EECS 553: Machine Learning (ECE) **University of Michigan**

Support Vector Machines

Winter 2023 **Clayton Scott**

1 Optimal Soft Margin Hyperplane

The support vector machine (SVM) is one of the most useful general purpose classifiers in the machine learning toolbox.[1] This nonlinear classifier is obtained by kernelizing the optimal soft-margin hyperplane classifier. Recall that the latter is the solution of

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \ & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{n}\sum_{i=1}^n \xi_i \\ \text{s.t.} \ & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geqslant 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i \geqslant 0, \quad i = 1, \ldots, n. \end{aligned}$$

To kernelize this algorithm, we will use Lagrange multiplier theory. In particular, we will derive the dual optimization problem, and then recover the solution of the OSM hyperplane classifier from its dual. The dual can easily be kernelized, which allows us to kernelize the OSM hyperplane classifier.

2 The Optimal Soft Margin Dual

Let α_i be the Lagrange multiplier (dual variable) corresponding to the constraint $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geqslant 1 - \xi_i$, and let β_i be the Lagrange multiplier corresponding to the constraint $\xi_i \geq 0$. Also denote $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)$. The Lagrangian is

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{n}\sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^n \beta_i\xi_i. \tag{1}$$

The dual problem is

$$\max_{\boldsymbol{\alpha}\geqslant 0, \boldsymbol{\beta}\geqslant 0} L_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{2}$$

where

$$L_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\boldsymbol{w},b,\boldsymbol{\xi}} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}). \tag{3}$$

and we use the notation $\boldsymbol{\alpha} \geq \mathbf{0}$ to mean that $\alpha_i \geq 0 \ \forall i$, and similary for $\boldsymbol{\beta}$.

The optimization problem defining the dual function is an unconstrained minimization with a convex, differentiable objective function. Therefore, for fixed $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, the values of $\boldsymbol{w}, b$ and $\boldsymbol{\xi}$ achieving the minimum satisfy

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{w}} &= \boldsymbol{w} - \sum \alpha_i y_i \boldsymbol{x}_i = \mathbf{0} \\ \frac{\partial L}{\partial b} &= -\sum \alpha_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i} &= \frac{C}{n} - \alpha_i - \beta_i = 0 \quad \forall i. \end{aligned}$$

[1] https://jmlr.org/papers/v15/delgado14a.html

Therefore, if $\sum_i \alpha_i y_i = 0$ and $\alpha_i + \beta_i = \frac{C}{n}\ \forall i$, then

$$\begin{aligned} L_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \frac{1}{2}\Big\| \sum_i \alpha_i y_i \boldsymbol{x}_i \Big\|^2 - \sum_i \alpha_i y_i \Big\langle \sum_j \alpha_j y_j \boldsymbol{x}_j, \boldsymbol{x}_i \Big\rangle + \sum_i \alpha_i \\ &= \frac{1}{2}\Big\langle \sum_i \alpha_i y_i \boldsymbol{x}_i, \sum_j \alpha_j y_j \boldsymbol{x}_j \Big\rangle - \Big\langle \sum_i \alpha_i y_i \boldsymbol{x}_i, \sum_j \alpha_j y_j \boldsymbol{x}_j \Big\rangle + \sum_i \alpha_i \\ &= -\frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j + \sum_i \alpha_i. \end{aligned}$$

Furthermore note that if $\sum_i \alpha_i y_i \neq 0$ or $\alpha_i + \beta_i \neq \frac{C}{n}$ for some i, then $L_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\infty$. To see this, suppose $\sum_i \alpha_i y_i \neq 0$. Then we may send b to $+\infty$ or $-\infty$, whichever has the opposite sign of $-\sum_i \alpha_i y_i$, and the Lagrangian will tend to $-\infty$. Similarly, if $\alpha_i + \beta_i \neq \frac{C}{n}$, we can send ξ_i to $\pm\infty$. In summary, the dual function is[2]

$$L_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \begin{cases} -\frac{1}{2}\sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j + \sum_i \alpha_i, & \text{if } \sum_i \alpha_i y_i = 0 \text{ and } \alpha_i + \beta_i = \frac{C}{n}\ \forall i, \\ -\infty, & \text{otherwise.} \end{cases}$$

