Support Vector Machines and Kernel Methods

Daniel Khashabi

Fall 2012
Last Update: September 26, 2016

1 Introduction

In *Support Vector Machines* the goal is to find a separator between data which has the largest margin, as it can be seen in the Figure 1. Note that, in the *Perceptron* algorithms, the goal is just to find a separator for the data, although such separator might not be a *large-margin* separator.

In the following sections, we will start with the basic formulation of the SVM, and continue to the more advanced representations of the model.

2 Simple classification using Support Vector Machines

Suppose we choose a group of data points, which could reasonably separate information regions. These data points that lie close to separation regions, selected among all the input data, are commonly called "*support vectors*". Assume that we have group of data $\{\mathbf{x}_i, y_i\}$that could be separated by a hyperplane. Thus we can write the following statements about the separating hyperplanes,

$$\begin{cases} \boldsymbol{\beta}.\mathbf{x}_i + \beta_0 \geq +1, & \text{if } y_i = +1 \\ \boldsymbol{\beta}.\mathbf{x}_i + \beta_0 \leq -1, & \text{if } y_i = -1. \end{cases}$$

Equivalently we could write the above separating equations as follows:

$$y_i.\left(\boldsymbol{\beta}.\mathbf{x}_i + \beta_0\right) \geq 1, \ \forall i.$$

In the above formulation, β_0 is the bias weight. To continue with simpler formulation, we do the following reformulations:

$$\begin{cases} \boldsymbol{\beta} \leftarrow [\boldsymbol{\beta}, \quad \beta_0] \\ \mathbf{x}_i \leftarrow [\mathbf{x}_i, \quad +1] \end{cases}$$

Now we will and the problem becomes,

$$y_i \boldsymbol{\beta}.\mathbf{x}_i \geq 1, \ \forall i.$$

In this formulation, 1 is the size of the margin. Instead of fixing this value, we can optimize over it:

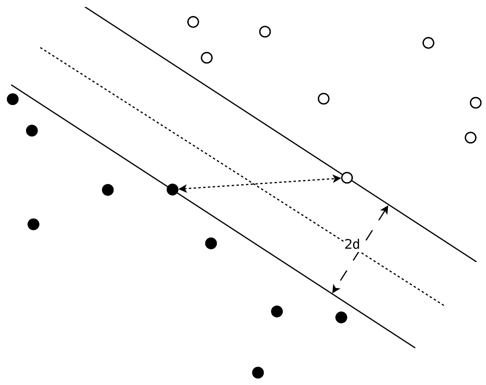

Figure 1: Max-margin scheme for support vector classification.

$$\begin{cases} \max_{\gamma,\boldsymbol{\beta}} \gamma \\ \frac{y_i \boldsymbol{\beta}.\mathbf{x}_i}{\|\boldsymbol{\beta}\|} \geq \gamma, \ \forall i. \end{cases}$$

Since only $\boldsymbol{\beta}$ is important, and to reduce the number of the parameters, we can set $\gamma = \frac{1}{\|\boldsymbol{\beta}\|}$. Then the previous program could be written in the following form:

$$\begin{cases} \min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{\beta}\|^2 \\ y_i \boldsymbol{\beta}.\mathbf{x}_i \geq 1, \ \forall i. \end{cases} \tag{1}$$

This is the optimality criterion for separation of two hyperplanes. The minimization criterion, $\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{\beta}\|^2$, maximizes the coefficient vector, $\boldsymbol{\beta}$, while preserving the separation constraints.

