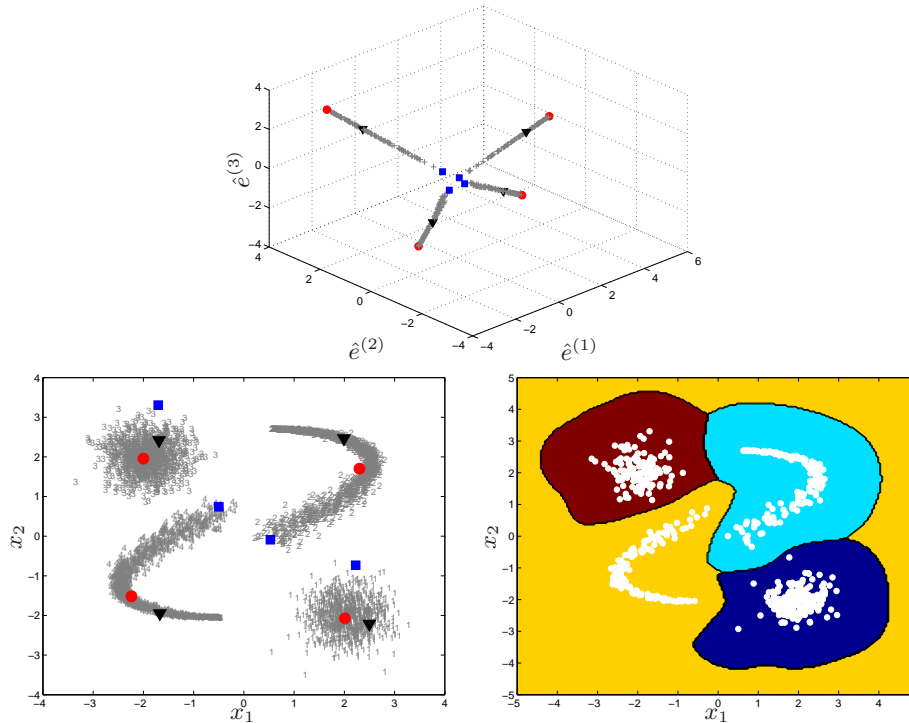


FIG 6. *Highly sparse representations in kernel spectral clustering, illustrated on a toy data set: (Top) the kernel based model is represented in terms of 12 support vectors, by inspecting the line structures in the space $(\hat{e}^{(1)}, \hat{e}^{(2)}, \hat{e}^{(3)})$; (Bottom) clustering results obtained from the predictive model. The different colors indicate the estimated regions for the four obtained clusters, obtained by making out-of-sample extensions.*

tures that represent the clusters in the projections space. Since the estimated cluster membership depends on the orthant of the projected data, points that are far away from the origin are more certain to belong to the corresponding cluster. The tips of the lines can therefore serve as prototypes of the clusters. As shown in [6], the pivots can be chosen by selecting both the endpoints of the lines and the median point. This leads to highly sparse kernel-based models where the predictive model is based upon $3k$ data points with k denoting the number of clusters. In Figure 6 a toy example with 4,000 data is shown to illustrate the method. The tuning parameter σ of the Gaussian kernel is selected using the BLF criterion evaluated on a validation set of 800 points, 600 data points were used for training.

7. Dimensionality reduction and data visualization

While traditionally methods as principal component analysis, multi-dimensional scaling and self-organizing maps [44, 46] are frequently applied for dimensionality reduction and data visualization, in recent years there has been many

interest in exploring new avenues. More recent approaches include e.g. locally linear embedding, Hessian locally linear embedding, Laplacian eigenmaps, diffusion maps and others [9, 18, 66]. For many of these approaches which are commonly known under the umbrella of kernel eigenmap methods and manifold learning, the solution is characterized by an eigenvalue problem. However, most methods require setting regularization and/or tuning constants for which it is often unclear how to select them [10]. This can easily lead to discovering fake structures when projecting high dimensional data to two dimensional or three dimensional coordinates. Most of the proposed techniques are also only formulated on the training data. The issue of making out-of-sample extensions of the method is then unclear or one has to rely on approximate techniques for this purpose [11]. Another relevant issue is also the computational complexity of the scheme. While convex optimization methods with semi-definite programs [87] have been studied, these have the drawback that the method does not scale well in terms of the number of training data.

In [80] a method of kernel maps with a reference point has been proposed. This reference point converts the eigenvalue problem into solving a linear system, which is desirable from a computational complexity point of view [73]. The formulation makes use of an LS-SVM core model for mapping the input data to the unknown coordinates to the low dimensional space. It takes a modified form of locally linear embedding as an additional regularization term. The method enables to make out-of-sample extensions exactly. The determination of all regularization and tuning constants has been successfully performed by cross-validation approaches [80].

