1 Logistic Regression and Regularization

Let us consider a binary logistic regression model.

(a) Given n training examples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)$, please write down the negative log likelihood (as loss function):

$$\mathcal{L}(\mathbf{w}) = -\log\left(\prod_{i=1}^{n} P(Y = y_i|\mathbf{X} = \mathbf{x}_i)\right).$$

(b) Is this loss function convex? Provide your reasoning.

(c) Show that the magnitude of the optimal $\mathbf{w}$ can go to infinity when the training samples are *linearly seperable* (i.e. the data can be *perfectly* classified by a linear classifier).

(d) A convenient way to resolve the problem described in (c) is to add a regularization term to the likelihood function as follows:

$$\mathcal{L}(\mathbf{w}) = -\log\left(\prod_{i=1}^{n} p(Y = y_i|X = x)\right) + \lambda\|\mathbf{w}\|_2^2, \tag{1}$$

where $\|\mathbf{w}\|_2^2 = \sum_i w_i^2$ and $\lambda > 0$. Please compute the gradient respect to w_i, i.e. $\frac{\partial \mathcal{L}(\mathbf{w})}{\partial w_i}$.

(e) Show that the problem in Eq. (1) has a unique solution.

2 Sparsity via Regularization

Given a set of training data $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, $\cdots$, $(\mathbf{x}_N, y_N)$ where $\mathbf{x}_i \in \mathcal{R}^D$, linear regression learns the weight vector $\mathbf{w}$ (assuming the bias term is absorbed into $\mathbf{w}$) by solving

$$\min_{\mathbf{w}} \sum_{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 \tag{2}$$

For many real-world applications (e.g. text and image classification, gene sequence analysis), the dimensionality D can be very high. In practice, however, only a small portion of features may be actually useful in predicting the target varible y. This motivates us to enforce certain sparse structure on $\mathbf{w}$: many elements of $\mathbf{w}$ are zeros.

(a) One way to incorporate sparsity into linear regression is to add a regularization term as follows

$$\min_{\mathbf{w}} \sum_{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda\|\mathbf{w}\|_0 \tag{3}$$

where $\lambda > 0$, and $\|\mathbf{w}\|_0 = \#\{i : \mathbf{w}_i \neq 0\}$ is called ℓ_0-norm of $\mathbf{w}$ which is equal to number of nonzeros in $\mathbf{w}$. The challenge is that $\|\mathbf{w}\|_0$ is not a convex function of $\mathbf{w}$. Provide an example to show the nonconvexity of $\|\mathbf{w}\|_0$.

(b) One way to overcome the nonconvexity of ℓ_0-norm is to use ℓ_1-norm: $\|\mathbf{w}\|_1 = \sum_i |\mathbf{w}_i|$ which is the sum of absoluate value of each element. Although ℓ_1-norm is an approximation of ℓ_0-norm, it does often lead to sparse solution in practice. Prove that the $\|\mathbf{w}\|_1$ is convex in $\mathbf{w}$.

(c) Due to the nondifferentibility of $\|\mathbf{w}\|_1$, solving

$$\min_{\mathbf{w}} \sum_n (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_1 \tag{4}$$

becomes challenging. It no longer has an analytic solution. One method to optimize the objective function is to transform it into a quadratic programming (QP) which has fast solvers. A QP has the following standard form

$$\min_{\mathbf{u}} \frac{1}{2}\mathbf{u}^\top \mathbf{Q}\mathbf{u} + \mathbf{c}^\top \mathbf{u} \tag{5}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{u} \leq \mathbf{b} \tag{6}$$

where $\mathbf{u} \in \mathcal{R}^P$ is the vector to be optimized (P can be different from D). $\mathbf{c} \in \mathcal{R}^P, \mathbf{b} \in \mathcal{R}^G$, $\mathbf{A} \in \mathcal{R}^{G \times P}$, and $\mathbf{Q} \in \mathcal{R}^{P \times P}$ are coefficients. Show that (4) can be converted into a Standard QP.
(***Hint***: introduce D additional variables $t_1, \cdots, t_D$ where $t_d \geq \boldsymbol{w}_d$ and $t_d \geq -\boldsymbol{w}_d$ for $d = 1, \cdots, D$. Then show that minimizing $\|\boldsymbol{w}\|_1$ is equivalent to minimizing $\sum_{d=1}^D t_d$ which is an upper bound of $\|\boldsymbol{w}\|_1$. To further transform into a standard QP, stack $\boldsymbol{w}$ and $\boldsymbol{t}$ into a vector $\boldsymbol{u} \in \mathcal{R}^{2D}$, and then form the corresponding $\boldsymbol{Q} \in \mathcal{R}^{2D \times 2D}, \boldsymbol{A} \in \mathcal{R}^{2D \times 2D}, \boldsymbol{c} \in \mathcal{R}^{2D}$ and $\boldsymbol{b} \in \mathcal{R}^{2D}$.)

3 Kernel Ridge Regression

In this problem, we will provide guidelines for you to derive kernel ridge regression, a nonlinear extension of linear ridge regression. You will be asked to implement it in the programming part.

