UIUC-CS412 "Introduction to Data Mining" (Fall 2015)

Final Exam Solution

Friday, Dec. 11, 2015
180 minutes, 150 points

Name: NetID:

1 [30']	2 [10']	3 [10']	4 [10']	5 [25']	6 [40']	7 [22']	8 [3']	Total [150']

1. Short answer questions [30'].

(a) [3'] What is the advantage of cosine similarity over Euclidean distance in measuring the document similarity?

Solution.
Cosine similarity focuses on orientation but not magnitude. It normalizes the similarity value to a specific range. It will not be influenced by the document length. It is very efficient to evaluate, especially for sparse vectors, as only the non-zero dimensions need to be considered.

(b) [3'] Consider a transaction database in which for each single item x, there is at least one transaction that only contains x. Can the number of closed frequent patterns be smaller than the number of frequent patterns? (Answer yes or no; no need to explain.)

Solution.
Yes. The number of closed patterns is equal to the number of all frequent patterns if for each item (not only single items) x, there is at least one transaction that only contains x.

(c) [3'] In graph mining algorithm gSpan, what key steps were used to avoid redundant computation but not to miss any possible frequent graph?

Solution.
minimum FDS code are used to identify graphs already mined. Rightmost extension promises mining is complete.

(d) [3'] In GSP, for a sequence pattern of length 100, what is the lower bound of the size of length-1 candidates? How about that of length-2? And what is the number for all candidates? (Use the tightest lower bound estimation you can get.)

Solution.
choose 1 out of 100, choose 2 out of 100, and $2^100 - 1$.

(e) [3'] The larger k is, the more accurate the kNN classifier will be. (Choose from *True*/*False*, and briefly explain.)

Solution.
Too large a k results in predictions less specific to the query point.

(f) [3'] Decision trees are superior to Naive Bayes classifiers since they give more interpretable results. (Choose from *True*/*False*, and briefly explain.)

Solution.
Interpretability is only one pro of decision trees but does not make it superior. The probabilities provided by Naive Bayes can be used as confidence measure.

(g) [3'] A paper suggests running a single-link hierarchical clustering algorithm for a few iterations on the K-means clustering result. List one advantage of this approach over using K-Means alone.

Solution.
It helps to detect non-convex clusters, because when we merge clusters from K-Means using AGNES, the final clusters can have non-convex shapes. The other accepted answer is (1) it helps to alleviate the issues when previously choosing K too big because AGNES will reduce the number of clusters in that case (2) it helps to eliminate outliers if we run AGNES within each cluster from K-Means because it will detect extremely small clusters in each cluster from K-Means, which has the effect of detecing outliers.

(h) [3'] In some cases, using PCA to reduce dimension can make classification much less precise than it would have been in the original feature space. Explain how this can happen.

Solution.
That happens when the dimensions with low variance actually contains useful information for classification. One of PCA's key assumptions is dimensions with low variances are unimportant but it is possible that in reality those dimensions are actually important for a particular task like classification. Anyway, this situation does not happen offen. That is the reason why PCA is used widely.

(i) [3'] In our guest lecture by Matt Ahrens, according to the speaker, what is the takeaway of the lecture, if there is only one thing?

Solution.
Do not be the fraud writer who creates fraud ads on the web.

(j) [3'] In our guest lecture by Matt Ahrens, he explained that fraudulent advertisements are rather significant. What is the typical rate he reported? a) 1-3% b) 5-10% c) 15-20% d) 30-40%

Solution.
c)

2. Data preprocessing [10'].

(a) [4'] Some researchers want to study if lung cancer correlates with smoking. They interviewed 1000 people, and got the following statistics:

- cancer and smoking: 400
- cancer and non-smoking: 100
- no-cancer and smoking: 100
- no-cancer and non-smoking: 400

(i) [2′] Should we use correlation coefficient (Pearson's product moment coefficient) to measure the correlation? If yes, explain your answer; if no, explain and suggest a better measure.

Solution.
No, because they are categorical data, while correlation coefficient is applicable only for numerical data (like heights of students)!

(ii) [2′] Calculate the correlation using the measure you chose.

Solution.
$(400-250)^2/250+(100-250)^2/250+(400-250)^2/250+(100-250)^2/250 = 4*150^2/250 = 360$ Someone chooses to use Lift or Jaccard coefficient. Both are acceptable!