The primal problem for realizing a dimensionality reduction $\mathbb{R}^d \rightarrow \mathbb{R}^p : x \mapsto z$ with coordinates z in a $p = 2$ dimensional space (a 3D projection goes similarly) is formulated as follows

$$\begin{aligned}
 \min_{z, w_1, w_2, b_1, b_2, e_{i,1}, e_{i,2}} \quad & \frac{1}{2} (z - P_D z)^T (z - P_D z) + \frac{\nu}{2} (w_1^T w_1 + w_2^T w_2) + \\
 & \frac{\eta}{2} \sum_{i=1}^N (e_{i,1}^2 + e_{i,2}^2) \\
 \text{subject to} \quad & c_{1,1}^T z = q_1 + e_{1,1} \\
 & c_{1,2}^T z = q_2 + e_{1,2} \\
 & c_{i,1}^T z = w_1^T \varphi_1(x_i) + b_1 + e_{i,1}, \quad i = 2, \dots, N \\
 & c_{i,2}^T z = w_2^T \varphi_2(x_i) + b_2 + e_{i,2}, \quad i = 2, \dots, N.
 \end{aligned} \tag{7.1}$$

The non-zero reference point is denoted by $q = [q_1; q_2] \in \mathbb{R}_0^2$ and is chosen by the user. It approximately fixes the z coordinates of the first data point x_1 . This point x_1 is sacrificed in the visualization. The first feature map $\varphi_1(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{n_{h1}}$ is used in mapping the x data to the first component of the z coordinates. A second feature map $\varphi_2(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{n_{h2}}$ is used for mapping to the second component of the z coordinates. The $c_{i,1}, c_{i,2}$ vectors consist of 0 and 1 elements to specify the projections for each of the data points, where $z = [z_1; z_2; \dots; z_N] \in \mathbb{R}^{pN}$. The additional regularization term equals $(z -$

$P_D z)^T(z - P_D z) = \sum_{i=1}^N \|z_i - \sum_{j=1}^N s_{ij} D z_j\|_2^2$ with D a diagonal matrix and $s_{ij} = \exp(-\|x_i - x_j\|_2^2/\sigma^2)$ which encourages that input data that are close in x coordinates will also be close to each other in their z coordinates [80]. The mapping to the z coordinates admit errors, which are characterized by $e_{i,1}, e_{i,2}$ for the training data. The model also contains bias terms b_1, b_2 for optimal centering. Finally, η, ν are positive regularization constants.

Note that in this estimation problem one jointly optimizes over the unknown mappings, the training data errors and the coordinates z . The unique solution to this problem first involves solving the linear system

$$\left[\begin{array}{c|c|c} U & -V_1 M_1^{-1} \mathbf{1} & -V_2 M_2^{-1} \mathbf{1} \\ \hline -1^T M_1^{-1} V_1^T & 1^T M_1^{-1} \mathbf{1} & 0 \\ \hline -1^T M_2^{-1} V_2^T & 0 & 1^T M_2^{-1} \mathbf{1} \end{array} \right] \begin{bmatrix} z \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \eta(q_1 c_{1,1} + q_2 c_{1,2}) \\ 0 \\ 0 \end{bmatrix} \quad (7.2)$$

with $U = (I - P_D)^T(I - P_D) - \gamma I + V_1 M_1^{-1} V_1^T + V_2 M_2^{-1} V_2^T + \eta c_{1,1} c_{1,1}^T + \eta c_{1,2} c_{1,2}^T$, $M_1 = \frac{1}{\nu} \Omega_1 + \frac{1}{\eta} I$, $M_2 = \frac{1}{\nu} \Omega_2 + \frac{1}{\eta} I$, $V_1 = [c_{2,1} \dots c_{N,1}]$, $V_2 = [c_{2,2} \dots c_{N,2}]$ and kernel matrices $\Omega_1, \Omega_2 \in \mathbb{R}^{(N-1) \times (N-1)}$ where $\Omega_{1,ij} = K_1(x_i, x_j) = \varphi_1(x_i)^T \varphi_1(x_j)$, $\Omega_{2,ij} = K_2(x_i, x_j) = \varphi_2(x_i)^T \varphi_2(x_j)$ with positive definite kernel functions $K_1(\cdot, \cdot), K_2(\cdot, \cdot)$.

