1 Bias Variance Trade-off (10 Points)

Consider a dataset with n data points $(\mathbf{x}_i, y_i)$, $\mathbf{x}_i \in \mathbb{R}^{p\times 1}$, drawn from the following linear model:

$$y = \mathbf{x}^\top \boldsymbol{\beta}^\star + \varepsilon,$$

where ε is a Gaussian noise and the star sign is used to differentiate the true parameter from the estimators that will be introduced later. Consider the L_2 regularized linear regression as follows:

$$\widehat{\boldsymbol{\beta}}_\lambda = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \boldsymbol{\beta})^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \right\},$$

where $\lambda \geq 0$ is the regularization parameter. Let $X \in \mathbb{R}^{n\times p}$ denote the matrix obtained by stacking $\mathbf{x}_i^\top$ in each row.

(a) Find the closed form solution for $\widehat{\boldsymbol{\beta}}_\lambda$ and its distribution. (2 points)

(b) Calculate the bias term $\mathbb{E}[\mathbf{x}^\top \widehat{\boldsymbol{\beta}}_\lambda] - \mathbf{x}^\top \boldsymbol{\beta}^\star$ as a function of λ and some feature vector $\mathbf{x}$. (3 points)

(c) Calculate the variance term $\mathbb{E}\left[\left(\mathbf{x}^\top \widehat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\mathbf{x}^\top \widehat{\boldsymbol{\beta}}_\lambda]\right)^2\right]$ as a function of λ and some feature vector $\mathbf{x}$. (3 points)

(d) Use the results from parts (b) and (c) and the bias–variance theorem to analyze the impact of λ in the squared error. (i.e. which term dominates when λ is small or large?) (2 points)

2 Kernel Construction (15 Points)

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 and k_5 are valid kernel functions (using Mercer theorem or otherwise):

(a) $k_3(\mathbf{x}, \mathbf{x}') = a_1 k_1(\mathbf{x}, \mathbf{x}') + a_2 k_2(\mathbf{x}, \mathbf{x}')$ where $a_1, a_2 \geq 0$. (5 points)

(b) $k_4(\mathbf{x}, \mathbf{x}') = f(\mathbf{x}) f(\mathbf{x}')$ where $f(\cdot)$ is a real valued function. (5 points)

(c) $k_5(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') k_2(\mathbf{x}, \mathbf{x}')$. (5 points)

3 Kernel Regression (15 Points)

In this problem, we will provide guidelines for you to derive kernel ridge regression, a nonlinear extension of linear ridge regression.

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

$$\min_{\mathbf{w}} \sum_n (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_2^2 \tag{1}$$

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

$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_D)^{-1} \mathbf{X}^\top \mathbf{y} \tag{2}$$

where $\mathbf{I}_D$ denotes the identity matrix with size $d \times d$. (3 points)

(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

$$\mathbf{w}^* = \mathbf{\Phi}^\top (\mathbf{\Phi}\mathbf{\Phi}^\top + \lambda \mathbf{I}_N)^{-1} \mathbf{y} \tag{3}$$

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*

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

(5 points)

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

$$\hat{y} = \mathbf{y}^\top (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \kappa(\mathbf{x}) \tag{4}$$

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 $\{\mathbf{\Phi_i}\}$. (2 points)

(d) Compare the computational complexity for obtaining the above closed-form solutions between linear ridge regression and kernel ridge regression. Assume that matrix multiplication of two $N \times N$ matrices and inverting an $N \times N$ matrix takes $O(N^3)$ time each. (5 points)

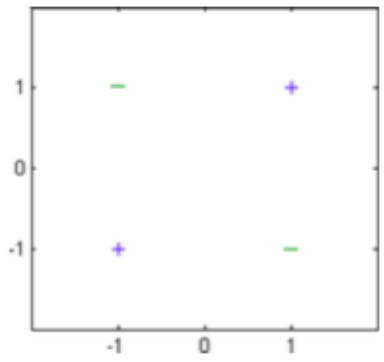
4 Support Vector Machine (5 Points)

Consider a supervised learning problem in which the training examples are points in 2-dimensional space. The positive examples are $(1, 1)$ and $(-1, -1)$. The negative examples are $(1, -1)$ and $(-1, 1)$.

