CSCI 567, Fall 2019
Prof. Victor Adamchik

Theory Assignment 3

Due: 11:59 pm, Nov 8th, 2019

Instructions

Submission: Assignment submission will be via `courses.uscden.net`. By the submission date, there will be a folder named 'Theory Assignment 3' set up in which you can submit your files. Please be sure to follow all directions outlined here.

You can submit multiple times, but only *the last submission* counts. That means if you finish some problems and want to submit something first and update later when you finish, that's fine. In fact you are encouraged to do this: that way, if you forget to finish the homework on time or something happens (remember Murphy's Law), you still get credit for whatever you have turned in.

Problem sets must be typewritten or neatly handwritten when submitted. In both cases, your submission must be a single PDF. It is strongly recommended that you typeset with LaTeX. There are many free integrated LaTeX editors that are convenient to use (e.g Overleaf, ShareLaTeX). Choose the one(s) you like the most. This tutorial Getting to Grips with LaTeX is a good start if you do not know how to use LaTeX yet.

Please also follow the rules below:

- The file should be named as `firstname_lastname_USCID.pdf` e.g., `Don_Quijote_de_la_Mancha_8675309045.pdf`).
- Do not have any spaces in your file name when uploading it.
- Please include your name and USCID in the header of your report as well.

Collaboration: You may discuss with your classmates. However, you need to write your own solutions and submit separately. Also in your report, you need to list with whom you have discussed for each problem. Please consult the syllabus for what is and is not acceptable collaboration. Review the rules on academic conduct in the syllabus: a single instance of plagiarism can adversely affect you significantly more than you could stand to gain.

Notes on notation:

- Unless stated otherwise, scalars are denoted by small letter in normal font, vectors are denoted by small letters in bold font and matrices are denoted by capital letters in bold font.
- $\|.\|$ means L2-norm unless specified otherwise i.e. $\|.\| = \|.\|_2$

Problem 1 Principle Component Analysis (25 points)

In this problem, we use proof by induction to show that the M-th principle component corresponds to the M-th eigenvector of $\boldsymbol{X}^T\boldsymbol{X}$ sorted by the eigenvalue from largest to smallest. Here $\boldsymbol{X}$ is the centered data matrix and we denote the sorted eigenvalues as $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d$. In the lecture, the results was proven for $M = 1$. Now suppose the result holds for a value M, and you are going to show that it holds for $M+1$. Note that the $M+1$ principle component corresponds to the solution of the following optimization problem:

$$\max_{\boldsymbol{v}} \quad \boldsymbol{v}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{v} \tag{1}$$

$$\text{s.t.} \quad \|\boldsymbol{v}\|_2 = 1 \tag{2}$$

$$\boldsymbol{v}^T\boldsymbol{v}_i = 0, i = 1, ..., M \tag{3}$$

where $\boldsymbol{v}_i$ is the i-th principle component. Write down the Lagrangian of the optimization problem above, and show that the solution $\boldsymbol{v}_{M+1}$ is an eigenvector of $\boldsymbol{X}^T\boldsymbol{X}$. Then show that the quantity in (1) is maximized when the $\boldsymbol{v}_{M+1}$ is the eigenvector with eigenvalue λ_{M+1}.

Ans: The Lagrangian of 1 is:

$$\mathcal{L}(\boldsymbol{v}; \eta_1, ..., \eta_M, \eta) = \boldsymbol{v}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{v} + \eta(1 - \boldsymbol{v}^T\boldsymbol{v}) + \sum_{i=1}^{M} \eta_i \boldsymbol{v}^T\boldsymbol{v}_i \qquad \textbf{(4 points)}$$

Taking derivative w.r.t $\boldsymbol{v}$ and set it to 0, we get:

$$2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{v} - 2\eta\boldsymbol{v} + \sum_{i=1}^{M} \eta_i \boldsymbol{v}_i = 0 \qquad \textbf{(4 points)}$$

