CIS 520, Machine Learning, Fall 2017: Assignment 4
Due: Sunday, October 15th, 11:59pm, PDF to Canvas

For Problems 1-4

- **Instructions.** Please write up your responses to the following problems clearly and concisely. We encourage you to write up your responses using LaTeX; we have provided a LaTeX template, available on Canvas, to make this easier. **Submit your answers in PDF form to Canvas. We will not accept paper copies of the homework.**
- **Collaboration Policy.** You are allowed and encouraged to work together. You may discuss the homework to understand the problem and reach a solution in groups up to size **two students.** However, *each student must write down the solution independently, and without referring to written notes from the joint session.* **In addition, each student must write on the problem set the names of the people with whom you collaborated.** You must understand the solution well enough in order to reconstruct it by yourself. (This is for your own benefit: you have to take the exams alone.)

For Problem 5

- **Instructions.** This is a MATLAB programming problem. You can submit your code to be automatically checked for correctness to receive feedback ahead of time.

  We are providing you with codebase / templates / dataset that you will require for this problem. Download the file `hw4_kit.zip` from Canvas **before** beginning the assignment. **Please read through the documentation provided in ALL Matlab files before starting the assignment.** The instructions for submitting your homeworks and receiving automatic feedback are online on the wiki:

  http://alliance.seas.upenn.edu/~cis520/wiki/index.php?n=Resources.HomeworkSubmission

  If you are not familiar with Matlab or how Matlab functions work, you can refer to Matlab online documentation for help:

  http://www.mathworks.com/help/matlab/

  In addition, please use built-in Matlab functions rather than external library functions. Without proper reference to external library, auto-grader may fail even if your code runs perfectly on your local machine. Also, please DO NOT include data file in your submission.
- **Collaboration Policy.** You are allowed and encouraged to work together. You may discuss the homework to understand the problem and reach a solution in groups up to size **two students.** You can write **one copy** of solution code per group. However, each group member needs to submit plots separately (can be the same as your collaborator's) in the PDF submitted to canvas. For the auto-grader you need to submit **one copy** of your code per group. Be sure to include **your and your collaborator's** pennkey and name in the **group.txt** file. **We will be using automatic checking software to detect blatant copying of other groups' assignments, so, please, don't do it.**

Collaborators:

Type Collaborator Name Here

0 Survey [10 bonus points]

Complete the online survey on course backgrounds (see post from the instructors on piazza).

1 Convolutional Neural Network [20 points]

Convolutional Neural Network (CNN) is considered superior to regular neural networks when dealing with image classification problems. In this question, we will work through the details of a CNN model. For simplicity, we will assume there is no bias in this model.

1. Regular neural networks use fully connected layers. If we use a regular neural net to classify a set of 105×154 RBG images, how many weights do we need for each single neuron in the first hidden layer?

2. CNNs use a convolutional layer to help reduce the number of parameters in the model. Each neuron only connects to a small local spatial region of the images. If we use a filter (weights on the local region) of size $21 \times 14 \times 3$, how many weights do we need now for a single neuron in this convolutional layer?

3. We can adjust the size of the filters as well as the stride to limit the number of neurons we need in a convolutional layer. In the previous example of 105×154 RBG images, if we choose to use a stride of 7 in both x and y dimensions, how many neurons will be there in this layer? Note that we greatly reduce the dimension of the original image parameter space after this convolutional layer.

4. Using the following toy example, let's compute by hand exactly how convolutional layer works.

$$\text{input image} = \begin{bmatrix} 2 & 1 & 0 & 1 & 2 \\ 1 & 0 & -1 & 0 & 1 \\ 0 & -1 & -2 & -1 & 0 \\ 1 & 0 & -1 & 0 & 1 \\ 2 & 1 & 0 & 1 & 2 \end{bmatrix} \quad \text{filter 1} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \text{filter 2} = \begin{bmatrix} -3 & -2 & 1 \\ -2 & 1 & -2 \\ 1 & -2 & -3 \end{bmatrix}$$

   Here we have a $5 \times 5 \times 1$ input image, and we are going to use 2 different filters with size $3 \times 3 \times 1$ and stride 1 as our first convolutional layer. Compute and write the exact output from this simple convolutional layer (Hint: the output dimension is $3 \times 3 \times 2$).

