Homework 2

Logistic Regression, Naive Bayes, SVM

CMU 10-701: Introduction to Machine Learning (Spring 2018)

OUT: Feb. 12, 2018
DUE: **Feb. 26, 2018, 10:30 AM**

START HERE: Instructions

- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. It is also OK to get clarification (but not solutions) from books or online resources, again after you have thought about the problems on your own. There are two requirements: first, cite your collaborators fully and completely (e.g., "Jane explained to me what is asked in Question 3.4"). Second, write your solution independently: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only.
- **Submitting your work:** Assignments should be submitted as PDFs using Gradescope unless explicitly stated otherwise. Each derivation/proof should be completed on a separate page. Submissions can be handwritten, but should be labeled and clearly legible. Else, submissions can be written in LaTeX. Upon submission, label each question using the template provided by Gradescope. Please refer to Piazza for detailed instruction for joining Gradescope and submitting your homework.
- **Programming**: All programming portions of the assignments should be submitted to Gradescope as well. We will not be using this for autograding, meaning you may use any language which you like to submit.

1 Logistic Regression (20 pts) [Otilia and George]

Consider a binary classification problem where the goal is to predict a class $y \in \{0, 1\}$, given an input $x \in \mathcal{R}^p$. A method that you can use for this task is *Logistic Regression.* Recall that in *Logistic Regression*, the conditional log likelihood probability can be written as follows:

$$\mathcal{L}(w) = \log(y|\mathbf{X}, w) = \sum_{i=1}^{n} [y_i w^T x_i - \log\big(1 + \exp\big(w^T x_i\big)\big)]$$

where:

- $\mathbf{X} \in \mathcal{R}^{n \times (1+p)}$ is a data matrix, with the first column composed of all ones
- $w \in \mathcal{R}^{(p+1) \times 1}$ is the weight vector, with the first index w^1 acting as the bias term
- x_i is a column vector of the i^{th} row of $\mathbf{X}$
- $y \in \mathcal{R}^{n \times 1}$ is a column vector of labels $y_i \in \{0, 1\}$
- p is the number of features in each observation

Our goal is to find the weight vector w that maximizes this likelihood. Unfortunately, for this model, we cannot derive an analytic solution. An alternative way to solve for w is to use gradient ascent, and "walk" towards the optimal w step by step. In this question, you will prove that the conditional log likelihood probability is a concave function, and thus gradient ascent will converge to the optimal solution w that maximizes $\mathcal{L}$.

(a) **[4 pts]** A real-valued function $f : S \to \mathcal{R}$ defined on a convex set S, is said to be *convex* if,

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2), \forall x_1, x_2 \in S, \forall t \in [0, 1].$$

Show that a linear combination of n convex functions, $f_1, f_2, ..., f_n$, $\sum_{i=1}^{n} a_i f_i(x)$ is also a convex function $\forall a_i \in R^+$.

(b) **[5 pts]** Another property of convex functions is that the second derivative is non-negative. In fact, having a non-negative second derivative is a sufficient condition for a function to be convex. Using this property, show that $f(x) = \log(1 + \exp x)$ is a convex function.

(c) **[4 pts]** Given two convex functions $f : \mathcal{R} \to \mathcal{R}$ and $g : \mathcal{R} \to \mathcal{R}$, where g is nondecreasing, show that their composition $g \circ f$ is also convex. In other words, show that:

$$(g \circ f)\,(tx_1 + (1-t)x_2) \leq t(g \circ f)(x_1) + (1-t)(g \circ f)(x_2)$$

Note that $(g \circ f)(x) = g(f(x))$.

(d) **[7 pts]** Using (a), (b) and (c), show that the log likelihood of *Logistic Regression* is a concave function. Recall that if a function $f(x)$ is convex, then $-f(x)$ is concave.

2 Naive Bayes: Theory (20 pts) [Dimitris and Wenhao]

Recall the difference in the modeling assumptions in a discriminative and a generative classifier. In a generative model, $P(X, Y)$ is estimated by initially modeling the conditional $P(X|Y)$, since, $P(X, Y) = P(Y)P(X|Y)$. The joint probability with Bayes rule is then used to calculate $P(Y|X)$ for each class label. On the other hand, a discriminative classifier directly estimates $P(Y|X)$. In this problem, we will explore the relation between Naive Bayes and logistic regression classifiers, by focusing on the class conditional $P(Y|X)$.

When Y is Boolean and $X = \langle X_1 ... X_n \rangle$ is a vector of continuous variables, and each $P(X_i|Y = y_k)$ is modelled with a Gaussian distribution, then the assumptions of the Gaussian Naive Bayes classifier imply that $P(Y|X)$ is given by the logistic function with appropriate parameters $w_0, w_1, ...w_n$. In particular:

$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

and

$$P(Y = 0|X) = \frac{\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

Consider instead the case where Y is Boolean and $X = \langle X_1 ... X_n \rangle$ is a vector of Boolean variables. Show that the derived expression of $P(Y|X)$ has the same form as that in the Logistic Regression classifier model (by getting an appropriate expression, which can be substituted for weights in the $P(Y|X)$ equation for Logistic Regression).