The solution to (7.2) is finally used to find the Lagrange multipliers to the last two set of constraints in (7.1). One solves the dual problem in $\alpha_1, \alpha_2 \in \mathbb{R}^{N-1}$

$$\begin{aligned} M_1 \alpha_1 &= V_1^T z - b_1 \mathbf{1}_{N-1} \\ M_2 \alpha_2 &= V_2^T z - b_2 \mathbf{1}_{N-1} \end{aligned} \quad (7.3)$$

which are linear systems with a unique solution. In a dimensionality reduction to a 2-dimensional space for any point x_* , the estimated coordinates used for visualization are $\hat{z}_* = [\hat{z}_{*,1}; \hat{z}_{*,2}]$. These are delivered by the kernel-based presentations of the model at the dual level:

$$\begin{array}{ccc} & (P) : & \hat{z}_{*,1} = w_1^T \varphi_1(x_*) + b_1 \\ & & \hat{z}_{*,2} = w_2^T \varphi_2(x_*) + b_2 \\ \mathcal{M} & \begin{array}{c} \nearrow \\ \searrow \end{array} & \downarrow \\ & (D) : & \hat{z}_{*,1} = \frac{1}{\nu} \sum_{i=2}^N \alpha_{i,1} K_1(x_*, x_i) + b_1 \\ & & \hat{z}_{*,2} = \frac{1}{\nu} \sum_{i=2}^N \alpha_{i,2} K_2(x_*, x_i) + b_2. \end{array} \quad (7.4)$$

An illustration of the method on 3D visualization of microarray data is given in Figure 7 (see [80] for details).

8. Kernel CCA and ICA

A link between correlation in the feature space and independence in the input space was first discussed in [8]. If two random variables are uncorrelated in the feature space induced by a universal kernel, then these variables are independent

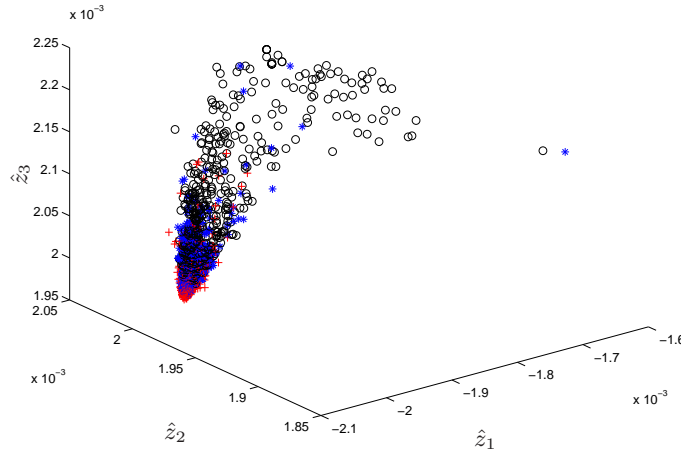


FIG 7. 3D visualization of the Alon colon cancer microarray data set using kernel maps with a reference points [80]: (blue) training genes; (black) validation genes; (red) test genes. An advantage of this approach is that one can validate or cross-validate the underlying model for visualization.

in the input space. Several contrast functions for kernel-based independent component analysis (ICA) have been proposed [1, 8, 39]. For the case of more than two variables, the contrast functions are characterizing pairwise independence.

Regularization is a critical aspect in kernel canonical correlation analysis (kernel CCA). It is well known that un-regularized kernel CCA yields too non-informative correlation estimates: ill-conditioning takes place since the kernel matrices can be singular or near-singular. Typical kernel CCA algorithms start from an un-regularized scheme and adding regularization afterwards in an ad-hoc manner. The scheme proposed in [1] corresponds to a multivariate version of the LS-SVM formulation to kernel CCA. One of the advantages of this method lies in the fact that regularization is incorporated naturally into the primal problem leading to a better conditioned generalized eigenvalue problem in the dual.

8.1. Multivariate kernel CCA

Given a number of m variables (called sources), the primal problem can be written as [1]

$$\begin{aligned} \min_{w^{(l)}, e^{(l)}} \quad & \frac{1}{2} \sum_{l=1}^m w^{(l)T} w^{(l)} + \frac{1}{2} \sum_{l=1}^m \nu_l e^{(l)T} e^{(l)} - \frac{\gamma}{2} \sum_{l=1}^m \sum_{k \neq l}^m e^{(l)T} e^{(k)} \\ \text{subject to} \quad & e^{(l)} = \Phi_{N \times n_{h_l}}^{(l)} w^{(l)}, \quad l = 1, \dots, m. \end{aligned} \tag{8.1}$$

The objective function can be interpreted as an extension to the expression for two sources $\|e^{(1)} - e^{(2)}\|_2^2$ towards $\sum_{l=1}^m \sum_{k \neq l}^m \|e^{(l)} - e^{(k)}\|_2^2$ for m sources