1. Are the positive examples linearly separable from the negative examples in the original space? (1 point)

2. Consider the feature transformation $\Phi(x) = [1, x_1, x_2, x_1x_2]$, where x_1 and x_2 are, respectively, the first and second coordinates of a generic example x. The prediction function is $y(x) = w^T \times \Phi(x)$ in this feature space. Give the coefficients, w, of a maximum-margin decision surface separating the positive examples from the negative examples. (You should be able to do this by inspection, without any significant computation.) (1 point)

3. Add one training example to the graph so the total five examples can no longer be linearly separated in the feature space $\Phi(x)$ defined in part 2. (1 point)



4. What kernel $K(x, x)$ does this feature transformation correspond to? (2 points)

5 SVMs and the slack penalty C (15 Points)

The goal of this problem is to correctly classify test data points, given a training data set. You have been warned, however, that the training data comes from sensors which can be error-prone, so you should avoid trusting any specific point too much. For this problem, assume that we are training an SVM with a quadratic kernel that is, our kernel function is a polynomial kernel of degree 2. You are given the data set presented in Figure 1. The slack penalty C will determine the location of the separating hyperplane. Please answer the following questions qualitatively. Give a one sentence answer/justification for each and draw your solution in the appropriate part of the Figure at the end of the problem.

1. Where would the decision boundary be for very large values of C (i.e., $C \to \infty$)? (remember that we are using an SVM with a quadratic kernel.) Draw on the figure below. Justify your answer. (3 points)

2. For $C \approx 0$, indicate in the figure below, where you would expect the decision boundary to be? Justify your answer. (3 points)

3. Which of the two cases above would you expect to work better in the classification task? Why? (3 points)

4. Draw a data point which will not change the decision boundary learned for very large values of C. Justify your answer. (3 points)

