Theory Assignment 1 (Practice Version)

COMP 451 - Fundamentals of Machine Learning

Prof. William L. Hamilton

Winter 2021

Question 1 [6 points]

Recall that the k-NN model is defined by the prediction function

$$f_{\text{k-NN}}(\mathbf{x}) = \texttt{MAJ}\left(\{y_i : (\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{trn}} \wedge \exists_{<k}(y_j, \mathbf{x}_j) \in \mathcal{D}_{\text{trn}} : d(\mathbf{x}, \mathbf{x}_i) > d(\mathbf{x}, \mathbf{x}_j)\}\right), \tag{1}$$

where `MAJ` is the majority vote function. Assume that we are using the Euclidean distance function, considering a binary 0-1 classification task with two-dimensional features, and that we are evaluating accuracy using the 0-1 loss:

$$L(y, \hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

Lastly, assume that ties in the majority vote function are broken randomly with a 50/50 probability (i.e., if we have equal positive and negative classes in the nearest neighbor set, then we flip a coin to make the prediction), and assume that we are evaluating the expected accuracy in light of this randomness.

Prove or provide a counter-example to the following claim: if we assume that our dataset is linearly separable with geometric margin γ, then the expected training error of a k-NN monotonically increases as a function of k for $k \geq 1$.

Hint: Remember that the nearest neighbor of a training point is always itself!

Solution. *The claim is false. For example, suppose we have the following training dataset:*

point 1:	$([0, 1], 1)$
point 2:	$([0, 2.1], 1)$
point 3:	$([0, 2.1], 1)$
point 4:	$([0, 0], 0)$
point 5:	$([0, -0.2], 0)$
point 6:	$([0, -0.2], 0)$

Now, for $k = 2$ we will have that point 1 is misclassified with 50% probability, since its two nearest neighbors are itself and point 4. All other points will be correctly classified: points 2/3 and points 5/6 are identical and from the same class, guaranteeing correct classification; point 4 is correctly classified because its nearest neighbors are points 5/6. Moving up to $k = 3$, point 1 is now correctly classified, since points 2/3 are included in its set of nearest neighbors. All the other points remain correctly classified, since the points added to their nearest neighbor set are from the same class.

Question 2 [6 points]

For this question, you should refer to the details and notation for the perceptron algorithm (i.e., Algorithm 1) in Chapter 3 of the notes. Provide a proof for the following lemma, which we we used to prove the perceptron convergence theorem:

Lemma 1. *Assume that there exists some $\gamma > 0$ and some set of optimal parameters $\mathbf{w}^*$ such that $y_i(\mathbf{w}^*)^\top \mathbf{x}_i \geq \gamma$ for all $(\mathbf{x}_i, y_i) \in \mathcal{D}_{trn}$. The norm of the weight vector $\|\mathbf{w}^{(k)}\|$ increases at most linearly with each update in Algorithm 1. In particular, if assume that $\|\mathbf{x}_i\| < R, \forall i \in \mathcal{D}_{trn}$, then $\|\mathbf{w}^{(k)}\|^2 \leq R^2 k$, where k denotes the number of updates in Algorithm 1.*

Solution. *By the definition of the perceptron update on a point $(\mathbf{x}, y)$ we have that*

$$\begin{aligned}
\|\mathbf{w}^{(k)}\|^2 &= (\mathbf{w}^{(k)})^\top \mathbf{w}^{(k)} && (3)\\
&= \left(\mathbf{w}^{(k-1)} + y\mathbf{x}\right)^\top \left(\mathbf{w}^{(k-1)} + y\mathbf{x}\right) && (4)\\
&= (\mathbf{w}^{(k-1)})^\top \mathbf{w}^{(k-1)} + 2y\mathbf{x}^\top \mathbf{w}^{(k-1)} + y^2 \mathbf{x}^\top \mathbf{x} && (5)\\
&\leq (\mathbf{w}^{(k-1)})^\top \mathbf{w}^{(k-1)} + \mathbf{x}^\top \mathbf{x} && (6)\\
&\leq \|\mathbf{w}^{(k-1)}\|^2 + R^2 && (7)
\end{aligned}$$

Note that we only make updates when we make mistakes, so we can safely assume that $2y\mathbf{x}^\top \mathbf{w}^{(k-1)} < 0$, since y and $\mathbf{x}^\top \mathbf{w}^{(k-1)}$ must have opposite signs. The proof is completed by simple induction on k. The inductive step is given by Equation 7 and the base case for $k = 0$ is given by $\|\mathbf{w}^{(0)}\|^2 = \|\mathbf{0}\|^2 = 0 = 0R^2$.

