Primal and dual model representations in kernel-based learning

Paul Hofmarcher

Institute for Statistics and Mathematics
WU Vienna, University of Economics and Business

Content I

- We want to discuss the role of primal and Lagrange dual model representations following Suykens and Alzate (2010). Hereby we discuss:
  - kernels,
  - Reproducing Kernel Hilbert Spaces (RKHS),
  - Support Vector Machines (SVM),
  - Least Square SVM,
  - Ridge Regression,
  - and Kernel PCA.

Kernels

- Let $\mathcal{X}$ be a (non-empty) set. A mapping

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \ \ (x, x') \to k(x, x'), \qquad (1)$$

  is called a kernel if k is symmetric, i.e., $k(x, x') = k(x', x)$
- A kernel k is *positive definite*, if its Gram Matrix $K_{i,j} := k(x_i, x_j)$ is positive definite $\forall x$.
- The Cauchy-Schwarz inequality holds for p.d. kernels.
- Define a *reproducing kernel map:*

$$\Phi : x \to k(\cdot, x), \qquad (2)$$

  i.e., to each point x in the original space we associate a function $k(\cdot, x)$.
- Another way to characterize p.d. kernels on a compact set is via Mercer's Theorem.

Kernels II

- linear kernel: $k(x, x') = \langle x, x' \rangle$.
- Gaussian kernel: Each point x maps to a Gaussian distribution centered at that point.

$$k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}} \tag{3}$$

- Linear combinations of kernels are kernels.
- Construct a vector space containing all linear combinations of functions $k(\cdot, x)$:

$$f(\cdot) = \sum_i \alpha_i k(\cdot, x_i). \tag{4}$$

Reproducing Kernel Hilbert Spaces (RKHS)

- $k(\cdot, \cdot)$ is a reproducing kernel of a Hilbert space $\mathcal{H}$ if $\forall f \in \mathcal{H}$, $f(x) = \langle k(x, \cdot), f(\cdot)\rangle$
- A RKHS is a Hilbert space H with a reproducing kernel whose span is dense in H.
- Construct a vector space via

$$\{f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i) : n \in \mathbb{N}, x_i \in \mathcal{X}, \alpha_i \in \mathbb{R}\} \tag{5}$$

  and define $\langle f, g\rangle = \sum_{i,j} \alpha_i \beta_j k(x_j, x_i)$
- Note that $\langle f, k(\cdot, x)\rangle = \sum_i \alpha_i k(x, u_i) = f(x)$, i.e., k has the reproducing property.

RKHS II

Given a training data set $\{(x_i, y_i)_{i=1}^N\}$ of N training data, with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. Find a function f that minimizes

$$\min_f \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i)) + \nu \|f\|_K^2 \quad (6)$$

- $L(\cdot, \cdot)$ denotes the chosen loss function and $\|f\|_K$ the norm in the RKHS $\mathcal{H}$ defined by kernel K.
- ν cost parameter.

RKHS III

- f belongs to $\mathcal{H}$.
- For any convex loss function L the solution of 6 has the form (*representer Theorem*, Kimeldorf & Wahba, 1971)

$$f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i) \tag{7}$$

- The model has the reproducing property

$$f(x) = \langle f, K(x, \cdot) \rangle_K \tag{8}$$

Loss functions

- Plugging-In different loss functions one obtains:
  - regularization network:

$$L(y, f(x)) = (y - f(x))^2 \tag{9}$$

  - support vector regression:

$$L(y, f(x)) = |y - f(x)|_\epsilon \tag{10}$$

  - SVM classification:

$$L(y, f(x)) = [1 - yf(x)]_+ \tag{11}$$

Loss functions II

- $|\cdot|_\epsilon$ denotes the ϵ-insensitive loss function with $\epsilon \geq 0$. For $|y - f(x_i)| \leq \epsilon$ it is set to 0. Otherwise it is $|y - f(x_i)| - \epsilon$.
- This result in a sparse solution, meaning that many α_i are 0. For $\epsilon = 0$ it corresponds to an L_1 estimator.
- Regularization constant ν controls the bias-variance trade-off. For ν to small, might result in overfitting the data, while ν to large might give inflexible models.
- Usually ν is estimated via cross validation.

