CS 189:289A Introduction to Machine Learning
Spring 2017

Homework 4: Regression
Due Mon., March 13

Instructions

- We prefer that you typeset your answers using LaTeX or other word processing software. Neatly handwritten and scanned solutions will also be accepted.
- Please make sure to start **each question on a new page**, as grading (with Gradescope) is much easier that way!
- See the end of this assignment for the list of deliverables.
- Due **Monday, March 13, 2017 at 11:59 PM.**

Q1. Logistic Regression with Newton's Method

Consider sample points $X_1, X_2, \ldots, X_n \in \mathbb{R}^d$ and associated values $y_1, y_2, \ldots, y_n \in \{0, 1\}$, an $n \times d$ design matrix $X = [X_1 \quad \ldots \quad X_n]^\top$ and an n-vector $y = [y_1 \quad \ldots \quad y_n]^\top$.

If we add ℓ_2-regularization to logistic regression, the cost function is

$$J(w) = \lambda \, |w|_2^2 - \sum_{i=1}^{n} \Big(y_i \ln s_i + (1 - y_i) \ln(1 - s_i) \Big)$$

where $s_i = s(X_i \cdot w)$, $s(\gamma) = 1/(1 + e^{-\gamma})$, and $\lambda > 0$ is the regularization parameter. As in lecture, the vector $s = [s_1 \quad \ldots \quad s_n]^\top$ is a useful shorthand.

In this problem, you will use Newton's method to minimize this cost function on the four-point, two dimensional training set

$$X = \begin{bmatrix} 0 & 3 \\ 1 & 3 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \qquad y = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

You may want to draw these points on paper to see what they look like. The y-vector implies that the first two sample points are in class 1, and the last two are in class 0.

These sample points cannot be separated by a decision boundary that passes through the origin. As described in lecture, append a 1 to each X_i vector and use a weight vector $w \in \mathbb{R}^3$ whose last component is the bias term (the term we call α in lecture).

1. Derive the gradient of the cost function $J(w)$. Your answer should be a simple matrix-vector expression. Do NOT write your answer in terms of the individual components of the gradient vector.

2. Derive the Hessian of $J(w)$. Again, your answer should be a simple matrix-vector expression.

3. State the update equation for one iteration of Newton's method for this problem.

4. We are given a regularization parameter of $\lambda = 0.07$ and a starting point of $w^{(0)} = \begin{bmatrix} -2 & 1 & 0 \end{bmatrix}^\top$.

   (a) State the value of $s^{(0)}$ (the value of s before any iterations).

   (b) State the value of $w^{(1)}$ (the value of w after one iteration).

   (c) State the value of $s^{(1)}$.

   (d) State the value of $w^{(2)}$ (the value of w after two iterations).

Q2. ℓ_1- and ℓ_2-Regularization

Consider sample points $X_1, X_2, \ldots, X_n \in \mathbb{R}^d$ and associated values $y_1, y_2, \ldots, y_n \in \mathbb{R}$, an $n \times d$ design matrix $X = [X_1 \quad \ldots \quad X_n]^\top$ and an n-vector $y = [y_1 \quad \ldots \quad y_n]^\top$. For the sake of simplicity, assume that the sample data has been centered and whitened so that each feature has mean 0 and variance 1 and the features are uncorrelated; i.e., $X^\top X = nI$. For this question, we will not use a fictitious dimension nor a bias term; our linear regression function will be zero for $x = 0$.

Consider linear least-squares regression with regularization in the ℓ_1-norm, also known as Lasso. The Lasso cost function is

$$J(w) = |Xw - y|^2 + \lambda \, \|w\|_{\ell_1}$$

where $w \in \mathbb{R}^d$ and $\lambda > 0$ is the regularization parameter. Let $w^* = \operatorname{argmin}_{w \in \mathbb{R}^d} J(w)$ denote the weights that minimize the cost function.

In the following steps, we will show that whitened training data decouples the features, so that w_i^* is determined by the i^{th} feature alone (i.e., column i of the design matrix X), regardless of the other features. This is true for both Lasso and ridge regression.

1. We use the notation $X_{*1}, X_{*2}, \ldots, X_{*d}$ to denote column i of the design matrix X, which represents the i^{th} feature. (Not to be confused with row i of X, the sample point $X_i^\top$.) Write $J(w)$ in the following form for appropriate functions g and f.

$$J(w) = g(y) + \sum_{i=1}^{d} f(X_{*i}, w_i, y, \lambda)$$

2. If $w_i^* > 0$, what is the value of w_i^*?

3. If $w_i^* < 0$, what is the value of w_i^*?

4. Considering parts 2 and 3, what is the condition for w_i^* to be zero?

5. Now consider ridge regression, which uses the ℓ_2 regularization term $\lambda \, |w|^2$. How does this change the function $f(\cdot)$ from part 1? What is the new condition in which $w_i^* = 0$? How does it differ from the condition you obtained in part 4?