One other interpretation for the above optimization criterion is as it follows. Consider the Figure 1, and the two data points on the margin of each area, $(\mathbf{x}_1, y_1 = +1)$ and $(\mathbf{x}_2, y_2 = -1)$. For these two data points we have,

$$\begin{cases} \boldsymbol{\beta}.\mathbf{x}_1 = +1, & \text{if } y_1 = +1 \\ \boldsymbol{\beta}.\mathbf{x}_2 = -1, & \text{if } y_2 = -1. \end{cases} \Rightarrow \boldsymbol{\beta}.\left(\mathbf{x}_1 - \mathbf{x}_2\right) = 2 \Rightarrow \|\mathbf{x}_1 - \mathbf{x}_2\| = \frac{2}{\boldsymbol{\beta}}$$

Here is a nice interpretation: In order to maximize the separating margin $\|\mathbf{x}_1 - \mathbf{x}_2\|$ between data points, it suffices to minimize $\|\boldsymbol{\beta}\|$ or minimize $\|\boldsymbol{\beta}\|^2$ [1].

The formulation in the Equation 1 is called *hard-margin* SVM, and it is the *primal* form. The objective is a quadratic function, and linear constraints, and therefore we have a *quadratic* optimization (and hence a convex problem). Would it be enough to use a standard quadratic solver to solve the SVM problem? Indeed one can use quadratic solver for SVM, but many early studies showed that, since SVM problem is a special case of general quadratic programs, adhoc solutions to SVM, usually give better and faster solutions to SVM, than general solvers. Here we will derive multiple direct solutions to the problem.

[1]Using $\|\boldsymbol{\beta}\|^2$ is just for simplicity and ease of notation

One can optimize the constrained program in the Equation 1 using Lagrange multilpiers. Now first form the *Lagrangian*, with Lagrange multilpiers $\{\lambda_i \geq 0\}$ added, as following:

$$\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\lambda}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_i \lambda_i \left(y_i \boldsymbol{\beta}.\mathbf{x}_i - 1\right). \tag{2}$$

Note that we wish to find saddle point of $\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\lambda})$:

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\lambda})$$

The *complementary slackness* condition says that essentially:

$$\lambda_i \left(y_i \boldsymbol{\beta}.\mathbf{x}_i - 1\right) = 0, \quad \forall i$$

In other words:

- If $y_i \boldsymbol{\beta}.\mathbf{x}_i > 0$, then essentially $\lambda_i = 0$.
- If $\lambda_i > 0$, then essentially $y_i \boldsymbol{\beta}.\mathbf{x}_i = 1$. Such points are called *support vectors*.

The above Lagrange function satisfies the necessary conditions,

$$\begin{cases} \nabla_{\boldsymbol{\beta}} \mathcal{L} = 0 & \Rightarrow \boldsymbol{\beta}^* = \sum_i \lambda_i y_i \mathbf{x}_i, \\ & \lambda_i \geq 0, \\ & \lambda_i y_i \boldsymbol{\beta}.\mathbf{x}_i - 1 \geq 0, \\ & \lambda_i \left\{\lambda_i y_i \boldsymbol{\beta}.\mathbf{x}_i - 1\right\} = 0. \end{cases}$$

By replacing $\boldsymbol{\beta}^*$ into the Lagrangian, one can find the dual problem which essentially has the same solution as the main problem,

$$\mathcal{L}(\boldsymbol{\beta}^*, \boldsymbol{\lambda}) = \sum_i \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j \left(\mathbf{x}_i.\mathbf{x}_j\right)$$

The full dual program is the following:

$$\begin{cases} \max_{\lambda_i} \sum_i \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j \left(\mathbf{x}_i.\mathbf{x}_j\right) \\ \text{subject to: } \sum_i \lambda_i y_i = 0, \ \lambda_i \geq 0 \end{cases} \tag{3}$$

Now it suffices to solve the dual problem for $\lambda_i > 0$, and find the coefficient vector $\boldsymbol{\beta}^* = \sum_i \lambda_i y_i \mathbf{x}_i$. For prediction on new points, we can now do *sign* $(\boldsymbol{\beta}^*.\mathbf{x})$.

