Assignment 2: Regression

CS485/685 – Winter 2012

Out: February 2, 2012
Due: February 16, 2012, at the beginning of the lecture. Late assignments may be submitted in the pink drop off box on the third floor of MC within 24 hrs for 50% credit.

Be sure to include your name and student number with your assignment.

1. **[30 pts]** In class, we discussed several loss functions for linear regression. However all the loss functions that we discussed assume that the error contributed by each data point have the same importance. Consider a scenario where we would like to give more weight to some data points. Our goal is to fit the data points (x_n, y_n) in proportion to their weights r_n by minimizing the following objective:

$$L(w,b) = \sum_{n=1}^{m} r_n(y_n - wx_n + b)^2$$

where w and b are the model parameters, the training data pairs are (x_n, y_n). To simplify things, feel free to consider 1D data (i.e., x_n and w are scalars).

   (a) **[15 pts]** Derive a closed-form expression for the estimates of w and b that minimize the objective. Show the steps along the way, not just the final estimates.

   (b) **[15 pts]** Show that this objective is equivalent to the negative log-likelihood for linear regression where each data point may have a different Gaussian measurement noise. What is the variance of measurement i in this model?

2. **[20 pts]** Consider a two-layer neural network of the form

$$y_k(x,w) = \sigma(\sum_j w_{kj}^{(2)} h(\sum_i w_{ji}^{(1)} x_i + w_{j0}^{(1)}) + w_{k0}^{(2)})$$

in which the hidden unit nonlinear activation functions $h(\cdot)$ are given by logistic sigmoid functions of the form

$$\sigma(a) = \frac{1}{1+e^{-a}}.$$

Show that there exists an equivalent network, which computes exactly the same function, but with hidden unit activation functions given by

$$tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}.$$

Hint: first find the relation between $\sigma(a)$ and $tanh(a)$ and then show that the paramneters of the two neural networks differ by linear transformations.

3. **[50 pts]** Non-linear regression techniques.

   Implement the following regression algorithms. A dataset will be posted on the course web page. The input and output spaces are continuous (i.e., $x \in \Re^d$ and $y \in \Re$).

   (a) **[12 pts]** Regularized generalized linear regression: perform least square regression with the penalty term $\lambda w^T w$. Use monomial basis functions up to degree d: $\{\prod_i (x_i)^{n_i} | \sum_i n_i \leq d\}$

   (b) **[12 pts]** Bayesian generalized linear regression: use monomial basis function up to degree d as described above.

   (c) **[12 pts]** Gaussian process regression: use the following kernels:

   - Identity: $k(x, x') = x^T x'$
   - Gaussian: $k(x, x') = e^{-||x-x'||^2/2\sigma^2}$
   - Polynomial: $k(x, x') = (x^T x' + 1)^d$ where d is the degree of the polynomial

   (d) **[14 pts]** Neural network: learn the weights of a two-layer neural network with a sigmoid activation function for the hidden nodes and the identity function for the output node.