

## CS534 — Midterm

**Name:**

### 1. (Perceptron)

- a. (6pt) Define the hinge loss objective function that is optimized by perceptron. What was the reason for using the hinge loss instead of 0/1 loss?

*Hinge loss:*

$$J(w) = \frac{1}{N} \sum_{i=1}^N \max(0, -y_i w \cdot x_i)$$

*0/1 loss function is piece-wise constant, thus gradient descent optimization can not be applied to optimize 0/1 loss.*

- b. (6pt) What is the key difference between the batch learning and online learning algorithms for perceptron? Which one is more sensitive to the order that training examples are received?

*Batch learning updates the weight vector after seeing all training examples, whereas the online algorithm takes an update every time a mistake is made. Online learning is more sensitive.*

- c. (6pt) When the training data is not linearly separable, we can apply the voted perceptron algorithm, where we store all intermediate linear separators ( $\mathbf{w}$ 's) and take a weighted vote as described by the following function:

$$h(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^N c_i \cdot \text{sign}(\mathbf{w}_i \cdot \mathbf{x})\right)$$

Please explain how  $c_i$  is computed, and what is the rationale behind using  $c_i$  to weight the decision of each  $\mathbf{w}_i$ .

*$c_i$  is the “survival time” of the weight vector  $w_i$ , which measures how many training examples  $w_i$  was able to correctly classify before being updated due to a mistake.*

*Rationale behind using  $c_i$  to weight the vote of  $w_i$ : larger  $c_i$  suggests higher accuracy of  $w_i$ .*

2. (Generative vs. discriminative classifier)

- a. (6pts) What is the key distinction between generative and discriminative methods? Provide an example for each class of methods.

*Generative: learns  $P(X, y)$  ( $P(X|y)$  and  $p(y)$ ) - e.g. Naive Bayes*

*Discriminative: learns  $P(y|X)$  - e.g., Logistic regression*

- b. (6pts) In class, we showed that under the LDA model (i.e., Gaussian class conditional distributions, shared covariance matrix), we can show that  $P(y|X)$  can be represented in the form of

$$\frac{1}{1 + \exp(\theta^T \mathbf{x})}$$

where  $\theta$  is some functions of LDA model parameters. Note that this form is identical to what is assumed by Logistic regression. Does this mean that both methods make the same modeling assumptions? If your answer is no, which one makes stronger modeling assumptions (more restrictive, harder to satisfy)?

*NO they are not the same, LDA make stronger modeling assumptions.*

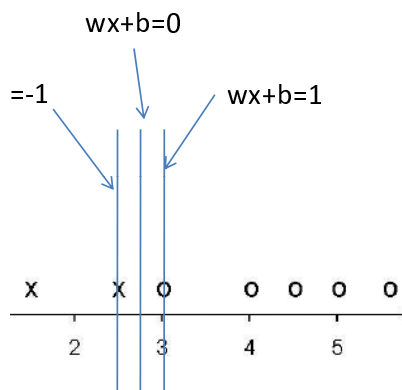
3. (Support Vector Machines)

- a. (6pts) In class we introduced two notions of margin: the *functional margin* and the *geometric margin*. When deriving the SVM optimization problem why did we choose to maximize the geometric margin rather than the functional margin? Specifically, what would go wrong if we simply tried to maximize the functional margin and how does maximizing the geometric margin correct this problem?

*Functional margin can be scaled arbitrarily without changing the decision boundary itself. Geometric margin measures the distance between the training examples to the decision boundary, thus will not be influenced by scaling factors.*

- b. (6pts) **SVM.** Consider the following one dimensional data set. 'x' denotes negative examples and 'o' denotes positive examples. The exact location of the data points and their class labels are given in the table under the figure.

Circle the support vectors of this data set and mark the decision boundary as well as the  $w \cdot x + b = 1$  and  $w \cdot x + b = -1$  lines.



4. Short questions.

- a. (4pts) Explain what does it mean when we overfit the training data?  
*Overfitting happens when a learning algorithm fits to the peculiarities of the training data, and performs poorly on unseen data.*
- b. (6pts) Explain how each of the following methods can lead to overfitting:
- \* Decision tree: *growing the decision tree until every leaf node contains only examples of one class.*
  - \* Nearest neighbor: *k=1*
  - \* Neural networks: *large number of hidden units or overtraining*
- c. (6pts) In decision tree learning, a categorical feature with more than two possible values can be used to create a multi-way split, where each possible value of the feature leads to one branch. Explain why such features tend to be favored when comparing against binary features using the information gain criterion. Suggest a solution to the above problem.  
*Higher branching factor will generally lead to lower uncertainty in the leaf nodes, thus higher mutual information. To fix this bias, we can either normalize the mutual information with the entropy of the feature, or test on one possible value.*
- d. (6pts) What is the Naive Bayes assumption? Consider the following data set with two input features (*temperature* and *season*). Is the naive bayes assumption satisfied for this problem?

Temperature	Season	Electricity Usage (Class)
Below Average	Winter	High
Above Average	Winter	Low
Below Average	Summer	Low
Above Average	Summer	High

*Naive Bayes assumption: features are conditionally independent given class label (note that conditional independence  $\neq$  independence. Also  $x_1 \perp x_2$  does not imply conditional independence between  $x_1$  and  $x_2$ )*

*For this problem, the assumption does not hold. Because for electricity usage to be high, if  $S=winter$ , the temperature needs to be below average, whereas if  $S=summer$ , then temperature needs to be above average. They are clearly correlated given class label.*

5. **(Naive Bayes)** Consider the following training data set with class label  $Y$  and input features  $X_1$ ,  $X_2$ , and  $X_3$ .

$X_1$	$X_2$	$X_3$	Y
1	1	1	0
1	1	0	0
0	0	0	0
0	1	0	1
1	0	1	1
0	1	1	1

- a. [10pt] Please apply the Naive Bayes classifier without Laplace smoothing to this data set and compute  $P(y = 1|X)$  for  $X = (1, 0, 0)$ . Show your work.

$$p(y = 1|X) = \frac{p(1, 0, 0|y = 1)p(y)}{p(X)} = \frac{\frac{1}{3} \frac{1}{3} \frac{1}{3} \frac{1}{2}}{p(X)} = \frac{1}{54 * P(X)}$$

$$p(y = 0|X) = \frac{p(1, 0, 0|y = 0)p(y)}{p(X)} = \frac{\frac{2}{3} \frac{1}{3} \frac{2}{3} \frac{1}{2}}{p(X)} = \frac{4}{54 * P(X)}$$

Note that the above two needs to add up to 1, thus we have  $p(y = 1|X) = 1/5$ .

- b. [5pt] Given the following cost matrix, what is the optimal prediction for the given  $X$ ? (Note that Y is the true label and  $\hat{Y}$  is the predicted label.)

	Y = 1	Y = 0
$\hat{Y}=1$	0	1
$\hat{Y}=0$	5	0

*Expected loss for  $\hat{y} = 1$ :  $0 * P(y = 1|X) + 1 * P(y = 0|X) = 4/5$*

*Expected loss for  $\hat{y} = 0$ :  $5 * P(y = 1|X) + 0 * P(y = 0|X) = 5/5 = 1$*

*Thus we predict  $\hat{y} = 1$  because it minimizes expected loss.*

- c. [5pt] What is the potential problem with using a Naive Bayes classifier that was learned without Laplace smoothing?

*Given limited training data, we could end up with zero probabilities for some  $p(x|y)$ , leading to extreme or uninformed probabilities for  $p(y|X)$ .*