(b) [6'] We want to cluster 20000 documents based on the words they contain.

(i) [2′] Describe what would be the instances and features in this data mining problem.

Solution.
Instances are documents and features are words in dictionary. Some students do not answer the questions explicitly, which results in point deduction!

(ii) [4′] You should realize an issue with the number of dimensions of the dataset. What is the issue? Why does the dataset has that issue? Suggest a method to alleviate the issue.

Solution.
The problem is there are a big number of dimensions, because the number of documents is big, and each document should have many words. One possible solution is using PCA to preprocess the data. Someone suggests some other methods to select the best features. They are accpetable if the explanations are reasonable.

3. Data cube [10'].

(a) [6'] Consider a base cuboid of 5 dimensions that contains only four base cells:

$$(a_1, a_2, c_3, c_4, c_5)$$
$$(a_1, b_2, c_3, c_4, c_5)$$
$$(b_1, a_2, c_3, c_4, c_5)$$
$$(b_1, b_2, c_3, c_4, c_5)$$

where $a_i \neq b_i$, $\forall i = 1, 2$. There is **no** dimension with concept hierarchy. The measure of the cube is count. The count of each base cell is 1.

(i) [2'] How many cuboids are there in the full data cube?

Solution.
2^5.

(ii) [4'] List at least four **nonempty aggregated** (i.e., count ≥ 1 and non-base) **closed** cells in the full cube.

Solution.
$(a_1, *, c_3, c_4, c_5) : 2, (*, a_2, c_3, c_4, c_5) : 2, (*, b_2, c_3, c_4, c_5) : 2, (b_1, *, c_3, c_4, c_5) : 2, (*, *, c_3, c_4, c_5) : 4$

(b) [4'] Suppose that a data array has 3 dimensions A, B , C with the following sizes: $|A| = 400$, $|B| = 100$ and $|C| = 80$. The 3-D data array is divided into small chunks. Dimension A is divided into 4 equally sized partitions. Dimensions B and C are divided into 2 equally sized partitions respectively. Thus, there are totally 16 3-D chunks. The sizes of each 3-D chunk on dimensions A, B, and C are 100, 50, and 40 respectively. See Figure 1.

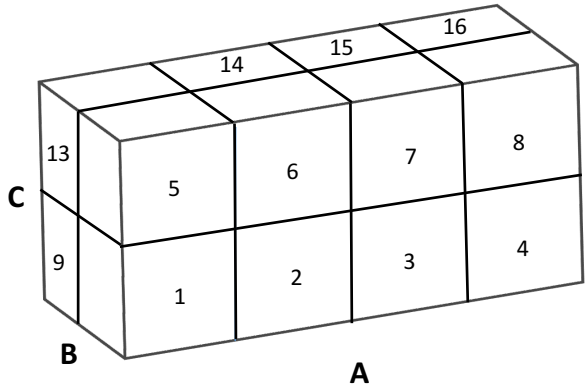

Figure 1: A 3-D data array with dimensions A, B and C. This data array is divided into 16 smaller chunks.

Now we want to use **Multiway Array Aggregation Computation** to materialize the 2-D cuboids AB, AC and BC. If we scan the chunks in the order of $1, 2, 3, 4, 5...., 15, 16$ when materializing the 2-D cuboids AB, AC and BC, to avoid reading a 3-D chunk into memory repeatedly, what is the minimum memory requirement for holding all the related 2-D planes? Show your result with important intermediate steps in calculation. (*Hint: When calculating the minimum memory requirement, you are* ***not*** *required to consider the memory cost for the* ***3-D chunk*** *which is read into the memory for scanning.*)

Solution.
If we scan the cube base on the order of $1, 2, 3...16$, we have: 400×80 (for the whole AC plane) + 400×50 (for one row of the AB plane) $+ 50 \times 40$ (for one BC plane chunk) $= 54,000$ memory units. (Do not consider the 3-D chunk)

4. Frequent pattern mining [10'].
Based on the tiny database of 5 transactions in Table 1, use the Apriori algorithm to find the frequent patterns and then find closed patterns, maximal patterns and association rules based on the frequent patterns you find with *relative min_sup* $= 0.6$, *min_conf* $= 0.9$.

Trans.	Items
1	abcde
2	bde
3	abef
4	bcde
5	ce

Table 1: A tiny transaction database

(a) [6'] Use Apriori to compute and list all frequent itemsets (including 1-itemsets, 2-itemsets and so on, you do not need to show the steps). Identify the maximal patterns within the frequent patterns you have found.

