CS 260: Machine Learning **Spring 2020**

Homework 5

Hand Out: May.25 *Due: June.7*

1. **Exercise 15.1**

   Let $\mathcal{H}$ be the class of halfspaces in $\mathbb{R}^d$, and let $S = ((\mathbf{x}_i, y_i)_{i=1}^m)$ be a linearly separable set. Let $\mathcal{G} = \{(\mathbf{w}, b) : ||\mathbf{w}|| = 1, (\forall i \in [m])\ y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0\}$. Our assumptions imply that this set is non-empty. Note that for every $(\mathbf{w}, b) \in \mathcal{G}$,

   $$\min_{i \in [m]} y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$$

   On the contrary, for every $||\mathbf{w}|| = 1$ and $(\mathbf{w}, b) \notin \mathcal{G}$,

   $$\min_{i \in [m]} y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq 0$$

   It follows that

   $$\operatorname*{argmax}_{(\mathbf{w},b):||\mathbf{w}||=1} \min_{i \in [m]} y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \subseteq \mathcal{G}$$

   Hence, solving the second optimization problem is equivalent to the following problem:

   $$\operatorname*{argmax}_{(\mathbf{w},b)\in\mathcal{G}} \min_{i \in [m]} y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)$$

   Finally, since for every $(\mathbf{w}, b) \in \mathcal{G}$, and every $i \in [m]$, $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) = |\langle \mathbf{w}, \mathbf{x}_i \rangle + b|$, we obtain that the second optimization problem is equivalent to the first optimization problem. ■

2. **Exercise 15.2**

   Let $S = ((\mathbf{x}_i, y_i))_{i=1}^m \subseteq (\mathbb{R}^d \times \{-1, 1\}^m)$ be a linearly separable set with a margin γ, such that $\max_{i \in [m]} ||\mathbf{x}_i|| \leq \rho$ for some $\rho > 0$. The margin assumption implies that there exists $(\mathbf{w}, b) \in \mathbb{R}^d \times \mathbb{R}$ such that $||\mathbf{w}|| = 1$, and

   $$(\forall i \in [m]) \qquad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq \gamma$$

   Hence,

   $$(\forall i \in [m]) \qquad y_i(\langle \mathbf{w}/\gamma, \mathbf{x}_i \rangle + b/\gamma) \geq 1$$

   Let $\mathbf{w}^* = \mathbf{w}/\gamma$. We have $||\mathbf{w}^*|| = 1/\gamma$. Applying Theorem 9.1, we obtain that the number of iterations of the perceptron algorithm is bounded above by $(\rho/\gamma)^2$. ■

3. **Exercise 16.2**

   In the Kernelized Perceptron, the weight vector $\mathbf{w}^{(t)}$ will not be explicitly maintained. Instead, our algorithm will maintain a vector $\boldsymbol{\alpha}^{(t)} \in \mathbb{R}^m$. In each iteration we update $\boldsymbol{\alpha}^{(t)}$ such that

   $$\mathbf{w}^{(t)} = \sum_{i=1}^{m} \alpha_i^{(t)} \psi(\mathbf{x}_i) \tag{1}$$

Assuming that Equation (1) holds, we observe that the condition

$$\exists i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \psi(\mathbf{x}_i) \rangle \leq 0$$

is equivalent to the condition

$$\exists i \text{ s.t. } y_i \sum_{i=1}^{m} \alpha_j^{(t)} K(\mathbf{x}_i, \mathbf{x}_j) \leq 0$$

which can be verified while only accessing instances via the kernel function. ■

We will now detail the update $\boldsymbol{\alpha}^{(t)}$. At each time t, if the required update is $\mathbf{w}_{t+1} = \mathbf{w}_t + y_i \mathbf{x}_i$, we make the update

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} + y_i \mathbf{e}_i$$

A simple inductive argument shows that Equation (1) is satisfied. Finally, the algorithm returns $\boldsymbol{\alpha}^{(T)}$. Given a new instance $\mathbf{x}$, the prediction is calculated using $\text{sign}(\sum_{i=1}^{m} \alpha_i^{(T)} K(\mathbf{x}_i, \mathbf{x}))$. ■

