CSCI 567, Spring 2021
Sirisha Rambhatla

Homework #4

Released: March 5, 2021
Due: 11:59 pm, March 19, 2021
Total scores: 40 points

Instructions

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 Decision Trees **(15 points)**

Consider a binary dataset with 400 examples, where half of them belongs to class A and another half belongs to class B. Next consider two decision stumps (i.e. trees with depth 1) $\mathcal{T}_1$ and $\mathcal{T}_2$, each with two children. For $\mathcal{T}_1$, its left child has 150 examples in class A and 50 examples in class B; for $\mathcal{T}_2$, its left child has 0 example in class A and 100 examples in class B. (You can infer what are in the right child.)

1.1 For *each leaf* of $\mathcal{T}_1$ and $\mathcal{T}_2$, compute the corresponding classification error, entropy (base e) and Gini impurity. **Note**: Prediction for each leaf is the majority class of examples belonging to the leaf. **(10 points)**

Classification error **(2 points: 0.5 per equation points)**

$$\begin{aligned}
\epsilon_{1,L} &= \frac{50}{150+50} = 0.25 \\
\epsilon_{1,R} &= \frac{50}{50+150} = 0.25 \\
\epsilon_{2,L} &= \frac{0}{0+100} = 0 \\
\epsilon_{2,R} &= \frac{100}{200+100} \approx 0.33
\end{aligned}$$

Entropy **(4 points: 1 per equation points)**

$$\begin{aligned}
H_{1,L} &= -\frac{150}{150+50}\ln(\frac{150}{150+50}) - \frac{50}{150+50}\ln(\frac{50}{150+50}) \approx 0.56 \\
H_{1,R} &= -\frac{50}{150+50}\ln(\frac{50}{150+50}) - \frac{150}{150+50}\ln(\frac{150}{150+50}) \approx 0.56 \\
H_{2,L} &= -\frac{0}{0+100}\ln(\frac{0}{0+100}) - \frac{100}{0+100}\ln(\frac{100}{0+100}) = 0 \\
H_{2,R} &= -\frac{200}{200+100}\ln(\frac{200}{200+100}) - \frac{100}{100+200}\ln(\frac{200}{200+100}) \approx 0.64
\end{aligned}$$

Gini impurity **(4 points: 1 per equation points)**

$$
\begin{aligned}
G_{1,L} &= 1 - (\frac{150}{150+50})^2 - (\frac{50}{150+50})^2 = 0.375 \approx 0.38 \\
G_{1,R} &= 1 - (\frac{50}{150+50})^2 - (\frac{150}{150+50})^2 = 0.375 \approx 0.38 \\
G_{2,L} &= 1 - (\frac{0}{0+100})^2 - (\frac{100}{0+100})^2 = 0 \\
G_{2,R} &= 1 - (\frac{200}{200+100})^2 - (\frac{100}{200+100})^2 \approx 0.44
\end{aligned}
$$

1.2 Compare the quality of $\mathcal{T}_1$ and $\mathcal{T}_2$ (that is, the two different splits of the root) based on classification error, conditional entropy (base e), and weighted Gini impurity respectively.

Note: By compare the quality, we mean that in your solution (1) mention what is the value of classification error/ conditional entropy/ weighted Gini impurity for each *tree* $\mathcal{T}_1$, $\mathcal{T}_2$, and (2) which tree is better *for each* measure based on the calculated values. **(5 points)**

Let $p_1 = \frac{150+50}{400} = 0.5$ be the fraction of examples that belong to left leaf of $\mathcal{T}_1$, and $p_2 = \frac{0+100}{400} = 0.25$ be the fraction of examples that belong to left leaf of $\mathcal{T}_2$. Then the total classification error for $\mathcal{T}_1$ and $\mathcal{T}_2$ are respectively:

$$
\begin{aligned}
\epsilon_1 &= p_1\epsilon_{1,L} + (1-p_1)\epsilon_{1,R} = 0.25 \\
\epsilon_2 &= p_2\epsilon_{2,L} + (1-p_2)\epsilon_{2,R} = 0.25
\end{aligned}
$$