Therefore, the dual optimization problem

$$\max_{\boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\beta} \geq \mathbf{0}} L_D(\boldsymbol{\alpha}, \boldsymbol{\beta})$$

may be written

$$\begin{aligned} \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \ & -\frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \sum_{i=1}^n \alpha_i \\ \text{s.t} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \\ & \alpha_i + \beta_i = \frac{C}{n} \quad \forall i \\ & \alpha_i \geqslant 0, \beta_i \geqslant 0 \quad \forall i. \end{aligned}$$

We can simplify this problem by eliminating $\boldsymbol{\beta}$. This is because the constraints $\alpha_i + \beta_i = \frac{C}{n}, \alpha_i \geq 0$ and $\beta_i \geq 0$ are equivalent to $0 \leq \alpha_i \leq \frac{C}{n}$. This leads to an alternate form of the optimal soft-margin dual:

$$\begin{aligned} \max_{\alpha} \ & -\frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \sum_{i=1}^n \alpha_i \\ \text{s.t} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \\ & 0 \leqslant \alpha_i \leqslant \frac{C}{n} \quad \forall i. \end{aligned}$$

Both versions of the dual (with and without $\boldsymbol{\beta}$) are useful and so we will keep both in mind. The latter is the one we solve numerically, but the former is useful for recoving the primal solution. Note that given a solution $\boldsymbol{\alpha}^*$ to the latter formulation, a solution to the former is given by $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ where $\beta_i^* = \frac{C}{n} - \alpha_i^*$. Note that both formulations of the dual are quadratic program (QP).

The main takeaway message from this section is that the dual optimization problem can be solved efficiently by modern optimization solvers, and can easily be kernelized owing to the training data appearing exclusively in inner products.

[2]The dual function is incorrectly stated in essentially every textbook that treats SVMs, because the second case is not stipulated. The second case is essential for logically arriving at a tractable formulation of the dual problem.

2.1 Recovery of primal solution

Let $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ be dual optimal, and let $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*)$ be primal optimal.[3] Since the primal is convex and satisfies a constraint qualification, strong duality holds. The KKT necessity theorem then implies that $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ satisfies the KKT conditions. We may use these conditions to determine $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*)$ in terms of $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$.

We first obtain $\boldsymbol{w}^*$ from $\boldsymbol{\alpha}^*$. From the first KKT condition,

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \boldsymbol{x}_i$$

and so the optimal normal vector is a linear combination of data points.

To find b^*, consider any j such that $0 < \alpha_j^* < \frac{C}{n}$. By complementary slackness, $0 < \alpha_j^*$ implies that

$$y_j({\boldsymbol{w}^*}^T \boldsymbol{x}_j + b^*) = 1 - \xi_j^*. \tag{4}$$

Again by complementary slackness, $\alpha_j^* < \frac{C}{n}$ implies $\xi_j^* = 0$. To see this, recall that we must have $\beta_j^* \xi_j^* = 0$, and $\alpha_j^* < \frac{C}{n}$ is equivalent to $\beta_j^* > 0$. Putting these observations together, we need b^* to satisfy

$$y_j({\boldsymbol{w}^*}^T \boldsymbol{x}_j + b^*) = 1 \tag{5}$$

for any j such that $0 < \alpha_j^* < \frac{C}{n}$. Solving for b^* results in (recall $y_j = \pm 1$ so $y_j^2 = 1$)

$$\begin{aligned} b^* &= y_j - {\boldsymbol{w}^*}^T \boldsymbol{x}_j \\ &= y_j - \sum_i \alpha_i^* y_i \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle. \end{aligned}$$

Note that b^* can be computed using any j such that $0 < \alpha_j^* < \frac{C}{n}$, and therefore all such j lead to the same value for b^*. In practice, it is common to average over several such j to counter numerical errors.

