CS340 Winter 2010: HW5
Out Monday 21st March, due Monday 4th April

1 Gradient and Hessian of log-likelihood for logistic regression

1. Let $g(z) = (1+e^{-z})^{-1}$ be the logistic function. Show that

$$\frac{dg(z)}{dz} = g(z)(1-g(z)).$$

2. Consider the following logistic regression model where

$$\begin{aligned} p(y=1|\mathbf{x},\mathbf{w}) &= 1-p(y=0|\mathbf{x},\mathbf{w}) \\ &= g\left(\mathbf{w}^{\mathrm{T}}\Phi(\mathbf{x})\right) \end{aligned}$$

with

$$\mathbf{w}^{\mathrm{T}}\Phi(\mathbf{x}) = \sum_{k=1}^{m} w_k \Phi_k(\mathbf{x}).$$

Assuming, as always, that the data $\{\mathbf{x}^i, y^i\}_{i=1}^N$ are independent, establish that the gradient

$$\nabla L(\mathbf{w}) := \left(\frac{\partial L(\mathbf{w})}{\partial w_1} \cdots \frac{\partial L(\mathbf{w})}{\partial w_m}\right)^{\mathrm{T}}$$

of the conditional log-likelihood

$$L(\mathbf{w}) = \log\ p\left(\left\{y^i\right\}_{i=1}^N \middle| \left\{\mathbf{x}^i\right\}_{i=1}^N, \mathbf{w}\right)$$

is given by

$$\nabla L(\mathbf{w}) = \sum_{i=1}^{N} \left(y^i - g\left(\mathbf{w}^{\mathrm{T}} \Phi\left(\mathbf{x}^i\right)\right)\right) \Phi\left(\mathbf{x}^i\right) = \Phi^{\mathrm{T}}(\mathbf{y} - \boldsymbol{\mu})$$

where $[\Phi]_{i,j} = \Phi_j(\mathbf{x}^i)$, $\mathbf{y} = \left(y^1 \ \cdots \ y^N\right)^{\mathrm{T}}$ and $\boldsymbol{\mu} = \left(g\left(\mathbf{w}^{\mathrm{T}}\Phi\left(\mathbf{x}^1\right)\right) \ \cdots \ g\left(\mathbf{w}^{\mathrm{T}}\Phi\left(\mathbf{x}^N\right)\right)\right)^{\mathrm{T}}$.

3. It can be shown that the Hessian matrix $\nabla^2 L(\mathbf{w})$ defined by

$$\left[\nabla^2 L(\mathbf{w})\right]_{k,l} = \frac{\partial^2 L(\mathbf{w})}{\partial w_k \partial w_l}$$

for $k, l = 1, ..., m$ satisfies

$$\nabla^2 L(\mathbf{w}) = -\Phi^{\mathrm{T}} U \Phi$$

with U a diagonal matrix with diagonal element

$$[U]_{i,i} = g\left(\mathbf{w}^{\mathrm{T}}\Phi\left(\mathbf{x}^i\right)\right)\left[1 - g\left(\mathbf{w}^{\mathrm{T}}\Phi\left(\mathbf{x}^i\right)\right)\right].$$

Show that $\nabla^2 L(\mathbf{w})$ is negative semi-definite; i.e. for any column vector $\mathbf{v}$

$$\mathbf{v}^{\mathrm{T}} \ \nabla^2 L(\mathbf{w}) \ \mathbf{v} \leq 0.$$

4. Newton's method is a generic (second order) optimization algorithm which converges faster than the simple gradient algorithm discussed in class. Applied to the maximization of the conditional log-likelihood $L(\mathbf{w})$, Newton's algorithm proceeds as follows at iteration t

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} - \left[\nabla^2 L\left(\mathbf{w}^{(t-1)}\right)\right]^{-1} \nabla L\left(\mathbf{w}^{(t-1)}\right).$$

Show that Newton's algorithm can be rewritten as

$$\mathbf{w}^{(t)} = \left(\Phi^{\mathrm{T}} U^{(t-1)} \Phi\right)^{-1} \Phi^{\mathrm{T}} U^{(t-1)} \left(\Phi \mathbf{w}^{(t-1)} + \left[U^{(t-1)}\right]^{-1} \left(\mathbf{y} - \boldsymbol{\mu}^{(t-1)}\right)\right)$$

where $U^{(t-1)}$ and $\boldsymbol{\mu}^{(t-1)}$ corresponds to U and $\boldsymbol{\mu}$ computed using $\mathbf{w}^{(t-1)}$.

