HOMEWORK 9
SVMs, K-MEANS, ADABOOST, PCA[1]

CMU 10-601: MACHINE LEARNING (FALL 2018)
https://piazza.com/cmu/fall2018/10601bd
OUT: Friday, November 30, 2018
DUE: Friday, December 7, 2018, 11:59pm EDT
TAs: Rongye, Rawal, Jeremy

START HERE: Instructions

Homework 9 covers topics on SVMs, K-Means, PCA, AdaBoost. The homework includes multiple choice, True/False, and short answer questions.

- **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 the course site for more information: http://www.cs.cmu.edu/~mgormley/courses/10601bd-f18/about.html#7-academic-integrity-policies
- **Late Submission Policy:** See the late submission policy here: http://www.cs.cmu.edu/~mgormley/courses/10601bd-f18/about.html#6-general-policies
- **Submitting your work:**
  - **Gradescope:** For written problems such as short answer, multiple choice, derivations, proofs, or plots, we will be using Gradescope (https://gradescope.com/). Please use the provided template. Submissions can be handwritten onto the template, 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 opportunity to regrade your entire paper, meaning if additional mistakes are found then points will be deducted. Each derivation/proof should be completed on a separate page. For short answer questions, you **should not** include your work in your solution. If you include your work in your solutions, your assignment may not be graded

[1] Compiled on Sunday 2$^{\text{nd}}$ December, 2018 at 15:21

correctly by our AI assisted grader. In addition, please tag the problems to the corresponding pages when submitting your work.

For multiple choice or select all that apply questions, shade in the box or circle in the template document corresponding to the correct answer(s) for each of the questions. For LaTeXusers, use ■ and ●for shaded boxes and circles, and don't change anything else.

Instructions for Specific Problem Types

For “Select One” questions, please fill in the appropriate bubble completely:

Select One: Who taught this course?

● Matt Gormley

○ Marie Curie

○ Noam Chomsky

If you need to change your answer, you may cross out the previous answer and bubble in the new answer:

Select One: Who taught this course?

● Matt Gormley

○ Marie Curie
⊗ Noam Chomsky

For “Select all that apply” questions, please fill in all appropriate squares completely:

Select all that apply: Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton

□ I don’t know

Again, if you need to change your answer, you may cross out the previous answer(s) and bubble in the new answer(s):

Select all that apply: Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton
☒ I don’t know

For questions where you must fill in a blank, please make sure your final answer is fully included in the given space. You may cross out answers or parts of answers, but the final answer must still be within the given space.

Fill in the blank: What is the course number?

10-601

10-~~7~~601

1 Support Vector Machines [19 pts]

In class, we discussed the properties and formulation of hard-margin SVMs, where we assume the decision boundary to be linear and attempt to find the hyperplane with the largest margin. Here, we introduce a new class of SVM called soft margin SVM, where we introduce the slack variables e_i to the optimization problem and relax the assumptions. The formulation of soft margin SVM with no Kernel is

$$\begin{aligned} \underset{\mathbf{w},b,e}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\left(\sum_{i=1}^{N} e_i\right) \\ \text{subject to} \quad & y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - e_i, \ \forall\, i = 1, \dots, N \\ & e_i \geq 0, \ \forall\, i = 1, \dots, N \end{aligned}$$

1. **[3pts]** Consider the ith training example $(\mathbf{x}^{(i)}, y^{(i)})$ and its corresponding slack variable e_i. Assuming $C > 0$ and is fixed, what would happen as $e_i \to \infty$?

   Select all that apply:

   ■ the constraint $y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - e_i$ would hold for almost all $\mathbf{w}$.

   □ there would be no vector that satisfies the constraint $y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - e_i$

   ■ the objective function would approach infinity.

   With this in mind, we hope that you can see why soft margin SVM can be applied even when the data is not linearly separable.

2. **[5pts]** What **could** happen as $C \to \infty$? Do **not** assume that the data is linearly separable unless specified.

   Select all that apply:

   ■ When the data is linearly separable, the solution to the soft margin SVM would converge to the solution of hard margin SVM.

   □ There is no solution $\mathbf{w}, b$ satisfying all the constraints in the optimization problem.

   □ Any arbitrary vector $\mathbf{w}$ and scalar b can satisfy the constraints in the optimization problem.

   ■ The optimal weight vector would converge to the zero vector $\mathbf{0}$.

   ■ When C approaches to infinity, it could help reduce overfitting.