Hints

(a) Simple notation will help. Since the X_i are Boolean variables, you need only one parameter to define $P(X_i|Y = y_k)$. Define $\theta_{i1} \equiv P(X_i = 1|Y = 1)$, in which case $P(X_i = 0|Y = 1) = (1 - \theta_{i1})$. Similarly, use θ_{i0} to denote $P(X_i = 1|Y = 0)$.

(b) Notice with the above notation you can represent $P(X_i|Y = 1)$ as follows

$$P(X_i|Y = 1) = \theta_{i1}^{(X_i)}(1 - \theta_{i1})^{(1-X_i)}$$

Note when $X_i = 1$ the second term is equal to 1 because its exponent is zero. Similarly, when $X_i = 0$ the first term is equal to 1 because its exponent is zero.

3 SVM (20 pts) [Satya]

3.1 Ridge regression [10 Points]

In contrast to ordinary least squares which has a cost function,

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(\theta^T x^{(i)} - y^{(i)})^2$$

we can also add a term that penalizes large weights in θ. In ridge regression, our least squares cost is regularized by adding a term $\lambda||\theta||^2$, where $\lambda > 0$ is a fixed (known) constant. The ridge regression cost function is then

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(\theta^T x^{(i)} - y^{(i)})^2 + \lambda||\theta||^2$$

Suppose that we transform our feature vectors using a feature mapping ϕ, so that the ridge regression cost function becomes

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(\theta^T \phi(x^{(i)}) - y^{(i)})^2 + \lambda||\theta||^2. \tag{1}$$

(a) Compute the minimizer $\widehat{\theta}_\phi$ of the new ridge regression objective in Eq. (1).

(b) Suppose that you are not given the actual feature map $\phi(\cdot)$, but only have access to the matrix $K \in \mathbb{R}^{m\times m}$ where $K_{ij} = (\phi(x^{(i)}))^T\phi(x^{(j)})$. Can you still compute $\widehat{\theta}_\phi$?

3.2 Support Vector Regression [10 points]

In this question, we consider using the methodology of support vector machines for regression. Your training data is $(x_1,\ y_1)$,...., $(x_n,\ y_n)$, where $x_i\,\epsilon\,\mathbb{R}^m$, $y_i\,\epsilon\,\mathbb{R}$. A candidate loss function for regression then is the so-called epsilon sensitive loss:

$$L_\epsilon(x, y, f) = |y - f(x)|_\epsilon = \max(0, |y - f(x)| - \epsilon)$$

Here x is the input, y is the output, and f is the function used for predicting the label. Using this notation, the cost function is defined as

$$\frac{1}{2}\left\|w\right\|^2 + C\sum_{i=1}^{n}L_\epsilon(x_i, y_i, f)$$

where $f(x) = w^T x$, and C, $\epsilon > 0$ are parameters.

Using slack variables $\xi_i := |y - f(x)| - \epsilon$, rewrite this problem as a quadratic problem (i.e. quadratic objective with linear constraints, just as we did for Support Vector Machines for classification).

4 Multiple Choice Questions (10 pts) [Sreena]

There might be one or more right answers. Please explain your choice in one or two sentences.

1. If your SVM with linear decision boundary is underfitting, which of these would you consider? [2 points]
(A) Increase number of features
(B) Decrease number of features
(C) Increase hyper-parameter C
(D) Decrease Hyper-parameter C

2. Which of the following is commonly used to evaluate the performance of a logistic regression model? [2 points]
(A) Accuracy
(B) Log loss
(C) Mean Squared Error
(D) Hinge Loss

3. Which of the following methods uses a generative model of the input given the output? [2 points]
(A) Support Vector Machines
(B) Logistic Regression
(C) Naive Bayes
(D) Linear Regression

4. Which of the following models are used to solve a classification problem? [2 points]
(A) Logistic Regression
(B) SVM
(C) Linear Regression
(D) Naive Bayes

5. If your polynomial regression model is overfitting, which of these would you consider? [2 points]
(A) Increase the degree of the polynomial.
(B) Remove regularization if any.
(C) Decrease the degree of the polynomial.
(D) Use classification instead.

5 Programming Exercise (20 pts) [Shaojie and Lam]

Note: Your code for all of the programming exercises should also be submitted to Gradescope. In particular, there is a separate 'programming assignment' in Gradescope to which you should upload your code, while visualizations and written answers should still be submitted within the primary Gradescope assignment. In your code, **please use comments to point out primary functions that compute the answers to each question.**

Feel free to use any programming language, as long as your TAs can read your code. Turn in your code in a single zipped folder that might contain multiple source code files.