Although not needed for the final classifier, the optimal slack variables ξ_i^* can be recovered from the formula

$$\begin{aligned} \xi_i^* &= \max(0, 1 - y_i(\langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle + b^*)) \\ &= \max(0, 1 - y_i(\sum_j \alpha_j^* y_j \langle \boldsymbol{x}_j, \boldsymbol{x}_i \rangle + b^*)) \end{aligned}$$

which we saw when studying empirical risk minimization.

The final classifier is therefore

$$\begin{aligned} f(\boldsymbol{x}) &= \operatorname{sign}\left(\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle + b^*\right) \\ &= \operatorname{sign}\left(\sum_{i=1}^{n} \alpha_i^* y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b^*\right) \end{aligned} \tag{6}$$

where

$$\begin{aligned} b^* &= y_j - \langle \boldsymbol{w}^*, \boldsymbol{x}_j \rangle \\ &= y_j - \sum_i \alpha_i^* y_i \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \end{aligned} \tag{7}$$

for any j such that $0 < \alpha_j^* < \frac{C}{n}$.

The main takeaways are that the final classifier can be computed in terms of the dual solution, and that both the final classifier and the dual are clearly kernelizable, i.e., the training and test instances appear only in inner products of the form $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ and $\langle \boldsymbol{x}_i, \boldsymbol{x} \rangle$.

[3] For this argument to be complete, we must show that a maximizer and minimizer exist for the dual and primal problems, respectively. This is not true for all constrained optimization problems but it is in the present case. This is left as an exercise.

2.2 Support vectors

From complementary slackness,

$$\alpha_i^*(1 - \xi_i^* - y_i(\boldsymbol{w}^{*T}\boldsymbol{x}_i + b^*)) = 0. \tag{8}$$

If $\boldsymbol{x}_i$ satisfies

$$y_i(\boldsymbol{w}^{*T}\boldsymbol{x}_i + b^*) = 1 - \xi_i^*, \tag{9}$$

we call $\boldsymbol{x}_i$ a *support vector*. Therefore, if $\boldsymbol{x}_i$ is not a support vector, then $\alpha_i^* = 0$. This means $\boldsymbol{w}^*$ depends only on the support vectors:

$$\boldsymbol{w}^* = \sum_{\substack{\text{support} \\ \text{vectors}}} \alpha_i^* y_i \boldsymbol{x}_i. \tag{10}$$

There is a geometric interpretation:

1. If $y_i(\boldsymbol{w}^{*T}\boldsymbol{x}_i + b^*) > 1$, then $\xi_i^* = 0$ and $\boldsymbol{x}_i$ is not a support vector.
2. If $y_i(\boldsymbol{w}^{*T}\boldsymbol{x}_i + b^*) = 1$, then $\xi_i^* = 0$ and $\boldsymbol{x}_i$ is a support vector.
3. If $0 \leqslant y_i(\boldsymbol{w}^{*T}\boldsymbol{x}_i + b^*) < 1$, then $\xi_i^* > 0$ and $\boldsymbol{x}_i$ is a support vector.
4. If $y_i(\boldsymbol{w}^{*T}\boldsymbol{x}_i + b^*) < 0$, then $\xi_i^* > 0$ and $\boldsymbol{x}_i$ is a support vector.

Based on which case occurs, we say

1. $\boldsymbol{x}_i$ is outside the margin
2. $\boldsymbol{x}_i$ is on the margin
3. $\boldsymbol{x}_i$ is within the margin
4. $\boldsymbol{x}_i$ is misclassified

Cases (b)-(d) are called *margin errors*, and here we have $\xi_i^* = 1 - y_i(\boldsymbol{w}^{*T}\boldsymbol{x}_i + b^*)$. As an exercise you are asked to show that for margin errors, ξ_i^* is proportional to the distance from $\boldsymbol{x}_i$ to the corresponding margin hyperplane. This can be visualized as in Figure 1. The stems correspond to margin errors/ support vectors. In summary, the support vectors consist of all training data points that are not on the correct side of the decision boundary by more than the margin.