4. **Exercise 16.3**

The representor theorem tells us that the minimizer of the training error lies in $\text{span}(\{\psi(\mathbf{x}_1), \cdots, \psi(\mathbf{w}_m)\})$. That is, the ERM objective is equivalent to the following objective:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^m} \lambda || \sum_{i=1}^{m} \alpha_i \psi(\mathbf{x}_i) ||^2 + \frac{1}{2m} \sum_{i=1}^{m} \left(\left\langle \sum_{j=1}^{m} \alpha_j \psi(\mathbf{x}_j), \psi(\mathbf{x}_i) \right\rangle - y_i \right)$$

Denoting the gram matrix by G, the objective can be rewritten as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^m} \lambda \boldsymbol{\alpha}^T G \boldsymbol{\alpha} + \frac{1}{2m} \sum_{i=1}^{m} (\langle \boldsymbol{\alpha}, G_{\cdot,i} \rangle - y_i)^2 \tag{2}$$

Note that the objective (Equation (2)) is convex[1]. It follows that a minimizer can be obtained by differentiating Equation (2), and comparing to zero. Define $\lambda' = 2m \cdot \lambda$. We obtain

$$(\lambda' G + G G^T) \boldsymbol{\alpha} - G^T \mathbf{y} = \mathbf{0}$$

Since G is symmetric, this can be rewritten as

$$G(\lambda' I + G) \boldsymbol{\alpha} = G \mathbf{y}$$

A sufficient (and necessary in case that G is invertible) condition for the above to hold is that

$$(\lambda' I + G) \boldsymbol{\alpha} = \mathbf{y}$$

Since G is positive semi-definite and $\lambda' > 0$, the matrix $\lambda' I + G$ is positive definite, and thus invertible. We obtain that $\boldsymbol{\alpha}^* = (\lambda' I + G)^{-1} \mathbf{y}$ is a minimizer of our objective. ■

[1] The term $\frac{2}{m} \sum_{i=1}^{m} (\langle \alpha, G_{\cdot,i} \rangle - y_i)^2$ is simply the least square objective, and thus it is convex, as we have already seen. The Hessian of $\boldsymbol{\alpha} G^T \boldsymbol{\alpha}$ is G, which is positive semi-definite. Hence, $\boldsymbol{\alpha} G^T \boldsymbol{\alpha}$ is also convex. Our objective is a weighted sum, with non-negative weights, of the two convex terms above. Thus, it is convex.

5. **Exercise 16.4**

Define $\psi : \{1, \cdots, N\} \to \mathbb{R}^N$ by

$$\psi(j) = (\mathbf{1}^j; \mathbf{0}^{N-j})$$

where $\mathbf{1}^j$ is the vector in $\mathbb{R}^j$ with all elements equal to 1, and $\mathbf{0}^{N-j}$ is the zero vector in $\mathbb{R}^{N-j}$. Then, assuming that the standard inner product, we obtain that $\forall (i,j) \in [N]^2$,

$$\langle \psi(i), \phi(j) \rangle = \langle (\mathbf{1}^i; \mathbf{0}^{N-i}), (\mathbf{1}^j; \mathbf{0}^{N-j}) \rangle = \min\{i, j\} = K(i, j)$$

■

6. **Exercise 16.6**

a. We will work with the label set $\{\pm 1\}$.