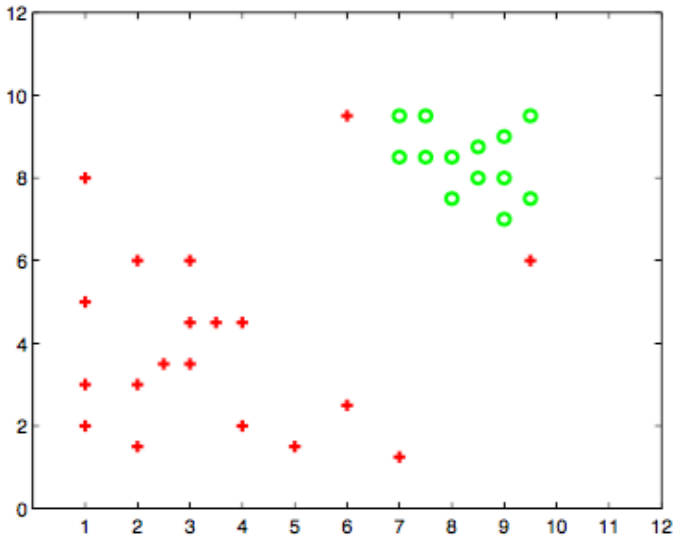
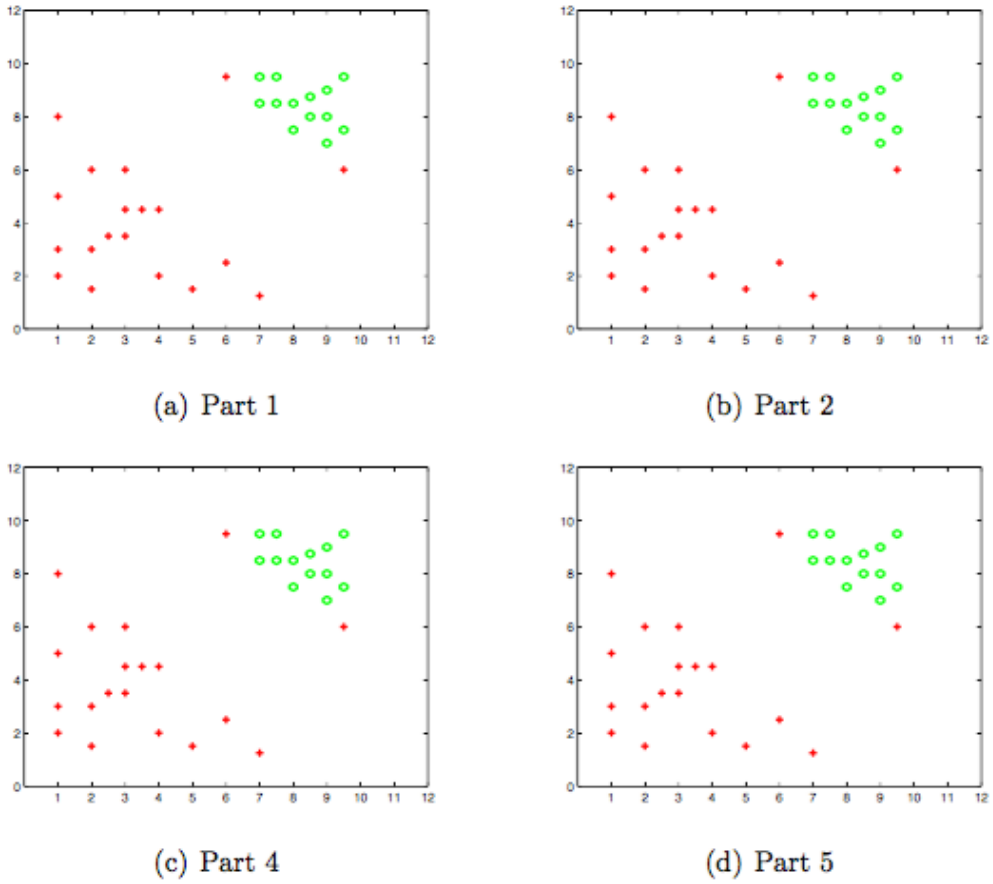


Figure 1: Dataset for SVM slack penalty selection task in Question 2.

5. Draw a data point which will significantly change the decision boundary learned for very large values of C. Justify your answer. (3 points)



(a) Part 1 (b) Part 2

(c) Part 4 (d) Part 5

Figure 2: Draw your solutions here.

Programming (40 Points)

Bias Variance Trade-off (20 Points)

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 (with no regularization) 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. (5 points)

(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. (5 points)

(c) Given your results in (a) and (b), discuss how the model complexity and sample size affect the bias^2 and variance. (5 points)

(d) Consider function $h_\lambda(x) = w_0 + w_1 x + w_2 x^2$ where $\lambda \geq 0$ is the L2 regularization parameter used during linear regression i.e. your regression loss function will have an additional $\lambda(w_0^2 + w_1^2 + w_2^2)$ regularization term. Following (b) (i.e. 100 samples per dataset), estimate the bias^2 and variance of each $h_\lambda(x)$ when λ is set to $0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1.0$ respectively. Analyze your obtained values and discuss how λ affects the bias^2 and variance. (5 points)

Linear and Kernel SVM (20 Points)

In this part, you will experiment with SVMs on a real-world dataset. You will implement linear SVM (i.e. SVM using the original features, namely the nonlinear mapping is the identity mapping). Also, you will use a widely used SVM toolbox called LIBSVM to experiment with kernel SVMs.