Q3. Regression and Dual Solutions

a) For a vector w, derive $\nabla |w|^4$. Then derive $\nabla_w |Xw - y|^4$.

b) Consider sample points $X_1, X_2, \ldots, X_n \in \mathbb{R}^d$ and associated values $y_1, y_2, \ldots, y_n \in \mathbb{R}$, an $n \times d$ design matrix $X = [X_1 \quad \ldots \quad X_n]^\top$ and an n-vector $y = [y_1 \quad \ldots \quad y_n]^\top$, and the regularized regression problem

$$w^* = \text{argmin}_{w \in \mathbb{R}^d} |Xw - y|^4 + \lambda |w|^2,$$

which is similar to ridge regression, but we take the fourth power of the error instead of the squared error. (It is not possible to write the optimal solution w^* as the solution of a system of linear equations, but it can be found by gradient descent or Newton's method.)

Show that the optimum w^* is unique. By setting the gradient of the objective function to zero, show that w^* can be written as a linear combination $w^* = \sum_{i=1}^n a_i X_i$ for some scalars $a_1, \ldots, a_n$. Write the vector a of dual coefficients in terms of X, y, and the optimal solution w^*.

c) Consider the regularized regression problem

$$w^* = \text{argmin}_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} L(w^\top X_i, y_i) + \lambda |w|^2$$

where the loss function L is convex in its first argument. Prove that the optimal solution has the form $w^* = \sum_{i=1}^n a_i X_i$. If the loss function is not convex, does the optimal solution always have the form $w^* = \sum_{i=1}^n a_i X_i$? Justify your answer.

Q4. Franzia Classification + Logistic Regression = Party!

Daylen is planning the frat party of the semester. He's completely stocked up on Franzia. Unfortunately, the labels for 497 boxes (test set) have been scratched off, and he needs to quickly find out which boxes contain Red wine (label 1) and White wine (label 0). Fortunately, for him the boxes still have their Nutrition Facts (features) intact and detail the chemical composition of the wine inside the boxes (the description of these features and the features themselves are provided in `data.mat`). He also has 6,000 boxes with Nutrition Facts and labels intact (train set). Help Daylen figure out what the labels should be for the 497 mystery boxes.

IMPORTANT: Do NOT use any software package for logistic regression that you didn't write yourself!

1. Derive and write down the batch gradient descent update equation for logistic regression with ℓ_2 regularization.

   Choose a reasonable regularization parameter value and a reasonable learning rate. Run your algorithm and plot the cost function as a function of the number of iterations. (As this is batch descent, one "iteration" should use every sample point once.)

2. Derive and write down the stochastic gradient descent update equation for logistic regression with ℓ_2 regularization. Choose a suitable learning rate. Run your algorithm and plot the cost function as a function of the number of iterations—where now each "iteration" uses *just one* sample point.

   Comment on the differences between the convergence of batch and stochastic gradient descent.

3. Instead of a constant learning rate ϵ, repeat part 2 where the learning rate decreases as $\epsilon \propto 1/t$ for the t^{th} iteration. Plot the cost function vs. the number of iterations. Is this strategy better than having a constant ϵ?

4. Finally, train your classifier on the entire training set. Submit your predictions for the test set to Kaggle. You can only submit twice per day, so get started early! In your writeup, include your Kaggle display name and score and describe the process you used to decide which parameters to use for your best classifier.

Q5. Real World Spam Classification

Motivation: After taking CS 189 or CS 289A, students should be able to wrestle with "real-world" data and problems. These issues might be deeply technical and require a theoretical background, or might demand specific domain knowledge. Here is an example that a past TA encountered.

Daniel (a past CS 189 TA) interned as an anti-spam product manager for an email service provider. His company uses a linear SVM to predict whether an incoming spam message is spam or ham. He notices that the number of spam messages received tends to spike upwards a few minutes before and after midnight. Eager to obtain a return offer, he adds the timestamp of the received message, stored as number of milliseconds since the previous midnight, to each feature vector for the SVM to train on, in hopes that the ML model will identify the abnormal spike in spam volume at night. To his dismay, after testing with the new feature, Daniel discovers that the linear SVM's success rate barely improves.

Why can't the linear SVM utilize the new feature well, and what can Daniel do to improve his results? Daniel is unfortunately limited to a quadratic kernel i.e. the features are at most polynomials of degree 2 over the original variables. This is an actual interview question Daniel received for a machine learning engineering position!

Write a short explanation. This question is open ended, and there can be many correct answers.

Submission Instructions

Please submit

- a PDF write-up containing your *answers, plots, and code* to Gradescope. Be sure to include your Kaggle display name and score in the PDF.
- a .zip file of your *code* and a README explaining how to run your code to Gradescope.
- your CSV file of predictions to Kaggle.