Assignment 4
CSE 446: Machine Learning

University of Washington

Due: November 14, 2017

November 13, 2017: updates in red.

Since there is no programming portion for this assignment, you will simply submit:

- Your **report** as a **pdf file** named `A4.pdf` containing answers to written questions. You must include the plots and explanation for programming questions (if required) in this document only. We *strongly* recommend typesetting your scientific writing using LaTeX. Some free tools that might help: ShareLaTeX (`www.sharelatex.com`), TexStudio (Windows), MacTex (Mac), TexMaker (cross-platform), and Detexify^2 (online). If you want to type, but don't know (and don't want to learn) LaTeX, consider using a markdown editor with real-time preview and equation editing (e.g., `stackedit.io`, `marxi.co`). Writing solutions by hand is fine, as long as they are neat; you will need to scan them into a single pdf.
  Part of the training we aim to give you in this advanced class includes practice with technical writing. Organize your report as neatly as possible, and articulate your thoughts as clearly as possible. We prefer quality over quantity. Do not flood the report with tangential information such as low-level documentation of your code that belongs in code comments or the README. Similarly, when discussing the experimental results, do not copy and paste the entire system output directly to the report. Instead, create tables and figures to organize the experimental results.

1 Probability Warmup [20 points]

1. You are just told by your doctor that you tested positive for a serious disease. The test has 99% accuracy, which means that the probability of testing positive given that you have the disease is 0.99, and also that the probability of testing negative given that you do not have the disease is 0.99. The good news is that this is a rare disease, striking only 1 in 10,000 people.
   (a) Why is it good news that the disease is rare? [2 points]
   (b) What is the probability that you actually have the disease? Show your work. [8 points]
2. A group of students were classified based on whether they are senior or junior (random variable S) and whether they are taking CSE446 or not (random variable C). The following data was

obtained.

	Junior ($S = 0$)	Senior ($S = 1$)
taking CSE446 ($C = 1$)	23	34
not taking CSE446 ($C = 0$)	41	53

Suppose a student was randomly chosen from the group. Calculate the following probabilities. Show your work.

(a) $\hat{p}(C = 1, S = 1)$ [3 points]
(b) $\hat{p}(C = 1|S = 1)$ [3 points]
(c) $\hat{p}(C = 0|S = 0)$ [3 points]

3. Why are there "hats" on the p (for probability) symbols above? [1 point]

2 Maximum Likelihood Estimation [20 points]

This question uses a discrete probability distribution known as the Poisson distribution. A discrete random variable X follows a Poisson distribution with parameter λ if

$$p(X = k) = \frac{\lambda^k}{k!}e^{-\lambda} \qquad \forall k \in \{0, 1, 2, \dots\}$$

You work for the city of Seattle picking up ballots from ballot dropoff boxes around town. You visit the box near UW every hour on the hour and pick up the ballots that have been left there. Here are the number of ballots you picked up this morning, starting at 1am.

time	1am	2am	3am	4am	5am	6am	7am	8am
new ballots picked up	6	4	2	7	5	1	2	5

Let $\boldsymbol{G} = \langle G_1, \dots, G_N \rangle$ be a random vector where G_n is the number of ballots picked up on iteration n.

1. Give the log-likelihood function of $\boldsymbol{G}$ given λ. [6 points]
2. Compute the MLE for λ in the general case. [8 points]
3. Compute the MLE for λ using the observed $\boldsymbol{G}$. [6 points]

3 Naïve Bayes [40 points]

In class, we discussed naïve Bayes classifiers, which have the decision rule:

$$f(\mathbf{x}) = \begin{cases} +1 & \text{if } \hat{\pi} \cdot \prod_{j=1}^{d} \hat{p}(X_j = \mathbf{x}[j] \mid Y = +1) > (1 - \hat{\pi}) \cdot \prod_{j=1}^{d} \hat{p}(X_j = \mathbf{x}[j] \mid Y = -1) \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

where $\hat{\pi}$ is the estimate of the class prior $p(Y = +1)$.

In the case where every feature is binary and we assume conditional Bernoulli distributions for $p(X_j \mid Y)$, we can substitute:

$$\begin{aligned}
\hat{p}(X_j = 1 \mid Y = +1) &= \hat{\theta}_{j,+1} \\
\hat{p}(X_j = 0 \mid Y = +1) &= 1 - \hat{\theta}_{j,+1} \\
\hat{p}(X_j = 1 \mid Y = -1) &= \hat{\theta}_{j,-1} \\
\hat{p}(X_j = 0 \mid Y = -1) &= 1 - \hat{\theta}_{j,-1} \qquad (2)
\end{aligned}$$

1. Show that the decision boundary for naïve Bayes with binary features will be **linear**. To do this, you will need to prove that there exist $\mathbf{w}$ and b such that the function f in Equation 1 above is equivalent to sign$(\mathbf{w} \cdot \mathbf{x} + b)$. Here is a hint. This problem is much easier if you temporarily (for the purpose of your derivation) double the number of features, so that, for $j \in \{1, \ldots, d\}$,

   $$\begin{aligned}
   \mathbf{x}'[2j-1] &\leftarrow \mathbf{x}[j] \\
   \mathbf{x}'[2j] &\leftarrow 1 - \mathbf{x}[j]
   \end{aligned}$$

