Machine Learning: Exercises for Block III
Kernel Methods

Isabel Valera

Exercise 1: Eigenvalues (Exercise 2.4 [2])

Definition Positive Definite Matrix *A complex $m \times m$ matrix K satisfying*

$$\sum_{i,j} c_i \bar{c}_j K_{ij} \geq 0 \tag{1}$$

for all $c_i \in \mathbb{C}$ is called positive definite. The bar in $\bar{c}_j$ denotes complex conjugation; for real numbers, it has no effect. Similarly, a real symmetric $m \times m$ matrix K satisfying (1) for all $c_i \in \mathbb{R}$ is called positive definite.

Prove that a symmetric matrix is positive definite if and only if all its eigenvalues are non-negative.

Exercise 2: Dot products are kernels (Exercise 2.5 [2])

Definition Dot Product *A dot product on a vector space $\mathcal{H}$ is a symmetric bilinear form,*

$$\begin{aligned} \langle ., . \rangle : \mathcal{H} \times \mathcal{H} &\to \mathbb{R} \\ (\mathbf{x}, \mathbf{x}') &\mapsto \langle \mathbf{x}, \mathbf{x}' \rangle \end{aligned}$$

that is strictly positive definite; in other words, it has the property that for all $\mathbf{x} \in \mathcal{H}$, $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$ with equality only for $\mathbf{x} = 0$.

Prove that dot products are positive definite kernels.

Exercise 3: Positive diagonal (Exercise 2.7 [2])

Definition (Positive Definite) Kernel *Let X be a nonempty set. A function k on $x \times X$ which for all $m \in \mathbb{N}$ and all $x_1 \ldots, x_m \in X$ gives rise to a positive definite Gram matrix is called a positive definite (pd) kernel. Often, we shall refer to it simply as a kernel.*

From the definition of a (positive definite) kernel, prove that a kernel k satisfies $k(x, x) \geq 0$ for all $x \in \mathcal{X}$.

Exercise 4: Squared error SVM (Exercise 7.13 [2])

Derive a version of the soft margin classification algorithm which penalizes the errors quadratically.

i) Start from the objective of a soft-margin classifier with $C > 0$:

$$\min_{\boldsymbol{w}\in\mathcal{H},\boldsymbol{\xi}\in\mathbb{R}^n} \frac{1}{2}||\boldsymbol{w}||^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i, \tag{2}$$

replace the second term by $\frac{1}{n}\sum_{i=1}^{n}\xi_i^2$, and derive the dual. Compare the result to the soft-margin support vector machine presented in class, both in terms of algorithmic differences and in terms of robustness properties.

ii) Which algorithm would you expect to work better for Gaussian-like noise, which one for noise with longer tails (thus more outliers)?

Exercise 5: Group error penalty (Exercise 7.14 [2])

Suppose the training data are partitioned into l groups:

$$\begin{matrix} (\boldsymbol{x}_1^1, y_1^1), & ..., & (\boldsymbol{x}_1^{m_1}, y_1^{m_1}) \\ \vdots & & \vdots \\ (\boldsymbol{x}_l^1, y_l^1), & ..., & (x_l^{m_l}, y_l^{m_l}), \end{matrix} \tag{3}$$

where $\boldsymbol{x}_i^j \in \mathcal{H}$ and $y_i^j \in \{-1,+1\}$ with $i = 1,...,l$ and $j = 1,...,m_i$.
Suppose, moreover, that we would like to count a point as misclassified already if one point belonging to the same group is misclassified. Design a soft-margin support vector algorithm where each group's penalty equals the slack of the worst point in that group.

i) Use the following objective and constraints:

$$\begin{aligned} &\min \frac{1}{2}||\boldsymbol{w}||^2 + \sum_i C_i\xi_i \\ &\text{s.t.} \quad y_i^j\left(\langle w, \boldsymbol{x}_i^j\rangle + b\right) \geq 1 - \xi_i \\ &\qquad\ \xi_i \geq 0. \end{aligned} \tag{4}$$

Show that the corresponding dual problem is given by:

$$\begin{aligned} &\max W(\boldsymbol{\alpha}) = \sum_{i,j}\alpha_i^j - \frac{1}{2}\sum_{i,j,i',j'}\alpha_i^j\alpha_{i'}^{j'}y_i^jy_{i'}^{j'}\langle\boldsymbol{x}_i^j, \boldsymbol{x}_{i'}^{j'}\rangle \\ &\text{s.t.} \quad \sum_{i,j}\alpha_i^jy_i^j = 0 \\ &\qquad\ \alpha_i^j \geq 0 \\ &\qquad\ \sum_j \alpha_i^j \leq C_i \ i = 1,...,l. \end{aligned} \tag{5}$$

ii) Argue that typically, only one point per group will become a support vector. Show that the formulation of soft-margin support vector machines given in class is a special case of this algorithm.

