10-701/15-781 Machine Learning, Fall 2005
Homework 3

Out: 10/20/05 *Due: beginning of the class 11/01/05*

Instructions. Contact questions-10701@autonlab.org for question

Problem 1. *Regression and Cross-validation* [40 points]

Part 1: Multiple regression [15 points]

The multiple regression model is $Y = X\beta + \epsilon$ where

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \\ . \\ . \\ . \\ Y_r \end{pmatrix}, \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ . \\ . \\ . \\ \epsilon_r \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ . \\ . \\ . \\ \beta_n \end{pmatrix} \text{ and } X = \begin{bmatrix} 1 & x_{11} & x_{12} & ... & x_{1n} \\ 1 & x_{21} & x_{22} & ... & x_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ 1 & x_{r1} & x_{r2} & ... & x_{rn} \end{bmatrix}.$$

Assume $Y \sim N(X\beta, \sigma^2 I)$ and $\epsilon \sim N(0, \sigma^2 I)$ where I is the $n \times n$ identity matrix.

From the class we know that the least square estimator $\widehat{\beta} = SY$ where $S = (X^T X)^{-1} X^T$.

(a) prove that $\widehat{\beta}$ is unbiased, i.e., $\mathbf{E}(\widehat{\beta}) = \beta$.

(b) find the covariance matrix of $\widehat{\beta}$: $\mathbf{V}(\widehat{\beta})$ (hint: $\mathbf{V}(Cx) = C\mathbf{V}(x)C^T$ if C is a constant matrix.)

The estimator $\widehat{Y} = X\widehat{\beta} = HY$ where $H = X(X^T X)^{-1} X^T$ (H is called the hat matrix).

(c) prove H is symmetric ($H = H^T$) and idempotent ($H^2 = H$).

(d) prove the trace of H equals the rank of X, i.e., $\text{tr}(H) = n + 1$ (hint: what is the relationship between $\text{tr}(AB)$ and $\text{tr}(BA)$ if AB and BA are defined?)

Part 2: Leave-one-out cross-validation [25 points] The least square estimator minimizes the sums of squared errors:

$$\text{SSE} = \sum_{i=1}^{r} (Y_i - \widehat{Y}_i)^2$$

Recall the definition of leave-one-out cross-validation score

$$LOOCV = \sum_{i=1}^{r}(Y_i - \widehat{Y}_i^{(-i)})^2$$

where $\widehat{Y}^{(-i)}$ is the estimator of Y after removing i-th observation (i.e., it minimizes $\sum_{j\neq i}(Y_j - \widehat{Y}_j^{(-i)})^2$). In particular, $\widehat{Y}_i^{(-i)}$ is the estimated value of Y_i after removing i-th observation.

(a) write $\widehat{Y}_i$ in terms of H and Y.

(b) prove that $\widehat{Y}^{(-i)}$ is also the estimator that minimizes SSE for Z where $Z_j = \begin{cases} Y_j, & j \neq i \\ \widehat{Y}_i^{(-i)}, & j = i \end{cases}$

(c) prove that $\widehat{Y}_i^{(-i)} = \widehat{Y}_i - H_{ii}Y_i + H_{ii}\widehat{Y}_i^{(-i)}$

(d) prove that

$$LOOCV = \sum_{i=1}^{r}(\frac{Y_i - \widehat{Y}_i}{1 - H_{ii}})^2$$

Problem 2. *Kernelization* [40 points]

In the lecture on SVM, we learned a trick called *kernelization* for classification. The idea is to map a feature vector in low dimensional space $\mathcal{X}$ into a higher dimensional space $\mathcal{Z}$. This can yield a more flexible classifier while retaining computational simplicity. In other words: a linear classifier in a higher dimensional space corresponds to a non-linear classifier in the original space.

In general, *kernelization* involves finding a mapping $\phi : \mathcal{X} \rightarrow \mathcal{Z}$ such that

1. $\mathcal{Z}$ has a higher dimension than $\mathcal{X}$;
2. the computation in $\mathcal{Z}$ only uses inner product;
3. there is a function K called kernel such that the inner product of $\phi(x_i)$ and $\phi(x_j)$ is $K(x_i, x_j)$[1].

The standard logistic regression has the following form:

$$P(Y = 1|X) = g(\omega_0 + \sum_{i=1}^{n} \omega_i X_i)$$

$$P(Y = 0|X) = 1 - P(Y = 1|X)$$

where $g(a) = 1/(1 + e^{-a})$.

(a) Consider a function ϕ maps X from a low dimensional space $\mathcal{X}$(dimensionality=n) into a high dimensional space $\mathcal{Z}$ (dimensonality is m, $m > n$). The logistic regression becomes

$$P(Y = 1|\phi(X)) = g(\omega_0 + \sum_{i=1}^{m} \omega_i \phi(X)_i)$$

where m is the dimension of $\mathcal{Z}$[2].