5. Newton's algorithm can be unstable when $\nabla^2 L(\mathbf{w})$ is singular or close to singular. Assume you are introducing a Gaussian prior on the parameters $\mathbf{w}$ so that

$$p(\mathbf{w}|\lambda) = \prod_{k=1}^{m} p(w_k|\lambda)$$

where w_k follows a Gaussian distribution of mean 0 and variance λ^{-1}. Establish the expression of the Newton's algorithm to maximize the associated log-posterior density of the weights.

2 Regularizing separate terms in logistic regression

1. Consider the training data in Figure 1. We fit the model $p(y = 1|\mathbf{x}, \mathbf{w}) = g(w_0 + w_1 x_1 + w_2 x_2)$ where $g(\cdot)$ is the logistic function. Suppose we fit the model by maximum likelihood. Sketch a possible decision boundary corresponding to $\widehat{\mathbf{w}}$. Copy the figure first (a rough sketch is enough), and then superimpose your answer on your copy, since you will need multiple versions of this figure). Is your answer (decision boundary) unique? How many classification errors does your method make on the training set?

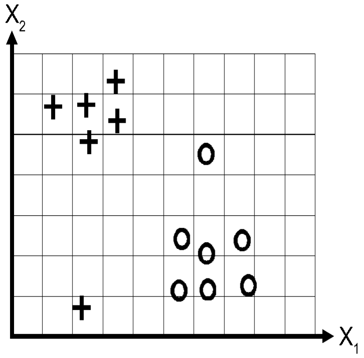

Figure 1: Training Data with Two Possible Classes: cross and circle

2. Now suppose we regularize only the w_0 parameter; i.e. we now minimize

$$-L(\mathbf{w}) + \lambda w_0^2$$

where $L(\mathbf{w})$ is the conditional log-likelihood. Suppose λ is a very large number, so we regularize w_0 all the way to 0, but all other parameters are unregularized. Sketch a possible decision boundary. How many classification errors does your method make on the training set? Hint: consider the behavior of simple linear regression, $w_0+w_1x_1+w_2x_2$ when $x_1 = x_2 = 0$.

3. Now suppose we heavily regularize only the w_1 parameter, i.e., we minimize

$$-L(\mathbf{w}) + \lambda w_1^2$$

Sketch a possible decision boundary. How many classification errors does your method make on the training set?

4. Now suppose we heavily regularize only the w_2 parameter, i.e., we minimize

$$-L(\mathbf{w}) + \lambda w_2^2$$

Sketch a possible decision boundary. How many classification errors does your method make on the training set?

3 Spam Classification using Logistic Regression

Consider the email spam data set `spamData.mat` downloadable on the course webpage. This consists of 4601 email messages, from which 57 features have been extracted. These are as follows:

- 48 features giving the percentage (0 to 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 to 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.

Load the data from `spamData.mat`, which contains a training set (of size 3065) and a test set (of size 1536).

One can imagine performing several kinds of preprocessing to this data. Try each of the following separately:

1. Standardize the columns so they all have mean 0 and unit variance.
2. Transform the features using $x_{ij} \leftarrow \log(x_{ij} + 0.1)$.
3. Binarize the features using $x_{ij} \leftarrow \mathbb{I}(x_{ij} > 0)$.

For each version of the data, fit a logistic regression model. Use cross validation to choose the strength of the Gaussian prior regularizer. Report the mean error rate on the training and test sets. You should get numbers similar to this:

method	train	test
standardized	0.082	0.079
log	0.052	0.059
binary	0.065	0.072

(The precise values will depend on what regularization value you choose.) Turn in your code and numerical results.