Solution.
$b, c, d, e, bd, be, ce, de, bde$; ce, bde.
1 point is deducted if one item is missing or wrong.

(b) [2'] How many times of database scanning in total does Apriori require to find all those frequent patterns? Will FP-Growth require fewer times of scanning in this case?

Solution.
3; yes.
FP-Growth requires only 2 scans.

(c) [2'] For mining maximal patterns, do you think the MaxMiner algorithm is more efficient than mining the frequent patterns first using basic Apriori and then compute maximal patterns based on the mined frequent patterns? Why?

Solution.
Yes; as long as something related to extra pruning is mentioned, you will get full scores.
If the reasons are wrong, 2 scores are deducted. Common wrong reasons include less db-scanning (how?) and utilizing fp-tree (that is closet).

5. Advanced pattern mining [25'].

In natural language processing, it is important to identify "phrases" from text. By considering phrases as word sequences of fixed order that are frequent in the corpus, one can apply the sequential pattern mining algorithm Prefix-Span to solve the problem.

ID	Text
1	Clustering and classification are important problems in machine learning.
2	There are many machine learning algorithms for classification and clustering problems.
3	Classification problems require training data.
4	Most clustering problems require user-specified group number.
5	*SVM*, *LogisticRegression*, and *NaiveBayes* are machine learning algorithms for classification problems.
6	*k-means*, *AGNES* and *DBSCAN* are clustering algorithms.
7	Dimension reduction methods such as *PCA* are also learning algorithms for clustering problems.

Table 2: Dataset for phrase mining

(a) [5'] Table 2 records 7 raw text sentences. As the first step, please remove less important words (*stop words*, including "are", "in", "for"), and convert all words to lower case except proper nouns (algorithm names). Then, build the sequence database. Specifically, process conjunction joiners ("and") by grouping joined words into itemsets. That is, for example, make "A, B, and C" into *(A, B, C)* in sequence transaction.

(b) [6'] Set $min\_sup = 3$, scan the database once, and generate length-1 frequent prefix list P_1 and its projected database.

(c) [7'] Again, compute length-2 frequent prefix list P_2 and its projected database. Is there any frequent prefix of length more than 2? Answer yes or no. If yes, list all frequent prefixes that have length more than 2.

(d) [7'] Some of the phrases are less interpretable than others because often additional words are filled in between to form meaningful semantics. Give such an example from your result above. Also, show one usage of the phrase in the dataset (phrase with necessary words filled in). For larger volume of text this kind of human inspection is not practical, what suggestions can you think of to remedy the problem? Specifically, we want to 1) identify phrases of this type. 2) find their typical usages. (*Hint:* the solution can be a post-processing step of sequence mining after above computation.)

Solution.

ID	Sequence
1	(clustering, classification), important, problems, machine, learning
2	there, many, machine, learning, algorithms, (classification, clustering), problems
3	classification, problems, require, training, data
4	most, clustering, problems, require, user-specified, group, number
5	(SVM, LR, NB), machine, learning, algorithms, classification, problems
6	(kNN, AGNES, DBSCAN), clustering algorithms
7	dimension, reduction, methods, such, as, PCA, also, learning, algorithms, clustering, problems

Prefix	Projected Database
clustering	(_, classification), important, problems, machine, learning (classification, _), problems problems, require, user-specified, group, number algorithms problems
classification	(clustering, _), important, problems, machine, learning (_, clustering), problems problems, require, training, data problems
problems	machine, learning NULL require, training, data require, user-specified, group, number NULL NULL
machine	learning learning, algorithms, (classification, clustering), problems learning, algorithms, classification, problems
learning	NULL algorithms, (classification, clustering), problems algorithms, classification, problems algorithms, clustering, problems
algorithms	(classification, clustering), problems classification, problems NULL clustering, problems

Prefix	Projected Database
clustering problems	machine, learning NULL require, user-specified, group, number NULL
classification problems	machine, learning NULL require, training, data NULL
machine learning	NULL algorithms, (classification, clustering), problems algorithms, classification, problems
learning algorithms	(classification, clustering), problems classification, problems clustering, problems
learning problems	NULL NULL NULL
algorithms problems	NULL NULL NULL
learning algorithms problems	NULL NULL NULL

After generating phrases from prefix-span, find their matches in text and compute if it is more likely to be used with additional words filled in. For those phrases and matched sentences (better to be chunked to their boundary), run prefix-span again with a smaller min_sup value.

