CMPSC 448: Machine Learning and Algorithmic AI
Homework 3

Haorui Lyu

October 2022

Logistic Regression

Problem 1. [30 points] Consider a binary training data $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$ where the feature vectors are $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}, i = 1, 2, \ldots, n$. Note that in the lectures we assumed $y_i \in \{-1, +1\}$.

1. Show that
$$\mathbb{P}[y|\boldsymbol{x}; \boldsymbol{w}] = \mathbb{P}[y = 1|\boldsymbol{x}; \boldsymbol{w}]^y \cdot \mathbb{P}[y = 0|\boldsymbol{x}; \boldsymbol{w}]^{(1-y)} \tag{1}$$

2. Following the derivation of logistic regression in lectures, derive the log-likelihood for the training data when the label of each training example is set to be $y_i \in \{0, 1\}$ and sigmoid function is used to covert the linear predictions to probabilities.

3. Then, write down the gradient descent (GD) for obtained optimization problem and discuss the contribution of each training example to updated solution in every iteration of GD. In particular, compare the contribution of a misclassified example with the contribution of a correctly classified example to the gradient.

Solution. 1. We know that

$$\mathbb{P}[y|\boldsymbol{x}; \boldsymbol{w}] = \frac{1}{1+e^{-yw^\top x}}$$

Since for the training data in this problem, $y_i \in \{0, 1\}$, therefore

$$\mathbb{P}[y = 1|\boldsymbol{x}; \boldsymbol{w}] = \sigma(w^\top x) = \frac{1}{1+e^{-w^\top x}}$$
$$\mathbb{P}[y = 0|\boldsymbol{x}; \boldsymbol{w}] = 1 - \sigma(w^\top x) = \frac{e^{-w^\top x}}{1+e^{-w^\top x}}$$

Therefore

$$\begin{aligned}&\mathbb{P}[y = 1|\boldsymbol{x}; \boldsymbol{w}]^y \cdot \mathbb{P}[y = 0|\boldsymbol{x}; \boldsymbol{w}]^{(1-y)} \\ &= \sigma(w^\top x)^y \cdot (1 - \sigma(w^\top x))^{1-y} \\ &= \mathbb{P}[y|\boldsymbol{x}; \boldsymbol{w}]\end{aligned}$$

2. The likelihood for this training data is

$$\prod_{i=1}^n \mathbb{P}[y|\boldsymbol{x}; \boldsymbol{w}]$$

Then take the log of this training data

$$
\begin{aligned}
&log(\textstyle\prod_{i=1}^{n} \mathbb{P}[y_i|\boldsymbol{x_i}; \boldsymbol{w}]) \\
&= \textstyle\sum_{i=1}^{n} log\mathbb{P}[y_i|\boldsymbol{x_i}; \boldsymbol{w}] \\
&= \textstyle\sum_{i=1}^{n} \sigma(w^\top x_i)^{y_i} \cdot (1 - \sigma(w^\top x_i))^{1-y_i} \\
&= \textstyle\sum_{i=1}^{n} y_i log\sigma(w^\top x_I) + (1 - y_i) \cdot log(1 - \sigma(w^\top x_i))
\end{aligned}
$$

The final MLE objective becomes (or minimizing the negative likelihood):

$$
\begin{aligned}
\arg max_w &= \textstyle\sum_{i=1}^{n} y_i log\sigma(w^\top x_i) + (1 - y_i) \cdot log(1 - \sigma(w^\top x_i)) \\
\arg min_w &= -\textstyle\sum_{i=1}^{n} y_i log\sigma(w^\top x_i) + (1 - y_i) \cdot log(1 - \sigma(w^\top x_i))
\end{aligned}
$$

3. We can take the gradient for the equation above. Since we know that $y_i \in \{0, 1\}$ and the gradient descent is to find minimizer. Therefore we use the negative likelihood and take the gradient:

$$
\begin{aligned}
\nabla \arg min_w &= -\textstyle\sum_{i=1}^{n} y_i \nabla log\sigma(w^\top x_i) + (1 - y_i)\nabla log(1 - \sigma(w^\top x_i)) \\
&= -\textstyle\sum_{i=1}^{n} y_i(\frac{e^{-w^\top x_i} x_i}{1+e-w^\top x_i}) + (1 - y)(-\frac{x_i}{1+e^{-w^\top x_i}}) \\
&= -\textstyle\sum_{i=1}^{n} \frac{e^{-w^\top x_i} y_i x_i - x_i(-y_i+1)}{1+e^{-w^\top x_i}} \\
&= -\textstyle\sum_{i=1}^{n} \frac{(e^{-w^\top x_i}+1) y_i x_i - x_i}{1+e^{-w^\top x_i}} \\
&= \textstyle\sum_{i=1}^{n} \sigma(w^\top x_i) x_i - y_i x_i
\end{aligned}
$$