Now let's assume that we want to do classification on a non-linear space. Something important to notice is that, the input variables enter to the optimization via inner product. We can use this fact, and project the variables $\mathbf{x}$ into another space, which has inner product. More specifically we define the function:

$$\Phi : \mathcal{X} \rightarrow \mathcal{F},$$

Using this function, we replace the variable "$\mathbf{x}$", with the new high-dimension variable "$\Phi(\mathbf{x})$". Now we define notion of *kernel* which appears in different occasions, and gets a more practical interpretations. We define a *kernel* as $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$. To get more intuition into *kernel*s and convince ourselves about usefulness of this definition let's go back and see the formulations based on new feature space, $\Phi(\mathbf{x})$. The dual formulation and the prediction formulations are,

$$\begin{cases} \max_{\lambda_i} \sum_i \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i\lambda_j y_i y_j \overbrace{\left(\Phi(\mathbf{x}_i) . \Phi(\mathbf{x}_j)\right)}^{k(\mathbf{x}_i,\mathbf{x}_j)}, \\ \text{subject to: } \sum_i \lambda_i y_i = 0,\ \lambda_i \geq 0, \end{cases}$$

The above formulation of problem, shows how in new formulation the inner product of variables appear in conjunction with each other which we call it *kernel.* Now we can interpret the prediction, as linear combination of kernels, defined by a subset of input data:

$$\begin{aligned} f(\mathbf{x}) &= sign\left(\boldsymbol{\beta}^*.\Phi(\mathbf{x})\right) \\ &= sign\left(\sum_i \lambda_i y_i \underbrace{\left(\Phi(\mathbf{x}_i).\Phi(\mathbf{x})\right)}_{k(\mathbf{x}_i,\mathbf{x})}\right). \\ &= sign\left(\sum_i \lambda_i y_i k(\mathbf{x}_i,\mathbf{x})\right). \end{aligned}$$

Remark 1. *The definition of the standard SVM, has two important main points:*

- *Max-margin criterion*
- *Projection of features into arbitrary space (the kernel trick)*

Example 1 (Gaussian Kernel)**.** *The following is the definition of a Gaussian kernel:*

$$k_G(u,v) = \exp\left(-\frac{\|u-v\|^2}{2\sigma^2}\right) = \langle \phi(u), \phi(v)\rangle$$

It can be shown that, for the above Gaussian kernel, the projection function $\phi(u)$ is of infinite dimension!

Example 2 (Polynomial Kernel)**.** *The following is the definition of a Polynomial kernel:*

$$k_(u,v) = (1 + u.v)^d, \quad \text{for any } d \tag{4}$$

Exercise 1. *Given the following kernels*

$$\begin{cases} K(x,x') = \phi(x).\phi(x') \\ K'(x,x') = \phi'(x).\phi'(x') \end{cases}$$

Prove that:

- *For any constant c, it is a valid kernel.*
- *For any constant c, cK is a valid kernel.*
- *$K \times K'$ is a valid kernel.*
- *$K + K'$ is a valid kernel.*
- *The polynomial kernel (defined in the Equation 4) is a valid kernel.*

2.1 A simple gurantee

Here we give a simple error gurantee based on the number of *support vectors.* Suppose h_S is the hypothesis returned by some algorithm, learned on dataset S. Then leave-one-out error of the algorithm on data S is defined by averaging the error of the algorithm on instance x, when it is trained on the rest of the instances $S \setminus \{x\}$:

$$\hat{R}_{loo} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{h_{S \setminus \{i\}}(x_i) \neq y_i\}$$

Lemma 1. *The* expected *leave-one-out error on m instances is an unbiased estimate of the* expected *generalization error over over $m-1$ instances.*

$$\mathbb{E}_{S \sim D^m} \left[\hat{R}_{loo} \right] = \mathbb{E}_{S' \sim D^{m-1}} \left[R_{S'} \right]$$

proof sketch. Distribute the expectation over the sum and decompose it into two independent expectations. □