3. **[5pts]** What **could** happen as $C \to 0$? Do **not** assume that the data is linearly separable unless specified.

   Select all that apply:

   ■ When the data is linearly separable, the solution to the soft margin SVM would converge to the solution of hard margin SVM.

   □ There is no solution $\mathbf{w}, b$ satisfying all the constraints in the optimization problem.

   □ Any arbitrary vector $\mathbf{w}$ and scalar b can satisfy the constraints in the optimization problem.

   ■ The optimal weight vector would converge to be the zero vector $\mathbf{0}$.

   □ When C approaches to 0, doing so could help reduce overfitting.

4. **[3pts]** An extension to soft margin SVM (or, an extension to the hard margin SVM we talked in class) is the 2-norm SVM with the following primal formulation

$$\begin{aligned} \underset{\mathbf{w},b,e}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\left(\sum_{i=1}^{N} e_i^2\right) \\ \text{subject to} \quad & y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - e_i, \ \forall\, i = 1, \ldots, N \\ & e_i \geq 0, \ \forall\, i = 1, \ldots N \end{aligned}$$

   Which of the following is true about the 2-norm SVM? (Hint: think about ℓ_1-regularization versus ℓ_2 regularization!)

   Select one:

   ○ If a particular pair of parameters $\mathbf{w}^*, b^*$ minimizes the objective function in soft margin SVM, then this pair of parameters is guaranteed to minimize the objective function in 2-norm SVM.

   ● 2-norm SVM penalizes large e_i's more heavily than soft margin SVM.

   ○ One drawback of 2-norm SVM is that it cannot utilize the kernel trick.

   ○ None of the above.

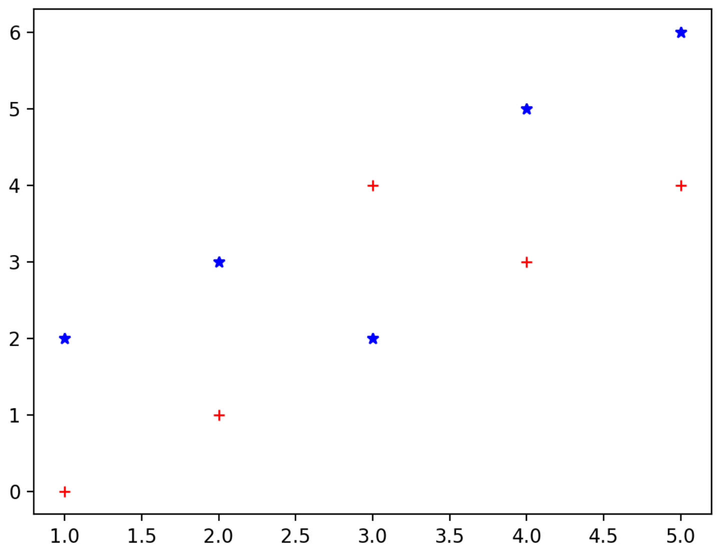


Figure 1: SVM dataset

5. [**3pts**] Consider the dataset shown in Figure 1. Which of the following models, when properly tuned, could correctly classify **ALL** the data points?

Select all that apply:

□ Logistic Regression without any kernel

□ Hard margin SVM without any kernel

■ Soft margin SVM without any kernel

■ Hard margin SVM with RBF Kernel

■ Soft margin SVM with RBF Kernel

2 Kernels [19pts]

1. **[2pt]** Consider the following kernel function:

$$K(x, x') = \begin{cases} 1, \text{ if } x = x' \\ 0, \text{ otherwise} \end{cases}$$

True or False: In this kernel space, any labeling of points from any training data X will be linearly separable.

○ True

● False

2. **[3pts]** Suppose that input-space is three-dimensional, $x = (x_1, x_2, x_3)^T$. The feature mapping is defined as -

$$\phi(x) = (x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3)^T$$

What is the corresponding kernel function, i.e. $K(x, z)$? **Select one.**

○ $(x_1z_1)^2 + (x_2z_2)^2 + (x_3z_3)^2$

○ $(x^Tz)^3$

● $(x^Tz)^2$

○ x^Tz

3. **[3pts]** Suppose that input-space is three-dimensional, $x = (x_1, x_2, x_3)^T$. The feature mapping is defined as -

$$\phi(x) = (x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3)^T$$

Suppose we want to compute the value of kernel function $K(x, z)$ on two vectors $x, z \in \mathbb{R}^3$. We want to check how many additions and multiplications are needed if you map the input vector to the feature space and then perform dot product on the mapped features. Report $\alpha + \beta$, where α is the number of multiplications and β is the number of additions.