So they are as good in terms of classification error. **(1 points)**

The conditional entropy for $\mathcal{T}_1$ and $\mathcal{T}_2$ are respectively:

$$
\begin{aligned}
\epsilon_1 &= p_1 H_{1,L} + (1-p_1)H_{1,R} \approx 0.56 \\
\epsilon_2 &= p_2 H_{2,L} + (1-p_2)H_{2,R} = 0.48
\end{aligned}
$$

So $\mathcal{T}_2$ is better in terms of conditional entropy. **(2 points. Partial credit if only one of the equations is correct, or both equations are correct but conclusion is wrong, or equations are correct but values substituted are wrong of 1 points)**

The weighted Gini impurity for $\mathcal{T}_1$ and $\mathcal{T}_2$ are respectively:

$$
\begin{aligned}
\epsilon_1 &= p_1 G_{1,L} + (1-p_1)G_{1,R} \approx 0.38 \\
\epsilon_2 &= p_2 G_{2,L} + (1-p_2)G_{2,R} \approx 0.33
\end{aligned}
$$

So $\mathcal{T}_2$ is also better in terms of weighted Gini impurity. **(2 points. Partial credit if only one of the equations is correct, or both equations are correct but conclusion is wrong, or equations are correct but values substituted are wrong of 1 points)**

Problem 2 Boosting **(8 points)**

2.1 We discussed in class that AdaBoost minimizes the exponential loss greedily. In particular, the derivation of β_t is by finding the minimizer of $\epsilon_t(e^{\beta_t} - e^{-\beta_t}) + e^{-\beta_t}$ where ϵ_t is the weighted classification error of

h_t and is fixed. Show that $\beta_t^* = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ is the minimizer. **(2 points)**

Set the derivative to 0:

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

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

$$e^{2\beta_t} = \frac{1}{\epsilon_t} - 1.$$

Solving for β_t finishes the proof.

2.2 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}, i.e., **(6 points)**

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

in other words, the update is so that D_{t+1} is the "hardest" weighting for h_t.

By the algorithm we have

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

and similarly

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

Note that $\sum_{n:h_t(\mathbf{x}_n)\neq y_n} D_{t+1}(n) + \sum_{n:h_t(\mathbf{x}_n)= y_n} D_{t+1}(n) = 1$. We thus have

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

Partial credit of 2 points if substantial part of the proof is correct, but the whole proof is not.

Problem 3 Support Vector Machines (17 points)

Consider the dataset consisting of points (x, y), where x is a real value, and $y \in \{-1, 1\}$ is the class label. Let's start with three points $(x_1, y_1) = (-1, -1)$, $(x_2, y_2) = (1, -1)$, $(x_3, y_3) = (0, 1)$.

3.1 Can three points shown in Figure 1, in their current one-dimensional feature space, be perfectly separated with a linear separator? Why or why not? **(1 points)**

No.A 1D linear model is effectively produces a decision boundary $x \geq \theta$ for class (+1) and $<$ for class (-1) for some threshold $\theta \in \mathbb{R}$. It is clear that if we want points x_1 and x_2 to be correctly classified, then x_3 must be incorrectly classified. (Full credit for briefly illustrating this explanation is justified).

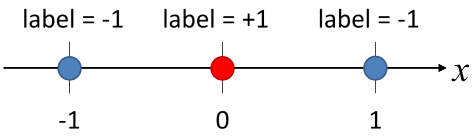
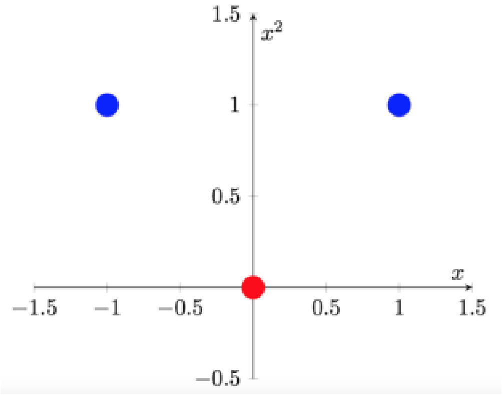


Figure 1: Three data points considered in Problem 3

3.2 Now we define a simple feature mapping $\phi(x) = [x, x^2]^T$ to transform the three points from one- to two-dimensional feature space. Plot the transformed points in the new two-dimensional feature space. Is there a linear decision boundary that can separate the points in this new feature space? Why or why not? **(2 points)**

Yes. Any horizontal line with intercept between 0 and 1 will work to separate them (1 pt). 1 pt for Fig.



