CPSC-540 Machine Learning 2011

Homework # 4

NAME:_____

Signature:_____

STD. NUM: _____

General guidelines for homeworks:

You are encouraged to discuss the problems with others in the class, but all write-ups are to be done on your own.
Homework grades will be based not only on getting the "correct answer," but also on good writing style and clear presentation of your solution. It is your responsibility to make sure that the graders can easily follow your line of reasoning.
Try every problem. Even if you can't solve the problem, you will receive partial credit for explaining why you got stuck on a promising line of attack. More importantly, you will get valuable feedback that will help you learn the material.
Please acknowledge the people with whom you discussed the problems and what sources you used to help you solve the problem (e.g. books from the library). This won't affect your grade but is important as academic honesty.
When dealing with python exercises, please attach a printout with all your code and show your results clearly.

1. **Logistic regression derivations** Derive the expressions, shown in class, for the gradient and Hessian of the negative log-likelihood of the logistic regression model.

2. **Logistic regression implementation**

In this question you will implement logistic regression in order to classify emails as spam or not spam. The dataset you will use for this problem consists of 4601 labelled emails along with 57 features. After downloading the data from the course website, load the data using

```
# load the data.
X = np.loadtxt('spambase.data', delimiter=',', skiprows=1)

# split X/y and add a constant column to X.
y = X[:,-1]
X = X[:,:-1]
X = np.c_[np.ones(X.shape[0]), X]
```

Here X consists of the 57 features and an additional constant feature to act as the bias and y consists of a label for each email, where $y_i = 1$ means that the email is spam. Next, split the data into a test and training set, using the first 4000 elements as our training set.

```
Xtrain, Xtest = X[0:4000], X[4000:]
ytrain, ytest = y[0:4000], y[4000:]
```

Now, implement a logistic regression solver using *iteratively reweighted least squares* using the following code stub:

```
def irls(X, y):
    theta = np.zeros(X.shape[1])
    theta_ = np.inf
    while max(abs(theta-theta_)) > 1e-6:
        theta_ = theta.copy()
        ????
    return theta
```

What feature is most indicative of an email not being spam? Note: the label for feature i is given by `names[i]` where

```
names = open('spambase.data').readline().split(',')
```

If we use a cutoff of .5, what is the misclassification rate of the training set and of the test set?

3. **Convolution of two Gaussians**

Prove the following statement about the convolution of two Gaussians:

$$\begin{aligned} p(\mathbf{y}|X) &= \int \mathcal{N}(\mathbf{y}|\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}_0, \boldsymbol{\Sigma}_y)\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)d\boldsymbol{\theta} \\ &\vdots \\ &= \mathcal{N}(\mathbf{y}|\mathbf{X}\boldsymbol{\mu}_\theta + \boldsymbol{\theta}_0, \boldsymbol{\Sigma}_y + \mathbf{X}\boldsymbol{\Sigma}_\theta\mathbf{X}^T) \end{aligned}$$

Hint: Create the vector $\mathbf{z} = [\mathbf{y} \;\; \boldsymbol{\theta}]^T$, group the exponents of the prior and likelihood so that you have a multivariate Gaussian in terms of $\mathbf{z}$. This will enable you to know the mean and **inverse** covariance of $\mathbf{z}$. However, what you need is the covariance in order to be able to read off the marginal distribution of $\mathbf{y}$ using the definitions of mean and Covariance of a multivariate Gaussian. To get an expression for the covariance, in terms of the 4 blocks of the inverse covariance, use the Schur Complement (see next page and Wikipedia derivation).

Consider a general partitioned matrix

$$\mathbf{M} = \begin{pmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{pmatrix}$$

where we assume $\mathbf{E}$ and $\mathbf{H}$ are invertible. We have

$$\begin{aligned} \mathbf{M}^{-1} &= \begin{pmatrix} (\mathbf{M}/\mathbf{H})^{-1} & -(\mathbf{M}/\mathbf{H})^{-1}\mathbf{F}\mathbf{H}^{-1} \\ -\mathbf{H}^{-1}\mathbf{G}(\mathbf{M}/\mathbf{H})^{-1} & \mathbf{H}^{-1} + \mathbf{H}^{-1}\mathbf{G}(\mathbf{M}/\mathbf{H})^{-1}\mathbf{F}\mathbf{H}^{-1} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{E}^{-1} + \mathbf{E}^{-1}\mathbf{F}(\mathbf{M}/\mathbf{E})^{-1}\mathbf{G}\mathbf{E}^{-1} & -\mathbf{E}^{-1}\mathbf{F}(\mathbf{M}/\mathbf{E})^{-1} \\ -(\mathbf{M}/\mathbf{E})^{-1}\mathbf{G}\mathbf{E}^{-1} & (\mathbf{M}/\mathbf{E})^{-1} \end{pmatrix} \end{aligned}$$

where

$$\begin{aligned} \mathbf{M}/\mathbf{H} &:= \mathbf{E} - \mathbf{F}\mathbf{H}^{-1}\mathbf{G} \\ \mathbf{M}/\mathbf{E} &:= \mathbf{H} - \mathbf{G}\mathbf{E}^{-1}\mathbf{F} \end{aligned}$$

We say that $\mathbf{M}/\mathbf{H}$ is the Schur complement of $\mathbf{M}$ wrt $\mathbf{H}$. This result is useful for very fast matrix inversion via low rank updates.

Now, Suppose $\mathbf{x} = [\mathbf{x}_1 \;\; \mathbf{x}_2]^T$ is jointly Gaussian with parameters

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \;\; \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}$$

Then, $p(\mathbf{x}_2)$ is obtained by extracting the rows and columns from $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ corresponding to $\mathbf{x}_2$ (using the definitions of mean and Covariance of a multivariate Gaussian):

$$p(\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_2 | \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22})$$

The Schur complement allows us to derive the conditional distribution $p(\mathbf{x}_1|\mathbf{x}_2)$:

$$\begin{aligned} p(\mathbf{x}_1|\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_1 | \boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \\ \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \end{aligned}$$

In class, we completed squares to get this result, but it can also be derived via the Schur complement (though the derivation is longer and more tedious). However, once we know this result, we can use it anytime we need to quickly derive other techniques, such as Kalman filtering and Gaussian processes.