Note: Multiplication/Addition with constants should also be included in the counts.

29

4. **[3pts]** Suppose that input-space is three-dimensional, $x = (x_1, x_2, x_3)^T$. The feature mapping is defined as -

$$\phi(x) = (x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3)^T$$

Suppose we want to compute the value of kernel function $K(x, z)$ on two vectors $x, z \in \mathbb{R}^3$.We want to check how many additions and multiplications are needed if you do the computation through the kernel function you derived above. Report $\alpha + \beta$, where α is the number of multiplications and β is the number of additions.

Note: Multiplication/Addition with constants should also be included in the counts.

$\boxed{6}$

5. **[3pts]** Suppose one dataset contains four data points in $\mathbb{R}^1$ space, as shown in Figure 2

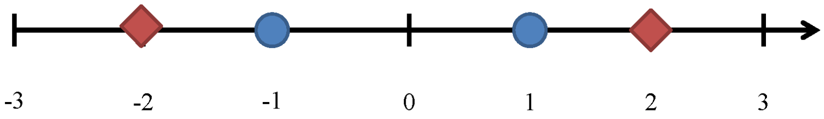


Figure 2: Data in $\mathbb{R}^1$

Different shapes of the points indicate different labels. If we train a linear classifier on the dataset, what is the lowest training error for a linear classifier on $\mathbb{R}^1$?

$\boxed{0.25}$

6. **[3pts]** Following the above question, we use feature mapping $\phi(x) = (x, x^2)$ to project the data to $\mathbb{R}^2$ space. Again we train a linear classifier on the projected dataset. What is the lowest traning error for a linear classifier on $\mathbb{R}^2$?

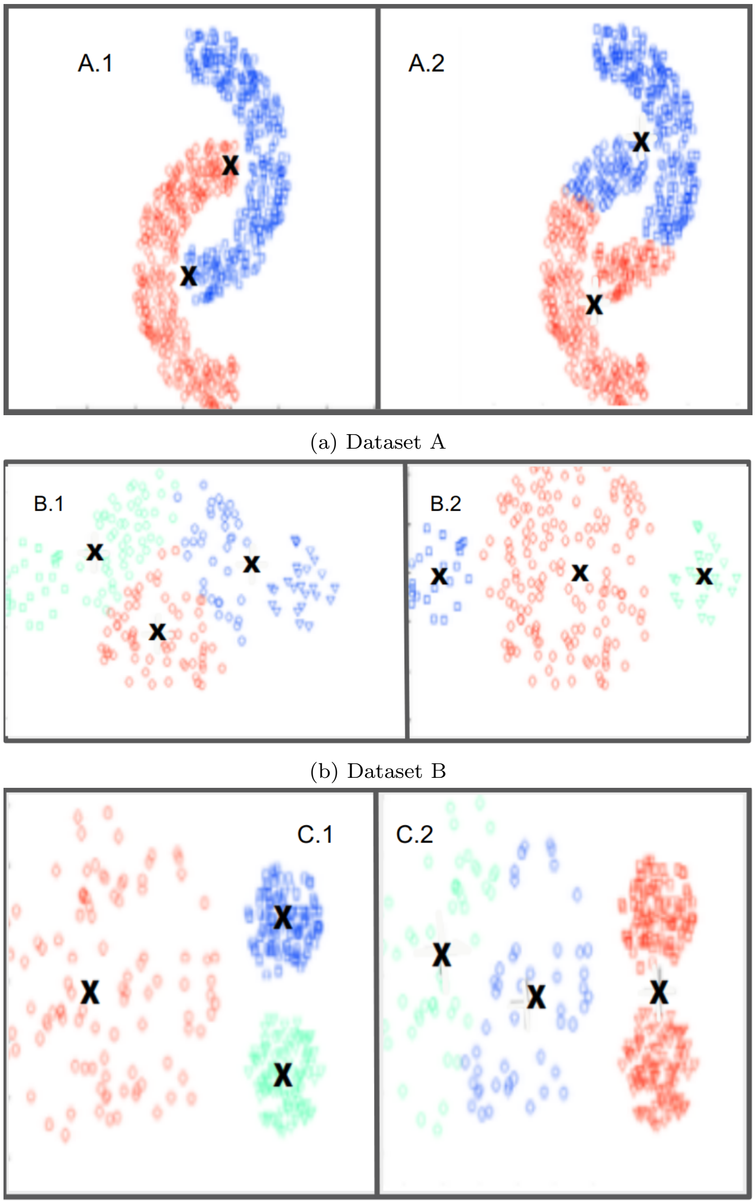
$\boxed{0}$

7. **[2pt] True or False:** Given the same training data, in which the points are linearly separable, the margin of the decision boundary produced by SVM will always be greater than or equal to the margin of the decision boundary produced by Perceptron.

