CSCI 567, Summer'20
Victor Adamchik

Problem Set #3

Due: July 27, 2020 11:59 pm

Instructions

Submission: Assignment submission will be via `courses.uscden.net`. By the submission date, there will be a folder named `Written 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 online LaTeX editors that are convenient to use (e.g Overleaf). You can also use offline editor such as TeXShop.

Please follow the rules below while submitting:

- The file should be named as Firstname_Lastname_USCID.pdf e.g., Jeff_Dean_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 written 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.

Note 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.

Problem 1 Lagrangian, Duality and Kernel Machines (30 points)

We are given N samples: $\{(\mathbf{x}_1, y_1)\dots(\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathcal{X}, y_i \in \{-1, 1\}$ $\forall i \in \{1 \dots N\}$. We say input $\mathbf{x}_i$ belongs to class $\mathcal{C}_1$ if its label y_i is 1 and it belongs to class $\mathcal{C}_{-1}$ if its label is -1. Mathematically, $\mathcal{C}_1 = \{(\mathbf{x}_i, y_i) : y_i = 1\}$ and $\mathcal{C}_{-1} = \{(\mathbf{x}_i, y_i) : y_i = -1\}$. Now, consider a two class classification problem formulation as follows:

We want to find a separating hyper-plane $\mathbf{w}$ such that if input $\mathbf{x}_i$ belongs to $\mathcal{C}_1$ then $\mathbf{w}^T\mathbf{x}_i \geq 0$ and if it belongs to $\mathcal{C}_{-1}$ then $\mathbf{w}^T\mathbf{x}_i \leq 0$. Therefore, we can find the optimal weights $\mathbf{w}^*$ by maximizing the objective

$$f(\mathbf{w}) = \sum_{i=1}^{N} y_i \mathbf{w}^T \mathbf{x}_i \tag{1}$$

Note that $f(\mathbf{w})$ can be arbitrarily maximized by increasing the magnitude of $\mathbf{w}$ once we have found a vector $\mathbf{w}$ such that $f(\mathbf{w}) > 0$. Therefore, we add an additional constraint that $\mathbf{w}^T\mathbf{w} = \|\mathbf{w}\|^2 \leq 1$.

1.1 Write the down the final constraint minimization problem clearly. Show that under above model, the optimal parameters of the model $\mathbf{w}^*$ follow:

$$\mathbf{w}^* \propto \sum_{i:\mathbf{x}_i \in \mathcal{C}_1} \mathbf{x}_i - \sum_{j:\mathbf{x}_j \in \mathcal{C}_{-1}} \mathbf{x}_j \tag{2}$$

where the constant of proportionality is positive. (10 points)

1.2 Show that to predict the class of a new sample $\mathbf{x}$, we only need to compute R.H.S in eq. 2 from the data and no other information is required from the training data. (4 points)

1.3 Suppose we use a transformation function $\phi : \mathcal{X} \to \mathbb{R}^K$ to transform inputs and the corresponding kernel function is $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}')$. Analogous to problem 1.1, write down the constraint minimization problem for this setup.
Hint: prediction would based on whether $\mathbf{w}^T\phi(\mathbf{x})$ *is greater than or less than zero.* (3 points)

** **1.4** Write down the dual of the optimization problem in 1.3. (7 points)

1.5 Can the optimization problem in 1.3 be kernelized? Also, can you kernelize the prediction rule? Explain why or why not. (6 points)

Problem 2 SVM by hand (20 points)

Consider a dataset with the following 4 data points each with $x \in \mathbb{R}$ and label $y_i \in \{-1, 1\}$:

$$\{(x_i, y_i)\} = \{(-3, 1), (-1, -1), (1, 1), (3, -1)\}$$

2.1 Is the data linearly separable? Explain your reasoning. (2 points)

2.2 Let us map these points to 2D using the transformation $\phi(x) = [x, \sin \frac{\pi}{2} x]$. Compute the kernel function $k(x, x')$. Is the data linearly separable after the transformation? (4 points)

2.3 Using the transformed dataset in 2.2, write down the primal and dual formulation of SVM. (6 points)

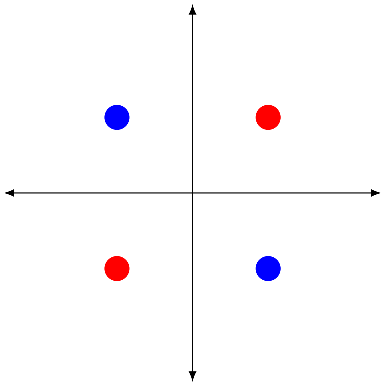
** **2.4** Solve the dual optimization from the previous problem and identify the support vectors. (8 points)
Hint: Use completion of squares to simplify the expressions and assume lagrange multipliers for $i = 1$ and $i = 4$ are same.