Given a set of training data $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, $\cdots$, $(\mathbf{x}_N, y_N)$ where $\mathbf{x}_i \in \mathcal{R}^D$, linear ridge regression learns the weight vector $\mathbf{w}$ (assuming the bias term is absorbed into $\mathbf{w}$) by optimizing the following objective function

$$\min_{\mathbf{w}} \sum_n (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_2^2 \tag{7}$$

where $\lambda \geq 0$ is the regularization coefficient.

(a) Let $\mathbf{X} \in \mathcal{R}^{N \times D}$ be a matrix whose nth row is $\mathbf{x}_n^\top$. $\mathbf{y} \in \mathcal{R}^N$ is a vector whose nth element is y_n. Show that the optimal $\mathbf{w}$ can be written as

$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_D)^{-1} \mathbf{X}^\top \mathbf{y} \tag{8}$$

where $\mathbf{I}_D$ denotes the identity matrix with size $d \times d$.

(b) Now we apply a nonlinear feature mapping to each sample $\mathbf{x}_i \to \mathbf{\Phi_i} = \phi_i(\mathbf{x}) \in \mathcal{R}^T$ where the dimensionality T is much larger than D. Define $\mathbf{\Phi} \in \mathcal{R}^{N \times T}$ as a matrix containing all $\mathbf{\Phi_i}$. Show that $\mathbf{w}^*$ can be written as

$$\mathbf{w}^* = \mathbf{\Phi}^\top (\mathbf{\Phi}\mathbf{\Phi}^\top + \lambda \mathbf{I}_N)^{-1}\mathbf{y} \tag{9}$$

Hint: you may use the following identity for matrices. For any matrix $\mathbf{P} \in \mathcal{R}^{p\times p}$, $\mathbf{B} \in \mathcal{R}^{q\times p}, \mathbf{R} \in \mathcal{R}^{q\times q}$ *and assume the matrix inversion is valid, we have*

$$(\mathbf{P}^{-1} + \mathbf{B}^\top \mathbf{R}^{-1}\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{R}^{-1} = \mathbf{P}\mathbf{B}^\top (\mathbf{B}\mathbf{P}\mathbf{B}^\top + \mathbf{R})^{-1}$$

(c) Given a testing sample $\phi(\mathbf{x})$, show that the prediction $\hat{y} = \mathbf{w}^{*\top}\phi(\mathbf{x})$ can be written as

$$\hat{y} = \mathbf{y}^\top (\mathbf{K} + \lambda \mathbf{I}_N)^{-1}\kappa(\mathbf{x}) \tag{10}$$

where $\mathbf{K} \in \mathcal{R}^{N\times N}$ is a kernel matrix defined as $\mathbf{K}_{ij} = \mathbf{\Phi_i}^\top \mathbf{\Phi_j}$, $\kappa(\mathbf{x}) \in \mathcal{R}^N$ is a vector with nth element $\kappa(\mathbf{x})_n = \phi(\mathbf{x}_n)^\top \phi(\mathbf{x})$. Now you can see that $\hat{y}$ only depends on the dot product (or kernel value) of $\{\mathbf{\Phi_i}\}$.

(d) Compare the computational complexity between linear ridge regression and kernel ridge regression.

4 Kernel Construction

The Mercer theorem can be used to construct valid kernel functions. The theorem states that, a bivariate function $k(\cdot,\cdot)$ is a positive definite kernel function, if and only if, for any N and any $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$, the matrix K, where $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, is positive semidefinite. That is, all the eigenvalues of the matrix are non-negative. An alternative (but equivalent) definition states that, for every positive semi-definite matrix $A \in \mathbb{R}^{n\times n}$ and arbitrary vector $\mathbf{x} \in \mathbb{R}^{n\times 1}$, we have $\mathbf{x}^\top A\mathbf{x} \geq 0$.

Suppose $k_1(\cdot,\cdot)$ and $k_2(\cdot,\cdot)$ are kernel functions, and $\mathbf{x}$ and $\mathbf{x}'$ are any two samples, prove that k_3, k_4, k_5, k_6 and k_7 are valid kernel functions using the Mercer theorem:

(a) $k_3(\mathbf{x}, \mathbf{x}') = a_1 k_1(\mathbf{x}, \mathbf{x}') + a_2 k_2(\mathbf{x}, \mathbf{x}')$ where $a_1, a_2 \geq 0$.

(b) $k_4(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})f(\mathbf{x}')$ where $f(\cdot)$ is a real valued function.

(c) $k_5(\mathbf{x}, \mathbf{x}') = g(k_1(\mathbf{x}, \mathbf{x}'))$ where $g(\cdot)$ is a polynomial function with positive coefficients.

(d) $k_6(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}')$.

(e) $k_7(\mathbf{x}, \mathbf{x}') = \exp\left(k_1(\mathbf{x}, \mathbf{x}')\right)$.