● True

○ False

3 K Means [22pts]



(a) Dataset A

(b) Dataset B

(c) Dataset C

Figure 3: Datasets

1. **[3pts]** Consider the 3 datasets A, B and C as shown in Figure 3. Each dataset is classified into k clusters as represented by different colors in the figure. For each dataset, determine which image with cluster centers (denoted by X) is generated by K-means method.The distance measure used here is the Euclidean distance.

1.1. [**1pt**] Dataset A **(Select one)**

○ A.1

● A.2

1.2. [**1pt**] Dataset B **(Select one)**

● B.1

○ B.2

1.3. [**1pt**] Dataset C **(Select one)**

○ C.1

● C.2

2. [**13pts**] Consider a Dataset **D** with 5 points as shown below. Perform a k-means clustering on this dataset with k as 3 using the Euclidean distance as the distance function. Remember that in the K-means algorithm, an iteration consists of performing following tasks: Assigning each data point to it's nearest cluster center followed by recomputation of those centers based on all the data points assigned to it. Initially, the 3 cluster centers are chosen randomly as $\mu 0$ = (5.3,3.5) (0), $\mu 1$ = (5.1, 4.2) (1), $\mu 2$ = (6.0, 3.9) (2).

$$D = \begin{bmatrix} 5.5 & 3.1 \\ 5.1 & 4.8 \\ 6.4 & 3.6 \\ 5.6 & 4.7 \\ 6.7 & 3.7 \end{bmatrix}$$

2.1. [**3pts**] Which of the following points will be the center for cluster 0 after the first iteration? **Select one:**

○ (5.4 , 3.2)

○ (5.5 , 3.2)

● (5.5 , 3.1)

○ (5.4 , 3.1)

2.2. [**3pts**] Which of the following points will be the center for cluster 1 after the first iteration? **Select one:**

○ (5.2 , 4.7)

○ (5.5 , 4.7)

○ (5.5 , 4.6)

● (5.3 , 4.7)

2.3. [**3pts**] Which of the following points will be the center for cluster 2 after the first iteration? **Select one:**

○ (6.5 , 3.5)

○ (6.5 , 3.9)

○ (6.5 , 3.8)

● (6.5 , 3.6)

2.4. [**2pt**] How many points will belong to cluster 1 after the first iteration? **Select one:**

● 2

○ 3

○ 5

○ 1

2.5. [**2pt**] How many points will belong to cluster 2 after the first iteration? **Select one:**

● 2

○ 3

○ 5

○ 1

3. [**6pts**] Recall that in k-means clustering we attempt to find k cluster centers $c_j \in \mathbb{R}^d, j \in \{1, \dots, k\}$ such that the total distance between each datapoint and the nearest cluster center is minimized. Then the objective function is,

$$\sum_{i=1}^{n} \min_{j \in \{1,\dots,k\}} ||x_i - c_j||^2 \tag{1}$$

In other words, we attempt to find $c_1, \dots, c_k$ that minimizes Eq. (1), where n is the number of data points. To do so, we iterate between assigning x_i to the nearest cluster center and updating each cluster center c_j to the average of all points assigned to the j th cluster. Instead of holding the number of clusters k fixed, your friend John tries to minimize Eq. (1) over k. Yet, you found this idea to be a bad one.

Specifically, you convinced John by providing two values α, the minimum possible value of Eq. (1), and β, the value of k when Eq. (1) is minimized.

3.1. [**3pts**] What is the value of $\alpha + \beta$ when $n = 100$?

$\boxed{100}$

3.2. [**3pts**] We want to see how k-means clustering works on a single dimension. Consider the case in which k = 3 and we have 4 data points $x_1 = 1, x_2 = 2, x_3 = 5, x_4 = 7$. What is the optimal value of the objective Eq. (1)?

0.5

4 PCA [13pts]

1. [**4pts**] Assume we are given a dataset X for which the eigenvalues of the covariance matrix are: (2.2, 1.7, 1.4, 0.8, 0.4, 0.2, 0.15, 0.02, 0.001). What is the smallest value of k we can use if we want to retain 90% of the variance (sum of all the variances in value) using the first k principal components?