Assume the weight vector ω is the linear combination of all input feature vector $\phi(X_i)$; more formally, $(\omega_1, \ldots, \omega_m)^T = \sum_{i=1}^{R} \alpha_i \phi(X^{(i)})$ and $\omega_0 = \alpha_0$ where R is the number of data points and $X^{(i)}$ is the i-th data point.

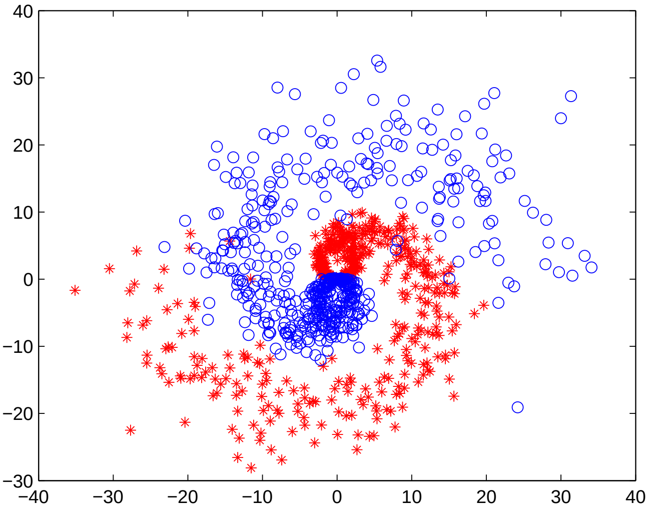
Use kernelization trick to compute $P(Y = 1|\phi(X))$ (i.e., to avoid explicitly computing in $\mathcal{Z}$)

(b) Write down the gradient descent update rule for kernel logistic regression.

[1] And K has to be positive definite, e.g. gaussian kernel is one of such kernel. And you don't have to worry it for this question.

[2] X is a n-dimensional feature vector; $\phi(X)$ is the corresponding m-dimensional vector; $\phi(X)_i$ is the i-th element of $\phi(X)$.

(c) Implement the kernel logistic regression using the gaussian kernel $K_\sigma(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$. And run your program on ds2.txt (first two columns are X, last column is Y) with $\sigma = 1$. Report the training error. Set stepsize to be 0.01 and maximum number of iterations 100 (Please use this setting and don't try alternative settings). The scatterplot of the ds2.txt is the follows:



(d) Use the 10-folds cross-validation to find the best σ and plot the total number of mistakes for $\sigma = \{0.5, 1, 2, 3, 4, 5, 6\}$.

Problem 3. *Computational Learning Theory* [20 points]

Part1:VC-dimension [12 points]

Consider the space of instances X corresponding to all points in the 2D plane. Give the VC-dimension of the following hypothesis spaces:

(a) H_r: the set of all axis-parallel rectangles in the 2D plane. Points inside the rectangle are positive examples.

(b) H_c: circles in the 2D plane. Points inside the circle are classified as positive examples.

(c) How many training examples suffice to assure with probability .9 that a consistent learner using H_c will learn the target function with accuracy of at least 0.95?

(d) What exactly does it mean in part (c) when we say the learner will succeed with probability 0.9? Answer this question by describing a simple experiment which you could run repeatedly, for which the success rate is expected to be at least 0.9.

Part2: Mistake bounds [8 points]

Consider learning a boolean valued function $f : X \rightarrow Y$, where $X = \langle X_1 \ldots X_N \rangle$, where Y and the X_i are all boolean valued variables. You decide to consider a hypothesis space H where each hypothesis is of the form

if $[(X_i = a) \wedge (X_j = b)]$ then $Y = 1$ else $Y = 0$.

where $i \neq j$, and where a and b can be either 0 or 1. Notice each hypothesis constrains exactly two of the features of X.

Please answer the following questions:

(a) How many distinct hypotheses are there in H?

(b) Consider the following Weighted Majority algorithm, applied to the entire space of hypotheses H: You begin with all hypotheses in H assigned an initial weight equal to 1. Every time you see a new example, you predict based on a weighted majority vote of the hypotheses in H. After each prediction, any hypothesis that made an incorrect prediction has its weight divided by two. How many mistakes will this Weighted Majority algorithm make when shown a sequence of training examples, as a function of the number of mistakes made by the most accurate hypothesis in H?

(c) Suppose X has N=1024 features, the training sequence contains 1000 examples, and the best hypothesis in H has a true error of 0.05. What bound can you given on the expected number of mistakes made by the Weighted Majority algorithm in this case?