Therefore the gradient descent update function is

$$
\begin{aligned}
w_{t+1} &= w_t - \eta_t \nabla f(w) \\
&= w_t - \eta_t(\tfrac{1}{n} \textstyle\sum_{i=1}^{n} \sigma(w^\top x_i) x_i - y_i x_i)
\end{aligned}
$$

We can slightly modify update function: $w_t - \eta_t(\frac{1}{n} \sum_{i=1}^{n} (\sigma(w^\top x_i) - y_i) x_i)$. Therefore we can find that the x_i is mainly determined by the previous equation $\sigma(w^\top x_i) - y_i$, which is predicted value minus label y_i. Therefore we know that the contribution of a misclassified example to the gradient is greater. □

Decision Trees

Problem 2. [30 points] In this problem, you will investigate building a decision tree for a binary classification problem. The training data is given in Table with 16 instances that will be used to learn a decision tree for predicting whether a mushroom is edible or not based on its attributes (Color, Size, and Shape). Please note the label set is a binary set {Yes, No}.

Instance	Color	Size	Shape	Edible
D1	Yellow	Small	Round	Yes
D2	Yellow	Small	Round	No
D3	Green	Small	Irregular	Yes
D4	Green	Large	Irregular	No
D5	Yellow	Large	Round	Yes
D6	Yellow	Small	Round	Yes
D7	Yellow	Small	Round	Yes
D8	Yellow	Small	Round	Yes
D9	Green	Small	Round	No
D10	Yellow	Large	Round	No
D11	Yellow	Large	Round	Yes
D12	Yellow	Large	Round	No
D13	Yellow	Large	Round	No
D14	Yellow	Large	Round	No
D15	Yellow	Small	Irregular	Yes
D16	Yellow	Large	Irregular	Yes

Table 1: Mushroom data with 16 instances, three categorical features, and binary labels.

1. Which attribute would the algorithm choose to use for the root of the tree. Show the details of your calculations. Recall from lectures that if we let $\mathcal{S}$ denote the data set at current node, A denote the feature with values $v \in \mathcal{V}$, H denote the entropy function, and $\mathcal{S}_v$ denote the subset of $\mathcal{S}$ for which the feature A has the value v, the gain of a split along the feature A, denoted $\mathsf{InfoGain}(S, A)$ is computed as:

$$\text{InfoGain}(\mathcal{S}, A) = H(\mathcal{S}) - \sum_{v \in \mathcal{V}} \left(\frac{|\mathcal{S}_v|}{|\mathcal{S}|} \right) H(\mathcal{S}_v)$$

   That is, we are taking the difference of the entropy before the split, and subtracting off the entropies of each new node after splitting, with an appropriate weight depending on the size of each node.

2. Draw the full decision tree that would be learned for this data (assume no pruning and you stop splitting a leaf node when all samples in the node belong to the same class, i.e., there is no information gain in splitting the node).

Solution. 1. We first calculate $H(S)$

$$H(S) = -\tfrac{9}{16} log_2(\tfrac{9}{16}) + (-\tfrac{7}{16} log_2(\tfrac{7}{16})) = 0.98869$$

Then we calculate entropy for all three roots of the three and find the optimal
case 1: take color to use for the root.

$$H(S|Yellow) = -\frac{8}{13}log_2(\frac{8}{13}) + (-\frac{5}{13}log_2(\frac{5}{13})) = 0.96123$$
$$H(S|Green) = -\frac{1}{3}log_2(\frac{1}{3}) + (-\frac{2}{3}log_2(\frac{2}{3})) = 0.91829$$
$$InforGain(S, Color) = H(S) - H(S|Color) = 0.98869 - (\frac{13}{16}(0.96123) + \frac{3}{16}(0.91829)) = 0.0355$$

