CSC2515 – Assignment #2

Due: Nov.8, 2005, 2pm at the **START** of class
Worth: 18%
Late assignments not accepted.

1 Polynomial Regression (3%)

One of the most basic generalized linear models is the K^{th}-order polynomial regression model:

$$\hat{y}(\mathbf{x}) = \sum_j w_j h_j(\mathbf{x})$$
$$h_j(\mathbf{x}) = x_1^{a_1} x_2^{a_2} \dots x_D^{a_D}$$

where j indexes over vectors $a_1, a_2, \dots, a_D$ such that $0 \leq \sum_{i=1}^{D} a_i \leq K$, x_i is the value of the i^{th} dimension (component) of the input vector $\mathbf{x}$, and $\hat{y}(\mathbf{x})$ is the predicted output given data case $\mathbf{x}$. For example, when $D = 2$ and $K = 0$ the model is constant: $h_0 = x_1^0 x_2^0 = 1$ and $\hat{y}(\mathbf{x}) = w_0 h_0$. When $D = 2$ and $K = 1$ the model is linear regression: $h_0 = x_1^0 x_2^0 = 1, h_1 = x_1^1 x_2^0, h_2 = x_1^0 x_2^1$ and $\hat{y}(\mathbf{x}) = w_0 h_0 + w_1 h_1 + w_2 h_2$. The higher model orders K, and other data dimensions D are similarly defined.

Assuming that our objective is to minimize squared prediction error (corresponding to a belief that there is Gaussian noise on our output), polynomial ridge regression on a finite dataset $\{\mathbf{x}^1, y^1, \dots, \mathbf{x}^N, y^N\}$ optimizes this function:

$$\phi^P = \sum_n (\hat{y}(\mathbf{x}^n) - y^n)^2 + \lambda \sum_j w_j^2 \langle h_j(\mathbf{x})^2 \rangle \quad (1)$$

$$\langle h_j(\mathbf{x})^2 \rangle = \frac{1}{N} \sum_{n=1}^{N} h_j(\mathbf{x}^n)^2 \quad (2)$$

Theory Questions:

1. Derive the gradient of ϕ^P with respect to $\mathbf{w}$, ie the partial derivative of ϕ^P with respect to w_j for all j. Show a few lines of work.

2. By setting the gradient above to zero, solve for the weights which minimize the (regularized) objective function. Show a few lines of work. This is nothing more than a trivial re-derivation of the linear-least squares formula. This emphasizes that polynomial regression is just linear regression in the extended input vector $\mathbf{h}(\mathbf{x}) = [h_0(\mathbf{x}), h_1(\mathbf{x}), \dots]$.

3. The $\langle h_j(\mathbf{x})^2 \rangle$ term in the weight decay helps to calibrate the weights when some basis elements are much larger/smaller than others. To get an idea of why this is needed, assume all the components x_i of each data vector n were drawn IID from a single *uniform* distribution $p(\cdot)$ with range $[0, 2\mu]$ and mean μ. Given $a_1, ..., a_D$, find the expected value under $p(\cdot)$ of $(\prod_{i=1}^{D} x_i^{a_i})^2$, ie: $E_P((\prod_{i=1}^{D} x_i^{a_i})^2 | a_1, ..., a_D)$. What does this tell you about how $h_j(\mathbf{x})^2$ scales as a function of $\sum_{i=1}^{D} a_i$, and the mean μ of $p(\cdot)$?

2 Radial Basis Network Regression (2%)

Recall that a radial basis network is a generalized linear model of the form

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^{K} w_j h_j(\mathbf{x})$$
$$h_j(\mathbf{x}) = \exp\left[-\alpha(\mathbf{x}-\mathbf{z}_j)^\top(\mathbf{x}-\mathbf{z}_j)\right]$$

Assuming again that we are trying to minimize squared error (maximize likelihood under a Gaussian noise model), our objective is:

$$\phi^R = \sum_{n=1}^{N} \left(\hat{y}(\mathbf{x}^n) - y^n\right)^2 + \lambda \sum_{j=1}^{K} w_j^2$$

As you know, for fixed $\alpha > 0$ and fixed $\mathbf{z}_j$, the optimal weights can be found in closed form (using linear least squares). But if $\mathbf{z}_j$ and α are unknown, they cannot be solved for in closed form. We would like to adapt them by doing numerical optimization.