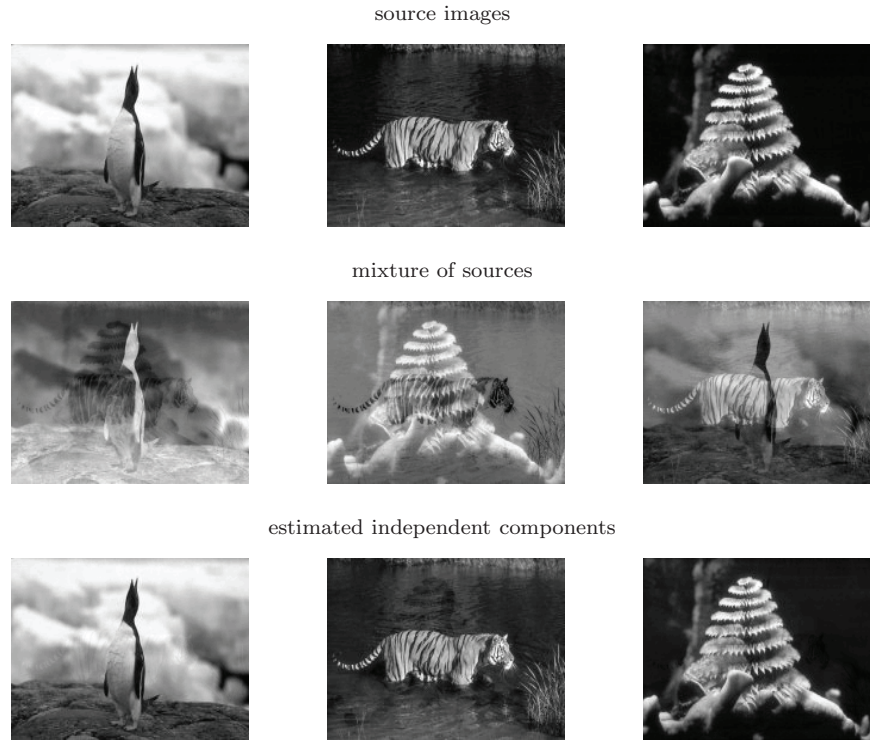


FIG 8. Image demixing using the KRC with parameters $\sigma^2 = 0.37, \nu = 0.68$ found using model selection on validation data. (Top) Source images; (Center) Mixtures of the sources using a random mixing matrix; (Bottom) Estimated independent components.

is $mN \times mN$. This can be problematic for large-scale problems. Therefore, approximation techniques via the incomplete Cholesky decomposition have been proposed in order to reduce the computational burden of computing the KRC. Figure 8 shows an image demixing experiment using the KRC with a Gaussian kernel and tuned parameters using validation data. Kernel-based measures for independence have been shown to perform better than other ICA algorithms with respect to near-Gaussian sources, increasing number of independent components and robustness to outliers [8]. The KRC outperforms well known methods for ICA such as Fast ICA and Jade and compares favorably in terms of Amari error and computation times to similar kernel-based measures for independence.

9. Conclusions

In this paper we have discussed problems in supervised and unsupervised learning which can be conceived in terms of primal and dual model representations, respectively involving a high dimensional feature map and positive definite kernel functions. It can be viewed as a methodology for kernel-based modelling

which is complementary to functional analysis approaches in RKHS and probabilistic approaches with Gaussian processes. Least squares support vector machine formulations play a central role as core problems in regression, classification, principal component analysis, canonical correlation analysis, spectral clustering and others². Other related developments are in ranking problems and survival analysis [83]. The model representations provide both connections to parametric statistics (the primal world) and non-parametric statistics (the dual world).

The core models, which naturally embody regularization in the primal problem for model complexity control, can be systematically extended by adding additional sets of constraints and additional regularization mechanisms. From the conditions for optimality follow both the optimal model representations and the model estimates. A future perspective is that this also enables developing a new generation of software tools where the optimal model representations and solutions can be derived in a symbolic way. Also from a computational point of view, the primal and dual characterizations have led to new efficient algorithms such as fixed-size kernel models for very large data sets.

Acknowledgements

Research supported by Research Council K.U. Leuven: GOA AMBioRICS, GOA-MaNet, CoE EF/05/006, OT/03/12, PhD/postdoc & fellow grants; Flemish Government: FWO PhD/postdoc grants, FWO projects G.0499.04, G.0211.05, G.0226.06, G.0302.07; Research communities (ICCoS, ANMMM, MLDM); AWI: BIL/05/43, IWT: PhD Grants; Belgian Federal Science Policy Office: IUAP DYSCO. The authors are grateful to Grace Wahba for inviting to contribute this manuscript. An earlier version of this work has been presented by Johan Suykens at the *Workshop on Learning Theory and Approximation* (organizers: Kurt Jetter, Steve Smale, Ding-Xuan Zhou) Mathematisches Forschungsinstitut Oberwolfach Germany www.mfo.de July 2008 [43].