   5

2. [**3pts**] Assume we apply PCA to a matrix $X \in R^{n \times m}$ and obtain a set of PCA features, $Z \in R^{n \times m}$ .We divide this set into two, $Z1$ and $Z2$.The first set, Z1, corresponds to the top principal components. The second set, Z2, corresponds to the remaining principal components. Which is more common in the training data: **Select one:**

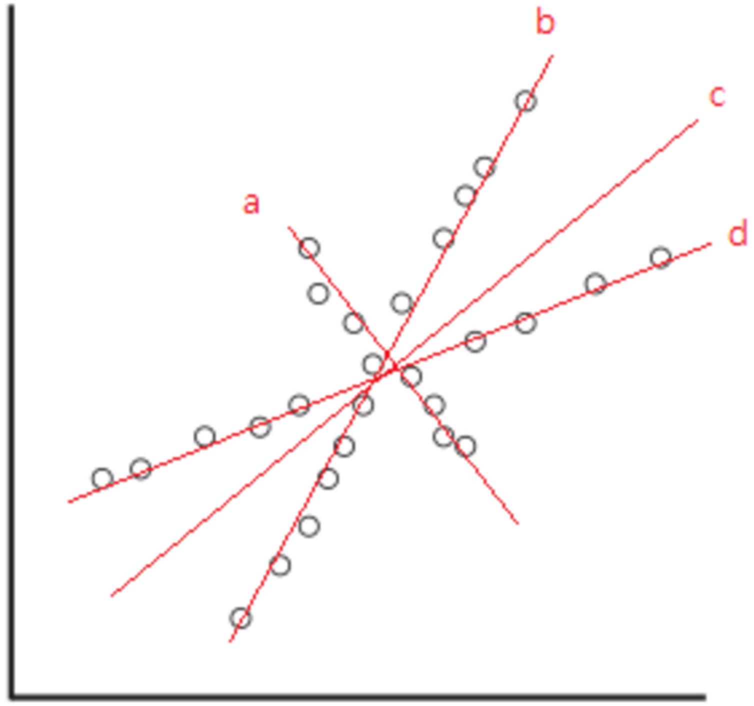
   ● a point with large feature values in $Z1$ and small feature values in $Z2$

   ○ a point with large feature values in $Z2$ and small feature values in $Z1$

   ○ a point with large feature values in $Z2$ and large feature values in $Z1$

   ○ a point with small feature values in $Z2$ and small feature values in $Z1$

3. [**2pts**] For the data set shown below, what will be it's first principal component?



Select one:

● d

○ b

○ c

○ a

4. [**2pts**] **NOTE : This is continued from the previous question.** What is the second principal component in the figure from the previous question? **Select one:**

   ○ d

   ● b

   ○ c

   ○ a

5. [**2pts**] **NOTE : This is continued from the previous question.** What is the third principal component in the figure from the previous question? **Select one:**

   ○ (a)

   ○ (b)

   ○ (c)

   ○ (d)

   ● None of the above

5 Lagrange Multipliers [12pts]

Lagrange multipliers are helpful in solving constrained optimization problems. In this problem, we will analyze how to apply this method to a simple question: finding the MLE of a categorical distribution. Let $\mathbf{1}$ be the indicator function, where

$$\mathbf{1}\{a = b\} = \begin{cases} 1 & \text{, if } a = b \\ 0 & \text{, otherwise} \end{cases}$$

The probability mass function for categorical distribution with K categories, assuming $K > 1$, can then be written as

$$\mathbb{P}(x) = \prod_{k=1}^{K} p_k^{\mathbf{1}\{x=k\}},$$

where $\sum_{k=1}^{K} p_k = 1$ and $0 \leq p_k \leq 1$. Let $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ be the samples we have, the log likelihood of the categorical distribution is then

$$\sum_{i=1}^{N} \sum_{k=1}^{K} \mathbf{1}\{x^{(i)} = k\} \log(p_k)$$

Finding the MLE of the categorical distribution is equivalent to the following optimization problem

$$\begin{aligned} \underset{p_1,\ldots,p_K}{\text{minimize}} \quad & -\sum_{i=1}^{N} \sum_{k=1}^{K} \mathbf{1}\{x^{(i)} = k\} \log(p_k) \\ \text{subject to} \quad & p_k \geq 0 \ \forall \ k = 1, \ldots, K \\ & p_k \leq 1, \ \forall \ k = 1, \ldots, K \\ & \sum_{k=1}^{K} p_i = 1 \end{aligned}$$

We attempt to solve this problem by solving its dual. To do this, recall first that the dual problem can be written as

$$\max_{\theta} \min_{p_1,\ldots,p_k} L(p_1, \ldots, p_k; \theta),$$

where θ is the variables we introduce when deriving the dual and $L(p_1, \ldots, p_k; \theta)$ is the Lagrangian function.