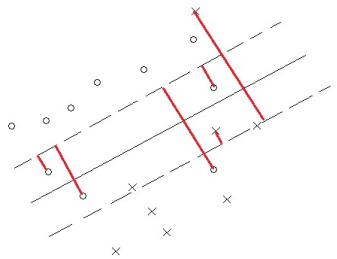

Figure 1: Margin errors indicated by stems.

3 The Support Vector Machine

From Eqns. (6) and (7), the final classifier only involves the data via inner products. Furthermore, the same is true of the dual QP. Therefore the optimal soft margin hyperplane can be kernelized. The resulting classifier is known as a *support vector machine*.

Let k be an inner product kernel. The SVM classifier is

$$f(\boldsymbol{x}) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i^* y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b^*\right) \tag{11}$$

where $\boldsymbol{\alpha}^*$ is the solution of

$$\begin{aligned} \max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum \alpha_i \\ s.t \quad \sum \alpha_i y_i = 0 \\ 0 \leqslant \alpha_i \leqslant \frac{C}{n}, \quad i = 1, ..., n \end{aligned} \tag{12}$$

and b^* is given by

$$b^* = y_j - \sum_{i=1}^{n} \alpha_i^* y_i k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{13}$$

for any j such that $0 < \alpha_j^* < \frac{C}{n}$. The final classifier depends only on those training data points that are support vectors. In many applications this is a small fraction of the full training set, which reduces the memory and time needed to store and evaluate the classifier, respectively. The size of the dual (i.e., the number of variables) is n, and in particular it is independent of the output dimension of the feature map Φ associated with k, which could be infinite.

4 The SMO Algorithm ⚠

SMO stands for sequential minimal optimization. In the context of SVMs, it is an algorithm to efficiently solve the SVM dual. The basic conceptual strategy is that of *coordinate ascent*.

For the moment consider a general constrained optimization problem

$$\max_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha})$$
$$\text{s.t. } \boldsymbol{\alpha} \in S$$

where S is the feasible set. Coordinate ascent does

Initialize $\alpha^0 = (\alpha_1^0, ..., \alpha_n^0) \in S$
$t \leftarrow 0$
repeat
  $t \leftarrow t + 1$
  for $i = 1 \ldots, n$ **do**
    $\alpha_i^t = \arg\max_\alpha f(\alpha_1^t, ..., \alpha_{i-1}^t, \alpha, \alpha_{i+1}^{t-1}, ..., \alpha_n^{t-1})$
  end for
until converged

If $n = 2$, we have the following picture:

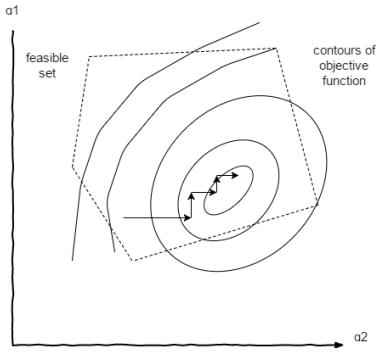

Figure 2: Coordinate ascent.

However, we cannot apply CA to the SVM as stated above. The reason is that the constraint $\sum \alpha_i y_i = 0$ determines each α_i as a function of the others, namely

$$\alpha_i = y_i(-\sum_{j \neq i} \alpha_j y_j), \tag{14}$$

where we used $y_i^2 = 1$. So instead, SMO updates two variables at a time:

Initialize α^0 (any feasible pt, e.g., the zero vector)
repeat
  Select α_i, α_j to be updated
  Update α_i, α_j by solving the SVM dual QP, holding all other α_k, $k \neq i, j$ fixed.
until termination criterion satisfied

The first step is usually accomplished with a heuristic that estimates which two variables will lead to the largest increase in the objective, such as a measure of how much the variables violate the KKT conditions.