References

- [1] ALZATE C. and SUYKENS J.A.K. (2008). “A Regularized Kernel CCA Contrast Function for ICA”, *Neural Networks*, **21**(2–3), 170–181.
- [2] ALZATE C. and SUYKENS J.A.K. (2008). “Kernel Component Analysis using an Epsilon Insensitive Robust Loss Function”, *IEEE Transactions on Neural Networks*, **19**(9), 1583–1598.
- [3] ALZATE C. and SUYKENS J.A.K. (2008). “Sparse Kernel Models for Spectral Clustering using the Incomplete Cholesky Decomposition”, *IEEE World Congress on Computational Intelligence (WCCI-IJCNN 2008)*, Hong Kong, pp. 3555–3562.

²For software see e.g. www.esat.kuleuven.be/sista/lssvmlab/ and www.kernel-machines.org/

- [4] ALZATE C. and SUYKENS J.A.K. (2010). “Multiway Spectral Clustering with Out-of-Sample Extensions through Weighted Kernel PCA”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(2): 335–347
- [5] ALZATE C. and SUYKENS J.A.K. (2009). “A Regularized Formulation for Spectral Clustering with Pairwise Constraints”, *International Joint Conference on Neural Networks (IJCNN 2009)*, Atlanta, 141–148.
- [6] ALZATE C. and SUYKENS J.A.K. (2010). “Highly Sparse Kernel Spectral Clustering with Predictive Out-of-Sample Extensions”, *Proc. of the 18th European Symposium on Artificial Neural Networks (ESANN 2010)*, Bruges, Belgium, pp. 235–240.
- [7] ARONSZAJN N. (1950). “Theory of reproducing kernels”, *Trans. American Mathematical Soc.*, **68**, 337–404. [MR0051437](#)
- [8] BACH F.R. and JORDAN M.I. (2002). “Kernel independent component analysis”, *Journal of Machine Learning Research*, **3**, 1–48. [MR1966051](#)
- [9] BELKIN M. and NIYOGI P. (2003). “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation”, *Neural Computation*, **15**(6): 1373–1396.
- [10] BELKIN M., NIYOGI P. and SINDHWANI V. (2006). “Manifold Regularization: a Geometric Framework for Learning from Labeled and Unlabeled Examples”, *Journal of Machine Learning Research*, **7**: 2399–2434. [MR2274444](#)
- [11] BENGIO Y., PAIEMENT J.-F., VINCENT P., DELALLEAU O., LE ROUX N. and OUIMET M. (2004). “Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering,” *Advances in Neural Information Processing Systems*, 16, 2004.
- [12] BISSACCO A., CHIUSO A. and SOATTO S. (2007). “Classification and Recognition of Dynamical Models: The Role of Phase, Independent Components, Kernels and Optimal Transport”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**(11): 1958–1972.
- [13] BOUSQUET O. and ELISSEEFF A. (2002). “Stability and Generalization”, *Journal of Machine Learning Research*, **2**, 499–526. [MR1929416](#)
- [14] BOYD S. and VANDENBERGHE L. (2004). *Convex Optimization*, Cambridge University Press. [MR2061575](#)
- [15] BRADLEY P.S. and MANGASARIAN O.L. (1998). “Feature Selection via Concave Minimization and Support Vector Machines”, *Machine Learning Proceedings of the Fifteenth International Conference (ICML98)*, (Ed. J. Shavlik), Morgan Kaufmann, San Francisco, California, 82–90.
- [16] CHAPELLE O., SCHÖLKOPF B. and ZIEN A. (Eds.) (2006). *Semi-Supervised Learning*, MIT Press. [MR2441315](#)
- [17] CHUNG F.R.K. (1997). *Spectral Graph Theory*, CBMS Regional Conference Series in Mathematics, 92. [MR1421568](#)
- [18] COIFMAN R.R. and LAFON S. (2006). “Diffusion maps”, *Applied and Computational Harmonic Analysis*, **21**(1), 5–30. [MR2238665](#)
- [19] CORTES C. and VAPNIK V. (1995). “Support vector networks”, *Machine Learning*, **20**, 273–297.