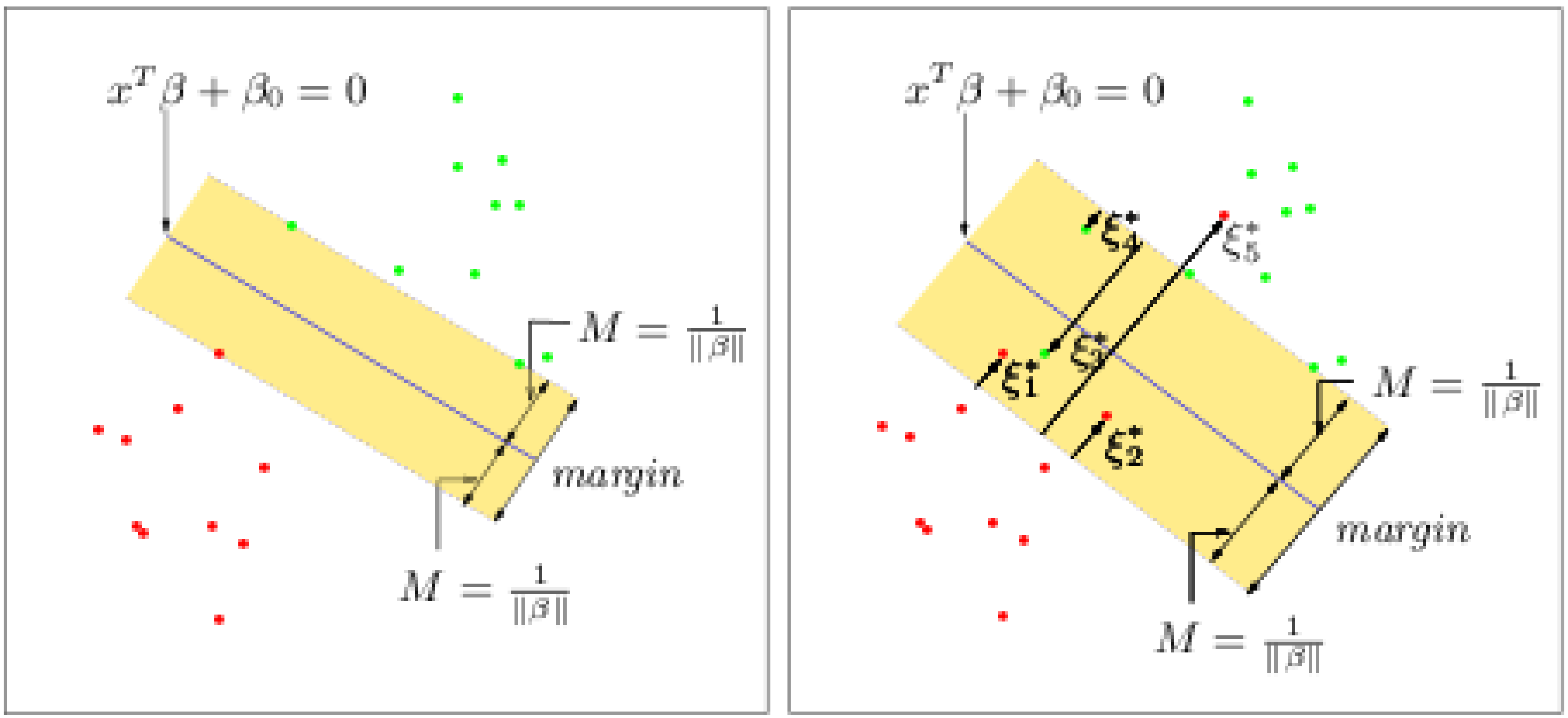
Support Vector Classifiers

- a feature map $\phi(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{n_d}$ maps the data from input space to higher dimensional space.
- The classifier model corresponds to

$$\widehat{y} = sign[\sum_{i=1}^{n_h} w_j \phi_j(x) + b]. \tag{12}$$

- feature map is usually not explicitly defined at the beginning, but implicitly through a p.d. kernel.