The point of SMO is that the second step can be performed efficiently. Suppose $\alpha_3, \ldots, \alpha_n$ are fixed. We need to solve

$$\begin{aligned} \max_{\alpha_1,\alpha_2} \; & -\frac{1}{2}[\alpha_1^2 k_{11} + \alpha_2^2 k_{22} + 2\alpha_1\alpha_2 y_1 y_2 k_{12}] + c_1\alpha_1 + c_2\alpha_2 \\ \text{s.t.} \; & \alpha_1 y_1 + \alpha_2 y_2 = -\sum_{l=3}^{n} y_l \alpha_l \\ & 0 \leqslant \alpha_1, \alpha_2 \leqslant \frac{C}{n}. \end{aligned}$$

Here c_1 and c_2 are constants depending on $\alpha_3, \ldots, \alpha_n$, and $k_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

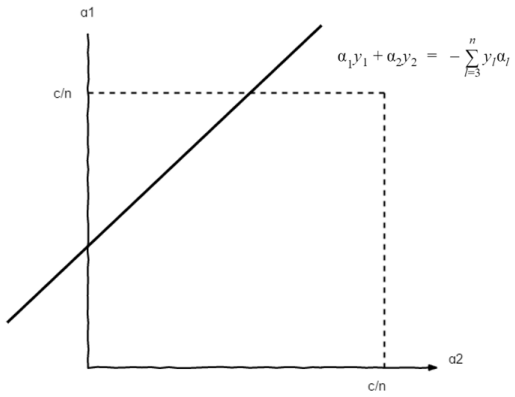

Figure 3: The feasible set for the SMO subproblem is the line segment given by the intersection of the line and the box in the figure.

We can use the linear equality constraint to solve for α_2 in terms of α_1, and then solve the resulting QP for α_1. The objective is just a parabola, and so the maximum occurs either at the critical point of the parabola, or the boundary of the feasible range for α_1. If the critical point is feasible, it is optimal. If not, the optimizer has to be one of the endpoints of the feasible interval. See Figures 3 and 4.

The SMO algorithm converges to the global optimum. The computational complexity is $O(n^3)$ worst case, but typically more like $O(n^2)$. For more on SMO, see the paper by Platt (1999). While the ideas in the original paper have been improved upon, state of the art solvers for SVMs still typically rely on some form of dual coordinate ascent.

Exercises

1. (★) True or false: $\boldsymbol{x}_i$ is a support vector if and only if $\alpha_i^* = 0$.
2. (★) Explain how to determine ξ_i^*, $i = 1, \ldots, n$, from $\boldsymbol{\alpha}^*$.
3. (★★) Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ be training data for a binary classification problem, where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. A *cost-sensitive support vector machine* is the solution to

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C_+}{n}\sum_{i:y_i=1}\xi_i + \frac{C_-}{n}\sum_{i:y_i=-1}\xi_i \\ \text{s.t.} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, n \\ & \xi_i \geq 0 \quad i = 1, 2, \ldots, n. \end{aligned}$$

