CS189: Introduction to Machine Learning

Homework 3

Due: October 11th, 2016, 12:00 noon, NOT MIDNIGHT

Problem 1: Maximum Likelihood Estimation of Multivariate Gaussian Distribution

Suppose that n samples $X_1, \cdots, X_n \in \mathrm{R}^d$ are random vectors which are drawn independently according to the following Gaussian distributions:

1. Let X have a Gaussian distribution with covariance matrix $\Lambda = \sigma^2 I_{d\times d}$ and mean μ. Find the maximum likelihood estimates of σ^2 and μ.

Solution:

$$\begin{aligned}
\mathcal{L}(X|\mu,\sigma) = &\quad \frac{1}{(2\pi)^{dn/2}\sigma^{dn}}\exp\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(X_i-\mu)^T(X_i-\mu)\}\\
l(\mu,\sigma|\mathcal{D}) = &\quad -\frac{nd}{2}\log 2\pi - dn\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(X_i-\mu)^T(X_i-\mu)\\
\frac{\partial l}{\partial \mu} = &\quad \frac{\sum_i X_i - \mu}{\sigma^2} = 0 \rightarrow \hat{\mu}_{ML} = \frac{1}{n}\sum_{i=1}^{n} X_i\\
\frac{\partial l}{\partial \sigma} = &\quad -\frac{nd}{\sigma} + \frac{\sum_{i=1}^{n}(X_i-\hat{\mu}_{ML})^T(X_i-\hat{\mu}_{ML})}{\sigma^3} = 0\\
\rightarrow \hat{\sigma}^2_{ML} = &\quad \frac{1}{nd}\sum_{i=1}^{n}(X_i-\hat{\mu}_{ML})^T(X_i-\hat{\mu}_{ML})
\end{aligned}$$

2. Let X have a Gaussian distribution with a diagonal covariance matrix Λ and mean μ. That is, assume that Λ is known to be a diagonal matrix of the form

$$\Lambda = \begin{bmatrix} \sigma_1^2 & & & & \\ & \sigma_2^2 & & & \\ & & \sigma_3^2 & & \\ & & & \ddots & \\ & & & & \sigma_d^2 \end{bmatrix}.$$

Find the maximum likelihood estimates of Λ and μ.

Solution: if $\Lambda_{ii} = \sigma_i^2$:

$$\begin{aligned}
\mathcal{L}(X|\mu,\Lambda) = &\quad \frac{1}{(2\pi)^{n/2}(\prod_{j=1}^d \sigma_j^2)^{n/2}} \exp\{-\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^d \frac{(X_i^j-\mu_j)^2}{\sigma_j^2}\} \\
l(\mu,\Lambda|X) = &\quad -\frac{n}{2}\log 2\pi - \frac{n}{2}\sum_{j=1}^n \log\sigma_j^2 - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^d \frac{(X_i^j-\mu_j)^2}{\sigma_j^2} \\
\frac{\partial l}{\partial \mu_j} = &\quad \sum_{i=1}^n \frac{(X_i^j-\mu_j)}{\sigma_j^2} = 0 \to \hat{\mu}_{ML} = \frac{1}{n}\sum_{i=1}^n X_i \\
\frac{\partial l}{\partial \sigma_j^2} = &\quad 0 \to \hat{\sigma}_{j,ML}^2 = \frac{\sum_{i=1}^n (X_i^j-\mu_j)^2}{n}
\end{aligned}$$

3. Let X have a Gaussian distribution with covariance matrix Λ and mean $A\mu$ where Λ and A are known $d\times d$ matrices. Find the maximum likelihood estimate of μ assuming A is invertible.