- [20] CRESSIE N. (1993). *Statistics for Spatial Data*, John Wiley & Sons, New York. [MR1239641](#)
- [21] CRISTIANINI N. and SHAWE-TAYLOR J. (2000). *An Introduction to Support Vector Machines*, Cambridge University Press.
- [22] CUCKER F. and SMALE S. (2002). “On the mathematical foundations of learning theory”, *Bulletin of the AMS*, **39**, 1–49. [MR1864085](#)
- [23] CUCKER F. and ZHOU D.-X. (2007). *Learning Theory: an Approximation Theory Viewpoint*, Cambridge University Press. [MR2354721](#)
- [24] DAUBECHIES I., DEVORE R., FORNASIER M. and GUNTURK S. (2010). “Iteratively re-weighted least squares minimization for sparse recovery”, *Communications on Pure and Applied Mathematics*, **63**(1): 1–38.
- [25] DE BRABANTER K., PELCKMANS K., DE BRABANTER J., DEBRUYNE M., SUYKENS J.A.K., HUBERT M. and DE MOOR B. (2009) “Robustness of Kernel Based Regression: a Comparison of Iterative Weighting Schemes”, *Proc. of the 19th International Conference on Artificial Neural Networks (ICANN 2009)*, Limassol, Cyprus, pp. 100–110.
- [26] DE BRABANTER K., DE BRABANTER J., SUYKENS J.A.K. and DE MOOR B. (2010), “Optimized Fixed-Size Kernel Models for Large Data Sets”, *Computational Statistics & Data Analysis*, **54**(6): 1484–1504.
- [27] DEBRUYNE M., HUBERT M. and SUYKENS J.A.K. (2008). “Model selection for kernel regression using the influence function”, *Journal of Machine Learning Research*, **9**, 2377–2400. [MR2452631](#)
- [28] DEBRUYNE M., CHRISTMANN A., HUBERT M. and SUYKENS J.A.K. (2010). “Robustness of reweighted least squares kernel based regression”, *Journal of Multivariate Analysis*, **101**(2): 447–463. [MR2564353](#)
- [29] DONOHO D.L. and GRIMES C. (2003). “Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data”, *Proc. Natl. Acad. Sci. U.S.A.*, **100**(10): 5591–5596. [MR1981019](#)
- [30] ESPINOZA M., SUYKENS J.A.K. and DE MOOR B. (2006). “LS-SVM Regression with Autocorrelated Errors”, *Proc. of the 14th IFAC Symposium on System Identification (SYSID 2006)*, Newcastle, Australia, pp. 582–587.
- [31] ESPINOZA M., SUYKENS J.A.K., BELMANS R. and DE MOOR B. (2007). “Electric Load Forecasting using kernel based modeling for nonlinear system identification”, *IEEE Control Systems Magazine*, **27**(5), 43–57. [MR2350944](#)
- [32] FALCK T., PELCKMANS K., SUYKENS J.A.K. and DE MOOR B. (2009). “Identification of Wiener-Hammerstein Systems using LS-SVMs”, *Proceedings of the 15th IFAC Symposium on System Identification (SYSID 2009)*, Saint-Malo, France, pp. 820–825.
- [33] FUNG G. and MANGASARIAN O.L. (2001). “Proximal support vector machine classifiers”, *Proceedings KDD-2001*, 77–86.
- [34] FUNG G. and MANGASARIAN O.L. (2005). “Multicategory proximal support vector machine classifiers”, *Machine Learning*, **59**: 77–97.
- [35] GIROLAMI M. (2002). “Orthogonal series density estimation and the kernel eigenvalue problem”, *Neural Computation*, **14**(3), 669–688.

