EECS 545 – Machine Learning - Homework #3

Due: 11:59pm 11/08/2017

Homework Policy: Working in groups is allowed, but each member must submit their own writeup. Please write the members of your group on your solutions (maximum allowed group size is 4). Homeworks will be submitted via Gradescope (https://gradescope.com/) as pdf files (including your code).

1) **Logistic Regression (20 pts).**

In logistic regression, we are given training data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^m, y_i \in \{0, 1\}$. We want to find parameter $\mathbf{w} \in \mathbb{R}^m$ that minimizes the negative log-likelihood function (a.k.a cross entropy loss):

$$\ell(\mathbf{w}) = \sum_{i=1}^{n} \ell_i(\mathbf{w}) \tag{1}$$

$$= \sum_{i=1}^{n} -y_i \log h(\mathbf{x}_i) - (1 - y_i) \log(1 - h(\mathbf{x}_i)) \tag{2}$$

Here $h(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})}$ is the predicted probability of $\mathbf{x}$ being from class 1. i.e, $P(y = 1|\mathbf{x}; \mathbf{w}) = h(\mathbf{x})$

(a) (5 pts) Find the expression for the gradient $\nabla\ell(\mathbf{w})$.

(b) (5 pts) Find the expression of Hessian matrix H for $\ell(\mathbf{w})$, and show that it is positive semi-definite and thus $\ell(\mathbf{w})$ is convex and has no local minima other than the global one.

(c) (10 pts) With the H you calculated in part(b), Implement the Newton's method

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - H^{-1}\nabla\ell(\mathbf{w}) \tag{3}$$

for optimizing $\ell(\mathbf{w})$ and apply it to binary classification problem using files in `classification.zip`. `train_features.dat` contains *two* columns, adding the intercept term, each training point can be represented as $\mathbf{x}_i = \begin{bmatrix}1 & x_1^{(i)} & x_2^{(i)}\end{bmatrix}^T$, `train_labels.dat` contains the corresponding labels $y_i \in \{0, 1\}$. Initialize $\mathbf{w}^{(0)} = 0 = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$. To determine convergence, the weights $\mathbf{w}$ are updated until $|\ell(\mathbf{w})^{(t+1)} - \ell(\mathbf{w})^{(t)}| < \epsilon$, where ϵ is a very small number. For this problem, set $\epsilon = 10^{-8}$. Report the coefficients $\mathbf{w}$, including the intercept term, resulting from your fit. Plot the training error $\ell(w^{(t)})$ and test error at each iteration using the files `test_features.dat` and `test_labels.dat`. How many steps does your algorithm take until convergence?

2) **Maximum Likelihood Estimation (15 pts).**

Consider a random variable $\mathbf{x}$ (possibly a vector) whose distribution (pdf or pmf) belongs to a parametric family. The density or mass function may be written as $f(\mathbf{x}; \theta)$, where θ is called the parameter, and can be either a scalar or vector. For example, in the univariate Gaussian distribution, θ can be a two dimensional vector consisting of the mean and the variance. Suppose the parametric family is known, but the value of the parameter is unknown. It is often of interest to estimate this parameter from observations of $\mathbf{x}$.

Maximum likelihood estimation is one of the most important parameter estimation techniques. Let $x_1, \dots, x_n$ be i.i.d. (independent and identically distributed) random variables distributed according to $f(\mathbf{x}, \theta)$. By independence, the joint distribution of the observations is the product

$$L(\theta) = \prod_{i=1}^{n} f(\mathbf{x}_i; \theta).$$

Viewed as a function of θ, this quantity is called the *likelihood* of θ. It is often more convenient to work with the *log-likelihood*,

$$\ell(\theta) = \sum_{i=1}^{n} \log f(\mathbf{x}_i; \theta).$$

A maximum likelihood estimate (MLE) of θ is any parameter

$$\hat{\theta} \in \arg\max_{\theta} \sum_{i=1}^{n} \log f(\mathbf{x_i}; \theta).$$

If the maximizer is unique, $\hat{\theta}$ is called the maximum likelihood estimate of θ.

(a) (5 points) Consider i.i.d. 1-dimensional Gaussian random variables $\mathbf{x}_1, \dots, \mathbf{x}_n$ each with mean α and variance σ^2. That is,

$$f(x; \alpha, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\alpha)^2}{2\sigma^2}\right).$$

Find the maximum likelihood estimates of α and σ^2.

(b) (10 pts) Let $\mathbf{x}_1, \dots, \mathbf{x}_n$ be i.i.d. d-dimensional Gaussian random variables distributed according to $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. That is,

$$f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right).$$

Find the maximum likelihood estimated of the mean vector $\boldsymbol{\mu}$. (You can assume the covariance matrix $\boldsymbol{\Sigma}$ is known.)

3) **Information Theory (20 pts).**

Many algorithms for learning probabilistic models are best understood in terms of *information theory.* Consequently, it is useful to understand and manipulate these quantities in different contexts.

(a) (5 pts) Show that

$$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

where $I(X, Y)$ is the mutual information of X and Y, $H(X)$ is the entropy of X, and $H(X|Y)$ is the conditional entropy of X given Y. The entropy and conditional entropy are defined as

$$H(X) \triangleq -\int p(X) \ln p(X) dX,$$

$$H(X|Y) \triangleq -\int p(X, Y) \ln p(X|Y) dX dY.$$

(b) (5 pts) Prove that if X and Y are related by a bijection f (i.e., $X = f(Y)$ and $Y = f^{-1}(X)$), then $I(X, Y) = H(X) = H(Y)$.

