CS4780 Midterm

Spring 2017

NAME:	
Net ID:	
Email:	

1 [??] General Machine Learning

Please identify if these statements are either True or False. Please justify your answer **if false**. Correct "True" questions yield 1 point. Correct "False" questions yield two points, one for the answer and one for the justification.

1. (**T/F**) Naïve Bayes makes the assumption that the individual features are independent, i.e. $p(\mathbf{x}) = \prod_\alpha p([\mathbf{x}]_\alpha)$.
   False, it assumes that they are conditionally independent, i.e. $P(\mathbf{x}|y) = \prod_\alpha p([\mathbf{x}]_\alpha|y)$.

2. (**T/F**) The Perceptron classifier provably converges after a finite number of iterations in all cases.
   False, only if the data is linearly separable.

3. (**T/F**) The best machine learning algorithm make no assumptions about the data.
   False, ML algorithms always make assumptions about the data.

4. (**T/F**) The higher dimensional the data, the less likely the Perceptron is to converge.
   False, it is more likely that there is a linear separating hyperplane if the data is high dimensional.

5. (**T/F**) If MAP is used with a prior that has non-zero support everywhere, it will converge to the same result as MLE, as $n \to \infty$ (where n is the number of training samples).
True.

6. (**T/F**) As $n \to \infty$, the error of the Perceptron classifier is at most twice the error of the Optimal Bayes classifier (where n is the number of training samples).
False, this is true for the 1-nearest neighbor classifier.

7. (**T/F**) The SVM classifier is just a fancy way of saying you use the logistic loss (aka as log-loss) with an $l2$-regularizer.
False, it uses the hinge-loss.

8. (**T/F**) The Naïve Bayes classifier is a discriminative classification algorithm.
False, it is generative.

9. (**T/F**) One advantage of Adagrad is that each dimension has effectively its own separate learning rate, which is set automatically.
True.

10. (**T/F**) If the Naive Bayes assumption holds, the Naive Bayes classifier becomes identical to the Bayes Optimal classifier.
True.

11. (**T/F**) MAP estimation with a Beta prior to estimate the probability of a coin leading to heads, is identical to "halluzinating" coin toss results.
True.

12. (**T/F**) In practice people often create *Train* and *Validation* sets out of their original data D. Each point $x_i \in D$ is placed either into $Train$, $Validation$ or into both.
False, each point is only placed into one of the two sets.

13. (**T/F**) As the validation set becomes extremely large, the validation error approaches the test error.
True.

14. (**T/F**) *Generalization error* is just another word for *validation error*.
False, the generalization error is the *expected* error obtained on new data drawn from the same data distribution - the validation error is the *empirical* error on the validation set.

15. (**T/F**) If a data set is linearly separable, the Perceptron and the SVM will often learn identical hyperplanes.
False, this will almost surely never happen - the Perceptron outputs a hyperplane the SVM always outputs the one maximizing the margin.

16. (**T/F**) If you were to use the "true" Bayesian way of machine learning you would put a prior over the possible models and draw one randomly during training.
False, you would integrate out all possible models.

2 [16] Short Questions

1. (3) Assume you observed n draws $x_1, \ldots, x_n$. State the log-likelihood function, and estimate the parameters for the distrubion

$$f(x; \lambda) = \lambda e^{-\lambda x}$$

using MLE.
The log-likelihood is

$$\log p(x_1, \ldots, x_n | \lambda) = -\lambda \sum_{i=1}^{n} x_i + n \log \lambda.$$

Taking the derivative with respect to λ and setting it to 0, we get

$$0 = -\sum_{i=1}^{n} x_i + \frac{n}{\lambda}.$$

Rearrange to get the MLE solution $\lambda = \left(\frac{1}{n} \sum_{i=1}^{n} x_i\right)^{-1}$.