Left Multiply by $\boldsymbol{v}_j$ on both sides, we get:

$$2\boldsymbol{v}_j^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{v} - 2\eta\boldsymbol{v}_j^T\boldsymbol{v} + \sum_{i=1}^{M} \eta_i \boldsymbol{v}_j^T\boldsymbol{v}_i = 0 \qquad \textbf{(3 points)}$$

$$2\boldsymbol{v}_j^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{v} + \eta_j = 0 \qquad (\text{by } \boldsymbol{v}_1, ..., \boldsymbol{v}_M, \boldsymbol{v} \text{ are orthonormal}) \qquad \textbf{(3 points)}$$

$$2\lambda_j\boldsymbol{v}_j^T\boldsymbol{v} + \eta_j = 0 \qquad (\boldsymbol{v}_j \text{ is eigenvector of } \boldsymbol{X}^T\boldsymbol{X}) \qquad \textbf{(3 points)}$$

$$\eta_j = 0 \qquad \textbf{(1 points)}$$

Therefore, $\eta_j = 0, j = 1, ..., M$, and we have: $\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{v} = \eta\boldsymbol{v}$ **(2 points)**

Thus $\boldsymbol{v}$ is an eigenvector of $\boldsymbol{X}^T\boldsymbol{X}$, and the value of equation 1 is $v^T\boldsymbol{X}^T\boldsymbol{X}v = \lambda\boldsymbol{v}^T\boldsymbol{v} = \lambda$, where λ is the eigenvalue of $\boldsymbol{v}$. **(2 points)**

Therefore, to optimize the value, we should let $\boldsymbol{v}$ to be the eigenvector with eigenvalue λ_{M+1}, since $\boldsymbol{v}$ is perpendicular to the first M principle components, which are eigenvectors with eigenvalues $\lambda_1, ..., \lambda_M$. **(3 points)**

Problem 2 Support Vector Regression (30 points)

In this problem, we derive an extension of support vector machine to regression problem, called Support Vector Regression (SVR). Define the regressor $f(\boldsymbol{x}) = \mathbf{w}^T\boldsymbol{\phi}(\boldsymbol{x}) + b$, and given a dataset $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N, y_n \in \mathbb{R}$. Intuitively, we want to find a regressor that has small weight $\mathbf{w}$ and also ensure small approximation error to $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$. The intuition can be formulated as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & |\mathbf{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n) + b - y_n| \le \epsilon \end{aligned}$$

For an arbitrary dataset, the ϵ-close constraint may not be feasible, Therefore, we optimize the "soft" version of the loss above:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^N E_\epsilon(y_n - f(x_n)) \tag{4}$$

E_ϵ is the ϵ-insensitive error function which gives zero error if the difference between prediction and ground truth is smaller than ϵ and incurs linear penalty otherwise. It is defined as follow:

$$E_\epsilon(x) = \begin{cases} 0 & |x| \le \epsilon \\ |x| - \epsilon & |x| > \epsilon \end{cases}$$

Question 1 Reformulate the unconstrained optimization problem in equation 4 as a constraint optimization problem by introducing slack variables for each data points. Hint: For each data point, introduce slack variables $\xi_n \ge 0, \xi_n' \ge 0$ such that $-\epsilon - \xi_n' \le y_n - f(\boldsymbol{x}_n) \le \epsilon + \xi_n$. Then replace E_ϵ with ξ_n, ξ_n'. **(12 points)**
Ans: Minimizing the loss function in equation 4 is equivalent to the following constraint optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^N (\xi_n + \xi_n') && \textbf{(6 points)} \\ \text{s.t.} \quad & \epsilon + f(x_n) - y_n + \xi_n \ge 0 && \textbf{(2 points)} \\ & \epsilon + y_n - f(x_n) + \xi_n' \ge 0 && \textbf{(2 points)} \\ & \xi_n \ge 0, \xi_n' \ge 0 && \textbf{(2 points)} \end{aligned}$$

Question 2 Write down the Lagrangian of the constrained optimization derived in Question 1, then minimize the Lagrangian by taking derivative w.r.t $\mathbf{w}, b, \xi_n, \xi_n'$ and set the gradient to 0, and simplify expressions. Hint: there are no b, ξ_n, ξ_n' in the final expressions. **(18 points)**
Ans: Introduce $\xi = [\xi_1, ..., \xi_n]^T, \xi' = [\xi_1', ..., \xi_n']^T$. The Lagrangian is as follows:

$$\mathcal{L}(\mathbf{w}, b, \xi, \xi') = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^N (\xi_n + \xi_n') - \sum_{n=1}^N (b_n\xi_n + b_n'\xi_n') - \sum_{n=1}^N a_n(\epsilon + \xi_n + f(x_n) - y_n) - \sum_{n=1}^N a_n'(\epsilon + \xi_n' - f(x_n) + y_n)$$