1. **[2pts]** Which of the following is the formulation for the Lagrangian function L? **Select one.**

   ○ $\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbf{1}\{x^{(i)}=k\}\log(p_k)+\sum_{k=1}^{K}\alpha_k p_k+\sum_{k=1}^{K}\beta_k(p_k-1)+\gamma(\sum_{k=1}^{K}p_k-1)$

   ● $-\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbf{1}\{x^{(i)}=k\}\log(p_k)-\sum_{k=1}^{K}\alpha_k p_k+\sum_{k=1}^{K}\beta_k(p_k-1)+\gamma(\sum_{k=1}^{K}p_k-1)$

   ○ $-\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbf{1}\{x^{(i)}=k\}\log(p_k)-\sum_{k=1}^{K}p_k+\sum_{k=1}^{K}\beta_k(p_k-1)+\gamma(\sum_{k=1}^{K}p_k-1)$

   ○ $-\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbf{1}\{x^{(i)}=k\}\log(p_k)-\sum_{k=1}^{K}p_k+\sum_{k=1}^{K}(p_k-1)+\sum_{k=1}^{K}p_k$

Now, we would like to use the method of Lagrange Multipliers to prove that maximizing the variance in PCA gives an eigenvector. In general, a constrained optimization problem has the following standard form:

$$\begin{aligned} \min_{w} \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \leq 0, i = 1, ..., k \\ & h_i(w) = 0, i = 1, ..., l \end{aligned}$$

This is also called the primal problem. We introduce new variables $\alpha_i \geq 0$ and β_i (no need to be non-negative) called Lagrange multipliers and study the generalized Lagrangian defined by

$$L(w, \alpha, \beta) = \quad f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

A discussion of solving constrained optimization is based on the fact that if w^* is a feasible local minimum of the objective function that satisfies all the constraints, then there exist Lagrange multipliers α and β, such that the Karush-Kuhn-Tucker (KKT) conditions are satisfied:

$$\begin{aligned} \frac{\partial L(w, \alpha, \beta)}{\partial w}|_{w=w^*} &= 0 & \\ h_i(w^*) &= 0, i = 1, ..., l & \\ \alpha_i g_i(w^*) &= 0, i = 1, ..., k & \text{Complementary slackness} \\ g_i(w^*) &\leq 0, i = 1, ..., k & \text{Primal feasibility} \\ \alpha_i &\geq 0, i = 1, ..., k & \text{Dual feasibility} \end{aligned}$$

The points that satisfy the KKT gives a set of all the local optimums. In the following examples, we can efficiently solve the optimization problem by finding the KKT points and investigate each of them to optimize the objective function globally.

From class, we know that minimizing reconstruction error is equivalent to maximizing variance, i.e.,

$$\begin{aligned} \max_{\mathbf{v}} \quad & \frac{1}{N} \sum_{i=1}^{N} (\mathbf{v}^T x^{(i)})^2 \\ \text{s.t.} \quad & \mathbf{v}^T \mathbf{v} = 1 \end{aligned}.$$

Let $\mathbf{X} = [x^{(1)}, x^{(2)}, ..., x^{(N)}]$. Using the covariance matrix $\mathbf{A} = \frac{1}{N}\mathbf{X}\mathbf{X}^T$, we get to the following standard constrained minimization problem

$$\begin{aligned} \min_{\mathbf{v}} \quad & -\mathbf{v}^T \mathbf{A} \mathbf{v} \\ \text{s.t.} \quad & \mathbf{v}^T \mathbf{v} - 1 = 0. \end{aligned}$$

2. [**2pts**] Please write down the Lagrangian $L(\mathbf{v}, \lambda)$, using the Lagrangian multiplier λ.

