CSCI 1950-F Homework 5: Logistic Regression

Brown University, Spring 2012

Homework due at 12:00pm on March 12, 2012

Question 1:

In this question, we consider a continuous estimation problem in which the input \( x \) and response variable \( y \) are both real numbers. Their joint probability density function \( p(x, y) \) is constant within the closed, shaded region shown in Figure 1, and zero elsewhere.

a) For any estimator \( \hat{y}(x) \), consider the quadratic loss function \( L(\hat{y}(x), y) = (\hat{y}(x) - y)^2 \).

Determine the estimator that minimizes the following posterior expected loss:

\[
E[L(\hat{y}(x), y) \mid x] = \int_{-\infty}^{\infty} L(\hat{y}(x), y) p(y \mid x) \, dy
\]

Simplify the form of your answer as much as possible.

b) Consider how the estimator \( \hat{y}(x) \) from part (a) behaves for three particular input variables: \( x = -1.5, x = 0, \) and \( x = 1.5 \). In each case, is \( \hat{y}(x) \) also a maximum a posteriori (MAP) estimator? Why or why not?

c) Suppose that rather than knowing the true density function \( p(x, y) \), we instead have \( N \) training examples \( \{(x_i, y_i)\}_{i=1}^{N} \), where \( (x_i, y_i) \) are independent samples from \( p(x, y) \).

Consider the following linear regression model:

\[
p(y \mid w, x, \beta) = \text{Normal}(y \mid w^T \phi(x), \beta^{-1})
\]

Suppose that we choose \( \phi(x) = [1, x]^T \) as our basis or feature functions, and estimate \( \hat{w} \) via maximum likelihood. As \( N \to \infty \), will the expected loss of the prediction function \( \hat{y}_w(x) = \hat{w}^T \phi(x) \) be lower than, higher than, or equal to that of the estimator from part (a)? Justify your answer.

d) Consider again the linear regression model from part (c), but with alternative features \( \phi'(x) = [1, |x|]^T \). As \( N \to \infty \), will the expected loss of \( \hat{y}'_w(x) = \hat{w}'^T \phi'(x) \) be lower than, higher than, or equal to that of the estimator from part (a)? Justify your answer.
**Figure 1:** Joint probability density function \( p(x, y) \) for question 1. The density is constant (uniform) in the shaded region, and zero elsewhere.

**Question 2:**

This problem investigates logistic regression classifiers. Let \( y \in \{1, \ldots, C\} \) denote the discrete class labels we want to predict, \( x \) the input or conditioning variables, and \( \phi(x) \in \mathbb{R}^m \) a fixed (possibly non-linear) feature function. One form of the multinomial logistic regression model is then

\[
p(y = c \mid x, w) = \frac{\exp(w_c^T \phi(x))}{\sum_{c'=1}^C \exp(w_{c'}^T \phi(x))},
\]

where \( w_c \in \mathbb{R}^m \) are the feature weights associated with class \( c \), and \( w \in \mathbb{R}^{Cm} \) is a vector concatenating the weights for all classes.

This parameterization is not identifiable: coordinated translations of the weight vectors \( w_c \) for different classes can produce an identical model. To avoid this ambiguity and improve robustness, we place a zero-mean, diagonal-covariance Gaussian prior on the weight vector:

\[
p(w) = \mathcal{N}(w \mid 0, \alpha^{-1} I_m) \propto \exp\left(-\frac{\alpha}{2} w^T w\right)
\]

Here, \( \alpha \) is a tunable precision parameter that controls the degree of regularization.

a) *Give an expression for* \(-\log p(w) - \log p(y \mid x, w)\), *the regularized negative log conditional likelihood of a generic data set* \( D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \), *ignoring any terms and factors that do not depend on* \( w \). *Also give an expression for the derivative of the regularized negative log conditional likelihood with respect to a particular feature weight* \( w_{ck} \). *A detailed derivation is not necessary, but be sure your notation is clear.*
We now reexamine the gamma ray data set from homework 2, but instead apply a logistic regression model for the binary classification of star showers. We have split the previous training set so that there are now 11,413 training examples, 3,804 development examples, and 3,804 test examples. Their class labels are stored in column vectors named \texttt{trainLabels}, \texttt{devLabels}, \texttt{testLabels}.

We will compare the performance of three different feature mappings of the $F = 10$ raw inputs. In all cases, the first feature should be a constant bias or offset term, $\phi_0(x_i) = 1$. The three feature sets are then:

1. $F + 1$ linear features, the bias feature plus the raw inputs $\phi_k(x_i) = x_{ik}, 1 \leq k \leq F$.