   Note that these new $2d$ features are all still binary. Suppose you work out $\mathbf{w}' \in \mathbb{R}^{2d}$, the weights for these new features. Now, if you work out what $\mathbf{w}'[2j-1]$ and $\mathbf{w}'[2j]$ are, then you will notice that these two terms in the activation function simplify:

   $$\begin{aligned}
   a &= \cdots + \mathbf{w}'[2j-1] \cdot \mathbf{x}'[2j-1] + \mathbf{w}'[2j] \cdot \mathbf{x}'[2j] + \cdots \\
   &= \cdots + \mathbf{w}'[2j-1] \cdot \mathbf{x}[j] + \mathbf{w}'[2j] \cdot (1 - \mathbf{x}[j]) + \cdots \\
   &= \cdots + \underbrace{(\mathbf{w}'[2j-1] - \mathbf{w}'[2j])}_{\mathbf{w}[j]} \cdot \mathbf{x}[j] + \underbrace{\mathbf{w}'[2j]}_{\text{constant w.r.t. } \mathbf{x}} + \cdots
   \end{aligned}$$

   You do not *have to* use this hint; the point is that you can express $\mathbf{w}[j]$ in terms of some of the $\hat{\theta}$s. Another hint: this question is easier if you take logarithms of both sides of $\hat{\pi} \cdot \prod_{j=1}^{d} \hat{p}(X_j = \mathbf{x}[j] \mid Y = +1) > (1 - \hat{\pi}) \cdot \prod_{j=1}^{d} \hat{p}(X_j = \mathbf{x}[j] \mid Y = -1)$. [10 points]
2. Suppose that you have some continuous features with values that are, at least in principle, unbounded.[1] Let X_j be the random variable for one of them. Choose a distribution for $p(X_j \mid Y)$ (say what it is called!), write out the formula you'll need to compute for $\hat{p}(X_j \mid Y)$ to implement f, and write out the maximum likelihood estimates for any parameter(s) entailed by your choice. [15 points]
3. Suppose that you have some discrete features with values that are not numerical (e.g., the Europe/Asia/America feature from the automobile dataset in the beginning of the class). Let X_j be the random variable for one of them, and let the values it can take be denoted $\mathcal{X}_j = \{x_j^1, x_j^2, \ldots, x_j^{m_j}\}$. Choose a distribution for $p(X_j \mid Y)$ (say what it is called!), write out the formula you'll need to compute for $\hat{p}(X_j \mid Y)$ to implement f, and write out the maximum likelihood estimates for any parameter(s) entailed by your choice. [15 points]

[1] Of course, in your training data, you will observe minimum and maximum values—but let's assume you believe even more extreme values are possible.

4 Hinge Loss [20 points]

For linear classification, we defined the perceptron loss as:

$$\max\{0, -y \cdot (\mathbf{w} \cdot \mathbf{x} + b)\}$$

The perceptron finds some separating hyperplane, eventually, if one exists, but any hyperplane will do (it doesn't have any preference among hyperplanes—unless, of course, you use regularization).

There is a more sophisticated approach that tries to find a hyperplane that maximizes the margin, if the data are linearly separable. It uses a loss function closely related to the perceptron loss, called the **hinge loss**, which is defined as:

$$\max\{0, 1 - y \cdot (\mathbf{w} \cdot \mathbf{x} + b)\}$$

Unlike the perceptron loss, the hinge loss function can be viewed as a relaxation of the zero-one loss, offering an elegant way to deal with training datasets that are not separable. In this exercise, we'll see why we consider it a "relaxation" of finding the maximum margin classifier when there's no way to linearly separate the data.

(Interestingly, minimizing the hinge loss, together with L_2 regularization, corresponds exactly to a model called the (soft-margin) **support vector machine** (SVM). SVMs/hinge loss try to find a separating hyperplane with a wide margin, with an implied penalty for every point that's on the wrong side of that hyperplane. To learn more about SVMs, read section 7.7 in the textbook.)

To simplify things in this problem, assume b is fixed at zero.

1. Show that the hinge loss is convex (as a function of $\mathbf{w}$). You may write a rigorous mathematical proof if you like, but a well-annotated picture with a clear explanation is also just fine. [5 points]
2. Suppose that for some $\mathbf{w}$ we have a correct prediction of y_n with $\mathbf{x}_n$, i.e. $y_n = \text{sign}(\mathbf{w} \cdot \mathbf{x}_n)$. What range of values can the hinge loss take on this correctly classified example? Points that are classified correctly and which have non-zero hinge loss are referred to as "margin mistakes." [5 points]
3. Let $M(\mathbf{w})$ be the number of mistakes made by $\mathbf{w}$ on our dataset (in terms of the zero-one loss). Show that:

$$\frac{1}{N} M(\mathbf{w}) \leq \frac{1}{N} \sum_{n=1}^{N} \max\{0, 1 - y_n \cdot \mathbf{w} \cdot \mathbf{x}_n\}. \tag{3}$$

   In other words, the average hinge loss on our dataset is an upper bound on the average number of mistakes we make on our dataset. [5 points]
4. To minimize the hinge loss, you can use stochastic subgradient descent. Derive the subgradient for the hinge loss with respect to one example, $(\mathbf{x}_n, y_n)$. [5 points]