Lemma 2. *Let h_S be the hypotheis returned by SVM algorithm when trained on S dataset of size m and let $\#_{SV}$ be the number of support vectors in this resut. Then*

$$\mathbb{E}_{S \sim D^{m-1}} \left[R_S \right] \leq \frac{\#_{SV}}{m+1}$$

proof sketch. If a point x is not support vector then h_S and $h_{S \setminus \{x\}}$ should be the same; in other words $h_{S \setminus \{x\}}$ wil gave a correct prediction on x. If a point x is a support vector then $h_{S \setminus \{x\}}$ might make a mistake on x. Replacing these results in the definition of leave-one-out error, using the previous lemma and followed by expectation we get the desired result. □

3 Soft SVM

Instead of having *hard* margins, in many cases we may want to compromise a little, to get more generalization power. So we impose slack variables $\xi \geq 0$ which imposes more flexibility on the separation margins,

$$\begin{cases} \boldsymbol{\beta}.\mathbf{x}_i \geq +1 - \xi_i, & \text{if } y_i = +1 \\ \boldsymbol{\beta}.\mathbf{x}_i \leq -1 + \xi_i, & \text{if } y_i = -1. \end{cases}$$

Also, we want to punish the algorithm whenever there is a non-zero slack:

$$\frac{1}{2} \|\boldsymbol{\beta}\|^2 + C \sum_i \xi_i$$

In other words, we let the algorithm to make a few mistakes, but pay for its cost. Similar to the Equation 2, one could solve the above program. The same problem could be written in the following form:

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|^2 + C \sum_i \max\{0, 1 - y_i \boldsymbol{\beta}.\mathbf{x}_i\} \tag{5}$$

Define the hinge loss to be the following:

$$\phi(\alpha) = (1-\alpha)_+ = \max\{0, 1-\alpha\}$$

One other interpretation of this model is that, we penalize margin violations with a hinge loss; as long as $y_i\boldsymbol{\beta}.\mathbf{x}_i > 1$ the model is not penalized. When $y_i\boldsymbol{\beta}.\mathbf{x}_i < 1$, it is penalized with weight C.

Similar to the previous case, we can form the Lagrangian, form the dual and find the updates of the model.

$$\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\eta}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_i \xi_i - \sum_i \lambda_i \left\{1 - y_i\boldsymbol{\beta}.\mathbf{x}_i - \xi_i\right\} - \sum_i \eta_i\xi_i$$

We first remove the primal variables from the above Lagrangian:

$$\nabla_{\boldsymbol{\beta}}\mathcal{L} = 0 \Rightarrow \boldsymbol{\beta} = \sum_i \lambda_i y_i \mathbf{x}_i$$
$$\nabla_{\boldsymbol{\xi}}\mathcal{L} = 0 \Rightarrow \lambda_i + \eta_i = C$$

By replacing the above equalities in the Lagrangian, we get the following:

$$\begin{cases} \max_{\lambda_i} \sum_i \lambda_i - \frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j \left(\mathbf{x}_i.\mathbf{x}_j\right) \\ \text{subject to:} \\ \qquad \sum_i \lambda_i y_i = 0 \\ \qquad \lambda_i \geq 0, \eta_i \geq 0 \\ \qquad \lambda_i + \eta_i = C \end{cases}$$

And we can easily eliminate η_i and end up with the following program:

$$\begin{cases} \max_{\lambda_i} \sum_i \lambda_i - \frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j \left(\mathbf{x}_i.\mathbf{x}_j\right) \\ \text{subject to:} \\ \qquad \sum_i \lambda_i y_i = 0 \\ \qquad \lambda_i \geq 0 \\ \qquad 0 \leq \lambda_i \leq C \end{cases} \tag{6}$$

How different is the dual of the soft-SVM (Equation 6) from dual of the hard-SVM (Equation 3)? The only difference is that, there is an upper bound C on the dual variables. The interpretation is that, we cannot put too much weight any point.