Solution:

$$\begin{aligned}
\mathcal{L}(X|\mu,\Lambda) = &\quad \frac{1}{(2\pi)^{n/2}|\Lambda|^{n/2}} \exp\{-\frac{1}{2}\sum_{i=1}^n (X_i - A\mu)^T\Lambda^{-1}(X_i - A\mu)\} \\
l(\mu,\Lambda|X) = &\quad -\frac{n}{2}\log 2\pi - \frac{n}{2}\log|\Lambda| - \frac{1}{2}\sum_{i=1}^n (X_i - A\mu)^T\Lambda^{-1}(X_i - A\mu) \\
\frac{\partial l}{\partial \mu} = &\quad = A^T\Lambda^{-1}\sum_{i=1}^n [(X_i - A\mu)] = 0 \\
\to &\quad \hat{\mu}_{ML} = A^{-1}\frac{\sum_{i=1}^n X_i}{n}
\end{aligned}$$

Problem 2: l_2-regularized Logistic/Linear Regression with Newton's Method

Let $\{(x_i, y_i)\}_{i=1}^n$ be a training set, where $x_i \in \mathbb{R}$ and $y_i \in \{0,1\}$.

- (a) Recall the negative log likelihood for l_2-regularized logistic regression:

$$l(\beta) = \lambda\|\beta\|_2^2 - \sum_{i=1}^n [y_i \log\mu_i + (1-y_i)\log(1-\mu_i)]$$

where $\mu_i = 1/(1+\exp(-\beta x_i))$, and $\lambda > 0$ is the regularization parameter.

- (b) Recall the l_2-regularized quadratic cost function (i.e. ridge regression):

$$J(\beta) = \lambda \|\beta\|_2^2 + \frac{1}{2}\sum_{i=1}^{N}(y_i - \beta x_i)^2$$

In this problem, you will use Newton's method to minimize (a) negative log likelihood and (b) the quadratic function on a small training set. Here's the setup: We have 14 data points (in $\mathbb{R}$), half of class 1, and half of class 0. Here is the data (you may want to draw this on paper to see what the data looks like):

$$X = \big[4, 5, 5.6, 6.8, 7, 7.2, 8, 0.8, 1, 1.2, 2.5, 2.6, 3, 4.3\big]$$

$$Y = \big[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0\big]$$

Standardize X to have mean 0 and variance 1. Notice that this data cannot be separated by a boundary that goes through the origin. To account for this, you should append 1 to each standardized X_i and fit a two-dimensional β vector that includes an offset term.

1. Derive the gradient of the negative log likelihood (problem (a)) as well as the quadratic cost function (problem (b)). Your answer should be two simple matrix-vector expressions. Do NOT write your answer in terms of the individual elements of the gradient vector. (Consider X as a $n \times 2$ matrix and β as a 2×1 vector.)

Solution:

$$(a) \quad \nabla_\beta \mathcal{L} = 2\lambda\beta - X^T(Y - \mu)$$
$$(b) \quad \nabla_\beta \mathcal{L} = 2\lambda\beta - X^T(Y - X\beta)$$

2. State the Hessian in both problems (a) and (b). Again, your answer should be simple matrix-vector expressions.

Solution:

$$(a) \quad \nabla_\beta^2 \mathcal{L} = 2\lambda I + X^T W X$$

where $W = \text{diag}(\mu_1(1-\mu_1), \mu_2(1-\mu_2), \ldots, \mu_n(1-\mu_n))$.

$$(b) \quad \nabla_\beta^2 \mathcal{L} = 2\lambda I + X^T X$$

3. State the update equations for Newton's method in both problems (a) and (b).

Solution:

$$(a) \quad \beta_{t+1} = \beta_t - (2\lambda I + X^T W X)^{-1}(2\lambda\beta_t - X^T(Y - \mu))$$
$$(b) \quad \beta^* = (2\lambda I + X^T X)^{-1} X^T Y$$

4. When $\lambda = 0.07$ and with $\beta_0 = [1, 0]^T$, plot the fit of logistic regression after three iterations of the Newton's algorithm. In the same figure, plot the fit of linear regression as well. Also, write down their corresponding parameter values β. Which one proposes a better fit to the data points?

Solution:

(a)

$$\beta = [2.9278, 0.0632]^T$$

resulting in the solid black line in the figure (b)

$$\beta = [0.4350, 0.4930]^T$$

resulting in the dotted black line in the figure

5. Add an additional data point $(X_{15}, Y_{15}) = (3, 1)$ to the standardized set (X, Y) and repeat the previous part. Briefly explain what you observe.