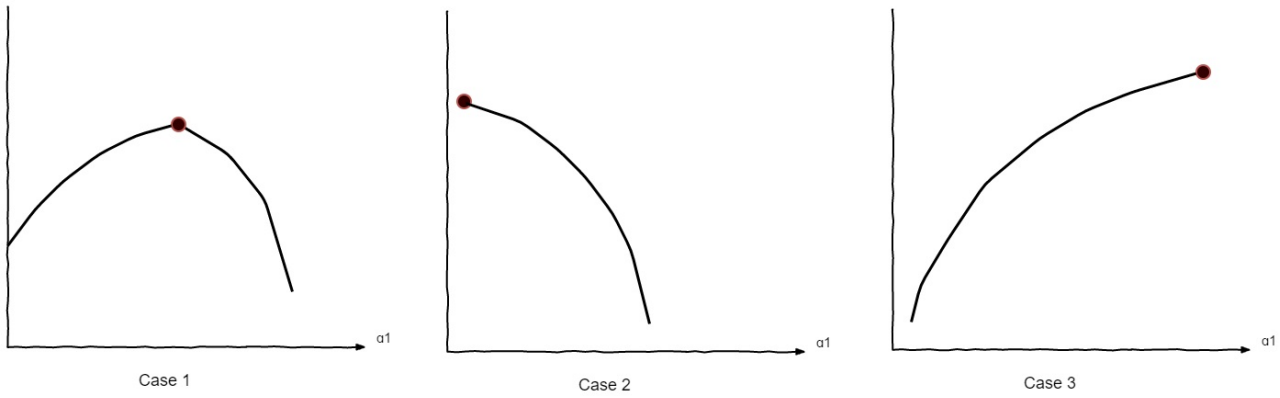

Figure 4: The objective of the SMO subproblem, after eliminating one of the variables, is just a parabola. The maximizer is either the critical point of the parabola (case 1) or at the boundary corresponding to $\alpha_1 = 0$ or $\alpha_1 = C/n$ (cases 2 and 3).

C_+ and C_- allow the user to assign different costs to margin errors (positive slack variables) from the two classes. This is useful when false positives and false negatives carry different importance in a given application. Determine the dual quadratic program of the above optimization problem, and show how to kernelize the classifier.

4. (★★) Suppose we drop the offset b, so that the linear decision boundary passes through the origin. The primal QP defining the *optimal soft margin hyperplane classifier without offset* is

$$\begin{aligned} \min_{\boldsymbol{w},\xi} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle) \geq 1 - \xi_i \quad \text{for } i = 1, 2, \ldots, n \\ & \xi_i \geq 0 \quad \text{for } i = 1, 2, \ldots, n \end{aligned}$$

   (a) Derive the dual quadratic program.

   (b) Explain how to kernelize this method, including how to evaluate the final classifier.

   (c) Suggest a kernel for which this method would be appropriate, i.e., for which there is little to no loss in dropping the offset term.

5. (★★) Consider the following variation on the soft-margin hyperplane:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{2n}\sum_{i=1}^{n}\xi_i^2 \\ \text{s.t.} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i \qquad \text{for } i = 1, 2, \ldots, n \\ & \xi_i \geq 0 \qquad \text{for } i = 1, 2, \ldots, n \end{aligned}$$

   where $C > 0$. Now we are penalizing margin violations quadratically instead of linearly. This hyperplane is called the optimal soft-margin hyperplane classifier with *squared hinge loss.*

   (a) Argue that the constraints $\xi_i \geq 0$ can be dropped.

(b) Determine the dual QP. Your result should look fairly similar to the dual of the soft-margin hyperplane with hinge loss.

(c) Argue that the above method can be kernelized, thus yielding a new method for nonlinear classification, which we could call the SVM with squared hinge loss.

(d) Given an inner product kernel k, find an inner product kernel k' such that the SVM with squared hinge loss with kernel k yields the same $\boldsymbol{\alpha}^*$ as the standard SVM with hinge loss and kernel k'.

(e) Explain how b^* can be recovered from the dual solution.

6. (★★) *Support vector regression* (SVR) is a method for regression analogous to the support vector classifier. Let $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \ldots, n$ be training data for a regression problem.

   In the case of *linear regression*, SVR solves

$$\begin{aligned} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{C}{n}\sum_{i=1}^n (\xi_i^+ + \xi_i^-) \\ \text{s.t.} \quad & y_i - \boldsymbol{w}^T\boldsymbol{x}_i - b \le \epsilon + \xi_i^+ \quad \forall i \\ & \boldsymbol{w}^T\boldsymbol{x}_i + b - y_i \le \epsilon + \xi_i^- \quad \forall i \\ & \xi_i^+ \ge 0 \quad \forall i \\ & \xi_i^- \ge 0 \quad \forall i \end{aligned}$$

   where $\boldsymbol{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, $\boldsymbol{\xi}^+ = (\xi_i^+, \ldots, \xi_n^+)^T$, and $\boldsymbol{\xi}^- = (\xi_i^-, \ldots, \xi_n^-)^T$.

   Here $C > 0$, $\epsilon > 0$ are fixed.