Remark 2. *If C is bigger than the biggest λ_i, then the soft-SVM is equivalent to the hard-SVM.*

Remark 3. *This form of SVM is usually known as C-SVM.*

4 Kernels and Hilbert spaces

Theorem 1 (Mercer's theorem)**.** *Suppose K is a continuous symmetric non-negative definite kernel. Then there is a set of orthonormal basis $\left\{\varphi_i \in L^2(\mathsf{X}, P)\right\}$ consisting of eigenfunctions of T_k, i.e. $T_K\varphi_j = \lambda_j\varphi_j$, such that the corresponding sequence of eigenvalues $\{\lambda_i\}$ is nonnegative. The eigenfunctions corresponding to non-zero eigenvalues are continuous on X and K have the representation*

$$K(s,t) = \sum_{j=1}^{\infty} \lambda_j\, \varphi_j(s)\, \varphi_j(t), \quad \forall s,t \in \mathsf{X}$$

where the convergence is absolute and uniform.

4.1 Reproducing Kernel Hilbert Spaces(RKHS)

The RKHS property says that, projecting any function in $\mathcal{L}_K(\mathsf{X})$ will produce exact the same function:

$$\begin{aligned}\langle f, K(x,.)\rangle_K &= \langle \sum_j c_j K(x_j,.), K(x,.)\rangle_K \\ &= \sum_j c_j \langle K(x_j,.), K(x,.)\rangle_K \\ &= \sum_j c_j K(x_j,x) = f(x)\end{aligned}$$

Another representation of RKHS is based on the eigen functions spanning the space of the kernels. Any function $f \in \mathcal{L}_K(\mathsf{X})$ can be represented as,

$$f(x) = \sum_i c_i K(x_i,x) = \sum_i c_i \sum_j \lambda_j\, \varphi_j(x_i)\, \varphi_j(x) = \sum_j \sum_i c_i \lambda_j\, \varphi_j(x_i)\, \varphi_j(x) = \sum_j d_j \varphi_j(x)$$

Example 3. *Let X be a compact (i.e. closed and bounded) subset of $\mathbb{R}^d$, and let $K : \mathsf{X} \times \mathsf{X} \to \mathbb{R}$ be Mercer kernel defined over X. With a fixed probability distribution P on X, consider the Hilbert space $L^2(\mathsf{X}, P)$ of functions $g : \mathsf{X} \to \mathbb{R}$, where,*

$$\int_{\mathsf{X}} g^2(x) P(dx) < \infty$$

with the norm defined as,

$$\langle g, g' \rangle = \int_{\mathsf{X}} g(x) g'(x) P(dx) = \mathbb{E}\left[g(X) g'(X)\right]$$

Also consider the operator T_K

$$[T_K \phi](x) = \int_{\mathsf{X}} K(x,t)\varphi(t) P(dt), \quad \forall x \in \mathsf{X}$$

which maps a function?? .

For a given kernel K, define $\mathcal{L}_K(\mathsf{X})$ to be the set of all functions such that,

$$f(x) = \sum_j c_j K(x_j, x)$$

Using Mercer's reproducing kernel theorem, prove that,

1. *Let $J \triangleq \{j \in \mathbb{N} : \lambda_j > 0\}$, and for each $j \in J$ define the function $\psi \triangleq \varphi_j \sqrt{\lambda_j}$. Then $\{\psi_j\}_{j \in J}$ is an orthonormal system in the RKHS $\mathcal{H}_K$, i.e. $\langle \psi_j, \psi_k \rangle_K = \delta_{jk}$, for all $j, k \in J$.*

2. *Let $\mathcal{F}$ be the unit ball of $\mathcal{H}_K$, and let $X_1, X_2, \ldots, X_n$ be drawn i.i.d. from P. Then*

$$\mathbb{E} R_n\left(\mathcal{F}(X^n)\right) \leq \sqrt{\frac{1}{n} \sum_{j=1}^{+\infty} \lambda_j}$$