Question 3 [6 points]

In class, we were introduced to Bernoulli Naive Bayes and the Gaussian Naive Bayes models. In this question, you will derive that maximum likelihood parameters for a Poisson Naive Bayes model. In a Poisson Naive Bayes model, the feature likelihoods are defined following distribution:

$$p(\mathbf{x}[j] \mid y = k) = \frac{\theta_{j,k}^{\mathbf{x}[j]} e^{-\theta_{j,k}}}{\mathbf{x}[j]!}. \tag{8}$$

As in the Bernoulli Naive Bayes model, the $\theta_{j,k}$ parameter determines the likelihood for the jth feature, assuming the point belongs to class k.

Part 1 [2 points]

Assume we are in a binary classification setting. Write an expression for the log-odds ratio of the Poisson Naive Bayes model. Use the notation from Equation 8 above and use $\theta_k = P(y = k)$ to denote the estimated class likelihoods.

Part 2 [4 points]

Derive the maximum likelihood estimates for the Poisson Naive Bayes parameters, i.e., give maximum likelihood estimates for the $\theta_{j,k}$ parameters.

Solution.

Part 1

The log-odds ratio is given by

$$\frac{\log(p(y = 1 \mid \mathbf{x})}{\log(p(y = 0 \mid \mathbf{x})} = \log(\theta_1) - \log(\theta_0) + \sum_{j=1}^{m} \mathbf{x}[j] \left(\log(\theta_{j,1}) - \log(\theta_{j,0})\right) - \theta_{j,1} + \theta_{j,0}$$

Part 2

To derive the maximum likelihood estimates for the $\theta_{j,k}$ parameters, we only need to consider the parts of the log-likelihood that depend on the $\theta_{j,k}$ term. All other terms will be zero. Moreover, without loss of generality we assume that $k = 1$. Given these simplifications we have that

$$\begin{aligned}\frac{\partial}{\partial \theta_{j,1}} \log \mathcal{L}(\mathcal{D}; \Theta) &= \frac{\partial}{\partial \theta_{j,1}} \sum_{(\mathbf{x},y)\in\mathcal{D}} y(\mathbf{x}[j] \log(\theta_{j,1}) - \theta_{j,1}) \\ &= \sum_{(\mathbf{x},y)\in\mathcal{D}} y \left(\frac{\mathbf{x}[j]}{\theta_{j,1}} - 1\right),\end{aligned}$$

and setting this to zero and solving we get

$$\sum_{(\mathbf{x},y)\in\mathcal{D}} y\left(\frac{\mathbf{x}[j]}{\theta_{j,1}} - 1\right) = 0$$

$$\sum_{(\mathbf{x},y)\in\mathcal{D}} y\left(\frac{\mathbf{x}[j]}{\theta_{j,1}}\right) = \sum_{(\mathbf{x},y)\in\mathcal{D}} y$$

$$\theta_{j,1} = \frac{\sum_{(\mathbf{x},y)\in\mathcal{D}} y\mathbf{x}}{\sum_{(\mathbf{x},y)\in\mathcal{D}} y}$$

And in general we get that

$$\theta_{j,k} = \frac{\sum_{(\mathbf{x},y)\in\mathcal{D}:y=k} \mathbf{x}[j]}{|(\mathbf{x},y)\in\mathcal{D} : y = k|}. \tag{9}$$

In other words, we just take the average value of the feature for points belonging to class k.

Question 4 [short answers; 2 points each]

Answer each question with 1-3 sentences for justification, potentially with equations/examples for support.

a) True or false: Bernoulli Naive Bayes always correctly classifies all training points if the dataset is linearly separable.

b) Consider the following dataset:

point 1:	$([0, 1], 1)$
point 2:	$([1, 0], 1)$
point 3:	$([0, 0], -1)$
point 4:	$([1, 1], -1)$

Is the perceptron algorithm guaranteed to converge on this dataset?

c) Consider the following dataset:

point 1:	$([0.5, 1], 1)$
point 2:	$([0.2, 0.5], 1)$
point 3:	$([0.9, 0.9], 0)$
point 4:	$([1.5, 1.5], 0)$

What class will a Gaussian Naive Bayes model predict for point $[1.1, 1.1]$?

Solution.

a) *This is false. If there are many more points from one class then the class priors can lead to a misclassification even if the data is separable. For example, suppose we have a dataset consisting of 5 identical points* $([1, 0], 1)$, *5 identical points* $([0, 1], 1)$, *and one point* $([1, 1], 0)$. *This dataset is linearly separable, but the model will predict class 1 for the training point* $([1, 1], 0)$, *since* $0.5 \times 0.5 \times \frac{10}{11} > 1 \times 1 \times \frac{1}{11}$.

b) *The data is not linearly separable. (It is the exclusive-or function). Thus, the perceptron is not guaranteed to converge.*

c) *The GNB model will predict class 0, since the input point is closer to the mean of the points from class 0.*