5 Programming - Bias/Variance Tradeoff

Let the function $f(x) = 2x^2 + \epsilon$, where ϵ is Gaussian noise drawn from $\mathcal{N}(0, 0.1)$. We are also given the following functions:

- $g_1(x) = 1$
- $g_2(x) = w_0$
- $g_3(x) = w_0 + w_1 x$
- $g_4(x) = w_0 + w_1 x + w_2 x^2$
- $g_5(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$
- $g_6(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$

(a) Randomly generate 100 datasets. Each dataset contains 10 samples (x_i, y_i), where x_i is uniformly sampled from $[-1, 1]$ and $y_i = f(x_i)$. For a given g function, estimate its parameters using linear regression and compute the sum-square-error on every dataset. Then plot the histogram of the mean-squared-error. Repeat this for each g function. Please provide the 6 histogram plots in the report. For each g function, estimate its bias^2 and variance, and report the results.

(b) In (a), change the sample size in each dataset to 100, and repeat the procedures. Plot the resulting histograms in the report. For each g function, estimate its bias^2 and variance, and report the results.

(c) Given your results in (a) and (b), discuss how the model complexity and sample size affect the bias^2 and variance.

(d) Consider function $h(x) = w_0 + w_1 x + w_2 x^2 + \lambda(w_0^2 + w_1^2 + w_2^2)$ where $\lambda \geq 0$ is a regularization parameter. Following (b) (i.e. 100 samples per dataset), estimate the bias^2 and variance of $h(x)$ when λ set to $0.01, 0.1, 1, 10$ respectively. Discuss how λ affect the bias^2 and variance.

6 Programming - Regression

In this problem, you will implement linear ridge regression and kernel ridge regression, and work on a regression dataset.

All programming should be done in MATLAB. You must implement certain functions which we specify. Otherwise, you may use built-in MATLAB functions. Do not use code from the internet or from other students. Record answers to questions in your written LaTeX report.

Dataset It contains 3,107 observations on U.S. county votes cast in the 1980 presidential election. Each observation contains 6 features related to population structure, ecomonical condition, as well as geographical info of the county. The goal is to predict the portion of the votes. Details of the dataset can be found at http://lib.stat.cmu.edu/datasets/space_ga.

Preprocessing Randomly split the dataset into the training (80% samples) and test sets (20% samples). Normalize each feature to have zero mean and unit variance (note that you should only use the mean and variance information in the training data). When experimenting with any model described below, do 10 random splits, and report the average test error.

Linear Ridge Regression Implement linear ridge regression described in Problem 2 using Matlab. For the regularization parameter λ, using 5-folder cross validation on the training set to pick the optimal one from $[0, 10^{-4}, 10^{-3}, 10^{-2}, \cdots, 10^{2}, 10^{3}]$. Note that for each random split of the data, you need to run cross validation to select λ. Report the optimal λ for each ramdom split. Additionally, report the average test error.

Kernel Ridge Regression Implement kernel ridge regression described in Problem 2 using Matlab. You need to try three types of kernels

(a) Linear kernel: $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$.

(b) Polynomial kernel: $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + a)^b$, where a and c are additional parameters. You need to choose a from $[-1, -0.5, 0, 0.5, 1]$, and choose c from $[1, 2, 3, 4]$.

(c) Gaussian RBF kernel: $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma^2}\right)$, where σ is an additional parameter. You need to choose σ^2 from $[0.125, 0.25, 0.5, 1, 2, 4, 8]$.

For each of the kernel type, use 5-folder cross validation on the training set to pick the optimal λ from $[0, 10^{-4}, 10^{-3}, 10^{-2}, \cdots, 10^{2}, 10^{3}]$ and the other kernel parameters. Report the optimal parameters for each ramdom split. Additionally, report the average test error.

Comparison Does kernel ridge regression with linear kernel give the same results as linear ridge regression? If not, why? Among three types of kernels, which one performs the best?

Submission Instructions: You need to provide the followings:

- Provide your answers to problems 1-6 in PDF file, named as `CSCI567_hw3_fall15.pdf`. You need to submit the homework in both hard copy (at Locker #19 at PHE, with a box labeled as CSCI567-homework by 6pm of the deadline date) and electronic version as pdf file on Blackboard. If you choose handwriting instead of typing all the answers, you will get 40% points deducted.
- Submit ALL the code and report via Blackboard by 6 pm of the deadline date. The only acceptable language is MATLAB. For your program, you MUST include the main function called `CSCI567_hw3_fall15.m` in the root of your folder. After running this main file, your program should be able to generate all of the results needed for this programming assignment, either as plots or console outputs. You can have multiple files (i.e your sub-functions), however, the only requirement is that once we unzip your folder and execute your main file, your program should execute correctly. Please double-check your program before submitting. You should only submit one `.zip` file. No other formats are allowed except `.zip` file. Also, please name it as `[lastname]_[firstname]_hw3_fall15.zip`.

Collaboration: You may collaborate. However, collaboration has to be limited to discussion only and you need to write your own solution and submit separately. You also need to list with whom you have discussed.