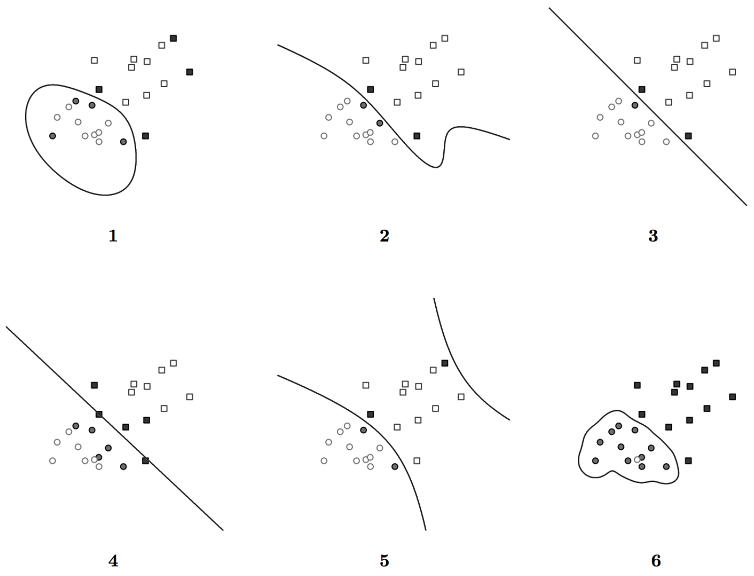

Figure 2: Decision Boundaries

5 Exercise Problems

Consider a dataset with 3 points in 1D:

$$\{(+, 0), (-, -1), (-, +1)\}$$

1. Are the classes $\pm$ linearly separable?
2. Consider mapping each point to 3D using new feature vectors $\Phi(x) = \left[1, x\sqrt{2}, x^2\right]^\top$. Are the classes now linearly separable? If so, find a separating hyperplane.
3. Consider the formulation of the soft-margin primal SVM, for a given training data:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p,\ y_i \in \{-1, 1\}\}_{i=1}^n$$

$$\begin{aligned} &\arg\min_{\mathbf{w},\xi,b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^n \xi_i \right\} \\ &\qquad y_i(\mathbf{w}\cdot\mathbf{x_i} - b) \geq 1 - \xi_i, \quad \forall i = 1, .., n \\ &\qquad \xi_i \geq 0, \quad \forall i = 1, .., n \end{aligned}$$

Also remember the hard-margin primal SVM $\sigma_i = 0, \forall i$. And remember that we can derive the dual formulation and replace each $\mathbf{x}.\mathbf{x}'$ with a kernel function $k(\mathbf{x}, \mathbf{x}')$.

Mach each of the followings with a decision boundary in Figure 2:

(a) A soft-margin linear SVM with $C = 0.1$.

(b) A soft-margin linear SVM with $C = 10$.

(c) A hard-margin kernel SVM with kernel $k(\mathbf{u}, \mathbf{v}) = \mathbf{u}.\mathbf{v} + (\mathbf{u}.\mathbf{v})^2$.

(d) A hard-margin kernel SVM with kernel $k(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{1}{4}\|\mathbf{u} - \mathbf{v}\|^2\right)$.

(e) A hard-margin kernel SVM with kernel $k(\mathbf{u}, \mathbf{v}) = \exp\left(-4\|\mathbf{u} - \mathbf{v}\|^2\right)$.

4. Define a class variable $y_i \in \{-1, +1\}$ which denotes the class of x_i and let $\mathbf{w} = (w_1, w_2, w_3)^\top$. The max-margin SVM classifier solves the following problem

$$\begin{aligned} &\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \\ &y_i(\mathbf{w} \cdot \phi(\mathbf{x_i}) - b) \geq 1, \quad \forall i = 1, .., n \end{aligned}$$

Using the method of Lagrange multipliers show that the solution is $\hat{\mathbf{w}} = (0, 0, -2)^\top$, $b = 1$ and the margin is $1/\|\hat{\mathbf{w}}\|$.

5. What happens if we change the constraints to

$$y_i(\mathbf{w} \cdot \phi(\mathbf{x_i}) - b) \geq \beta, \beta \geq 1$$

Solution:

1. No.

2. The points are mapped to $(1, 0, 0)$, $(1, -\sqrt{2}, 1)$, $(1, \sqrt{2}, 1)$, respectively. The points are now separable in 3-dimensional space. A separating hyperplane is given by the weight vector $(0, 0, 1)$.