- [36] GOLUB G.H., HEATH M. and WAHBA G. (1979). “Generalized cross-validation as a method for choosing a good ridge regression parameter”, *Technometrics*, **21**, 215–223. [MR0533250](#)
- [37] GOETHALS I., PELCKMANS K., SUYKENS J.A.K. and DE MOOR B. (2005). “Identification of MIMO Hammerstein Models using Least Squares Support Vector Machines”, *Automatica*, **41**(7), 1263–1272. [MR2160126](#)
- [38] GOETHALS I., PELCKMANS K., SUYKENS J.A.K. and DE MOOR B. (2005). “Subspace Identification of Hammerstein Systems using Least Squares Support Vector Machines”, *IEEE Transactions on Automatic Control*, **50**(10), 1509–1519. [MR2171870](#)
- [39] GRETTON A., HERBRICH R., SMOLA A., BOUSQUET O. and SCHÖLKOPF B. (2005). “Kernel Methods for Measuring Independence”, *Journal of Machine Learning Research*, **6**, 2075–2129. [MR2249882](#)
- [40] HASTIE T. and TIBSHIRANI R. (1990). *Generalized Additive Models*, Chapman & Hall/CRC. [MR1082147](#)
- [41] HASTIE T., TIBSHIRANI R. and FRIEDMAN J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer series in statistics. [MR1851606](#)
- [42] HUBER P.J. (1981). *Robust Statistics*, Wiley, New York. [MR0606374](#)
- [43] JETTER K., SMALE S. and ZHOU D.-X. (Eds.) (2008). *Learning Theory and Approximation*, Oberwolfach Reports, **5**(3): 1655–1706. [MR2524072](#)
- [44] JOLLIFFE I.T. (1986). *Principal Component Analysis*, Springer Series in Statistics, Springer-Verlag. [MR0841268](#)
- [45] KIMELDORF G.S. and WAHBA G. (1971). “A correspondence between Bayesian estimation on stochastic processes and smoothing by splines”, *Ann. Math. Statist.*, **2**, 495–502. [MR0254999](#)
- [46] KOHONEN T. (1990). “The self-organizing map,” *Proceedings of the IEEE*, **78**(9): 1464–1480.
- [47] LUENBERGER D.G. (1969). *Optimization by vector space methods*, Wiley, New York. [MR0238472](#)
- [48] LUTS J., SUYKENS J.A.K. and VAN HUFFEL S. (2007). “Semi-supervised learning: avoiding zero label assumptions in kernel based classifiers”, Internal Report 07-122, ESAT-SISTA, K.U.Leuven (Leuven, Belgium).
- [49] LUTS J., OJEDA F., VAN DE PLAS R., DE MOOR B., VAN HUFFEL S. and SUYKENS J.A.K. (2010). “A tutorial on support vector machine-based methods for classification problems in chemometrics”, *Analytica Chimica Acta*, **66**(2): 129–145.
- [50] MACKAY D.J.C. (1998). “Introduction to Gaussian processes” in *Neural networks and machine learning* (Ed. C.M. Bishop), Springer NATO-ASI Series F: Computer and Systems Sciences, Vol.168, 133–165.
- [51] MANGASARIAN O.L. and WILD E.W. (2008). “Nonlinear Knowledge-Based Classification”, *IEEE Transactions on Neural Networks*, **19**, 1826–1832.
- [52] MEILA M. and SHI J. (2001). “A random walks view of spectral segmentation”, *Artificial Intelligence and Statistics* (AISTATS 2001).

- [53] MERCER J. (1909). “Functions of positive and negative type and their connection with the theory of integral equations”, *Philos. Trans. Roy. Soc. London*, **209**, 415–446.
- [54] NG A.Y., JORDAN M.I. and WEISS Y. (2002). “On spectral clustering: Analysis and an algorithm”, in T. Dietterich, S. Becker and Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems (NIPS)*, 14.
- [55] OJEDA F., SUYKENS J.A.K. and DE MOOR B. (2008). “Low rank updated LS-SVM classifiers for fast variable selection”, *Neural Networks*, **21**(2–3), 437–449.
- [56] PARZEN E. (1970). “Statistical inference on time series by RKHS methods”, Dep. Statist. Stanford Univ. Tech. Rep.14, Jan. [MR0275616](#)
- [57] PELCKMANS K., ESPINOZA M., DE BRABANTER J., SUYKENS J.A.K. and DE MOOR B. (2005). “Primal-Dual Monotone Kernel Regression”, *Neural Processing Letters*, **22**(2), 171–182.
- [58] PELCKMANS K., GOETHALS I., DE BRABANTER J., SUYKENS J.A.K. and DE MOOR B. (2005). “Componentwise Least Squares Support Vector Machines”, Chapter in *Support Vector Machines: Theory and Applications*, (Wang L., ed.), Springer, 2005, pp. 77–98.
- [59] PELCKMANS K., SUYKENS J.A.K. and DE MOOR B. (2005). “Building Sparse Representations and Structure Determination on LS-SVM Substrates”, *Neurocomputing*, **64**, 137–159.
- [60] PELCKMANS K., SUYKENS J.A.K. and DE MOOR B. (2006). “Additive Regularization Trade-off: Fusion of Training and Validation levels in Kernel Methods”, *Machine Learning*, **62**(3), 217–252.
- [61] PEREZ-CRUZ F., BOUSONO-CALZON C. and ARTES-RODRIGUEZ A. (2005). “Convergence of the IRWLS Procedure to the Support Vector Machine Solution”, *Neural Computation*, **17**(1), 7–18.
- [62] POGGIO T. and GIROSI F. (1990). “Networks for approximation and learning”, *Proceedings of the IEEE*, **78**(9), 1481–1497.
- [63] POGGIO T., RIFKIN R., MUKHERJEE S. and NIYOGI P. (2004). “General conditions for predictivity in learning theory”, *Nature*, **428**(6981), 419–422.
- [64] RASMUSSEN C.E. and WILLIAMS C.K.I. (2006). *Gaussian Processes for Machine Learning*, MIT Press. [MR2514435](#)
- [65] ROUSSEEUW P.J. and LEROY A. (1997). *Robust Regression and Outlier Detection*, John Wiley & Sons, New York. [MR0914792](#)
- [66] ROWEIS S. and SAUL L. (2000). “Nonlinear dimensionality reduction by locally linear embedding”, *Science*, **290** (5500), 2323–2326.
- [67] SAUNDERS C., GAMMERMAN A. and VOVK V. (1998). “Ridge regression learning algorithm in dual variables”, *Proc. of the 15th Int. Conf. on Machine Learning*, Madison-Wisconsin, 515–521.
- [68] SCHÖLKOPF B., SMOLA A. and MÜLLER K.-R. (1998). “Nonlinear component analysis as a kernel eigenvalue problem”, *Neural Computation*, **10**, 1299–1319.
- [69] SCHÖLKOPF B. and SMOLA A. (2002). *Learning with Kernels*, MIT Press, Cambridge, MA.