2 Convex Sets and Convex Functions [10 points]

1. For each of the following sets in 3 dimensions, $\mathcal{C} \subseteq \mathbb{R}^3$, indicate whether or not $\mathcal{C}$ is a convex set and provide a brief explanation for your answer:

   (a) $\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{R}^3 \;\middle|\; x_1^2 + x_2^2 + x_3^2 \leq 8 \right\}$

   (b) $\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{R}^3 \;\middle|\; x_1^2 + x_2^2 + x_3^2 \geq 8 \right\}$

   (c) $\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{R}^3 \;\middle|\; \sqrt{|x_1|} + \sqrt{|x_2|} + \sqrt{|x_3|} \leq 2 \right\}$

   (d) $\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{R}^3 \;\middle|\; 0 \leq x_1 \leq 1,\, 2 \leq x_2 \leq 4,\, 3 \leq x_3 \leq 7 \right\}$

   (e) $\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{R}^3 \;\middle|\; \max(|x_1|, |x_2|, |x_3|) \leq 3,\, |x_1| + |x_2| + |x_3| \leq 5 \right\}$

2. For each of the following functions of 3 variables, $f : \mathbb{R}^3 \to \mathbb{R}$, indicate which of the following is true and provide a brief explanation for your answer:

   A. f is convex
   B. f is concave
   C. f is both convex and concave
   D. f is neither convex nor concave

(a) $f(\mathbf{x}) = x_1 - 2x_2 + 7x_3$
(b) $f(\mathbf{x}) = x_1^2 + x_2^2 - x_3^2$
(c) $f(\mathbf{x}) = -x_1^2 - 7x_3^2$
(d) $f(\mathbf{x}) = \max\left(x_1^2,\, x_2^2,\, x_3\right)$
(e) $f(\mathbf{x}) = |x_1| + |x_2| + |x_3|$

3 Optimization and Duality [10 points]

As a simple example of a constrained optimization problem that can be solved easily through duality, consider the following optimization problem where the goal is to minimize a linear function of two variables (x_1, x_2) subject to (x_1, x_2) lying in the Euclidean ball of radius 1:

$$\begin{array}{ll} \text{minimize} & 4x_1 + 3x_2 \\ \text{subject to} & x_1^2 + x_2^2 \leq 1\,. \end{array}$$

This is a convex optimization problem that satisfies Slater's condition, and so strong duality holds. You will find the solution by solving the dual problem.

1. Since there is just one inequality constraint, there will be one dual variable λ. Write down the Lagrangian function $\mathcal{L}(x_1, x_2, \lambda)$.

2. Write down the dual function $\phi(\lambda)$. (To obtain this, you will need to minimize $\mathcal{L}(x_1, x_2, \lambda)$ over x_1, x_2; you can do this by setting the gradient to zero, solving for x_1 and x_2 in terms of λ, and then plugging back these values of x_1 and x_2 into $\mathcal{L}(x_1, x_2, \lambda)$).

3. The dual problem is now

$$\begin{array}{ll} \text{maximize} & \phi(\lambda) \\ \text{subject to} & \lambda \geq 0\,. \end{array}$$

   Can you solve this problem? Note that if an *unconstrained* maximum of the dual objective is achieved at a value $\lambda \geq 0$, then this is automatically also a solution to the constrained dual problem. Once you have a solution λ^* to the dual problem, you can obtain a solution (x_1^*, x_2^*) to the primal problem by plugging in the value of λ^* into the expressions for x_1 and x_2 you derived in part (b) above. Does your solution (x_1^*, x_2^*) lie in the Euclidean ball of the primal constraint?

4 Kernel Functions [10 points]

Let $K_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $K_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be two symmetric, positive definite kernel functions, and for simplicity, assume that each implements dot products in some finite-dimensional space, so that there are vector mappings $\phi_1 : \mathcal{X} \to \mathbb{R}^{d_1}$ and $\phi_2 : \mathcal{X} \to \mathbb{R}^{d_2}$ for some $d_1, d_2 \in \mathbb{Z}_+$ such that

$$K_1(x, x') = \phi_1(x)^\top \phi_1(x')\,, \qquad K_2(x, x') = \phi_2(x)^\top \phi_2(x') \qquad \forall x, x' \in \mathcal{X}\,.$$

For each of the following functions $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, either find a vector mapping $\phi : \mathcal{X} \to \mathbb{R}^d$ for some suitable $d \in \mathbb{Z}_+$ such that $K(x, x') = \phi(x)^\top \phi(x')$ $\forall x, x'$, or explain why such a mapping cannot exist.

1. $K(x, x') = c \cdot K_1(x, x')$, where $c > 0$

2. $K(x, x') = K_1(x, x') + K_2(x, x')$

3. $K(x, x') = K_1(x, x') - K_2(x, x')$

4. $K(x, x') = K_1(f(x), f(x'))$, where $f : \mathcal{X} \to \mathcal{X}$ is any function.