6. Classification [40'].

(a) [10'] With the training data given in Table 3a, construct a decision tree using the *Gini index* measure for attribute selection. Then, evaluate your decision tree in terms of *precision* and *recall* for the class *Yes* with the testing data given in Table 3b. (*Hint*: The Gini index of a set of tuples D is defined by $Gini(D) = 1 - \sum_k p_k^2$, where p_k is the proportion of tuples with class k in D; the Gini index of a collection of sets of tuples is the weighted sum (by the relative size of each set) of the Gini index of each set.)

Solution.

See Figure 2. $precision = 0.5, recall = 0.5$.

(b) [7'] In order to make predictions using the *Naive Bayes* classifier, using the training data given in Table 3a, calculate the *prior probability* of each target class. Then calculate the *conditional probabilities* of the attribute values given the class. Finally, using the above calculated probabilities, predict the class for tuple $(X_1 = T, X_2 = T, X_3 = F)$. (*Note*: There is *no* need to calculate the exact probability for prediction, and do *not* use Laplacian correction).

X_1	X_2	X_3	Class
T	T	F	*Yes*
T	T	F	*Yes*
T	T	T	*Yes*
T	F	T	*No*
T	F	T	*No*
F	T	F	*No*
F	T	T	*No*
F	T	F	*No*
F	F	T	*No*
F	T	F	*No*

(a) Training data

X_1	X_2	X_3	Class
F	T	F	*No*
F	F	F	*No*
T	T	T	*Yes*
T	T	F	*No*
T	F	T	*Yes*

(b) Testing data

Table 3: Dataset for decision tree

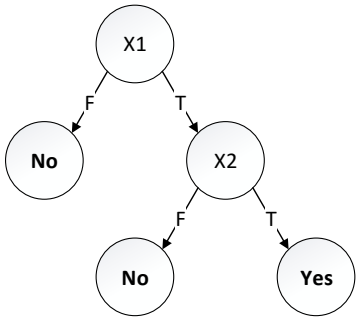

Figure 2: Solution: decision tree

Solution.
Predicted class is *Yes*. Calculations are as follows.

$$\frac{P(Yes|X)}{P(No|X)} = \frac{P(Yes)P(X|Yes)}{P(No)P(X|No)} = \frac{0.3 \times 1 \times 1 \times 2/3}{0.7 \times 2/7 \times 4/7 \times 3/7} > 1$$

(c) [6'] Design *either* a perceptron *or* a multilayer feedforwoard neural network to implement the EQUAL operation. That is, the classifier takes x_1 and x_2, both in $\{0, 1\}$, as input, and computes $h(x_1, x_2)$ as output, where $h(x_1, x_2) = 1$ if $x_1 = x_2$ and 0 otherwise. If you choose to design a neural network, assume the *threshold* function is used as the activation function for all units in the hidden and output layers.

Solution.
See Figure 3 for a possible solution.

(d) [7'] For each of the datasets D_1, D_2, and D_3 given in Figure 4, find the decision boundary of the 1NN (1-nearest neighbor) classifier, and draw it directly on the given figure. Then, for dataset D_4 in Figure 4, draw the decision boundary of the 3NN classifier. (*Hint*: Recall

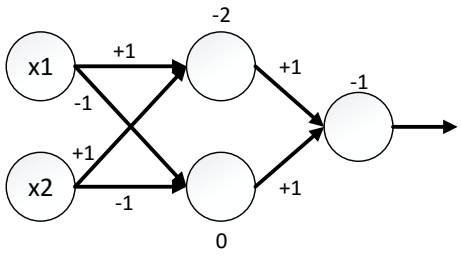

Figure 3: Solution: neural network

that the decision boundary of a binary classifier partitions the feature space into two disjoint areas, each area for one class.)