Problem 3 Boosting (20 points)

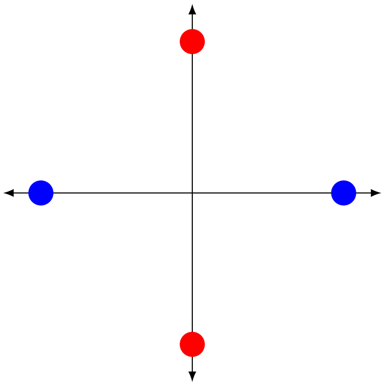
In the lecture, we learnt that we can use boosting to learn a good classifier from an ensemble of weak classifier. In particular, Adaboost algorithm (see algorithm 1), does this by iteratively reweighing the samples and fitting a weak classifier to the new data. The final classifier is weighted ensemble of all the weak classifiers.

Algorithm 1 AdaBoost Algorithm

```
Given: $\mathcal{H}$: A set of functions, where $h \in \mathcal{H}$ takes a D-dimensional vector as input and outputs $+1$ or $-1$
Given: A training set $\{(x_n \in \mathbb{R}^D, y_n \in \{+1, -1\})\}_{n=1}^N$
Goal:Learn $F(x) = sgn(\sum_{t=1}^T \beta_t f_t(x))$, where $f_t \in \mathcal{H}, \beta_t \in \mathbb{R}, sgn(a) = \begin{cases} +1, & \text{if } a \geq 0 \\ -1, & \text{otherwise} \end{cases}$
Initialization: $w_1(n) = 1/N$                                  ▷ Start with equal weights
for $t = 1 \ldots N$ do
    $f_t = \arg\min_{h \in \mathcal{H}} \sum_n w_t(n) \mathbb{I}[y_n \neq h(x_n)]$           ▷ Fit a weak classifier
train classifier
    $\epsilon_t = \sum_n w_t(n) \mathbb{I}[y_n \neq h(x_n)]$                       ▷ Compute the error
calc importance
    $\beta_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$     expression corresponds to min greedy exponential loss
update weights
    $w_{t+1}(n) = \begin{cases} w_t(n) \exp(-\beta_t) & \text{if } y_n = f_t(x_n) \\ w_t(n) \exp(\beta_t) & \text{if } y_n \neq f_t(x_n) \end{cases}$     ▷ Update weights
normalize weights
    $w_{t+1}(n) \leftarrow \frac{w_{t+1}(n)}{\sum_{n'} w_{t+1}(n')}$                     ▷ Normalization
return $F(x) = sgn(\sum_{t=1}^T \beta_t f_t(x))$
```



Data for problem 3.1, 3.2



Data for problem 3.3, 3.4, 3.5

In this problem, we consider weak classifier of following type:

$$h_{s,b,d} = \begin{cases} s & \text{if } x_d > b \\ -s & \text{otherwise} \end{cases}$$

where $s \in \{-1, 1\}, b \in \mathbb{R}, d \in \{1 \ldots D\}$. Such weak classifiers are called decision stumps as they can also be seen as one-level decision tree. Note that for this problem, if you have two classifiers achieving the same error, you can randomly pick any of the two classifiers.

We are given the following data:

$$\mathcal{D} = \{(x_1, y_1) = ([1,1], -1), (x_2, y_2) = ([-1,-1], -1), (x_3, y_3) = ([1,-1], 1), (x_4, y_4) = ([-1,1], 1)\}$$

We want to run adaboost upto $T = 3$ iterations.

3.1 Compute first iteration of adaboost algorithm. Clearly write down f_1, β_1, ϵ_1 and w_2. (4 points)

3.2 Compute second iteration of adaboost algorithm. Clearly write down f_2, β_2, ϵ_2 and w_3. Can you tell the outcome of this adaboost algorithm without doing the third step? (6 points)

For the next problems, we linearly transform the dataset by multiplying with the matrix $\mathbf{W} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$ Under this transformation, the data now becomes:

$$\overline{\mathcal{D}} = \{(x_1, y_1) = ([0,2], -1), (x_2, y_2) = ([0,-2], -1), (x_3, y_3) = ([2,0], 1), (x_4, y_4) = ([-2,0], 1)\}$$

We will run first two iterations of adaboost algorithm on the transformed data.

3.3 Compute first iteration of adaboost algorithm. Clearly write down f_1, β_1, ϵ_1 and w_2. (4 points)

3.4 Compute second iteration of adaboost algorithm. Clearly write down f_2, β_2, ϵ_2 and w_3. (4 points)

3.5 Write down $F(x)$ after two iterations. (2 points)