5 SVM and Kernels: Programming Exercise [50 points]

In this problem you will write a piece of MATLAB code to experiment with SVMs and kernel functions on both synthetic and real-world data. You will use the LIBSVM[1] library (version 3.22) for training your SVMs. You will be given binary classification datasets $(\mathbf{X}, \mathbf{y})$, where $\mathbf{X}$ is an $m \times d$ matrix (m instances, each of dimension d) and y is an m-dimensional vector (with $y_i \in \{\pm 1\}$ being a binary label associated with the i-th instance in $\mathbf{X}$). You will be experimenting with the polynomial kernel K_{poly}, which is defined as $K_{\text{poly}}(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + 1)^q$ with degree parameter q; and the RBF kernel K_{rbf}, which is defined as $K_{\text{rbf}}(\mathbf{x}, \mathbf{x}') = e^{-\sigma \|\mathbf{x} - \mathbf{x}'\|_2^2}$ with (inverse) width parameter σ. Note that the linear kernel is a special case of the polynomial kernel with $q = 1$.

1. **Polynomial kernel:** You are provided a 2-dimensional synthetic dataset (See folder `\Synthetic\` in hw4-data.zip). You will be training SVMs on the training set using polynomial kernels for degree $q \in \{1, 2, 3, 4, 5\}$. For each q, in order to select the SVM parameter C, do 5-fold cross-validation on the training data using the cross-validation folds provided to you. Note that the training data is already divided into 5 folds, and the resulting datasets for cross-validation are provided in the folder `\Synthetic\Cross-Validation`; you should not create your own folds. Use cross-validation to select C from the range $\{1, 10, 10^2, 10^3, 10^4, 10^5\}$.

   Task: For each $q \in \{1, 2, 3, 4, 5\}$, after selecting C via cross-validation as above, plot the decision boundary of the learned classifier using the code provided `decision_boundary_SVM.m`. Also, plot the training and test error achieved by each value of $q \in \{1, 2, 3, 4, 5\}$ (q on the x-axis and the classification error on the y-axis; the train and test data are provided in the folder `\Synthetic`). Which value of q achieves the lowest test error?

2. **RBF kernel:** You will now repeat part (1) for the RBF kernel. This time, you will be training SVMs with different values of the RBF parameter σ in $\{0.01, 1, 10, 10^2, 10^3\}$. Again, for each σ, select C from the range $\{1, 10, 10^2, 10^3, 10^4, 10^5\}$ using cross-validation on the folds provided.

   Task: For each $\sigma \in \{0.01, 1, 10, 10^2, 10^3\}$, after selecting C via cross-validation as above, plot the decision boundary of the learned classifier using the code provided `decision_boundary_SVM.m`. Also, plot the training and test error achieved by each value of $\sigma \in \{0.01, 1, 10, 10^2, 10^3\}$ ($\log(\sigma)$ on the x-axis and the classification error on the y-axis). Which value of σ achieves the lowest test error?

3. **Cross-validation over kernels:** Here you will again experiment with the polynomial and RBF kernels over the same synthetic dataset as in parts (1) and (2), but now you will see if you can select a good *kernel* using cross-validation. You will do this twice: once to select a polynomial kernel among those with degree parameters q as above, and once to select an RBF kernel among those with parameters σ as above. In each case, you will also need to select the SVM parameter C as before. In other words, in each case, you will be doing cross-validation over 25 parameter combinations (5 kernel parameters $\times$ 5 values of C parameter), using the same cross-validation folds provided to you; finally, for each kernel parameter q or σ, keep the best C value.

   Task: Re-plot the training and test error curves above, but now also add a curve for the cross-validation error (you will have two plots, one for polynomial kernels showing training, test, and cross-validation errors as a function of q, and the other for RBF kernels showing these errors as a function of σ). Can you conclude that selecting the kernel parameters by cross-validation is a good approach?

4. **Real dataset:** For this part you will be training SVMs on the breast cancer dataset provided in the folder `\Breast-Cancer` (which was also used in HW2), and will be applying the learned classifiers to some test data that will be provided through autograder.

   Task: Write some code that takes as input a kernel type, which is either 'poly' or 'rbf', and a test dataset. Given these inputs, your code should train an SVM classifier on the *training set provided in the above folder* with the polynomial kernel if the input parameter is 'poly', and an SVM classifier with the RBF kernel if the input parameter is 'rbf'. In both cases, the kernel parameter (q or σ) and SVM parameter (C) should be chosen through cross-validation as done for the synthetic data above (using

[1] LIBSVM – A Library for Support Vector Machines: https://www.csie.ntu.edu.tw/~cjlin/libsvm/.

the same ranges of values as above). The training data here is provided in `\Breast-Cancer`; the 5-fold cross-validation data for selecting parameters is provided in `\Breast-Cancer\Cross-Validation`. After training the SVM, your code should apply the learned classifier to the input test set, and should output +1/-1 predictions on the test set.

A template for the code is provided to you in `\Code\SVM_train.m`. You will submit your code to the autograder, and receive your test error for each of the two kernel types. You should report these two test errors in your homework solution.