Solution

$L(\mathbf{v}, \lambda) = -\mathbf{v}^T \mathbf{A}\mathbf{v} + \lambda(\mathbf{v}^T \mathbf{v} - 1)$

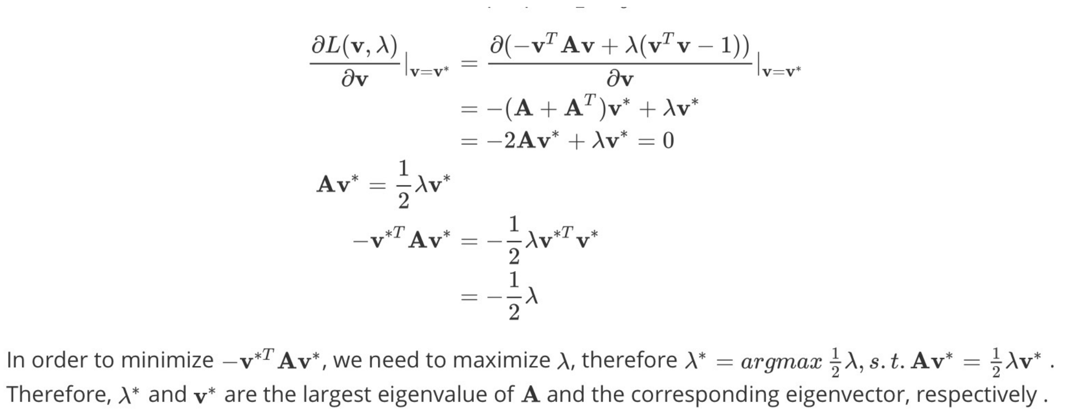
3. [**4pts**] Please write down the KKT conditions of the optimization problem.

Solution

$$\frac{\partial L(\mathbf{v}, \lambda)}{\partial \mathbf{v}}|_{\mathbf{v}=\mathbf{v}^*} = 0$$
$$\mathbf{v}^{*T}\mathbf{v}^* - 1 = 0$$

4. [**4pts**] Please solve the optimization problem by solving the KKT conditions and show that the optimal solution is $(\mathbf{v}^*, \lambda^*)$, where λ^* and $\mathbf{v}^*$ are the largest eigenvalue of $\mathbf{A}$ and the corresponding eigenvector, respectively .

Solution



$$\begin{aligned}
\frac{\partial L(\mathbf{v},\lambda)}{\partial \mathbf{v}}|_{\mathbf{v}=\mathbf{v}^*} &= \frac{\partial(-\mathbf{v}^T\mathbf{A}\mathbf{v}+\lambda(\mathbf{v}^T\mathbf{v}-1))}{\partial \mathbf{v}}|_{\mathbf{v}=\mathbf{v}^*} \\
&= -(\mathbf{A}+\mathbf{A}^T)\mathbf{v}^* + \lambda\mathbf{v}^* \\
&= -2\mathbf{A}\mathbf{v}^* + \lambda\mathbf{v}^* = 0 \\
\mathbf{A}\mathbf{v}^* = \frac{1}{2}\lambda\mathbf{v}^* \\
-\mathbf{v}^{*T}\mathbf{A}\mathbf{v}^* &= -\frac{1}{2}\lambda\mathbf{v}^{*T}\mathbf{v}^* \\
&= -\frac{1}{2}\lambda
\end{aligned}$$

In order to minimize $-\mathbf{v}^{*T}\mathbf{A}\mathbf{v}^*$, we need to maximize λ, therefore $\lambda^* = argmax\ \frac{1}{2}\lambda, s.t.\ \mathbf{A}\mathbf{v}^* = \frac{1}{2}\lambda\mathbf{v}^*$. Therefore, λ^* and $\mathbf{v}^*$ are the largest eigenvalue of $\mathbf{A}$ and the corresponding eigenvector, respectively .

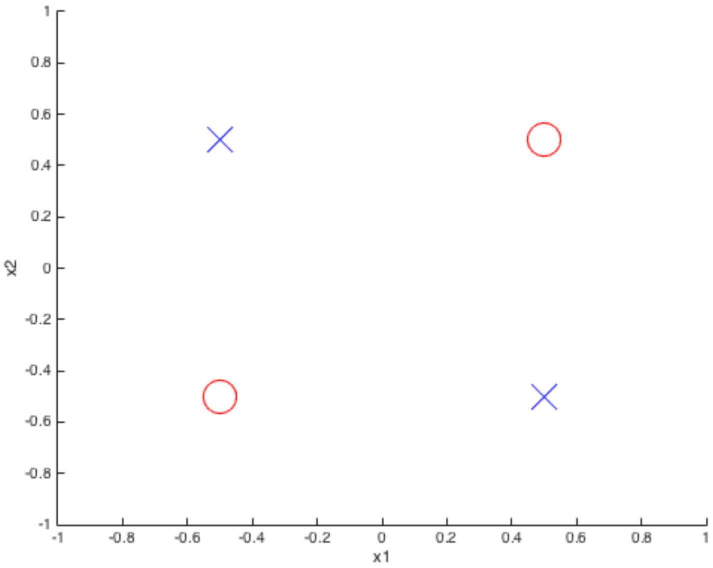


Figure 4: AdaBoost dataset.