Support Vector Classifiers II



Support Vector Classifiers, Primal vs. dual

- The Primal problem is stated as:

$$\min_{w,\beta,\xi} 0.5 w^T w + c \sum_i \xi_i, \tag{13}$$

$$s.t.\ y_i(\phi(x_i)^T w + \beta) \geq 1 - \xi_i \quad , \xi_i \geq 0 \forall i \quad ,$$

- The dual Problem:

$$\max_{\alpha} \quad -0.5 \sum_{i,j=1}^{N} y_i y_j K(x_i, x_j) \alpha_i \alpha_j + \sum_{j=1}^{N} \alpha_j, \tag{14}$$

$$s.t.\ \sum_{i=1}^{N} \alpha_i y_i = 0 \quad 0 \leq \alpha_i \leq c_i \forall i \quad ,$$

Support Vector Classifiers, Primal vs. dual II

- where we have to use a p.d. kernel, satisfying

$$K(x,z) = \langle \phi(x)^T \phi(z) \rangle = \sum_{j=1}^{n_d} \phi_j(x)\phi_j(z) \tag{15}$$

  for any pair $x, z \in \mathbb{R}^d$ (*kernel trick*).
- From optimality conditions we get $w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i)$, such that

$$\hat{y} = sign[\sum_{i \in \mathbb{SV}} \alpha_i y_i K(x, x_i) + b] \tag{16}$$

- where $\mathbb{SV}$ denotes the set of support vectors.

Support Vector Classifiers

- One can read equation 15 in two ways: left to right and right to left:
- left to right: One fixes the choice of the p.d. kernel. This guarantees the existence of an underlying feature map. So one does not know an explicit expression of the feature map.
- From right to left: One may also explicitly define a feature map and obtains the kernel via $K(x,z) = \phi(x)^T\phi(z)$.

LS-SVM core models

- Least square support vector machine works with equality constraints instead of inequality constraints and an L_2 loss function.
- characterizing the conditions for optimality becomes simpler.
- possible to extend methodology to a wide range of problems.
- it captures the simple essence while still providing high performant models.

LS-SVM

- LS-SVM is formulated as:

$$\min_{w,b,e_i} \; 0.5 w^t w + \gamma 0.5 \sum_{i=1}^{N} e_i^2, \tag{17}$$

$$s.t. \; y_i(\phi(x_i)^T w + b) = 1 - e_i \;\; \forall i \;\; ,$$

- Rewriting this into a Lagrangian problem with coefficients α_i and eliminating e, w one gets a square linear system.
- The classifier in the dual space has the form

$$\widehat{y} = sign[\sum_{i \in N} \alpha_i y_i K(x, x_i) + b] \tag{18}$$

LS-SVM II

- From optimality conditions yield:

$$w = \sum_{k=1}^{N} \alpha_k y_k \phi(x_k) \tag{19}$$

$$\sum_{k=1}^{N} \alpha_k y_k = 0 \tag{20}$$

$$\alpha_k = \gamma e_k \tag{21}$$

$$y_k[w^t \phi(x_k) + b] - 1 + e_k = 0 \tag{22}$$

LS-SVM III

- Elimination of w and e results in:

$$\left[\begin{array}{c|c} 0 & y^T \\ \hline y & \Omega + I/\gamma \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 1_N \end{array}\right] \tag{23}$$

- Now we apply the kernel rick to the matrix $\Omega := Z^T Z$, with

$$\Omega_{kl} = y_k y_l \phi(x_k)^T \phi(x_l) = y_k y_l K(x_k, x_l)$$

- For any point $x^* \in \mathcal{R}^d$ we get

$$(D:)\ \ \widehat{y^*} = sign[\sum_{i=1}^{N} \alpha_i y_i K(x^*, x_i) + b] \tag{24}$$