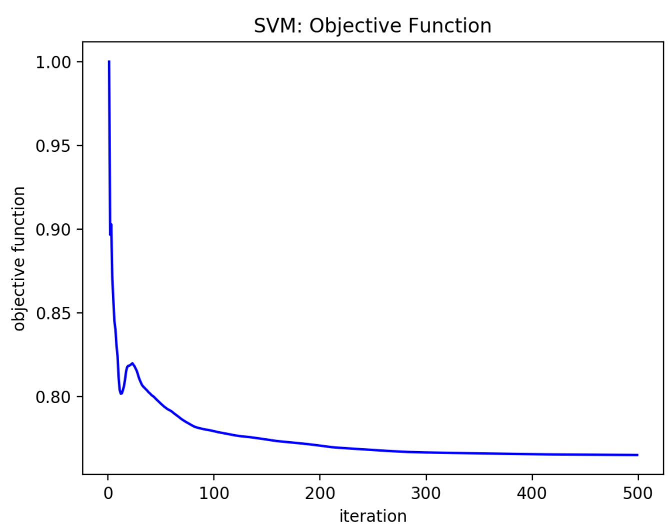
$$\begin{aligned} h(\mathbf{x}) &= \text{sign}(||\psi(\mathbf{x}) - c_-||^2 - ||\psi(\mathbf{x}) - c_+||^2) \\ &= \text{sign}(2\langle \psi(\mathbf{x}), c_+ \rangle - 2\langle \psi(\mathbf{x}), c_- \rangle + ||c_-||^2 - ||c_+||^2) \\ &= \text{sign}(2(\langle \psi(\mathbf{x}), \mathbf{w} \rangle + b)) \\ &= \text{sign}(\langle \psi(\mathbf{x}), \mathbf{w} \rangle + b) \end{aligned}$$

b. Simply note that

$$\begin{aligned} \langle \psi(\mathbf{x}), \mathbf{w} \rangle &= \langle \psi(\mathbf{x}), c_+ - c_- \rangle \\ &= \frac{1}{m_+} \sum_{i: y_i = 1} \langle \psi(\mathbf{x}), \psi(\mathbf{x}_i) \rangle - \frac{1}{m_-} \sum_{i: y_i = -1} \langle \psi(\mathbf{x}), \psi(\mathbf{x}_i) \rangle \\ &= \frac{1}{m_+} \sum_{i: y_i = 1} K(\mathbf{x}, \mathbf{x}_i) - \frac{1}{m_-} \sum_{i: y_i = -1} K(\mathbf{x}, \mathbf{x}_i) \end{aligned}$$

■

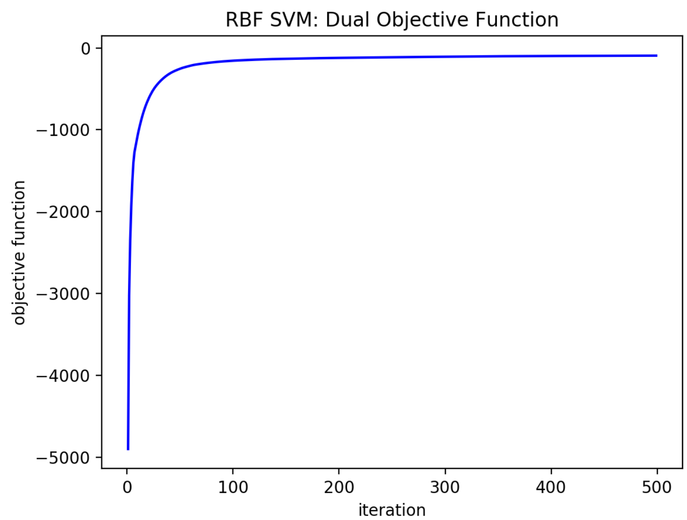
7.
- Training error of scikit-learn Linear SVM: 0.18
  Testing error of scikit-learn Linear SVM: 0.12
- The objective value with respect to the number of iterations during training is



Note that the figure can vary with implementation.

- Training error of LinearSVM: 0.17
  Testing error of LinearSVM: 0.12
  Note that error can vary with implementation.

8. - Training error of scikit-learn RBF SVM: 0.07
     Testing error of scikit-learn RBF SVM: 0.03
   - The dual objective value with respect to the number of iterations during training is



     Note that the figure can vary with implementation.
   - Training error of RBFSVM: 0.02
     Testing error of RBFSVM: 0.03
     Note that error can vary with implementation.