Solution:

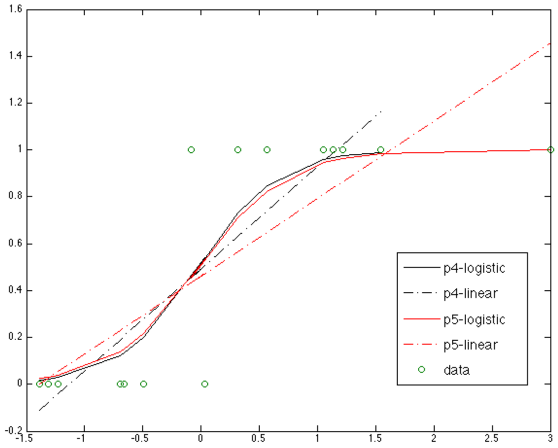
(a)

$$\beta = [2.9195, 0.0620]^T$$

resulting in the solid red line in the figure (b)

$$\beta = [0.3315, 0.4609]^T$$

resulting in the dotted red line in the figure



Refitting the linear regression changes the slope parameter significantly after adding one extra point, which was already classified correctly with the linear regressor in part 4. However, this extra point does not change the regression fit. In general, points that are already classified correctly don't affect the regression fit.

Problem 3: l_1-Regularized Linear Regression

l_1-norm is one of popular regularizers used to enhance the robustness of classification models; also noted as lasso regression. Such a penalty added to the classification loss function results in a spare parameter vector where less informative features get a zero weight.

Assume the training data points are denoted as the rows of a $n \times d$ matrix X and their corresponding output value as an $n \times 1$ vector $\mathbf{y}$. The parameter vector and its optimal value are represented by $d \times 1$ vectors β and β^*, respectively. For the sake of simplicity, assume columns of data have been standardized to have mean 0 and variance 1 as well as uncorrelated (i.e. $X^T X = nI$).

For lasso regression, the optimal parameter vector is given by:

$$\beta^* = \text{argmin}_\beta \{J_\lambda(\beta) = \frac{1}{2}\|\mathbf{y} - X\beta\|_2^2 + \lambda\|\beta\|_1\},$$

where $\lambda > 0$.

1. Show that standardized training data nicely decouples the features, making β^* determined by the i-th feature and the output regardless of other features. To show this, write $J_\lambda(\beta)$ in the following form for appropriate functions g and f:

$$J_\lambda(\beta) = g(\mathbf{y}) + \sum_{i=1}^{d} f(X_i, \mathbf{y}, \beta_i, \lambda)$$

where X_i is the i-th column of X.

Solution: Considering X_i as the i-th column of X:

$$\frac{1}{2}\|\mathbf{y} - X\beta\|_2^2 + \lambda\|\beta\|_1\} = \frac{1}{2}\mathbf{y}^T\mathbf{y} + \sum_{i=1}^{d}\{-\mathbf{y}^T X_i\beta_i + \frac{n}{2}\beta_i^2 + \lambda|\beta_i|\}$$

2. Assume that $\beta_i^* > 0$, what is the value of β_i^* in this case?

Solution: If $\beta_i^* > 0$, then we want to minimize

$$-\mathbf{y}^T X_i\beta_i + \frac{n}{2}\beta_i^2 + \lambda\beta_i$$

Take the derivative and equate to zero, we have:

$$\beta_i^* = \frac{1}{n}(\mathbf{y}^T X_i - \lambda)$$

3. Assume that $\beta_i^* < 0$, what is the value of β_i^* in this case?

Solution: If $\beta_i^* < 0$, then we want to minimize

$$-\mathbf{y}^T X_i \beta_i + \frac{n}{2}\beta_i^2 - \lambda \beta_i$$

Take the derivative and equate to zero, we have:

$$\beta_i^* = \frac{1}{n}(\mathbf{y}^T X_i + \lambda)$$

4. From 2 and 3, what is the condition for β_i^* to be zero?

Solution: From the previous parts, we know $\beta_i^* = 0$ if none of the above conditions hold, that is;

$$\mathbf{y}^T X_i + \lambda \geq 0, \quad \mathbf{y}^T X_i - \lambda \leq 0$$

Combining them, we get

$$-\lambda \leq y^T X_i \leq \lambda$$

5. Now consider the ridge regression problem, mentioned in problem 2, where the regularization term is replaced by $\lambda\|\beta\|_2^2$. What is the condition for $\beta_i^* = 0$? How does it differ from the condition you obtained in part 4?