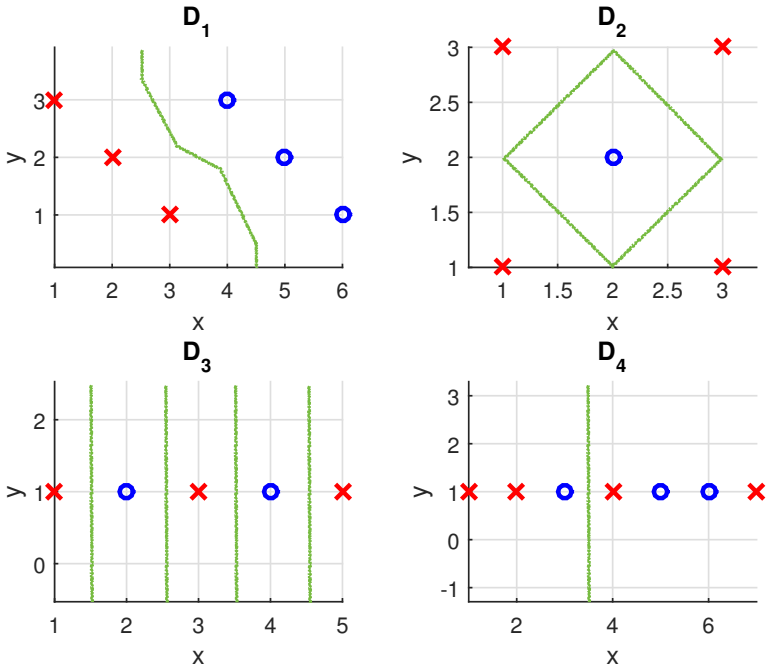

Figure 4: Data for kNN

Solution.
See Figure 4.

(e) [10'] For the set of data points in 1-D space, as shown in Table 4, use the AdaBoost algorithm to find an ensemble classifier h_e in which each weak classifier has the form of *either* "$h_i(x) = \mathbb{1}(x > a_i)$" *or* "$h_i(x) = \mathbb{1}(x < a_i)$", where $\mathbb{1}(cond)$ is the indicator function that equals 1 if the condition *cond* is satisfied or 0 otherwise, and $a_i \in \{-1.5, -0.5, 0.5, 1.5\}$. Also assume that the AdaBoost algorithm starts with uniform weight distribution, and the first weak classifier h_1 has *already* been given by $h_1(x) = \mathbb{1}(x < -0.5)$.

X	Class
-1	1
0	0
1	1

Table 4: Dataset for AdaBoost

Specifically, is it possible to find an ensemble classifier h_e that classifies all training data points correctly? If yes, find such a h_e and show your steps, i.e., show the data weight distribution and the weighted error in each iteration (pick an arbitrary one if the solution is not unique). Otherwise, please explain.

Solution.

There exists such an emsemble classifier that classifier all points correctly.
$h_1(x) = \mathbb{1}(x < -0.5)$, $= \mathbb{1}(x < 0.5)$, $h_3 = \mathbb{1}(x < 1.5)$.
Iteration-1: $w_1 = (1/3, 1/3, 1/3)$, $\epsilon_1 = 1/3$, $\alpha_1 = \ln 2$.
Iteration-2: $w_2 = (1/4, 1/4, 1/2)$, $\epsilon_2 = 1/4$, $\alpha_2 = \ln 3$.
Iteration-3: $w_3 = (1/2, 1/6, 1/3)$, $\epsilon_3 = 1/6$, $\alpha_3 = \ln 5$.

7. Clustering [22']. Consider the 9 data points given in Figure 5. The ground truth (correct clustering output) is also provided.

(a) [5'] Using the data in Figure 5, perform DBSCAN, a density-based clustering algorithm, with parameters $MinPts = 4$ and $\epsilon = 1.7$. You can visit the points in the dataset in any order. Show how clusters grow, and explicitly give the final clustering result.

Solution.

N : the data points we are going to process. N does not include itself. However, as mentioned in the slides and Piazza several times, when comparing the number of neighbors with MinPts, we must count itself as a neighbor. If excluding itself as a neighbor, the problem becomes trivial when all points are classified as noise. Students will get 3 points if doing that.

C1 = {P5}	N = {P2, P3, P6}
C1 = {P5, P2}	N = {P3, P6}
C1 = {P5, P2, P3}	N = {P6}
C1 = {P5, P2, P3, P6}	N = {P8, P9}
C1 = {P5, P2, P3, P6, P8}	N = {P9}
C1 = {P5, P2, P3, P6, P8, P9}	N = {}

All other points do not have enough neighbors, so they are noise. → final clusters:

- C1: P2, P3, P5, P6, P8, P9
- C2 (Noise): P1, P4, P7

Note: As shown in the sample questions, students might save some time by writing only the new member of the cluster in each step. For example, in line 3, instead of writing C1 = {P5, P2, P3}, students might write C1 = {..., P3}.