Theory Questions:

1. Derive the gradient of ϕ^R with respect to each centre $\mathbf{z}_j$ (ie compute the partial derivatives $\partial\phi^R/\partial\mathbf{z}_{ji}$) given a finite training set $\{\mathbf{x}^n, y^n\}$ and the current values of $\mathbf{z}_j, \alpha$ for all j.
2. Derive the gradient $\partial\phi^R/\partial \log\alpha$ with respect to $\log\alpha$, given the training set and the current values of $\mathbf{z}_j, \alpha$.

3 Locally Weighted Regression (1%)

A very useful and common special case of the above model is where we centre a basis function on every training input, and set the weights so that they interpolate the training outputs:

$$\hat{y}(\mathbf{x}) = \sum_{n=1}^{N} y^n h_n(\mathbf{x})$$
$$h_n(\mathbf{x}) = \frac{\exp\left[-\alpha(\mathbf{x}-\mathbf{x}^n)^\top(\mathbf{x}-\mathbf{x}^n)\right]}{\sum_{m=1}^{N}\exp\left[-\alpha(\mathbf{x}-\mathbf{x}^m)^\top(\mathbf{x}-\mathbf{x}^m)\right]}$$

This model is almost the ultimate "non-parametric" model; it has only a single parameter $\alpha > 0$ sometimes called the "bandwidth".

We'd like to set the bandwidth by optimizing an objective function numerically. We can't use just simple training error, since this would drive α to infinity. Instead, let's minimize the leave-one-out estimate of squared error:

$$\phi^L = \sum_{n=1}^{N} \left(\tilde{y}(\mathbf{x}^n) - y^n\right)^2$$

where $\tilde{y}(\mathbf{x}^n)$ is computed by leaving out the point $\mathbf{x}^n$ when making predictions, ie

$$\tilde{y}(\mathbf{x}^n) = \sum_{k\neq n} y^k \tilde{h}_k(\mathbf{x}^n)$$
$$\tilde{h}_k(\mathbf{x}^n) = \frac{\exp\left[-\alpha(\mathbf{x}^n-\mathbf{x}^k)^\top(\mathbf{x}^n-\mathbf{x}^k)\right]}{\sum_{m\neq n}\exp\left[-\alpha(\mathbf{x}^n-\mathbf{x}^m)^\top(\mathbf{x}^n-\mathbf{x}^m)\right]}$$

Theory Questions:

1. Find the derivative $d\phi^L/d\log\alpha$ with respect to $\log\alpha$, given the training set and the current value of α.

4 Experiments

The experimental part of this assignment is set up so that you can obtain a good grade by completing all of the basic experiments for 8%, and one of the two advanced experiments for an additional 2%. To receive full marks you must complete both of the advanced experiments. You may also attempt any of the bonus experiments for additional marks.

4.1 Datasets

On the course website you will find two datasets, `a2motorcycle` and `a2weather`, which are regression problems having a single scalar output. The motorcycle data set has data dimension 1, 103 training points, and 30 test points. The motorcycle data represents the acceleration force (in g's) on a motorcycle helmet as a function of time (in ms) during a collision. The weather data set has data dimension 2, 188 training points, and 50 test points. The weather data represents the annual precipitation (in mm) in the state of Washington, USA, as a function of grid location on a map.

4.2 Basic Experiments [8%]

Program learning procedures for each of the three regression models. For the polynomial model use either numerical minimization of ϕ^P, or solve for the optimal weights analytically. For the radial basis network use gradient minimization on the objective function ϕ^R. For locally weighted regression use gradient minimization on ϕ^L. You should restart your optimization on ϕ^R several times from different starting conditions. If you use numerical minimization on ϕ^P you should also use random restarts. For each of the two datasets:

1. Set $\lambda = 0$ and fit polynomial regression with $K = 1, 2, 3$.
2. Based on the $\hat{y}(x^n)$ and w_j values found in part 1, set $\lambda = .01 * \left[\sum_n \left(\hat{y}(\mathbf{x}^n) - y^n\right)^2\right] / \left[\sum_j w_j^2 \langle h_j(\mathbf{x})^2 \rangle\right]$ and refit the polynomial regression with $K = 1, 2, 3$ and this new λ.
3. Set $\lambda = 0$ and fit a radial basis network using $K = 10$ for motorcycle and $K = 18$ for weather. Try several random initializations for the parameters and select the one with the best objective (=training error, since $\lambda = 0$.).
4. Based on the $\hat{y}(x^n)$ and w_j values found in part 3, set $\lambda = .01 * \left[\sum_n \left(\hat{y}(\mathbf{x}^n) - y^n\right)^2\right] / \left[\sum_j w_j^2\right]$ and refit the radial basis network using the new λ and $K = 10$ for motorcycle and $K = 18$ for weather. Try several random initializations for the parameters and select the one with the best objective ($\neq$training error, since $\lambda \neq 0$).
5. Fit locally weighted regression. Set α by optimizing the leave-one-out error. If you carefully follow the gradient of leave-one-out error with respect to α (or $\log \alpha$) you should not need to do any random restarts. If you have some spare cycles on your hands you can check that your search is working by exhaustively evaluating α on a fine grid.

What to hand in:

1. Error Values: Report the average training error (for polynomial 1,2,3 and RBF only), the leave-one-out error (for locally weighted regression only) and the average test error for all methods. Make sure you do not include the weight decay term when computing average test/train errors. If you use random restarts for any of the methods, report results for the parameters that achieve the lowest value of the corresponding objective function.
2. Motorcycle Data Plots: For each regression model produce a plot showing the the training points as squares, the test points as circles and the interpolants $\hat{y}(\mathbf{x})$ as a curve. In a fourth figure, plot the training and testing data, and then plot each of the basis functions $h_j(\mathbf{x})$ of the radial basis network as a curve. Indicate on the plot the value of α that was learned by the radial basis network. Also indicate the value of α learned by locally

weighted regression. We will be providing some helper functions for plotting in Matlab and there will also be some help on this in the tutorial.

3. Weather Data Plots: Plot the training points in the input plane using squares and the test points using circles. For each regression model display the interpolant using contour lines overlaid on the data. In a fourth figure plot the training and test data. Show the centres and bandwidth α learned for your radial basis network, and also report the bandwidth learned for your locally weighted regression. We will be providing some helper functions for plotting in Matlab and there will also be some help on this in the tutorial.

4.3 Advanced Experiments [4%]

1. Try polynomial regression with K between 0 and 10. For each order k, compute the average leave-one-out error. Also compute the average training and testing errors when you train on the entire training set. Use the same value of λ you used in question 4.2.2. Hand in a plot of each error measure as a function of polynomial order k for both datasets. What order gives the best test error? What order would you have picked if you were using leave-one-out cross validation as a guide?

2. In the radial basis network, try all numbers of basis units in the set $k = [1, 2, 4, 8, 16, 32, ..., N]$ (where N is the number of training samples). For each number of units k, compute the average leave-one-out error. (Each time you leave out a point, try several random initializations and chose the one which leads to the smallest training error.) Also compute the average training and testing errors when you train on the entire training set (again, do several random initializations and chose the one with the lowest training error). Use the same value of λ you used in question 4.2.4. Hand in a plot of each error measure as a function of the number of units k. Do this for both datasets. What k gives the best test error? What number would you have picked if you were using leave-one-out cross validation as a guide?

4.4 Bonus Experiments [up to 6%]

1. Set both the polynomial order k *and* the value of the smoothing parameter λ simultaneously by cross-validation. Chose a sensible range for λ to search over, and display your results graphically, as well as reporting the best pair of (k,λ) for test and validation.

2. Set both the number of radial basis units k *and* the value of the smoothing parameter λ simultaneously by cross-validation. Chose a sensible range for λ to search over, and display your results graphically, as well as reporting the best pair of (k,λ) for test and validation.

3. Generalize the locally weighted regression model so that each basis function h_n has its own bandwidth α_n. Derive the gradient of the cross-validation error for this new model and re-fit it to the data, reporting your new results. Display the resulting bandwidths and the training data together.

IMPORTANT: please do your computing on CDF or your own machine and not on CSLAB machines since they have limited software licenses and other resources.