In this problem, you will implement the Naive Bayes (NB) algorithm on a pre-processed dataset that contains both **discrete** and **continuous** covariates. Recall from lecture that Naive Bayes algorithm takes a generative approach that models $P(X, y)$ using the Bayes rule. The key, in particular, is that NB classifiers assumes the attributes $x^1, x^2, \dots$ are conditionally independent of each other given a class label y. More formally, the prediction is made by $\hat{y} = \text{argmax}_y\, P(y|X)$, where:

$$P(y|X = (x^1, \dots, x^n)) \propto P(X, y) = P(X|y) \cdot P(y) = P(y) \cdot \prod_i P(x^i|y) \tag{2}$$

Consider the case where there are C classes (i.e. $y \in [C]$) and N different attributes.

- For a discrete attribute i that takes M_i different values (e.g., attribute "gender" can take only male or female, so $M_i = 2$) , the distribution $P(x^i|y = c)$ can be modeled by parameters $\alpha_{i,c,1}, \alpha_{i,c,2}, \dots, \alpha_{i,c,M_i}$, with $\sum_{j=1}^{M_i} \alpha_{i,c,j} = \sum_{j=1}^{M_i} P(x^i = j|y = c) = 1$. **Do NOT use smoothing**. Assume $\log(0) = \lim_{x \to 0} \log x = -\infty$.
- For a continuous attribute i, **in this question**, we can assume the conditional distribution is Gaussian; i.e. $P(x^i|y = c) = \mathcal{N}(\mu_{i,c}, \sigma^2_{i,c}) \approx \frac{1}{\sqrt{2\pi(\sigma^2_{i,c}+\varepsilon)}} \exp\left(-\frac{(x^i-\mu_{i,c})^2}{2(\sigma^2_{i,c}+\varepsilon)}\right)$, where $\mu_{i,c}$ and $\sigma^2_{i,c}$ are the mean and variance for attribute i given class c, respectively. In implementation, you should use $\mu_{i,c}$ =sample mean and $\sigma^2_{i,c}$ =sample variance to estimate these. **Meanwhile, take $\varepsilon = 10^{-9}$, which is a small value just to ensure the variance is not 0**.

You now need to implement a Naive Bayes algorithm that predicts whether a person makes over \$50K a year, based on various attributes about this person (e.g., age, education, sex, etc.). You can find the detailed description of the attributes, and download the data at

`https://archive.ics.uci.edu/ml/datasets/adult`.

You will need 2 files:

- `adult.data`[1]: Each line is a training data, with attributes listed in the same order as on the website and delimited by comma. For instance, the first entry of each line is the "age". The last entry of each line gives the correct label (>50K, ≤50K). There should be 32,561 training data.
- `adult.test`[2]: Same format as `adult.data`, but only used in evaluation of the model (i.e. testing), so you shouldn't use the label for training your NB classifier. There should be 16,281 testing data.

IMPORTANT: You should ignore (but don't delete) all incomplete data lines, which contains "?" as values for certain attributes in the line.

Hint: Because $P(y)\prod_i P(x^i|y)$ can get extremely small, you should use log-posterior for the computation:

$$\log\left[P(y)\prod_i P(x^i|y)\right] = \log P(y) + \sum_i \log P(x^i|y)$$

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data

[2] https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test

5.1 Report Parameters

For questions below, report only up to 4 significant digits after the decimal point.

(a) **[2 pts]** Report the prior probability of each class.

(b) **[8 pts]** For each class c, for each attribute i, print & report the following:

- If the attribute is discrete, report the value of $\alpha_{i,c,j}$ for every possible value j, **in the same order as on the website** (e.g., for attribute "sex", you should report the α for "Female" first, then "Male"). Clearly mark what the attribute is and what is the value of j.
- If the attribute is continuous, report the value of $\mu_{i,c}$ and $\sigma_{i,c}$.

One sample format of your answer looks like the following (the values are made up):

(1) Class "$> 50K$":

- age: mean=43.9591, var=105.4513
- workclass: Private=0.02, Self-emp-not-inc=0.0134, Self-emp-inc=0.0998, ...
- ...
- native-country: ...

(2) Class ...

(c) **[2 pts]** Report the log-posterior values (i.e. $\log[P(X|y)P(y)]$) for the first 4 test data (in the same order as the data), each with 4 significant digits.

5.2 Evaluation

(a) **[1 pts]** Evaluate the trained model on the training data. What is the accuracy of your NB model?

(b) **[1 pts]** Evaluate the trained model on the testing data. What is the accuracy of your NB model?

(c) **[6 pts]** Instead of training the NB using all training data, try to train only with the first n data [3], and then evaluate on the testing dataset. Report the testing accuracies for $n = \{2^i \text{ for } i = 5, \ldots, 13\}$ (i.e. $n = 32, \ldots, 8192$). Plot the training and testing accuracy vs. # of training data. What do you observe? At what value of n do testing accuracy and training accuracy attain maximum? **In general**, what would you expect to happen if we use only a few (say $n < 3$) training data for Naive Bayes? Explain briefly (hint: we did not use smoothing).

[3] The count includes those lines with "?", but you should ignore those lines when training.