Dataset: We have provided the *Phishing Websites Data Set* from UCI's machine learning data repository (https://archive.ics.uci.edu/ml/datasets/Phishing+Websites). The dataset is for a binary classification problem to detect phishing websites. The dimensionality of the input feature is 30, and the training and test sets contain 2,000 and 2,000 samples, respectively (they are provided on Blackboard as `phishing-train.mat` and `phishing-test.mat`). The datasets can be imported directly in MATLAB and using Scipy.io in Python.

Data preprocessing

All the features in the datasets are categorical. You need to preprocess the training and test data to make features with multiple values to features taking values only zero or one. If a feature f_i have value $\{-1, 0, 1\}$, we create three new features $f_{i,-1}, f_{i,0}$ and $f_{i,1}$. Only one of them can have value 1 and $f_{i,x} = 1$ if and only if $f_i = x$. For example, we transform the original feature 1 into $[0, 0, 1]$. In the given dataset, the features $2, 7, 8, 14, 15, 16, 26, 29$ (index starting from 1) take three different values $\{-1, 0, 1\}$. You need to transform each above feature into three 0/1 features.

Use linear SVM in LIBSVM

LIBSVM is widely used toolbox for SVM and has both a Matlab and a Python interface. Download LIBSVM from https://www.csie.ntu.edu.tw/~cjlin/libsvm/ and install it according to the README file. For each C from $\{4^{-6}, 4^{-5}, \cdots, 4, 4^2\}$, apply 3-fold cross validation (use `-v` option in LIBSVM) and report the cross validation accuracy and average training time. (5 points)

Kernel SVM in LIBSVM

LIBSVM supports a number of kernel types. Here you need to experiment with the polynomial kernel and RBF (Radial Basis Function) kernel.

(a) **Polynomial kernel**. Please tune C and `degree` in the kernel. For each combination of (C, `degree`), where $C \in \{4^{-3}, 4^{-2}, \cdots, 4^6, 4^7\}$ and `degree` $\in \{1, 2, 3\}$, report the 3-fold cross validation accuracy and average training time. (5 points)

(b) **RBF kernel**. Please tune C and `gamma` in the kernel. For each combination of (C, `gamma`), where $C \in \{4^{-3}, 4^{-2}, \cdots, 4^6, 4^7\}$ and `gamma` $\in \{4^{-7}, 4^{-6}, \cdots, 4^{-2}, 4^{-1}\}$, report the 3-fold cross validation accuracy and average training time. (5 points)

Based on the cross validation results of Polynomial and RBF kernel, which kernel type and kernel parameters will you choose? You need to write a script `libsvm.m`/`libsvm.py` using the best kernel and the parameters selected from cross-validation. Your script should train a SVM using libsvm with selected kernel and parameters and output the accuracy on the test set. (5 points)

Submission Instructions: You need to submit a soft copy and a hard copy of your solutions.

- All solutions must be typed into a `pdf` report (named `CSCI567_hw3_fall16.pdf`). If you choose handwriting instead of typing, you will get 40% points deducted.
- For code, the only acceptable language is MATLAB or Python2.7. You MUST include a main script called `CSCI567_hw3_fall16.m` or `CSCI567_hw3_fall16.py` in the root directory of your code folder. After running this main script, your program should be able to generate all the results needed for the programming assignment, either as plots or console outputs. You can have multiple files (i.e your sub-functions), however, once we execute the main file in your code folder, your program should execute correctly.
- The soft copy should be a single `zip` file named `[lastname]_[firstname]_hw3_fall16.zip`. It should contain your `pdf` report (named `CSCI567_hw3_fall16.pdf`) having answers to all the problems, and the folder containing all your code. It must be submitted via Blackboard by **11:59pm** of the deadline date.
- The hard copy should be a printout of the report `CSCI567_hw3_fall16.pdf` and must be submitted to locker #19 at PHE building 1st floor by **5:00pm** of the deadline date.

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