Exercise 6: Margin from multipliers (Exercise 7.4 [1])

Show that the value ρ of the margin for the hard-margin support vector machine is given by

$$\frac{1}{\rho^2} = \sum_{i=1}^{n}\alpha_i, \tag{6}$$

where $\boldsymbol{\alpha}$ is given by maximizing the dual representation of the maximum margin problem subject to constraints as defined in Lecture 14 on slide 18.

Exercise 7: Margin from Lagrangian (Exercise 7.5 [1])

i) Show that the values of ρ and $\boldsymbol{\alpha}$ of a hard-margin support vector machine satisfy

$$\frac{1}{\rho^2} = 2\widetilde{L}(\boldsymbol{\alpha}), \tag{7}$$

where $\widetilde{L}(\boldsymbol{\alpha})$ is defined as the objective of the dual representation of the maximum margin problem as defined in Lecture 14 on slide 18.

ii) Show that

$$\frac{1}{\rho^2} = ||\boldsymbol{w}||^2. \tag{8}$$

Exercise 8: Regression SVM (Exercise 7.7 [1])

Consider the Lagrangian of the regression support vector machine (see [1] chapter 7.1.4 on SVMs for regression):

$$\begin{aligned} L(\boldsymbol{w}, b, \xi_i, \hat{\xi}_i) =& C\sum_{i=1}^{n}(\xi_i + \hat{\xi}_i) + \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{n}(\beta_i\xi_i + \hat{\beta}_i\hat{\xi}_i) \\ &- \sum_{i=1}^{n}\alpha_i(\varepsilon + \xi_i + y(x_i) - y_i) - \sum_{i=1}^{n}\hat{\alpha}(\varepsilon + \hat{\xi}_i - y(x_i) + y_i), \end{aligned} \tag{9}$$

where E_ε is the epsilon-insensitive error function:

$$E_\varepsilon(y(x) - y) = \begin{cases} 0 & \text{if } |y(x) - y| < \varepsilon \\ |y(x) - y| - \varepsilon & \text{otherwise} \end{cases} \tag{10}$$

with the largest accepted error ε. We use Lagrange multipliers $\alpha, \hat{\alpha}$ for the constraints with slack variables $\xi_i, \hat{\xi}_i$:

$$\begin{aligned} y_i &\leq y(\boldsymbol{x}_i) + \varepsilon + \xi_i \\ y_i &\geq y(\boldsymbol{x}_i) - \varepsilon - \hat{\xi}_i \end{aligned}$$

and $\beta_i, \hat{\beta}_i$ to express the positivity constraints for $\xi_i, \hat{\xi}_i$.

By setting the derivatives of the of the Lagrangian with respect to $\boldsymbol{w}, b, \xi_i$ and $\hat{\xi}_i$ to zero and then back substituting to eliminate the corresponding variables, show that the dual Lagrangian is given by

$$\begin{aligned} \widetilde{L}(\mathbf{a}, \widehat{\mathbf{a}}) =& -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\alpha_i - \widehat{\alpha}_i\right)\left(\alpha_j - \widehat{\alpha}_j\right)k\left(\mathbf{x}_i, \mathbf{x}_j\right) \\ &- \varepsilon\sum_{i=1}^{i}\left(\alpha_i + \widehat{\alpha}_i\right) + \sum_{i=1}^{n}\left(\alpha_i - \widehat{\alpha}_i\right)y_i. \end{aligned} \tag{11}$$

with respect to $\boldsymbol{\alpha}$ and $\widehat{\boldsymbol{\alpha}}$. The kernel is defined as $k\left(\mathbf{x}, \mathbf{x}'\right) = \varphi(\mathrm{x})^{\mathrm{T}}\varphi\left(\mathrm{x}'\right)$.

References

[1] C. M. Bishop. *Pattern recognition and machine learning.* springer, 2006.

[2] B. Schölkopf, A. J. Smola, F. Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002.