3. —

4. First notice that all of the three points are support vectors. Therefore:

$$\begin{aligned} &\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \\ &y_i(\mathbf{w} \cdot \phi(\mathbf{x_i}) - b) = 1, \quad \forall i = 1, 2, 3 \end{aligned}$$

$$L(\mathbf{w}, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1,2,3} \alpha_i \left(y_i(\mathbf{w} \cdot \phi(\mathbf{x_i}) - b) - 1\right)$$

$$\begin{aligned} \frac{\partial L(\mathbf{w}, \alpha)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1,2,3} \alpha_i y_i \phi(\mathbf{x_i}) = 0 \\ \frac{\partial L(\mathbf{w}, \alpha)}{\partial b} &= \sum_{i=1,2,3} \alpha_i y_i = 0 \end{aligned}$$

Therefore:

$$\begin{aligned}
w_1 + \alpha_1 - \alpha_2 - \alpha_3 &= 0 \\
w_2 + \sqrt{2}\alpha_2 - \sqrt{2}\alpha_3 &= 0 \\
w_3 + \alpha_2 - \alpha_3 &= 0 \\
\alpha_1 - \alpha_2 - \alpha_3 &= 0
\end{aligned}$$

which would give us the desired result.

5. —

6 Bibliographical notes

Some intuitions are from David Forsyth's and Feng Liang's classes at UIUC. Peter Bartlett's class notes provided a very good summary of the main points.

7 Some Answers

7.1 Answer to example 3

7.1.1 First part:

The answer is inspired from the formulation in [1]. Based on the definitions we have

$$\begin{aligned}
\langle \psi_j(x), \psi_k(x) \rangle_K &= \left\langle \sqrt{\lambda_j}\varphi_j(x), \sqrt{\lambda_k}\varphi_k(x) \right\rangle_K \\
&= \left\langle \frac{1}{\sqrt{\lambda_j}} \int_{\mathsf{X}} K(x,t)\varphi_j(t)P(dt), \sqrt{\lambda_k}\varphi_k(x) \right\rangle_K && \text{projections} \\
&= \frac{\sqrt{\lambda_k}}{\sqrt{\lambda_j}} \int_{\mathsf{X}} \varphi_j(t) \langle K(x,t), \varphi_k(x) \rangle_K P(dt) \\
&= \frac{\sqrt{\lambda_k}}{\sqrt{\lambda_j}} \int_{\mathsf{X}} \varphi_j(t)\varphi_k(t)P(dt) && \text{RKHS property} \\
&= \frac{\sqrt{\lambda_k}}{\sqrt{\lambda_j}} \langle \varphi_j(t), \varphi_k(t) \rangle_{L^2(\mathsf{X},P)} \\
&= \frac{\sqrt{\lambda_k}}{\sqrt{\lambda_j}} \delta_{k,j} && \varphi_j\text{:orthonormal} \\
&= \delta_{k,j}
\end{aligned}$$

7.1.2 Second part:

We consider the ball of K:

$$\mathcal{F}_\lambda = \{ f \in \mathcal{H}_K : ||f||_K \leq 1 \}$$

Here I am just reviewing the procedure introduced for finding the risk for this,

$$\begin{aligned}
R_n(\mathcal{F}_\lambda(X^n)) &= \sup_{f:||f||_K \leq \lambda} \frac{1}{n}\mathbb{E}_{\sigma^n}\left|\sum_{i=1}^{n} \sigma_i f(X_i)\right| && (7)\\
&= \sup_{f:||f||_K \leq \lambda} \frac{1}{n}\mathbb{E}_{\sigma^n}\left|\sum_{i=1}^{n} \sigma_i \left\langle f, K_{X_i}\right\rangle_K\right| && (8)\\
&= \sup_{f:||f||_K \leq \lambda} \frac{1}{n}\mathbb{E}_{\sigma^n}\left|\left\langle f, \sum_{i=1}^{n} \sigma_i K_{X_i}\right\rangle_K\right| && (9)\\
&= \frac{\lambda}{n}\mathbb{E}_{\sigma^n}\left\|\sum_{i=1}^{n} \sigma_i K_{X_i}\right\|_K && (10)\\
&= \frac{\lambda}{n}\sqrt{\sum_{i=1}^{n} \|K_{X_i}\|_K^2} && (11)\\
&= \frac{\lambda}{n}\sqrt{\sum_{i=1}^{n} \left\langle K_{X_i}, K_{X_i}\right\rangle_K} && (12)\\
& && (13)
\end{aligned}$$

Now we first simplify $\langle K_{X_i}, K_{X_i}\rangle$ and plug in the results in the above bound. But before that, we use the result we found in the previous part. Previously we proved that, $\langle \psi_i, \psi_j\rangle_K = \delta_{i,j}$, we can use this result:

$$\langle \psi_i, \psi_j\rangle_K = \sqrt{\lambda_i\lambda_j}\,\langle \varphi_i, \varphi_j\rangle_K = \delta_{i,j} \Rightarrow \langle \varphi_i, \varphi_j\rangle_K = \frac{1}{\sqrt{\lambda_i\lambda_j}}\delta_{i,j} \tag{14}$$

Using this result, we simplify $\langle K_{X_i}, K_{X_i}\rangle$ in Equation 7.

$$\begin{aligned}
\sum_{i=1}^{n}\langle K_{X_i}, K_{X_i}\rangle &= \sum_{i=1}^{n}\left\langle \sum_{j=1}^{+\infty}\lambda_j\varphi_j(X_i)\varphi(X), \sum_{k=1}^{+\infty}\lambda_k\varphi_k(X_i)\varphi(X)\right\rangle_K\\
&= \sum_{i=1}^{n}\sum_{j=1}^{+\infty}\sum_{k=1}^{+\infty}\lambda_j\lambda_k\varphi_j(X_i)\varphi_k(X_i)\left\langle\varphi(X),\varphi(X)\right\rangle_K\\
&= \sum_{i=1}^{n}\sum_{j=1}^{+\infty}\sum_{k=1}^{+\infty}\frac{\lambda_j\lambda_k}{\sqrt{\lambda_j\lambda_k}}\varphi_j(X_i)\varphi_k(X_i)\delta_{j,k}\\
&= \sum_{i=1}^{n}\sum_{j=1}^{+\infty}\lambda_j\varphi_j^2(X_i)
\end{aligned}$$

Now we plug this result in the bound we found in Equation 7, with $\lambda = 1$ (the unit ball).

$$R_n(\mathcal{F}_\lambda(X^n)) \leq \frac{1}{n}\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{+\infty}\lambda_j\varphi_j^2(X_i)}$$

Now we take expectation with respect to samples,

$$\begin{aligned}
\mathbb{E}R_n(\mathcal{F}_\lambda(X^n)) &\leq \frac{1}{n}\mathbb{E}\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{+\infty}\lambda_j\varphi_j^2(X_i)} \\
&\leq \frac{1}{n}\sqrt{\mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{+\infty}\lambda_j\varphi_j^2(X_i)\right]} \\
&= \frac{1}{n}\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{+\infty}\lambda_j\mathbb{E}\left[\varphi_j^2(X_i)\right]} \\
&= \frac{1}{n}\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{+\infty}\lambda_j} \\
&= \frac{1}{n}\sqrt{n\sum_{j=1}^{+\infty}\lambda_j} \\
&= \sqrt{\frac{1}{n}\sum_{j=1}^{+\infty}\lambda_j}
\end{aligned}$$

Which gives the desired result.

References

[1] Felipe Cucker and Ding Xuan Zhou. *Learning theory: an approximation theory viewpoint.* Number 24. Cambridge University Press, 2007.