2. (3) Why would someone chose l_1 over l_2 regularization, and vice versa?
There are many ways to answer this question. Here is one
l_1: Advantage: Less sensitive to noise; Disadvantage: Not differentiable at 0
l_2: Advantage: Differentiable everywhere; Disadvantage: Somewhat sensitive to outliers/noise

3. (4) Recall the linear regression model

$$y_i = \mathbf{x}_i^\top \mathbf{w} + \epsilon_i \tag{1}$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ are input features, $\mathbf{w} \in \mathbb{R}^d$ is the model parameter and $\epsilon_i \sim N(0, \sigma^2)$ are iid Gaussian noise. Write down the loss function $\ell(\mathbf{w})$ whose minimizer results in the maximum a posteriori (MAP) estimate with Gaussian prior for the parameter vector $\mathbf{w}$, i.e. the entries of $\mathbf{w}$ are iid zero-mean Gaussians. Is the solution of this loss function unique if $d \gg n$?

$$\ell(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i^\top \mathbf{w} - y_i)^2 + \lambda\|\mathbf{w}\|_2^2$$

for some regularization parameter $\lambda > 0$. This loss function is strictly convex, hence it always admits a unique solution.

4. The newly elected president to the international machine learning society just announced some *awesome* simplifications to ERM.

   - From now on losses are not computed over the entire training data sets, but sample by sample.
   - All labels must be chosen from $\{-1, +1\}$
   - The only valid loss function is: $\ell(\mathbf{x}, y; \mathbf{w}) = \max(0, -y\mathbf{w}^\top \mathbf{x})$.
   - The learning rate is always 1.

   To be precise, his new version of empirical risk minimization becomes:

   1: Shuffle the data set D randomly and set $\mathbf{w} = \vec{0}$
   2: for $(\mathbf{x}_i, y_i) \in D$:
      3: compute gradient $\mathbf{g} = \frac{\partial \ell(\mathbf{x}_i, y_i)}{\partial \mathbf{w}}$
      4: $\mathbf{w} \leftarrow \mathbf{w} - \mathbf{g}$
   5: Goto 1: until convergence

   The community is excited! Clearly this algorithmic framework is much simpler and more elegant. How is it possible that nobody has thought of this before?

   (a) (3) Compute the gradient $\mathbf{g}$ as a function of $\mathbf{x}, y, \mathbf{w}$.

   $$\mathbf{g} = -\delta(y_i \mathbf{w}^\top \mathbf{x}_i < 0) y_i \mathbf{x}_i$$

   (b) (3) Imagine you have a binary data set D with $\mathbf{x}_i \in \mathcal{R}^d$. There exists a hyperplane $\mathbf{w}^*, b^*$, with $b^* \neq 0$ that perfectly separates the two classes. Would this algorithm be guaranteed to find a separating hyperplane? If not, what modification(s) would you have to make?
   No. There might not exist a hyperplane that goes through the origin that perfectly separates the two classes. A simple modification would be to add the constant 1 feature to $\mathbf{x}$, hence the last coordinate of $\mathbf{w}$ encodes the bias term. With this modification, the gradient is exactly the update rule for Perceptron, hence the ERM algorithm always finds the separating hyperplane if it exists.

3 [14] kNN

1. (1) What is the modeling assumption of kNN?
   Points that are close have similar label.

2. (4) Suppose there are n training points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. What is the time complexity of 1-NN (with $k = 1$) using the Euclidean distance during testing time (with a single test point) in terms of n and d? Explain.
   We need to compute the distance between the test point with every $\mathbf{x}_i$. Each computation involves d subtractions, d multiplications, $d - 1$ additions and 1 square root operation. The total time complexity is $O(dn)$.

3. (4) Suppose your features are age, number of years of education, and annual income in US dollars for an individual. Why is the Euclidean distance not a good measure of dissimilarity? Propose an alternative distance metric and explain why it solves the problem.
   The features are on a very different scale. Age is between 0 and 100 while annual income is in the 10,000s. Hence the Euclidean distance is dominated by annual income. One way around this is by scaling every feature appropriately, such as normalizing them to $[0, 1]$.