3.3 Given the feature mapping $\phi(x) = [x, x^2]^T$, write down the kernel function $k(x, x')$. Moreover, write down the 3×3 kernel (or Gram) matrix $\mathbf{K}$ based on $k(x_i, x_j)$ of the three data points. Verify that $\mathbf{K}$ is a positive semi-definite (PSD) matrix. **(6 points)**

The kernel function is $k(x, x') = \boldsymbol{\phi}(x)^T \boldsymbol{\phi}(x) = xx' + (xx')^2$. The Gram matrix is

$$\mathbf{K} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\forall \mathbf{z} \in \mathbb{R}^3, \qquad \mathbf{z}^T \mathbf{K} \mathbf{z} = 2z_1^2 + 2z_2^2 \geq 0$$

2 points for kernel function. 2 points for Gram matrix K. 2 points for proving positive semi-definite.

3.4 Write down the dual formulation of this problem (plugging in the kernel function values). **(2 points)**

General dual formulation of SVM is:

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

Plugging in the specific dataset gives:

$$\begin{aligned} \max_{\alpha_1,\alpha_2,\alpha_3 \geq 0} \quad & \alpha_1 + \alpha_2 + \alpha_3 - \alpha_1^2 - \alpha_2^2 \\ \text{s.t.} \quad & \alpha_1 + \alpha_2 = \alpha_3 \end{aligned}$$

3.5 Solve the dual form analytically. Then obtain primal solution $\mathbf{w}^*, b^*$ using dual solution. **(4 points)**

Eliminating the dependence on α_3 using the constraint $\alpha_1 + \alpha_2 = \alpha_3$, we arrive at the objective

$$\max_{\alpha_1,\alpha_2 \geq 0} 2\alpha_1 - \alpha_1^2 + 2\alpha_2 - \alpha_2^2.$$

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

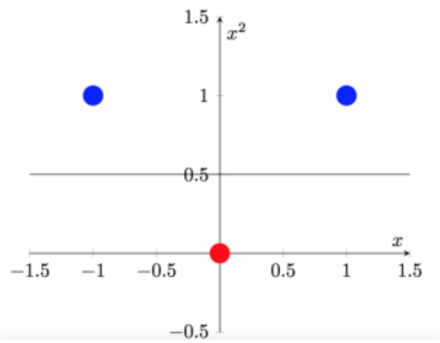
The primal solution can be found by

$$\begin{aligned} (w_1^*, w_2^*)^T &= \sum_{n=1}^{3} y_n \alpha_n^* \boldsymbol{\phi}(\mathbf{x}_n) = (0, -2)^T \\ b^* &= y_1 - \mathbf{w}^{*T} \boldsymbol{\phi}(\mathbf{x}_1) = 1 \qquad \text{(using any example works in this case)} \end{aligned}$$

2 points for the dual solution (1 point for partially correct solutions). Then 1 pt for $\mathbf{w}^*$ and 1 pt for b^*.

3.6 Let $\hat{y} = \mathbf{w}^{*T} \phi(\mathbf{x}) + b^*$ be the SVM prediction. Draw the decision boundary in the new two-dimensional feature space and circle all support vectors. (Set $\hat{y}$ to 0 to get the decision boundary). **(2 points)**

The decision boundary for the two-dimensional space is a horizontal line with intercept $1/2$. Everything above is classified as negative; everything below is classified as positive. All three training points are support vectors.



1 pt for decision boundary. 1 pt for identifying support vectors (data points with $\alpha_n^* > 0$).