   (a) Show that for an appropriate choice of λ, linear SVR solves

$$\min_{\boldsymbol{w}, b} \quad \frac{1}{n}\sum_{i=1}^n \ell_\epsilon(y_i, \boldsymbol{w}^T\boldsymbol{x}_i + b) + \lambda\|\boldsymbol{w}\|_2^2$$

   where $\ell_\epsilon(y, t) = \max\{0, |y - t| - \epsilon\}$ is the so-called *ϵ-insensitive loss*, which does not penalize prediction errors below a level of ϵ.

   Note: This problem is not used to solve the remaining problems.

   (b) Derive the dual optimization problem in a manner analogous to the support vector classifier. As in the SVC, you should eliminate the dual variables corresponding to the constraints $\xi_i^+ \ge 0$, $\xi_i^- \ge 0$.

   (c) Explain how to kernelize SVR. Be sure to explain how to determine b^*.

   (d) Argue that the final predictor will only depend on a subset of training examples, and characterize those training examples.

7. (★★) Let $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$, $i = 1, \ldots, n$, be classification training data. Consider the following variation on the soft-margin hyperplane:

$$\begin{aligned} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{2n}\sum_{i=1}^n \xi_i^2 \\ \text{s.t.} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) = 1 - \xi_i \quad \text{for } i = 1, 2, \ldots, n \end{aligned}$$

   Note the equality in the constraint, and also note that ξ_i is not constrained to be nonnegative. This formulation, whose kernelization is called the *least squares SVM*, has the disadvantage that correctly labeled points can be penalized. On the other hand, it is convenient from a computational perspective as you are asked to show.

(a) Write down the Lagrangian. Note that the Lagrange multipliers $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ can be negative since we have equality constraints.

(b) Write down the KKT conditions and express the optimal $\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}$ as the solution of a system of linear equations. You need not write the system in matrix form. The number of equations should equal the number of variables.

(c) Eliminate $\boldsymbol{w}$ and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)$ from this system, and express the optimal α and b as the solution of a system of linear equations in matrix form. That is, express

$$\begin{bmatrix} ? & ? & \cdots & ? \\ ? & \ddots & ? & \vdots \\ \vdots & ? & \ddots & \vdots \\ ? & \cdots & \cdots & ? \end{bmatrix} \begin{bmatrix} b \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ \vdots \\ ? \end{bmatrix}$$

by filling in the ?'s.

(d) Explain how to extend this linear classifier to a nonlinear classifier through the use of symmetric, positive definite kernels.

8. (★★★) Consider a classification problem where there are two classes, but we only have training data from one of the classes. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ denote the training data from this class. The goal is to design a good classifier even though we have no data from the other class. This problem is often referred to as one-class classification, anomaly detection, or novelty detection (the unobserved class is viewed as an anomaly or novelty).

Let $\phi(t) = \max(0, 1 - t)$ be the hinge loss. Consider the optimization problem

$$\min_{\boldsymbol{w}} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n} \phi(\boldsymbol{w}^T \boldsymbol{x}_i)$$

where $\lambda > 0$ is fixed. The solution $\boldsymbol{w}$ defines an anomaly detector, called the *one-class support vector machine* (OC-SVM), by the function

$$f(\boldsymbol{x}) = \text{sign}\{\boldsymbol{w}^T \boldsymbol{x} - 1\},$$

where a prediction of +1 corresponds to the observed class, and -1 to the unobserved class.

At first glance, it may not be clear why this is a good approach to one-class classification. Below, when we kernelize the algorithm, the utility of this classifier will be more apparent.

(a) Rewrite the above optimization problem as a quadratic program in the variables $\boldsymbol{w}$ and $\xi_1, \ldots, \xi_n$ where ξ_i are slack variables.

(b) Derive the dual optimization problem to the quadratic program from part (a). You do not need to explain how to solve the dual.

(c) Explain how to kernelize the OC-SVM. In the case of the Gaussian kernel, provide an intuitive interpretation of classifier.

9. (★) Determine c_1 and c_2 in the SMO algorithm. Then eliminate α_2 and state the QP used to optimize α_1.