CS 6780: Advanced Machine Learning
Homework 1

Due date: **Sep 21, 2010**
(Physical submission in the class or give to TA by noon.)

Note:
1. Write your name and net-id in BIG letters on the first page. Do not write anything else on the first-page of the homework.
2. For derivation questions, you need to show all the necessary steps. Just writing the answers won't do.
3. For data analysis / programming parts, you need to submit: (a) the values and plots requested, and (b) also provide the code in the appendix.

1 Logistic Regression

Assume $p(y|x;\theta) = (h_\theta(x))^y(1-h_\theta(x))^{1-y}$, where $h_\theta(x) = \frac{1}{1+\exp -\theta^T x}$. For learning the optimal value of θ, we will use gradient descent.

(a) Assume that the m training examples were generated independently. Use the maximum likelihood principle to derive the update for the parameters θ using (batch) gradient descent. (9 points)

(b) Write the algorithm (i.e., pseudo-code) for *stochastic* gradient descent for parameters θ. (6 points)

2 Exponential Families: Poisson

Consider the problem of website optimization, and the task is to estimate the hits y on the website in an hour (y is an integer, $y \geq 0$). A distribution

suitable for this problem is Poission distribution, which is given as: $p(y;\lambda) = \frac{e^{-\lambda}\lambda^y}{y!}$.

(a) Express the Poission distribution as a Exponential Family distribution, and write down the value of $T(y)$, $b(y)$ and $a(\eta)$. (6 points)

(b) Construct a GLM for this problem. I.e., assume that $y|x;\theta \sim \text{ExponentialFamily}(\eta)$, where the exponential family to be used would be Poisson. (6 points)

(c) Write six features (in informal words) that you would use as x in this problem. E.g., "time of the day". (3 points)

3 Regression: Weight priors

Often the amount of data we have (say m data-points) is not very large as compared to the number of dimensions n in the feature vector. I.e., $X \in \Re^{m\times n}$ and $n \approx m$ (or sometimes even $n > m$).

(a) In such cases, we use a method called "Ridge Regression." Here, instead of minimizing $(X\theta - y)^T(X\theta - y)$ as we did in the class, we will minimize $J_1(\theta) = (X\theta - y)^T(X\theta - y) + \lambda\theta^T\theta$, for $\lambda \in \Re, \lambda \geq 0$. Derive the value of θ that minimizes $J_1(\theta)$ in closed form. (5 points)

(b) In the maximum likelihood method, we choose θ as follows:

$$\theta_{ML} = \arg\max_\theta \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) \tag{1}$$

Here θ is considered *constant-valued but unknown* (called **frequentist** view).

An alternative view is to consider θ as a random variable whose value is unknown (called **Bayesian** view). In this philosophy, we can specify a **prior distribution** $p(\theta)$ on θ that expresses our "prior beliefs" about the parameters. We estimate θ by using an approximation called **MAP (maximum a posteriori)** estimate as:

$$\theta_{MAP} = \arg\max_\theta \prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta)p(\theta) \tag{2}$$

Assume $y|x,\theta \sim N(x^T\theta,\sigma^2)$ and $\theta \sim N(0,\tau^2 I)$. Is this maximization problem equivalent to minimizing $J_1(\theta)$ (in the previous part) for some value of λ? If yes, express λ as a function of σ and τ. (10 points)

(c) Another common method minimizes the sum of absolute of the terms in θ (as compared to sum of squares of terms in θ in part a). I.e., $J_2(\theta) = (X\theta - y)^T(X\theta - y) + |\theta|$.

- To get an optimal value of θ, we will use gradient descent. Unfortunately, the absolute value function θ is not differentiable; to address this we will use the following approximation:
  I.e., $J_2(\theta) \approx J_3(\theta) = (X\theta - y)^T(X\theta - y) + \sum_{i=1}^{n} f(\theta_i)$,
  where $f(z) = \frac{1}{\beta}\left[\log(1 + \exp(-\beta z)) + \log(1 + \exp(\beta z))\right]$
  for some large β (given and fixed) and $z \in \Re$.

  Derive an update rule that minimizes: $J_3(\theta)$ (8 points)

- Another method is to use a convex optimization software. Write the minimization of $J_2(\theta)$ as a standard convex minimization problem. Prove that it is a convex optimization problem.

  Write minimization of $J_2(\theta)$ as a Quadratic Program[1]! (7 points)

(d) Now we are trying to predict housing prices with 13 features.[2] Download the matlab file `regression.zip` from the course materials page. (We've given `regression.m`, a starter file that you could modify). You will get matrices X_{train}, X_{test} and y_{train}, y_{test}. Each row of X is a sample, and corresponding price is in y. Compute θ using ridge regression derived above. Compute root mean squared error $||y - \hat{y}||_2$ on the test data by predicting y_{test} using X_{test}. Report the error for several values of λ. Plot y_{test} and $\hat{y}_{test}$ on the same graph for the optimal value of λ. (8 points)

(e) So far, we were using the 13 features. Expand this feature set to 20 features by adding 7 more features (e.g., by taking squares, logs, exps, etc.) to try to reduce the training error. The parameters that you can modify here are choice of features (computed as a function of the given 13 features) and value of λ. Write down those 7 features, and also give the errors obtained on test and training set. (7 points)

[1] Quadratic program can actually be solved with a single function call in Matlab: `quadprog`.

[2] More info about data: http://archive.ics.uci.edu/ml/datasets/Housing

4 Gaussian Discriminant Analysis

We have

$$y \sim \text{Bernoulli}(\phi) \tag{3}$$
$$x|y = 0 \sim N(\mu_0, \Sigma_0) \tag{4}$$
$$x|y = 1 \sim N(\mu_1, \Sigma_1) \tag{5}$$

(a) Show that

$$p(y = 1|x; \phi, \Sigma_1, \Sigma_2, \mu_0, \mu_1) = \frac{1}{1 + \exp(-\theta^T x)} \tag{6}$$

for some θ as a function of $\phi, \Sigma_1, \Sigma_2, \mu_0, \mu_1$. Write this function.[3] (9 points)

(b) The previous part indicates that Gaussian Discriminant Analysis and Logistic Regression are the same. Is this true? If yes, why? If not, what is the difference. Provide explanation in less than 100 words. (4 points)

(c) After we have finished training (i.e., estimated $\mu_0, \mu_1, \Sigma_0, \Sigma_1, \phi$), we want to classify the new data point x at test time. Here is a *pseudo-code*:
`boolean classifyTestData(`μ_0`,` μ_1`,` Σ_0`,` Σ_1`,` ϕ`,` x`)`
`{ if ( XXXXXX` $\geq$ `0 ) return 1; }`
Replace XXXXXX with an expression of the input variables. (4 points)

(d) Consider $x \in \Re^2$, i.e., a classification problem with two features. What would the decision boundary (i.e., $p(y = 1|x) = 0.5$) look like for the following cases. (Draw contours on paper for the Gaussians and a line for the decision boundary; dont write code for this.) (8 points)

- $\mu_0 = (0, 0)$, $\mu_1 = (1, 1)$ and $\Sigma_0 = \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.
- $\mu_0 = (0, 0)$, $\mu_1 = (0, 1)$ and $\Sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\Sigma_1 = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$.
- $\mu_0 = \mu_1 = (0, 0)$ and $\Sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\Sigma_1 = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$.
- $\mu_0 = \mu_1 = (0, 0)$ and $\Sigma_0 = \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}$ and $\Sigma_1 = \begin{pmatrix} 4 & -1 \\ -1 & 4 \end{pmatrix}$.

[3] You need to assume that $\Sigma_1 = \Sigma_2$.