CSI 5325 Assignment 1

Greg Hamerly

Assigned: 1/20/2015; Due: 2/3/2015

Instructions

For this assignment, you should write your derivations in LaTeX. They should be concise and easy to follow, with appropriate English descriptions rather than just mathematics (i.e. explain what you are doing as necessary).

Write any programs in Matlab (or Octave), and attach their code with your assignment. Include any relevant figures and graphs (plotted using Matlab / Octave) in your writeup.

Submit your assignment in two ways: hardcopy (in class) and by email (to hamerly@cs.baylor.edu). The email should contain a single attachment as a ZIP file. It should be named "lastname-xx.zip", where lastname is your last name, and xx is the number of the assignment.

Finally, please keep your submitted email attachments small. In particular, make sure you are only submitting things that are necessary (omit datasets I gave you, compiled programs, etc.). Also, try to keep your graphics small by using vector (rather than bitmap) formats (e.g. PDF or EPS rather than JPG or BMP). Vector graphics are generally smaller in size and better quality than bitmap.

1 Maximum likelihood for linear regression (10 points)

Consider the simple linear model *with just one scalar parameter* θ:

$$h_\theta(x) = \theta x.$$

In other words, h has no intercept term, just a slope for a scalar input variable. Assume also that the noise $\epsilon = y - h_\theta(x)$ has distribution $\epsilon \sim N(0, \sigma^2)$; in other words, it is an i.i.d. Gaussian random variable with mean 0 and variance σ^2, so that $p(\epsilon) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$.

(a) Write down the likelihood function $L(\theta)$ for a dataset $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$. Use summation notation, rather than matrix notation. Then take the logarithm of the likelihood, $\ell(\theta) = \log(L(\theta))$, and simplify it.

(b) Take the derivative of $\ell(\theta)$ with respect to θ. Then use this to find the maximum likelihood value of θ. Compare your answer to the more general matrix-based solution we derived in class for the n-dimensional case: $\theta = \left(X^T X\right)^{-1} X^T y$

(c) Modify the Matlab code in `hwk_mllr.m` to implement your maximum likelihood estimator for linear regression. Can you make your estimator run in just one short line with no loops, using vector inner products? Provide a plot of your results in your writeup.

2 Locally-weighted linear regression (20 points)

Consider the log-likelihood function for the general (non-scalar) case of locally-weighted linear regression,

$$\begin{aligned}\ell(\theta) &= -m\log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{m} w^{(i)}\left(\theta^T x^{(i)} - y^{(i)}\right)^2 \\ &= -m\log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\left(X\theta - y\right)^T W\left(X\theta - y\right)\end{aligned}$$

where $X \in \mathbb{R}^{m\times n}$ is the design matrix, $y \in \mathbb{R}^{m\times 1}$ is output vector, and $W \in \mathbb{R}^{m\times m}$ is the diagonal weight matrix with entry $W_{ii} = w^{(i)}$ and zeros elsewhere. The closed-form maximum likelihood solution in matrix form is

$$\theta = \left(X^T W X\right)^{-1} X^T W y.$$

Use the maximum likelihood solution of θ to implement locally-weighted linear regression in the file `hwk_wls.m`. For the weights associated with prediction value x, use

$$w^{(i)} = \exp\left(-\frac{||x - x^{(i)}||^2}{2\tau^2}\right)$$

where $||z|| = \sqrt{z^T z}$ is the *norm* (length) of vector z. Plot your results, and describe the results you get for different values of τ.

3 Gradient descent, Newton's method, and logistic regression (30 points)

Consider the logistic regression method of classification, where the hypothesis is defined as

$$h_\theta(x) = g(\theta^T x)$$

where $g(z) = 1/(1+\exp(-z))$ is the sigmoid function. (To actually do the classification, we threshold the hypothesis value at, say, 0.5.) Assuming that $p(y = 1|x;\theta) = g(\theta^T x)$, we obtain the log-likelihood that we derived in class

$$\ell(\theta) = \sum_{i=1}^{m}\left[y^{(i)}\log g(\theta^T x^{(i)}) + (1-y^{(i)})\log\left(1 - g(\theta^T x^{(i)})\right)\right]$$

and the gradient is

$$\begin{aligned}\nabla_\theta \ell(\theta) &= \nabla_\theta \sum_{i=1}^{m}\left[y^{(i)}\log g(\theta^T x^{(i)}) + (1-y^{(i)})\log\left(1 - g(\theta^T x^{(i)})\right)\right] \\ &= \sum_{i=1}^{m}\left[y^{(i)} - g(\theta^T x^{(i)})\right]x^{(i)}\end{aligned}$$

and the partial derivative with respect to θ_j is the jth component of the gradient vector. Finally, the Hessian matrix entries are

$$\begin{aligned}H_{jk} &= \frac{\partial^2 \ell(\theta)}{\partial\theta_j \partial\theta_k} \\ &= -\sum_{i=1}^{m} x_j^{(i)} x_k^{(i)} g(\theta^T x^{(i)})\left(1 - g(\theta^T x^{(i)})\right)\end{aligned}$$

We can write this result in matrix form as

$$H = -X^T G(I - G)X$$

where X is the $m \times n$ design matrix, I is the identity matrix, and G is a diagonal $m \times m$ matrix where $G_{ii} = g(\theta^T x^{(i)})$, and 0 elsewhere.

Using the files `hwk_logistic_regression.m` and `hwk_sigmoid.m`, implement the two learning algorithms for logistic regression: gradient descent and Newton-Raphson. Compare their rates of convergence for the same data. Use graphics to illustrate the solutions you find and the rates of convergence.

Note that the data provided for this problem has overlap, and does not get perfect classification accuracy. For further exploration, create your own dataset for use in this problem (rather than using the one provided). Try creating a dataset that is completely separable (i.e. there is no overlap between the two classes), so that you can achieve perfect accuracy. What do you find happens to the values of θ as the learning algorithm(s) proceed? (You should remove the code's stopping condition of perfect accuracy to see the effect.)

4 Extra credit 1: deriving logistic regression (+5 points)

Show the derivations for the gradient and the Hessian of the log-likelihood for logistic regression.

5 Extra credit 2: weighted logistic regression (+5 points)

Consider applying the idea of locally-weighted examples to the logistic regression model. So for predicting $h_\theta(x)$, we would first construct a set of weights $w^{(i)}$ for each training example $x^{(i)}$, which are then used to learn θ via maximum likelihood (using gradient ascent or Newton's method).

Write down the likelihood (and log-likelihood) of theta for locally-weighted logistic regression. Then derive the gradient and Hessian of the log-likelihood.