Homework 9
Linear Regression, Kernel Methods, SVM

CMU 10-601: Machine Learning (Fall 2017)
https://piazza.com/cmu/fall2017/10601b/
OUT: November 9, 2017
DUE: November 16, 2017 11:59 PM
Authors: Rhea, Praneet, Mo

START HERE: Instructions

- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. It is also OK to get clarification (but not solutions) from books or online resources, again after you have thought about the problems on your own. There are two requirements: first, cite your collaborators fully and completely (e.g., "Jane explained to me what is asked in Question 2.1"). Second, write your solution *independently*: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only. See the Academic Integrity Section on piazza for more information: https://piazza.com/cmu/fall2017/10601b/home
- **Late Submission Policy:** See the late submission policy here: https://piazza.com/cmu/fall2017/10601b/home
- **Submitting your work:**
  - **Gradescope:** For this assignment, we will use an online system called Gradescope for short answer and multiple choice questions. You can access the site here: https://gradescope.com/. You should already have been added to Gradescope using your Andrew ID. **Students are required to sign up using their Andrew ids or else they may not receive credit for all of their work.** If for some reason you have not been added to Gradescope, the entry code for this course is **97Y649**. Solutions can be handwritten, but should be labeled and clearly legible. If your writing is not legible, you will not be awarded marks. Alternatively, submissions can be written in LaTeX. Regrade requests can be made, however this gives the TA the right to regrade your entire paper, meaning if additional mistakes are found then points will be deducted.

Problem 1: Training Linear Regression with MAP [37 points]

In class, you have derived that MLE corresponding to minimizing sum of squared prediction errors (SSE) in linear regression with Gaussian random noise. In this problem, you will derive on your own the MAP estimate of the parameters of linear regression model $\mathbf{w} = (w_0, w_1)$. Assume Y is some deterministic linear function of X, plus random noise

$$y = w_0 + w_1 x + \epsilon \qquad\qquad \text{where } \epsilon \sim N(0, \sigma^2)$$

1. **[6 pt]** Let's assume a Gaussian prior distribution with mean 0 and variance λ over the weights w_0 and w_1, respectively, and w_0 and w_1 are independently, identically distributed. Write down the mathematical expression for the posterior probability of $\mathbf{w}$ given a single training example $< x^t, y^t >$.
2. **[6 pt]** What is the expression of posterior probability of $\mathbf{w}$ given all N training examples?
3. **[6 pt]** Write down the log posterior probability of $\mathbf{w}$ given all N training examples. For simplicity, you can drop the constant terms in the expression.
4. **[3 pt]** What is the MAP estimate of the parameters $\mathbf{w}$ of this linear regression model? Write down the mathematical expression. Note that at this point you do not need to worry about solving the optimization problem.
5. **[8 pt]** The optimization problem of the MAP estimate can be solved using gradient descent/ascent. Write down the update rules for training. Assume we are using a fixed step size α for gradient descent/ascent.
6. **[8 pt]** Now we want to use a prior with zero-mean Laplace distribution. A general assumption to make here is that different parameters w_i are independently, identically distributed with the zero-mean Laplace prior with $\beta = \lambda$. What is the MAP estimate of the parameters $\mathbf{w}$ of this linear regression model given N training examples? Write down the mathematical expression including the derivation steps. Note that at this point you do not need to worry about solving the optimization problem. Recall that a Laplace distribution, Laplace(μ, β) has the probability density function of $p(x) = \frac{1}{2\beta} \exp(-\frac{|x-\mu|}{\beta})$.

Problem 2: Short Answers: Kernel SVM, Logistic Regression, and Neural Network [16 points]

Consider the following machine learning classifiers we have covered in class so far:

- Kernel SVM which learns the function $f : X \to \{-1, +1\}$ by learning a classifier in the transformed space $\Phi(X)$.
- Neural network with sigmoid hidden units and a single sigmoid output unit with cross-entropy loss.
- Logistic regression for classification problems.

1. **[2 pt]** True or False: Kernel SVM can have non-linear decision boundary in the original space X, but the decision surface of learned Kernel SVM in the transformed space is always linear.
2. **[2 pt]** True or False: A neural network with a single sigmoid output unit learns a non-linear decision surface over the vector space defined by that sigmoid unit's inputs.
3. **[2 pt]** True or False: Logistic regression always learns a linear decision surface over its inputs.
4. **[3 pt]** Multiple choice: Which of these classifiers learns its feature space explicitly rather than just takes it as a given input to the algorithm? Select all that apply.

   (a) Kernel SVM

   (b) Neural network

   (c) Logistic regression
5. **[3 pt]** Multiple choice: Which of the above three classifiers outputs the predicted probabilities of the class being positive/negative? Select all that apply.

   (a) Kernel SVM

   (b) Neural network

   (c) Logistic regression
6. **[4 pt]** Assume the neural network is capable of learning the kernel SVM projection. Given the same training data and test data, assume that global optimal conditions are satisfied for both neural network and kernel SVM, then what is your expectation of the relative magnitude of the training errors and test errors trained with kernel SVM and neural network? Briefly justify your answers in no more than 3 sentences.

Problem 3: Kernel Functions [24 points]

Consider the following two-dimensional dataset as shown in Figure 1. You are given 32 training example points which are positive (+) or negative (marked by circles). The coordinate axes are labelled as $X1$ and $X2$. Thus every point, i can be represented by the tuple $(x_1^{(i)}, x_2^{(i)})$. Answer the following questions based on the figure.