Ridge Regression

- In a similar way one can perform ridge regression in the feature space with additional bias term b

$$\min_{w,b,e_i} \; 0.5w^t w + \gamma 0.5 \sum_{i=1}^{N} e_i^2, \tag{25}$$

$$s.t. \; y_i = \phi(x_i)^T w + b + e_i \;\; \forall i \quad ,$$

- The corresponding primal and dual model representations are:
- Primal:

$$\widehat{y} = w^T \phi(x) + b \tag{26}$$

- Dual:

$$\widehat{y} = \sum_{i \in N} \alpha_i K(x, x_i) + b \tag{27}$$

Kernel Principal Component Analysis I

- Perform PCA for covariance matrix

$$\bar{C} = \frac{1}{l} \sum_{j=1}^{l} \phi(x_j)\phi(x_j)^T \tag{28}$$

- find eigenvalues $\lambda \geq 0$ and eigenvectors $V \in \mathcal{H}$ satisfying $\lambda V = \bar{C} V$
- We know that all solutions lie in the span of $\phi(x_1), \ldots, \phi(x_l)$.
- There exist coefficients $\alpha_1, \ldots, \alpha_l$ such that

$$V = \sum_{i=1}^{l} \alpha_i \phi(x_i), \tag{29}$$

KPCA II

- Kernel PCA can be obtained as the dual problem of the following LS-SVM formulation:

$$\max_{w,b,e_i} \; -0.5w^t w + \gamma 0.5 \sum_{i=1}^{N} e_i^2, \tag{30}$$

$$s.t.\ e_i = \phi(x_i)^T w + b \ \ \forall i \ \ ,$$

- The corresponding primal and dual model representations are:
- Primal:

$$\widehat{e} = w^T \phi(x) + b \tag{31}$$

- Dual:

$$\widehat{e} = \sum_{i \in N} \alpha_i K(x, x_i) + b \tag{32}$$

KPCA III

- The problem in the Lagrangian multipliers α_i related to the constraints is then given by:

$$\Omega\alpha = \lambda\alpha \;\; \Omega_{ij} = (\phi(x_i) - \widehat{\mu_\phi})(\phi(x_j) - \widehat{\mu_\phi}) \tag{33}$$

- Equation 30 describes the pool of of all candidate components. EV which are a solution of 33 lead to a value zero for $-0.5w^t w + \gamma 0.5 \sum_{i=1}^{N} e_i^2$.
- Relevant Components: Component corresponding to λ_{max} results in maximizing $\gamma 0.5 \sum_{i=1}^{N} e_i^2$.

Primal or Dual?

Solving the Primal or Dual Problem?

- In case the feature map is finite dimensional and explicitly known one has the choice between solving the primal or dual problem.
- E.g., for the Gaussian kernel one can only solve the dual problem.
- Consider linear regression $\hat{y} = w^T x + b$ with $w \in \mathbb{R}^d$
  - dual representation: $\hat{y} = \sum_{i=1}^{N} \alpha_i x_i^T x_i + b$ with $\alpha \in \mathbb{R}^N$.
  - d small and N large: solving the primal problem in $w \in \mathbb{R}^d$ is more convenient.
  - d large and N small: solving the dual problem in $\alpha \in \mathbb{R}^N$ is more convenient.

Kernel methods in R

R packages which allow to deal with kernel methods are e.g., e1071, klaR and kernlab.

- e1071 offers an interface to libsvm, a very efficient SVM implementation.
- klaR includes an interface to SVMlight.
- most of the libsvm and klaR SVM code is in C++.
- kernlab uses S4 class system.
- kernlab aims to allow the user to switch between kernels on an existing algorithm and even to create and use own kernel functions.

kernlab

- We will use economic growth data to illustrate the methods described above.
- kernlab includes 7 different kernels (`vanilladot`, `rbfdot`, `polydot`, `tanhdot`, `besseldot`, `laplacedot`, `anovadot`).
- `kernelMatirx` computes $k(x, x')$, i.e., it computes K where $K_{ij} = k(x_i, x_j)$.

kernlab **II**

- SVM can be estimated via `ksvm`
- LS-SVM via `lssvm`
- kernel PCA via `kpca`
- is there a function for kernel ridge regression?

References:

- Johan A.K. Suykens, Carlos Alzate (2010); *Primal and dual model representations in kernel-based learning*. Statistics Surveys, vol.4, 148-183