- [70] SCHÖLKOPF B., TSUDA K. and VERT J.P. (Eds.) (2004). *Kernel Methods in Computational Biology* 400, MIT Press.
- [71] SHAWE-TAYLOR J. and CRISTIANINI N. (2004). *Kernel Methods for Pattern Analysis*, Cambridge University Press.
- [72] SHI J. and MALIK J. (2000). “Normalized Cuts and Image Segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8), 888–905.
- [73] SMALE S. (1997), “Complexity theory and numerical analysis,” *Acta Numerica*, 523–551. [MR1489262](#)
- [74] SUYKENS J.A.K. and VANDEWALLE J. (1999). “Least squares support vector machine classifiers”, *Neural Processing Letters*, **9**(3), 293–300.
- [75] SUYKENS J.A.K. and VANDEWALLE J. (1999). “Multiclass Least Squares Support Vector Machines”, *Proc. of the International Joint Conference on Neural Networks (IJCNN'99)*, Washington DC, USA.
- [76] SUYKENS J.A.K., DE BRABANTER J., LUKAS L. and VANDEWALLE J. (2002). “Weighted least squares support vector machines: robustness and sparse approximation”, *Neurocomputing*, **48**(1–4), 85–105.
- [77] SUYKENS J.A.K., VAN GESTEL T., DE BRABANTER J., DE MOOR B. and VANDEWALLE J. (2002). *Least Squares Support Vector Machines*, World Scientific, Singapore.
- [78] SUYKENS J.A.K., HORVATH G., BASU S., MICCHELLI C. and VANDEWALLE J. (Eds.) (2003). *Advances in Learning Theory: Methods, Models and Applications*, vol. 190 NATO-ASI Series III: Computer and Systems Sciences, IOS Press.
- [79] SUYKENS J.A.K., VAN GESTEL T., VANDEWALLE J. and DE MOOR B. (2003). “A support vector machine formulation to PCA analysis and its kernel version”, *IEEE Transactions on Neural Networks*, **14**(2): 447–450.
- [80] SUYKENS J.A.K. (2008). “Data Visualization and Dimensionality Reduction using Kernel Maps with a Reference Point”, *IEEE Transactions on Neural Networks*, **19**(9), 1501–1517.
- [81] TIBSHIRANI R. (1996). “Regression shrinkage and selection via the lasso”, *J. Royal. Statist. Soc B.*, **58**(1), 267–288. [MR1379242](#)
- [82] TSUDA K., SHIN H.J. and SCHÖLKOPF B. (2005). “Fast protein classification with multiple networks”, *Bioinformatics (ECCB'05)*, **21**(Suppl.2): ii59–ii65.
- [83] VAN BELLE V., PELCKMANS K., SUYKENS J.A.K. and VAN HUFFEL S. (2010). “Additive survival least squares support vector machines”, *Statistics in Medicine*, **29**(2): 296–308.
- [84] VAN GESTEL T., SUYKENS J.A.K., LANCKRIET G., LAMBRECHTS A., DE MOOR B. and VANDEWALLE J. (2002). “Multiclass LS-SVMs: Moderated outputs and coding-decoding schemes”, *Neural Processing Letters*, **15**(1): 45–48.
- [85] VAPNIK V. (1998). *Statistical Learning Theory*, Wiley, New York. [MR1641250](#)
- [86] WAHBA G. (1990). *Spline Models for Observational Data*, Series in Applied Mathematics, 59, SIAM, Philadelphia. [MR1045442](#)

- [87] WEINBERGER K.Q. and SAUL L.K. (2004). “Unsupervised learning of image manifolds by semidefinite programming,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-04)*, Washington D.C.
- [88] WILLIAMS C.K.I. and SEEGER M. (2001). “Using the Nyström method to speed up kernel machines”, In T.K. Leen, T.G. Dietterich, and V. Tresp (Eds.), *Advances in neural information processing systems*, **13**, 682–688.