Solution: If the lasso is replaced by $\lambda\|\beta\|_2^2$, the optimization problem regarding β_i is given by:

$$-\mathbf{y}^T X_i \beta_i + \frac{n}{2}\beta_i^2 + \lambda \beta_i^2$$

take the derivative and equate to zero:

$$\beta_i^* = \frac{\mathbf{y}^T X_i}{n + 2\lambda}$$

It is equal to zero if $\mathbf{y}^T X_i = 0$ or λ goes to infinity. In contrast, $\beta_i^* = 0$ when $|\mathbf{y}^T X_i| < \lambda$ in Lasso regression. This is why the l_1-norm regularization encourages sparsity.

Problem 4: Spam classification using Logistic Regression

The spam dataset given to you as part of the homework in `spam.mat` consists of 4601 email messages, from which 57 features have been extracted as follows:

- 48 features giving the percentage (0 - 100) of words in a given message which match a given word on the list. The list contains words such as business, free, george, etc. (The data was collected by George Forman, so his name occurs quite a lot!)

- 6 features giving the percentage (0 - 100) of characters in the email that match a given character on the list. The characters are ; ([! $ # .
- Feature 55: The average length of an uninterrupted sequence of capital letters
- Feature 56: The length of the longest uninterrupted sequence of capital letters
- Feature 57: The sum of the lengths of uninterrupted sequence of capital letters

The dataset consists of a training set size 3450 and a test set of size 1151. One can imagine performing several kinds of preprocessing to this data. Try each of the following separately:

i) Standardize the columns so they all have mean 0 and unit variance.

ii) Transform the features using $log(x_{ij} + 0.1)$.

iii) Binarize the features using $\mathbb{I}(x_{ij} > 0)$.

Note that we haven't provided you with test labels for this homework. This means you won't be able to verify the test performance of your classifier like you've done before. Instead, we'll be using Kaggle, but more on that later. For this homework, you need to do the following:

1. Derive the gradient descent equations for logistic regression with l_2 regularization and write them down (you can just state it if your derivation is in the previous problem).

   Choose a reasonable regularization parameter value, and plot the training loss (the negative log likelihood of the training set) vs the number of iterations. You should have one plot for each preprocessing method.

   Note: One iteration here amounts to scanning through the whole training data and computing the full gradient.

2. Derive stochastic gradient descent equations for l_2 regularized logistic regression. Plot the training loss vs number of iterations (again, you should have one plot for each preprocessing method). Do you see any differences from the corresponding curve from (1)? If so, why?

   Note: One iteration here corresponds to processing just one data point.

3. Instead of a constant learning rate (η), repeat (2) where the learning rate decreases as $\eta \propto 1/t$ for the t^{th} iteration. Plot the training loss vs number of iterations. Is this strategy better than having a constant η?

4. Now, tune your classfier choosing the most appropriate of the 3 preprocessing methods and tuning the regularization parameter. Submit your results to Kaggle. Your classifier, when given the test points, should output a CSV file (there is a sample one on Kaggle). You'll upload this CSV file to Kaggle where it'll be scored with both a public test set, and a private test set. You will be able to see only your public score. You can only submit twice per day, so get started early! In your writeup, describe the process you used to decide which parameters to use for your best classifier.

   Beware of overfitting!

NOTE: You are NOT supposed to use any kind of software package for logistic regression!

Submission Instructions

You will submit:

- A PDF write-up containing your *answers, code, and plots* to Gradescope.
- A zip file of your *code* to Gradescope.
- Your predictions *CSV* to Kaggle.