4. (5) Briefly describe the curse of dimensionality. How does it affect the kNN classifier? Why can k-NN still function well on some high dimensional data sets (such as images of faces or handwritten digits)?
   As the dimensionality of the data d increases, points in $[0,1]^d$ become very far apart in Euclidean distance. Suppose you have n points uniformly sampled in $[0,1]^d$. To find the k nearest neighbors of a test point, you need a hypercube of length $l \approx \left(\frac{k}{n}\right)^{1/d}$, which becomes very close to 1 when d is large. Hence the entire feature space is needed to find the k nearest neighbors. As $d >> 0$, all points become equidistant, violating the kNN modeling assumption.
   This is not a problem for natural images since the intrinsic dimensionality of the data is still low, as the feature vector coordinates (i.e. pixels) are very highly correlated.

4 [17] Naive Bayes

You're building a spam classifier and you've compiled counts for certain keywords in both spam and authentic messages. You've also assembled a set of categorical features for those same messages. These two data sets are given below. (The $\mathbf{x}^c$ vectors are word counts, and the $\mathbf{x}^d$ vectors hold discrete categorical features within $\{a, b\}$.)

Training data:

$$\begin{aligned} \mathbf{x}_1^c &= [0, 4]^\top & \mathbf{x}_1^d &= [b, a]^\top & y_1 &= +1 \\ \mathbf{x}_2^c &= [0, 2]^\top & \mathbf{x}_2^d &= [a, a]^\top & y_2 &= +1 \\ \mathbf{x}_3^c &= [2, 0]^\top & \mathbf{x}_3^d &= [a, b]^\top & y_3 &= -1 \\ \mathbf{x}_4^c &= [1, 1]^\top & \mathbf{x}_4^d &= [a, b]^\top & y_4 &= -1 \end{aligned} \tag{2}$$

1. (2) You decide to model the discrete categorial features using Laplace (aka +1) smoothing. What is the probability for a test point $P([\mathbf{x}_t^d]_2 = a | y_t = -1)$, where $[\mathbf{x}_t^d]_2$ corresponds to the second feature value of $\mathbf{x}_t^d$?

   $P([\mathbf{x}_t^d]_2 = a | y_t = -1) = \frac{1+\Sigma_{i=1}^4 I([\mathbf{x}_i]=a \cap y_i=-1)}{2*1+\Sigma_{i=1}^4 I(y_i=-1)} = \frac{1+0}{2+2} = \frac{1}{4}$

2. (5) For the following test point with discrete categorical features

$$\mathbf{x}_t^d = [a, b] \tag{3}$$

   what is the posterior ratio

$$\frac{P(y_t = -1 | \mathbf{x}_t^d)}{P(y_t = +1 | \mathbf{x}_t^d)} \tag{4}$$

   (Hint: Note that you just need to compute the ratio and not the individual probabilities. You also do not need to reduce the fraction to the simplest form.)

   Because we are computing the posterior ratio, we can ignore the prior terms $P(y_t = 1)$ and $P(y_t = -1)$, since these values are the same. Thus: $\frac{P(y_t=-1|\mathbf{x}_t^d)}{P(y_t=+1|\mathbf{x}_t^d)} = \frac{P(\mathbf{x}_t^d|y_t=-1)}{P(\mathbf{x}_t^d|y_t=+1)} = \frac{P([\mathbf{x}_t^d]_1=a|y_t=-1)*P([\mathbf{x}_t^d]_2=a|y_t=-1)}{P([\mathbf{x}_t^d]_1=a|y_t=+1)*P([\mathbf{x}_t^d]_2=a|y_t=+1)} = \frac{\frac{3}{4}*\frac{1}{4}}{\frac{2}{4}*\frac{2}{4}} = \frac{1}{2}$ The first step comes from Bayes' Rule, and the second comes from the Naive Bayes assumption.

