CS489/698: Introduction to Machine Learning

Homework 2

Due: 11:59 pm, October 12, 2017, submit on LEARN.

Include your name, student number and session!

Submit your writeup in pdf and all source code in a zip file (with proper documentation). Write a script for each programming exercise so that the TAs can easily run and verify your results. Make sure your code runs! [Text in square brackets are hints that can be ignored.]

Exercise 1: k-nearest neighbors (30 pts)

Convention: The Matlab notation $X_{:,j}$ means the j-th column of X while $X_{i,:}$ means the i-th row of X.

Algorithm 1: k-nearest neighbors.

Input: $X \in \mathbb{R}^{n\times d}$, $\mathbf{y} \in \{-1,1\}^n$, $Z \in \mathbb{R}^{m\times d}$, distance metric $\mathsf{d}(\cdot,\cdot)$, $k \in \mathbb{N}$
Output: $\hat{\mathbf{y}} \in \{-1,1\}^m, D \in \mathbb{R}^{n\times m}$

1 **for** $t = 1, 2, \ldots, m$ **do**
2 **for** $i = 1, 2, \ldots, n$ **do**
3 compute $D_{i,t} = \mathsf{d}(X_{i,:}, Z_{t,:})$
4 find k smallest entries $t_1, \ldots, t_k$ in $D_{:,t}$ `// break ties arbitrarily`
5 $\hat{y}_t \leftarrow$ majority vote of $y_{t_1}, \ldots, y_{t_k}$ `// break ties arbitrarily`

Implement the k-nearest neighbor algorithm in Algorithm 1, and test it on the MNIST dataset (also available on course website). Specify the distance metric $\mathsf{d}(\cdot,\cdot)$ that you use. To select k, use 10-fold cross-validation but refrain from peeking at the test set. [A naive implementation would be too slow and the memory consumption would be huge. Try to break up the computation and recycle. For Matlab (and maybe python as well), avoid unnecessary for-loops by using matrix-matrix multiplication.] [If it is still too slow to run cross-validation, use a portion of the training set to do the job.] Plot the averaged validation error (y-axis) against k (x-axis), with the "best" k highlighted. After k is selected, report the error rate on the test set. [One could also cross-validate the distance metric if there are a few.]

An $O(n \log n)$ or $O(nk)$ implementation for line 4 will receive full credit but you should try to strike for $O(n)$ time complexity. [Some algorithm from CS341 might be useful.]

[Honesty rule: In your experiment, you can only inspect the test set once—when you actually compute the test error. If you want to try some ideas to improve performance or tune k, split the training set to do so but never peek the test set, please!]

Bonus (5 extra pts). Any implementation that respects the honesty rule and achieves lower than .52% error rate. You can transform the images, enhance the training set, or incorporate any crazy idea you might have, but the classifier has to be based on k-nn. Using a fancy deep neural net or convolutional net or nonlinear SVM or any other fancy classifier is considered cheating.

Exercise 2: Bayes rule and Bayes error (20 pts)

Let (X, Y) be a random sample from some joint distribution, with $X \in \mathcal{X}$ and $Y \in [c] := \{1, \ldots, c\}$. Prove that among all rules $f : \mathcal{X} \to [c]$, the Bayes rule (with ties breaking arbitrarily)

$$f^*(X) = \operatorname*{argmax}_{m=1,\ldots,c} \Pr(Y = m|X) \tag{1}$$

achieves the minimum classification error

$$P^* = \mathbf{E}(P^*(X)),\ P^*(X) = 1 - \max_{m=1,\ldots,c} \Pr(Y = m|X). \tag{2}$$

Exercise 3: ν-Support Vector Machines (50 pts)

Convention: In this exercise, if it helps simplifying your thoughts or presentation, you can always ignore the bias term b.

1. Consider the following "more general" formulation of C-SVM:

$$\min_{\mathbf{w},b} \ \tfrac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^n (\rho - y_i\hat{y}_i)_+, \ \text{where } \hat{y}_i = \mathbf{w}^\top \mathbf{x}_i + b, \tag{3}$$

   $(t)_+ := \max\{t, 0\}$, and $C \geq 0$ is a tuning hyper-parameter. Prove that (3), with any margin parameter $\rho > 0$, can be reduced to the simpler SVM with $\rho = 1$. [You may change the hyper-parameter C.]