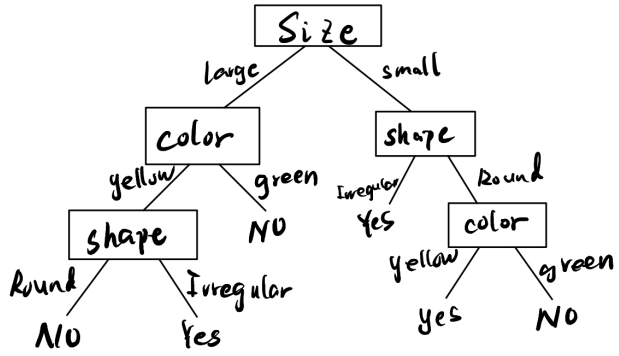
case 2: take size to use for the root.

$$H(S|Small) = -\frac{6}{8}log_2(\frac{6}{8}) + (-\frac{2}{8}log_2(\frac{2}{8})) = 0.81127$$
$$H(S|Large) = -\frac{3}{8}log_2(\frac{3}{8}) + (-\frac{5}{8}log_2(\frac{5}{8})) = 0.95443$$
$$InforGain(S, Size) = H(S) - H(S|Size) = 0.98869 - (\frac{1}{2}(0.81127) + \frac{1}{2}(0.95443)) = 0.106$$

case 3: take Shape to use for the root.

$$H(S|Round) = -\frac{6}{12}log_2(\frac{6}{12}) + (-\frac{6}{12}log_2(\frac{6}{12})) = 1$$
$$H(S|Irregular) = -\frac{3}{4}log_2(\frac{3}{4}) + (-\frac{1}{4}log_2(\frac{1}{4})) = 0.81127$$
$$InforGain(S, Shape) = H(S) - H(S|Shape) = 0.98869 - (\frac{12}{16}(1) + \frac{4}{16}(0.81127)) = 0.0359$$

Then we compare three InforGains and finally choose Size to use for the root.
2. The full decision tree is



□

Problem 3. [10 points] Handling real valued (numerical) features is totally different from categorical features in splitting nodes. This problem intends to discuss a simple way to decide good thresholds for splitting based on numerical features. Specifically, when there is a numerical feature in data, an option would be treating all numeric values of feature as discrete, i.e., proceeding exactly as we do with categorical data. What problems may arise when we use a tree derived this way to classify an unseen example?

Solution. It will cause overfitting since we proceed the data exactly as we do with categorical data. For categorical variable decision tree includes categorical target variables that are divided into categories. For example, the categories can be yes or no. The categories mean that every stage of the decision process falls into one category, and there are no in-betweens. Therefore that will be more node in the decision tree and cause overfitting. Meanwhile, the values in some unseen examples may not be in the tree, which means that the decision tree cannot classify these examples. So the decision tree may fail. □

Support Vector Machines

Problem 4. [30 points] Consider a data set with three data points in $\mathbb{R}^2$

$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 \\ 0 & -1 \\ -2 & 0 \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \end{bmatrix}$$

Manually solve the following optimization problem for hard-margin SVM stated as

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, i = 1, 2, \ldots, n \end{aligned}$$

to get the optimal hyperplane $(\boldsymbol{w}_*, b_*)$ and its margin

Solution. We implement constrain by $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, i = 1, 2, \ldots, n$

$$\begin{gathered} -1(0w_1 + 0w_2 + b) \geq 1 \\ -1(0w_1 - 1w_2 + b) \geq 1 \\ 1(-2w_1 + 0w_2 + b) \geq 1 \end{gathered}$$

By solve this constrain, we get

$$\begin{gathered} w_2 - b \geq 1 \\ w_2 \geq b + 1 \end{gathered}$$

and

$$\begin{gathered} -2w_1 + b \geq 1 \\ -2w_1 \geq 1 - b \\ w_1 \leq \tfrac{b-1}{2} \end{gathered}$$

Since we know that $-b \geq 1$, therefore $b \leq -1$, $\frac{b-1}{2} \leq -1, b + 1 \leq 0$.
Then we have

$$\begin{gathered} b \leq -1 \\ w_2 \geq b + 1 \\ w_1 \leq \tfrac{b-1}{2} \leq -1 \end{gathered}$$

For this problem we want to minimize $\frac{1}{2}\|\boldsymbol{w}\|_2^2$ which is $w_1^2 + w_2^2$. Therefore let $w_1 = -1, w_2 = 0, b = -1$.
Final result: can maximize margin by minimizing $\|\boldsymbol{w}\|_2^2$, then the margin $\frac{1}{\|\boldsymbol{w}\|_2^2} = \frac{1}{1+0} = 1$. □