3. (5) You now also observe the word count features of the test point

$$\mathbf{x}_t^c = [3, 2]^\top, \tag{5}$$

which you decide to model with a multinomial distribution and Laplace (aka +1) smoothing. If you use both the word count and categorical features in your final classifier, what is the posterior ratio

$$\frac{P(y_t = -1|\mathbf{x}_t)}{P(y_t = +1|\mathbf{x}_t)}, \tag{6}$$

based on both feature sets. (You do not need to reduce the fraction to its simplest form.)

Under the Naive Bayes assumption, $P(y|\mathbf{x}) = P(y)\prod_i P(x_i|y)$. There is no rule saying that all of the conditional probabilities in the product need to come from the same model, so we can take the categorical and multinomial estimates and multiply them together. We already calculated the categorical posterior ratio, so now we just find the multinomial ratio. Because we are computing the posterior ratio, we can ignore the constant $m!/[\mathbf{x}_t^c]_1![\mathbf{x}_t^c]_2!$ term in the multinomial PMFs. We continue to ignore the prior. Thus:

$$P(y_t = -1|\mathbf{x}_t^c) \propto P([\mathbf{x}_t^c]_1 = 3|y_t = -1) * P([\mathbf{x}_t^c]_2 = 2|y_t = -1) = (\theta_1^{-1})^3(\theta_2^{-1})^2 = (\frac{4}{6})^3(\frac{2}{6})^2$$

Where

$$\theta_1^{-1} = \frac{1 + \sum_{i=1}^4 I(y_t = -1)[\mathbf{x}_t^c]_1}{1 * 2 + \sum_{i=1}^4 I(y_i = -1)\sum_{j=1}^2 [\mathbf{x}_t^c]_j} = \frac{1+3}{2+4} = \frac{4}{6}$$

and

$$\theta_2^{-1} = \frac{1 + \sum_{i=1}^4 I(y_t = -1)[\mathbf{x}_t^c]_2}{1 * 2 + \sum_{i=1}^4 I(y_i = -1)\sum_{j=1}^2 [\mathbf{x}_t^c]_j} = \frac{1+1}{2+4} = \frac{2}{6}$$

Similarly,

$$P(y_t = +1|\mathbf{x}_t^c) \propto P([\mathbf{x}_t^c]_1 = 3|y_t = +1) * P([\mathbf{x}_t^c]_2 = 2|y_t = +1) = (\theta_1^{+1})^3(\theta_2^{+1})^2 = (\frac{1}{8})^3(\frac{7}{8})^2$$

Where

$$\theta_1^{+1} = \frac{1 + \sum_{i=1}^4 I(y_t = +1)[\mathbf{x}_t^c]_1}{1 * 2 + \sum_{i=1}^4 I(y_i = +1)\sum_{j=1}^2 [\mathbf{x}_t^c]_j} = \frac{1+0}{2+6} = \frac{1}{8}$$

and

$$\theta_2^{+1} = \frac{1 + \sum_{i=1}^4 I(y_t = +1)[\mathbf{x}_t^c]_2}{1 * 2 + \sum_{i=1}^4 I(y_i = +1)\sum_{j=1}^2 [\mathbf{x}_t^c]_j} = \frac{1+6}{2+6} = \frac{7}{8}$$

Therefore the final posterior ratio is equal to

$$\frac{\frac{3}{4} * \frac{1}{4} * (\frac{4}{6})^3 * (\frac{2}{6})^2}{\frac{2}{4} * \frac{3}{4} * (\frac{1}{8})^3 * (\frac{7}{8})^2}$$

4. (5) Show that the Naive Bayes classifier with Multinomial feature distributions is a linear classifier with some weight vector $\mathbf{w}$ and offset b.