6 AdaBoost [25pts]

Decision stumps are a popular family of binary classifiers used to implement the AdaBoost algorithm. Typically, they work as "weak classifiers." A decision stump is a linear separator that is axis-aligned. In other words, it is a 1-level decision tree. It only has a root node. More formally, suppose the inputs are vectors in $\mathbb{R}^d$. A decision stump is a function h where

$$h(\vec{x}) = \begin{cases} 1 & \text{if } x_i \geq c \\ -1 & \text{otherwise} \end{cases}$$

for some $i \in d$ and $c \in \mathbb{R}$. (You may also change $\geq$ to $\leq$, $<$, or $>$ and the resulting function is still a decision stump.)

1. [**2pts**] For the data in Figure 4, what is the lowest training error that can be achieved by a single decision stump?

Select one:

○ 25%

● 50%

○ 75%

○ 100%

Point	x_1	x_2	y
1	1	3	-1
2	1	1	+1
3	2	2	+1
4	3	3	+1
5	3	1	-1

Table 1: Data for AdaBoost Q2

2. **[23pts]** Now suppose we have the data in Table 1. On the first iteration of AdaBoost using decision stumps, we choose the classifier h where

$$h(\vec{x}) = \begin{cases} 1 & \text{if } x_1 \geq 1.5 \\ -1 & \text{otherwise} \end{cases}$$

. Fill in the blanks to complete this iteration of the algorithm. Use normalization factor $Z_t = 2\sqrt{err_t(1 - err_t)}$.

(a) **[3pts]** $err_1 = \boxed{0.4}$

(b) **[3pts]** $\alpha_1 = \boxed{0.203}$

(c) **[2pts]** Which weight(s) will be increased? (Note: $D(i)$ is the weight of point i)

Select all that apply:

□ $D(1)$

■ $D(2)$

□ $D(3)$

□ $D(4)$

■ $D(5)$

(d) **[7pts]** After updating, what are the weights $D(i)$? Fill in the table below, rounding to 4 decimal places.

$D(1)$	0.1667
$D(2)$	0.2500
$D(3)$	0.1667
$D(4)$	0.1667
$D(5)$	0.2500

(e) **[2pts]** On each iteration, AdaBoost will choose a weak classifier $h(\vec{x})$ which minimizes the error,

$$err_t = \frac{\sum_{i=1}^{m} D_t(i) * \mathbf{1}\{h(\vec{x}^{(i)}) \neq y^{(i)}\}}{\sum_{i=1}^{m} D_t(i)}$$

Continuing from above, which decision stump would we choose as our classifier in the second iteration?

Select one:

○ $h(\vec{x}) = \begin{cases} 1 & \text{if } x_2 \geq 1.5 \\ -1 & \text{otherwise} \end{cases}$

○ $h(\vec{x}) = \begin{cases} 1 & \text{if } x_2 \leq 2.5 \\ -1 & \text{otherwise} \end{cases}$

● $h(\vec{x}) = \begin{cases} 1 & \text{if } x_1 \leq 2.5 \\ -1 & \text{otherwise} \end{cases}$

○ $h(\vec{x}) = \begin{cases} 1 & \text{if } x_1 \geq 1.5 \\ -1 & \text{otherwise} \end{cases}$

(f) **[6pts]** What is the least number of iterations in which it is possible to achieve 0 training error on this dataset? (Hint: implementation of the AdaBoost is suggested to test your answer)

Select one:

○ 2

○ 3

○ 4

○ 5

● None of the above

Collaboration Questions Please answer the following:

After you have completed all other components of this assignment, report your answers to the collaboration policy questions detailed in the Academic Integrity Policies found here.

1. Did you receive any help whatsoever from anyone in solving this assignment? Is so, include full details.
2. Did you give any help whatsoever to anyone in solving this assignment? Is so, include full details.
3. Did you find or come across code that implements any part of this assignment ? If so, include full details.

Solution

1. No
2. No
3. No