   Here by reducing A to B we mean: if we can solve B then we can easily recover a solution for A.

2. Let us now introduce a penalty for ρ above and try to optimize it:

$$\min_{\mathbf{w},b,\rho} \ \tfrac{C}{2}\|\mathbf{w}\|_2^2 - \nu\rho + \sum_{i=1}^n (\rho - y_i\hat{y}_i)_+, \ \text{where } \hat{y}_i = \mathbf{w}^\top \mathbf{x}_i + b. \tag{4}$$

   Use a similar argument as above to prove that we can reduce (4) to the simpler case where $C = 1$. [Note that here ρ is an optimization variable instead of a tuning hyper-parameter.]

3. For the following we fix $C = 1$, with $\nu \geq 0$ being the only tuning hyper-parameter:

$$\min_{\mathbf{w},b,\rho} \ \tfrac{1}{2}\|\mathbf{w}\|_2^2 - \nu\rho + \sum_{i=1}^n (\rho - y_i\hat{y}_i)_+, \ \text{where } \hat{y}_i = \mathbf{w}^\top \mathbf{x}_i + b. \tag{5}$$

   Derive the Lagrangian dual of (5).

4. To understand what is really going on in (5), consider the following subproblem:

$$\min_{\rho} \ -k\rho + \sum_{i=1}^n (\rho - a_i)_+, \tag{6}$$

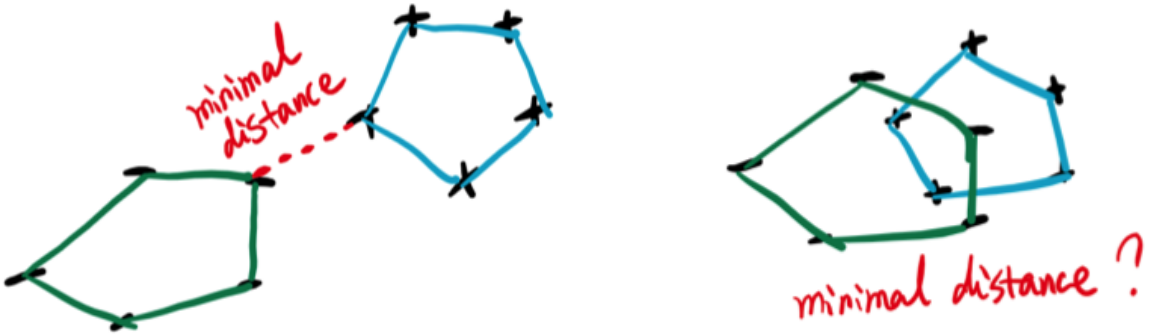
   where $\mathbf{a} \in \mathbb{R}^n$ is given and we have set $\nu = k \in \mathbb{N}$. Solve (6) in closed-form. [Sorting a_i might be helpful. Those familiar with financial risks might recognize CVaR here.]

   This is not required, but if you plug your closed-form solution into (5) you will perhaps find an intuitive explanation of the goal of (5).

5. Consider the following problem, where we take two sets $P := \{\mathbf{x}_i : y_i = 1\}$ and $N := \{\mathbf{x}_i : y_i = -1\}$ and we want to find the minimal distance between the convex hulls of P and N. Recall that any point $\mathbf{z}$ in the convex hull of a set $Q := \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m\}$ can be written as $\mathbf{z} = \sum_{i=1}^m \alpha_i \mathbf{z}_i$ with $\boldsymbol{\alpha} \geq \mathbf{0}$ and $\mathbf{1}^\top \boldsymbol{\alpha} = 1$. Thus, we can formulate the minimal (squared) distance between convex hulls of P and Q as:

$$\min_{\boldsymbol{\alpha} \geq \mathbf{0}, \mathbf{1}^\top \boldsymbol{\alpha} = 1, \boldsymbol{\beta} \geq \mathbf{0}, \mathbf{1}^\top \boldsymbol{\beta} = 1} \ \frac{1}{2}\Big\| \sum_{i:y_i=1} \alpha_i \mathbf{x}_i - \sum_{i:y_i=-1} \beta_i \mathbf{x}_i \Big\|_2^2. \tag{7}$$

   Explain how the dual problem in Ex3.3 is related to the minimal distance (7) above. [The two pictures below might help, or not.]


minimal distance
minimal distance ?