Let θ_i^{-1} and θ_i^{+1} represent the class-based multinomial parameter estimates for word i, similar to above. Note that the posterior ratio is greater than 1 iff the log of the ratio is greater than 0. Let x be a vector of multinomial feature counts. We then have:

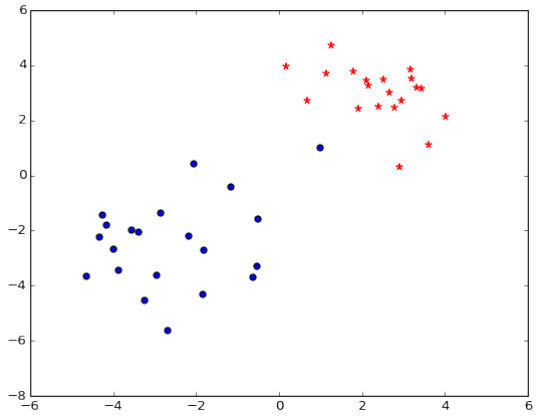
$$\begin{aligned} log\left(\frac{P(y=-1|x)}{P(y=+1|x)}\right) &= log\left(\frac{P(y=-1)\prod_i P(x_i|y=-1)}{P(y=+1)\prod_i P(x_i|y=+1)}\right) \\ &= log(P(y=-1)) + \sum_i x_i log(\theta_i^{-1}) - log(P(y=+1)) - \sum_i x_i log(\theta_i^{+1}) \\ &= log\left(\frac{P(y=-1)}{P(y=+1)}\right) + \sum_i x_i log\left(\frac{\theta_i^{-1}}{\theta_i^{+1}}\right) \\ &= b + x^\top w \end{aligned}$$

for $b = log(\frac{P(y=-1)}{P(y=+1)})$ and $w_i = log(\frac{\theta_i^{-1}}{\theta_i^{+1}})$ for all features i. As mentioned above, the posterior ratio is greater than 1 iff the log of the ratio is greater than 0, which occurs iff $x^\top w + b > 0$ for our chosen b and w. Thus the Naive Bayes classifier with Multinomial features is a linear classifier.

5 [16] Support Vector Machines

1. For this question, refer to the plot and dataset below, and recall that the soft-margin SVM optimization problem is equivalent to the following:

$$\min_{\mathbf{w},b} \mathbf{w}^\top \mathbf{w} + C \sum_{i=1}^{n} \max \left[1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b), 0\right]$$



(a) (3) Suppose you train an SVM on the dataset above with a very *large* value for the regularization parameter C (e.g., $C \to \infty$). Explain what would happen to the learned decision boundary as C increases, and draw as a bold solid line a decision boundary most likely to correspond to a very large value of C.

(b) (3) Suppose that instead you were to use a *small* value of C. Explain what would happen to the learned decision boundary as C *decreases*, and draw as a dashed line on the plot above a decision boundary most likely to correspond to a very small value of C.

(c) (1) On the plot, draw an additional *circle* data point that would not affect the decision boundary of a hard margin SVM trained on the dataset.

(d) (1) On the plot, draw an additional *star* data point that would make the hard margin SVM optimization problem infeasible.

2. Suppose you hand your dataset to the exciting new startup Bad Machine Learning Solutions, Inc., and they give you back a hard margin SVM classifier with parameters $\mathbf{w}$ and b. However, upon inspection, you discover that for every training data point $(\mathbf{x}_i, y_i)$, $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 2$.

(a) (1) Write down an equation for the separating hyperplane constraint used in the hard margin SVM problem.

(b) (2) Write down an equation for the margin of a linear classifier, $\gamma(\mathbf{w}, b)$.

(c) (5) Prove that the parameters this company gave you cannot possibly correspond to an optimal maximum margin classifier.

This page is left blank for scratch space.

This page is left blank for jokes. (Nothing dirty.)

Please do not write on this page. For administrative purposes only.

GML	
Short	
kNN	
NB	
SVM	
TOTAL	