(6 points)

Taking derivative w.r.t $\mathbf{w}, b, \xi_n, \xi'_n$ then gives:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \quad \Longrightarrow \quad \mathbf{w} = \sum_{n=1}^{N} (a_n - a'_n)\boldsymbol{\phi}(\boldsymbol{x}_n) \qquad \textbf{(3 points)}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \quad \Longrightarrow \quad \sum_{n=1}^{N} (a_n - a'_n) = 0 \qquad \textbf{(3 points)}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_n} = 0 \quad \Longrightarrow \quad a_n + b_n = C \qquad \textbf{(3 points)}$$

$$\frac{\partial \mathcal{L}}{\partial \xi'_n} = 0 \quad \Longrightarrow \quad a'_n + b'_n = C \qquad \textbf{(3 points)}$$

Problem 3 Support Vector Machine (25 points)

Consider the dataset consisting of points (x, y), where x is a real value, and $y \in \{-1, 1\}$ is the class label. There are only three points $(x_1, y_1) = (0, 1)$, $(x_2, y_2) = (\frac{\pi}{2}, -1)$, $(x_3, y_3) = (\pi, 1)$. Let the feature mapping $\boldsymbol{\phi}(x) = [\cos x, \sin x]^T$, corresponding to the kernel function $k(x, y) = \cos(x - y)$.

Question 1 Write down the primal and dual formulations of SVM for this dataset in the transformed two-dimensional feature space based on $\boldsymbol{\phi}(\cdot)$. Note that we assume the data points are separable and set the hyperparameter C to be $+\infty$, which forces all slack variables (ξ) in the primal formulation to be 0 (and thus can be removed from the optimization). **(12 points)**

Ans: General primal formulation of SVM for separable data is:

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & y_n[\mathbf{w}^T \boldsymbol{\phi}(x_n) + b] \geq 1, \forall n \end{aligned}$$

Plugging in the specific dataset gives:

$$\begin{aligned} \min_{w_1, w_2, b} \quad & \frac{1}{2}(w_1^2 + w_2^2) && \textbf{(3 points)} \\ \text{s.t.} \quad & w_1 + b \geq 1 && \textbf{(1 points)} \\ & -w_2 - b \geq 1 && \textbf{(1 points)} \\ & -w_1 + b \geq 1 && \textbf{(1 points)} \end{aligned}$$

(**Remark:** Give 2 points for partial credit if the points obtained in plugging in specific dataset is less than 2, but the general primal formulation is present.)

General dual formulation of SVM is:

$$\begin{aligned} \max_{\alpha} \quad & \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n k(x_m, x_n) \\ \text{s.t.} \quad & \alpha_n \geq 0, \forall n \\ & \sum_n a_n y_n = 0 \end{aligned}$$

Plugging in the specific dataset gives:

$$\begin{aligned} \max_{\alpha_1, \alpha_2, \alpha_3 \geq 0} \quad & \alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2}\alpha_1^2 - \frac{1}{2}\alpha_2^2 - \frac{1}{2}\alpha_3^2 + \alpha_1\alpha_3 && \textbf{(2 points)} \\ \text{s.t.} \quad & \alpha_2 = \alpha_1 + \alpha_3 && \textbf{(2 points)} \\ & \alpha_i \geq 0 && \textbf{(2 points)} \end{aligned}$$

(**Remark:** Give 2 points for partial credit if the points obtained in plugging in specific dataset is less than 2, but the general dual formulation is present.)

Question 2 Next, solve the dual formulation. Based on that, derive the primal solution. **(13 points)**

Ans: Eliminating the dependence on α_2 using the constraint $\alpha_1 + \alpha_3 = \alpha_2$, we arrive at the objective

$$\max_{\alpha_1, \alpha_2 \geq 0} \quad 2\alpha_1 - \alpha_1^2 + 2\alpha_2 - \alpha_2^2. \qquad \textbf{(3 points)}$$

Clearly, we can maximize over α_1 and α_2 separately, which gives $\alpha_1^* = \alpha_3^* = 1$ **(2 points)** and thus $\alpha_2^* = 2$ **(2 points)**.