Point	x	y	Cluster
P1	1	1	C3
P2	1	3	C1
P3	1	5	C1
P4	0	4	C1
P5	2	4	C1
P6	3	4	C2
P7	5	4	C2
P8	4	3	C2
P9	4	5	C2

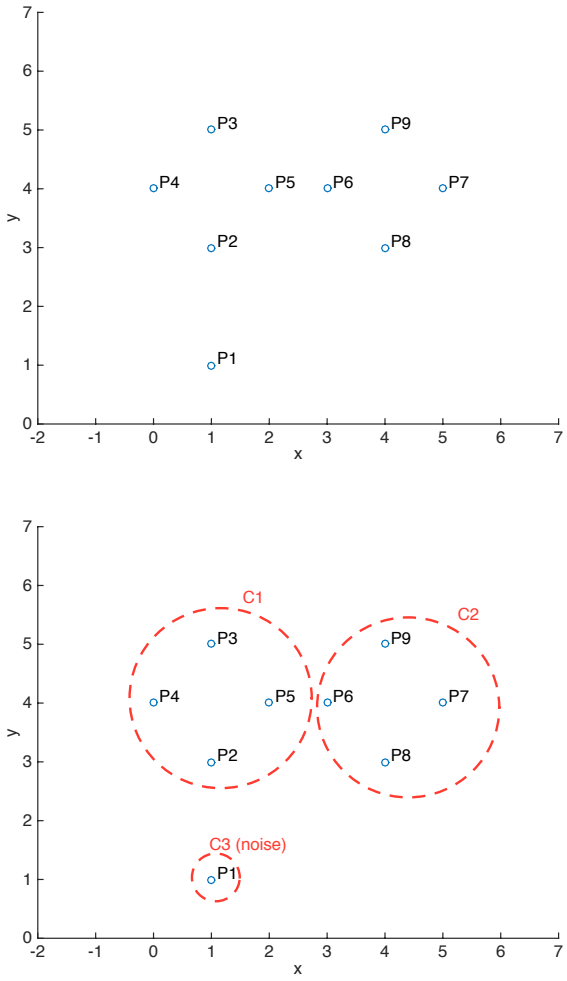

Figure 5: Data for clustering and ground truth

(b) [5'] Perform AGNES, a hierarchical clustering algorithm, using the data in Figure 5, and assume that the single-link method and Euclidean distance as the dissimilarity measure are used. (*Note*: You only need to draw the dendrogram.)

Solution.

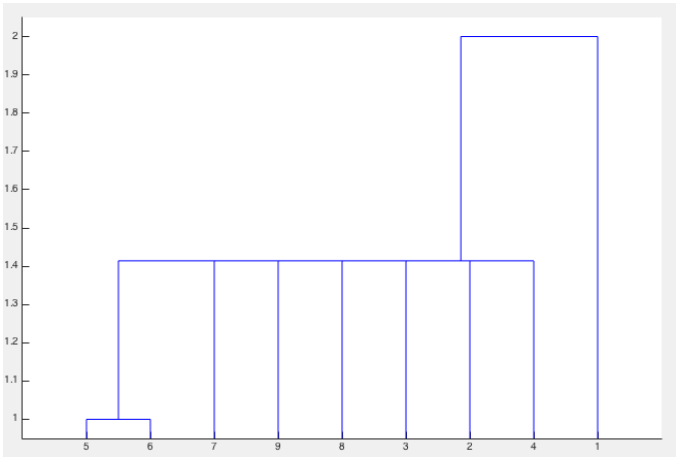

Figure 6: Dendrogram

(c) [3'] Using the same data in Figure 5, show the clustering result of K-Means for $K = 2$ by annotating Figure 7, with P3 and P9 as the initial centroids and using Euclidean distances. (*Note*: There is *no* need to show your computation or to illustrate the means.)