(c) (5 pts) Suppose we observe N samples $\mathcal{D} = (x_1, \dots, x_N)$ from some unknown distribution. Define $\hat{p}(x)$ to be the empirical probability density estimate,

$$\hat{p}(x) \triangleq \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[x = x_i]$$

Let $q(x|\theta)$ be the probability density corresponding to some known probabilistic model with parameter θ. Show that the minimum Kullback–Leibler divergence

$$\min_\theta D_{KL}(\hat{p}||q) \triangleq \min_\theta - \int \hat{p}(x) \ln \frac{q(x|\theta)}{\hat{p}(x)} dx$$

is obtained by the maximum likelihood estimate θ_{ML} given the data $\mathcal{D}$.

(d) (5 pts) Let $p = \mathcal{N}(\mu, \sigma^2)$ be a Gaussian distribution and q be any probability density with mean μ and variance σ^2. Prove that $H(q) \leq H(p)$, that is, the Gaussian distribution has maximum entropy among all distributions of the same variance. *Hint: Refer to the textbook (PRML by Bishop §1.6) for a proof outline.*

4) **Weighted Least Squares Regression (20 pts).**

In this problem, you will implement a locally-weighted version of the Least Squares Regression (Linear Regression) where different training points are *weighed* differently according to the query point. For this purpose, we find a solution for the following problem:

$$L(\beta, b) = \sum_{i=1}^{n} c_i (y_i - \beta^T \mathbf{x}_i - b) \tag{4}$$

$$\mathbf{w} = \begin{bmatrix} b & \beta^T \end{bmatrix}^T = \arg\min_{\beta, b} L(\beta, b) \tag{5}$$

$$= \arg\min_{\beta, b} \sum_{i=1}^{n} c_i (y_i - \beta^T \mathbf{x}_i - b) \tag{6}$$

(a) (10 pts) Find a closed form solution for $\mathbf{w}$ in terms of the data matrix X and the weight matrix C. Argue that when $c_i = 1 \forall i$, the above problem is the same as finding the Maximum Likelihood Estimate of $\mathbf{w} = \begin{bmatrix} b & \beta^T \end{bmatrix}^T$, with y_i being modeled as,

$$y_i = \beta^T \mathbf{x}_i + b + \epsilon_i \quad \forall i \tag{7}$$

where $\epsilon_i | \mathbf{x}_i \sim \mathcal{N}(0, \sigma^2)$ are i.i.d. random variables and $\mathbf{x}_i$ has been observed for each i.

(b) (10 pts) Repeat part (a), but this time with $\epsilon_i | \mathbf{x}_i \sim \mathcal{N}(0, \sigma_i^2)$, which means that the noise has different variance for each i

5) **Support Vector Machine (25 points).**

Recall that maximizing the soft-margin in SVM is equivalent to the following minimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & t^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \qquad (i = 1, \ldots, N) \end{aligned} \tag{8}$$

Equivalently, we can solve the following unconstrained minimization problem:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\max\left(0, 1 - t^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)\right) \tag{9}$$

(a) (5 pts) Prove that minimization problem (8) and (9) are equivalent.

(b) (7 pts) Let $(\mathbf{w}^*, b^*, \boldsymbol{\xi}^*)$ be the solution of minimization problem (9). Show that if $\xi_i^* > 0$, then the distance from the training data point $\mathbf{x}^{(i)}$ to the margin hyperplane $t^{(i)}((\mathbf{w}^*)^T\mathbf{x} + b^*) = 1$ is proportional to ξ_i^*.

(c) (3 pts) Prove that when $C \to \infty$, (9) is equivalent to the hard-margin SVM.

(d) Download the `diabetes_scaled.csv` file from Canvas. This file contains feature vectors $\mathbf{X}$ and labels $\mathbf{y}$ for a classification problem. The first column are labels and the other 8 columns are feature vectors. There are 768 labeled feature vectors. The features are

```
1.  Number of times pregnant
2.  Plasma glucose concentration
3.  Diastolic blood pressure
4.  Triceps skin fold thickness
5.  2-hour serum insulin
6.  Body mass index
7.  Diabetes pedigree function
8.  Age
```

The binary labels are

```
+1 = tested positive for diabetes
-1 = non-positive
```

(i) (7 pts) In this problem, we will train a soft-margin SVM using SVM solvers of Matlab toolbox or Scikit-learn package. You should first reserve the last 268 instances for testing. The goal is to minimize the probability of error on the test data. Train a soft-margin SVM using a 5-fold cross validation to select the parameter C. Search C over the grid `linspace(0.1, 2, 20)`. Report the best C and the best classification accuracy.

(ii) (3 pts) Train a hard-margin SVM by setting C as `1e6`. Report the classification accuracy and compare with the best accuracy you just got in the previous problem. Give a possible reason of the difference.

Comments:

- For Matlab users, please use `fitclinear` to train a SVM.
- For Python users, please use `svm.LinearSVC` of Scikit-learn package to train a SVM.
- For both Matlab and Python, set the random seed to 42 for repeatable results.
- Make sure you use ℓ_1 penalty and hinge loss.