The primal solution can be found by

$$(w_1^*, w_2^*)^T = \sum_{n=1}^{3} y_n \alpha_n^* \boldsymbol{\phi}(x_n) = [0, -2]^T \qquad \textbf{(3 points)}$$

$$b^* = y_1 - \mathbf{w}^{*T} \boldsymbol{\phi}(x_1) = 1 \qquad \text{(using any example works in this case)} \qquad \textbf{(3 points)}$$

Problem 4 Boosting (20 points)

Recall the procedure of AdaBoost algorithm described in class:

Algorithm 1: Adaboost

Given: A training set $\{(\boldsymbol{x}_n, y_n \in \{+1, -1\})\}_{n=1}^N$, and a set of classifier $\mathcal{H}$, where each $h \in \mathcal{H}$ takes a feature vector as input and outputs $+1$ or -1.

Goal: Learn $H(\boldsymbol{x}) = \text{sign}\left(\sum_{t=1}^T \beta_t h_t(\boldsymbol{x})\right)$

Initialization: $D_1(n) = \frac{1}{N}$, $\forall n \in [N]$.

for $t = 1, 2, \cdots, T$ **do**

Find $h_t = \arg\min_{h \in \mathcal{H}} \sum_{n: y_n \neq h(\boldsymbol{x}_n)} D_t(n)$.

Compute

$$\epsilon_t = \sum_{n: y_n \neq h_t(\boldsymbol{x}_n)} D_t(n) \qquad \text{and} \qquad \beta_t = \frac{1}{2} \log \frac{1 - \epsilon_t}{\epsilon_t}.$$

Compute

$$D_{t+1}(n) = \frac{D_t(n) e^{-\beta_t y_n h_t(\boldsymbol{x}_n)}}{\sum_{n'=1}^N D_t(n') e^{-\beta_t y_{n'} h_t(\boldsymbol{x}_{n'})}}$$

for each $n \in [N]$

Question 1 We discussed in class that AdaBoost minimizes the exponential loss greedily. In particular, Adaboost seeks the optimal β_t that minimizes

$$\epsilon_t (e^{\beta_t} - e^{-\beta_t}) + e^{-\beta_t}$$

where ϵ_t is the weighted classification error of h_t and is fixed. Show that $\beta^* = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ is the minimizer. **(8 points)**

Ans: Set the derivative to 0 **(1 points)**:

$$\epsilon_t (e^{\beta_t} + e^{-\beta_t}) - e^{-\beta_t} = 0$$

(3 points)

Multiplying both sides by e^{β_t} and rearranging gives

$$e^{2\beta t} = \frac{1}{\epsilon_t} - 1$$

(2 points)

Solving for β_t finishes the proof. **(2 points)**

(**Remark:** correctly calculate the gradient (3 points). Setting gradient to 0 (1 point) and get the final form of β_t(4 points).)

Question 2 Recall that at round t of AdaBoost, a classifier h_t is obtained and the weighting over the training set is updated from D_t to D_{t+1}. Prove that h_t is only as good as random guessing in terms of classification error weighted by D_{t+1}. That is **(12 points)**

$$\sum_{n:h_t(\mathbf{x}_n)\neq y_n} D_{t+1}(n) = \frac{1}{2}.$$

Hint: you can somehow ignore the denominator of $D_{t+1}(n)$ to simplify calculation.

Ans: By the update algorithm, we have

$$\sum_{n:h_t(\mathbf{x}_n)\neq y_n} D_{t+1}(n) \propto \sum_{n:h_t(\mathbf{x}_n)\neq y_n} D_t(n)e^{\beta t} = \epsilon_t e^{\beta t} = \sqrt{\epsilon_t(1-\epsilon_t)}$$

(3 points)

and similarly

$$\sum_{n:h_t(\mathbf{x}_n)=y_n} D_{t+1}(n) \propto \sum_{n:h_t(\mathbf{x}_n)=y_n} D_t(n)e^{-\beta t} = (1-\epsilon_t)e^{-\beta t} = \sqrt{(1-\epsilon_t)\epsilon_t}$$

(3 points)

Note that $\sum_{n:h_t(\mathbf{x}_n)\neq y_n} D_{t+1}(n) + \sum_{n:h_t(\mathbf{x}_n)=y_n} D_{t+1}(n) = 1$. **(3 points)**
Therefore,

$$\sum_{n:h_t(\mathbf{x}_n)\neq y_n} D_{t+1}(n) = \sum_{n:h_t(\mathbf{x}_n)=y_n} D_{t+1}(n) = \frac{1}{2}$$

(3 points)