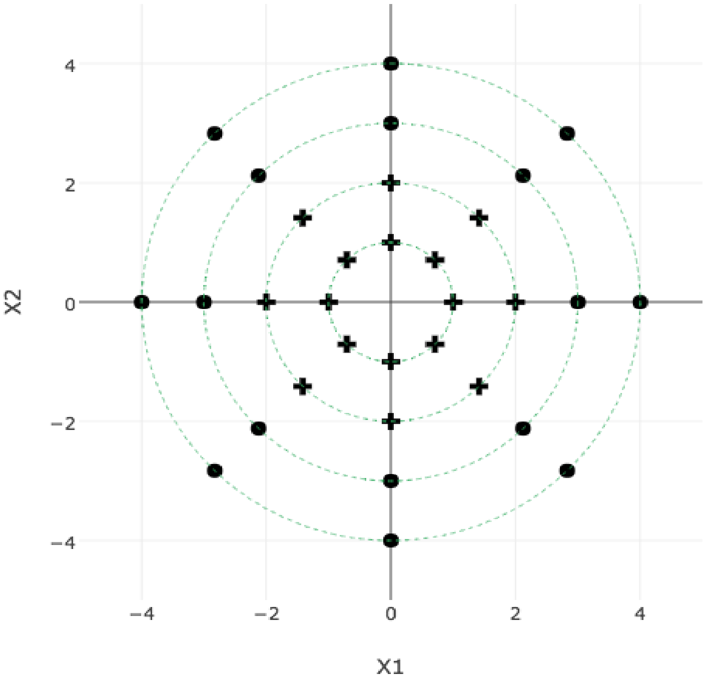
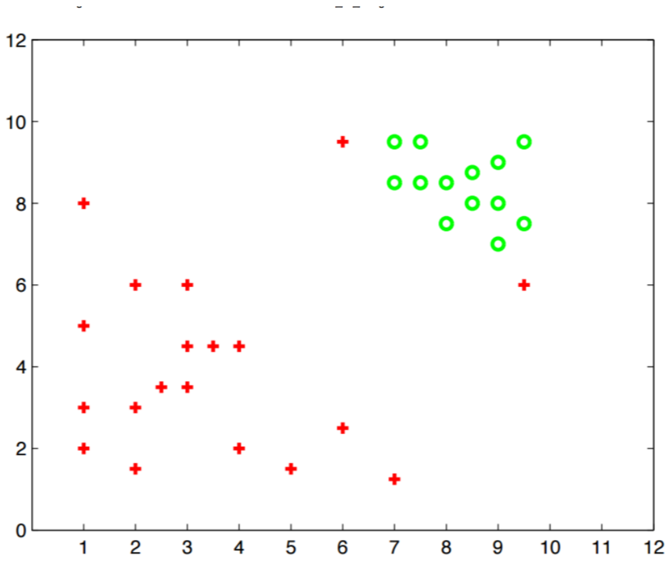


Figure 1

1. **[10 pt]** You can see that the given points are not linearly separable. Write down an one-dimensional feature transformation $f : \mathcal{R}^2 \rightarrow \mathcal{R}$ such that the transformed points $f(x_1^{(1)}, x_2^{(1)})$, $f(x_1^{(2)}, x_2^{(2)})$, $\ldots$, $f(x_1^{(32)}, x_2^{(32)})$ are linearly separable.
2. **[10 pt]** Draw your *transformed* points from the previous part on a one-dimensional line. Then draw the decision boundary which would be output by a hard-margin linear SVM in the *transformed* space. Also indicate which points in Figure 1 are the support vectors.
3. **[4 pt]** What is the VC dimension of the hard-margin linear SVM in this *transformed* feature space?

Problem 4: Hinge Loss and Maximum Margin [20 points]

1. [**4 pt**] What is an appropriate loss function for a classification problem and why? Choose one from the options below and justify your choice briefly.
   - Cross-entropy loss
   - Mean squared error loss
   - 0-1 loss
   - Hinge loss
2. [**4 pt**] What is an appropriate loss function for a regression problem and why? Choose one from the options below and justify your choice briefly.
   - Cross-entropy loss
   - Mean squared error loss
   - 0-1 loss
   - Hinge loss
3. [**12 pt**] Given the data set below, which of the following algorithms will be able to perfectly classify the 2 classes (green circles and red crosses) without a single mistake? Justify why/why not for each of them.



   (a) SVM with no kernel (Soft margin)

   (b) SVM with quadratic kernel with penalty on slack variable C=0

   (c) SVM with quadratic kernel with penalty on slack variable C=∞

   (d) Logistic regression (No kernel)

Collaboration Policy [3 points]

Reminder: you should submit your answers to collaboration policy questions to Gradescope with the rest of your solutions.

1. [**1 point**]:

   - Did you receive any help whatsoever from anyone in solving this assignment? Please answer *yes* or *no*.

   - If you answered *yes* on the previous question, please give full details below (e.g., *Christopher Nolan* explained to me what is asked in Question 3.4).

2. [**1 point**]:

   - Did you give any help whatsoever to anyone in solving this assignment? Please answer *yes* or *no*.

   - If you answered *yes* on the previous question, please give full details below (e.g., I pointed *Michael Bay* to section 2.3 since he didn't know how to proceed with Question 2).

3. [**1 point**]:

   - Did you find or come across code that implements any part of this assignment? Please answer *yes* or *no*.

   - If you answered *yes* on the previous question, please give full details below (book & page, URL & location, movies & scene, etc).