2. $2F + 1$ diagonal quadratic features, including the $F + 1$ linear features from set 1, as well as $F$ quadratic features $\phi_{F+k}(x_i) = x_{ik}^2, 1 \leq k \leq F$.

3. $(F + 1)F/2 + F + 1$ general quadratic features, including the $2F + 1$ features from set 2, as well as products of all pairs of input dimensions $x_k x_\ell, k \neq \ell$.

Given this data and the objective from part (a), we will use the \texttt{minFunc} function from the pmtk3 Matlab toolbox (http://code.google.com/p/pmtk3/) to find the weight vector $w$ that minimizes the regularized negative log conditional likelihood on the training data. You should provide the optimizer with the gradient of the regularized negative log conditional likelihood, and write your own function to compute the objective and its gradient. For all of the following questions, do all calculations for 10 logarithmically-spaced values of $\alpha$ between $10^{-8}$ and 10, and test each of the three feature sets above. There’s some sample code to get you started at /course/cs195f/asgn/hw5_logistic with instructions as to how to utilize the \texttt{minFunc} command, including appropriate convergence tolerance parameters.

b) Plot the negative log conditional likelihood of the train data as a function of $\alpha$ when training and evaluating on train. Also plot the accuracy of the classifier as a function of $\alpha$ when training and evaluating on train.

c) Plot the negative log conditional likelihood of the dev data as a function of $\alpha$. That is, for each value of $\alpha$ estimate the feature weight $w$ from the train data, and then calculate the negative log conditional likelihood of the dev data for that value of $w$. Find a value of $\alpha$ which minimizes the negative log conditional likelihood of the dev data. Does the learning output seem sensitive to the value of $\alpha$? Provide an explanation for your observations.

d) For each of the three feature sets, and the weight vectors $w$ corresponding to the values of $\alpha$ identified in part (c), evaluate and report the test accuracy of the corresponding classifiers.

e) Train a naive Bayes classifier on this train data using the Gaussian naive Bayes ML estimation code from homework 2. What is its performance on test? Compare to the logistic regression model.
Question 3:

This question uses synthetic data to compare the properties of logistic regression and linear regression for classification. Each data item has two continuous features \( x \in \mathbb{R}^2 \) and is labeled as one of either \( K = 2 \) or \( K = 3 \) classes. The data are stored as .mat files here:

/course/cs195f/asgn/hw5_logistic/toy

Your linear regression code from homework 4 can be adapted for classification as follows. For \( K \) classes, each response is encoded as a row vector \( Y_i = [Y_{i1}, \ldots, Y_{iK}] \), where \( Y_{ik} = 1 \) for an example of class \( k \), and zero otherwise. For \( N \) data samples we define the \( N \times K \) matrix \( Y \) as a matrix of 0’s and 1’s, with each row having a single 1. We fit a linear regression model to each of the columns of \( Y \) as follows:

\[
\hat{Y} = \Phi(\Phi^T\Phi)^{-1}\Phi^TY
\]

Here, \( \Phi \) is the \( N \times 3 \) model matrix of corresponding to the feature function \( \phi(x_i) = [1, x_{i1}, x_{i2}]^T \), i.e. the raw 2D input data augmented by a constant bias feature. The weights corresponding to the least squares prediction above equal

\[
\hat{W} = (\Phi^T\Phi)^{-1}\Phi^TY
\]

Here, \( \hat{W} \) is a \( 3 \times K \) matrix where the \( k^{th} \) column \( \hat{w}_k \) represents the linear regression fit for class \( k \). Finally, we can use this linear regression model to classify a new observation as

\[
\hat{y}(x) = \arg \max_k \phi(x)^T\hat{w}_k
\]

The supplied function \texttt{plotClassifier} can be used to visualize decision boundaries.

We compare the performance of this linear least squares classifier to a multinomial logistic regression classifier, both using the same features. To fit multinomial logistic regression models, use your implementation from question 2, with a small but nonzero regularization constant \( \alpha = 10^{-8} \) to ensure identifiability.

a) The first dataset contains two classes which lie in well-separated clouds. Implement the least squares classifier described above. Estimate weights from training data, and plot the learned decision boundary together with the training points. If implemented correctly, your test accuracy should be 100%. Is this the case?

b) The second dataset contains three classes with means arranged in a triangular pattern. Train a least squares classifier as above, as well as a multinomial logistic regression classifier using the same features. Plot the training decision boundaries for both classifiers, and report test accuracy for each. Explain any performance differences.

c) The third dataset contains three classes with means arranged in a straight line. Train a least squares classifier as above, as well as a multinomial logistic regression classifier using the same features. Plot the training decision boundaries for both classifiers, and report test accuracy for each. Explain any performance differences.