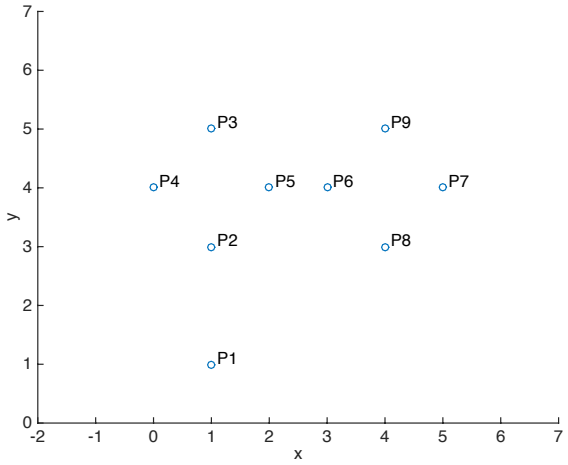

Figure 7: Annotate the output of K-Means in question (c) here

Solution.
As shown in Figure 8

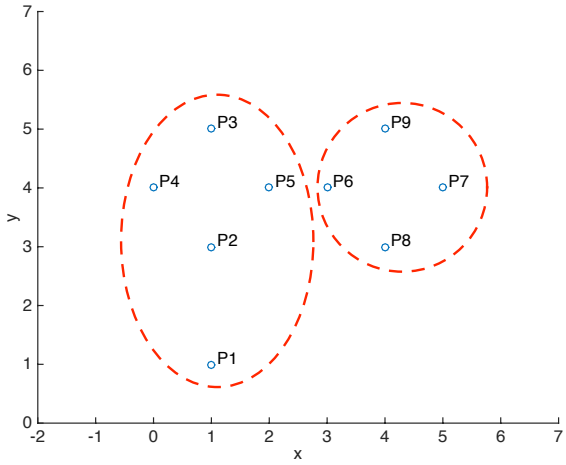

Figure 8: Solution for output of K-Means

(d) [3'] Do the same as in question (c), but with *different* initial centroids: P1 and P5. Show the clustering result of K-Means by annotating Figure 9.

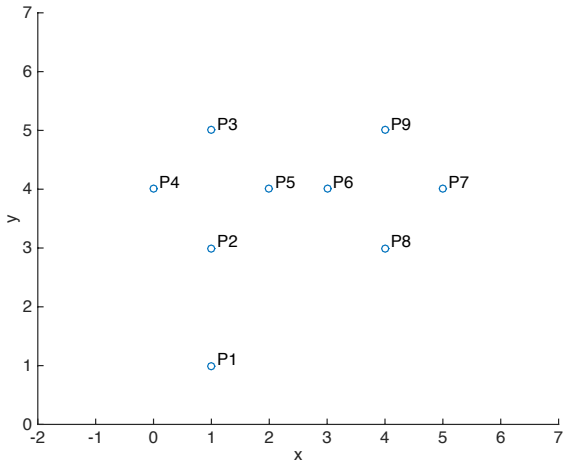

Figure 9: Annotate the output of K-Means in question (d) here

Solution.
As shown in Figure 10

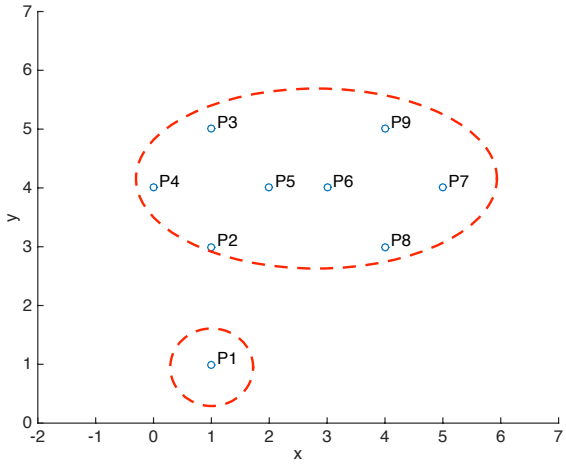

Figure 10: Solution for output of K-Means

(e) [2'] Based on the results of questions (c) and (d), you can see that the output of K-Means depends on the initial centroids. Suggest a general method (independent of specific datasets) to choose the best initial centroids for K-Means. Explain why your method is good.

Solution.
The most intuitive method is to run K-Means several times, and choose the best one according to some quality measure. Some students suggest to choose points that far away from each other. It may be not a good idea as you will select a lot of outliers, and the results could be very bad. Anyway, you still get full points for that answer (because it usually helps K-Means) as long as your explanation is clear enough.

(f) [4'] Suppose a particular algorithm outputs a clustering as shown in Figure 11. Based on the given ground truth, what are the B-Cubed precision and recall of the clustering? Show your calculations.

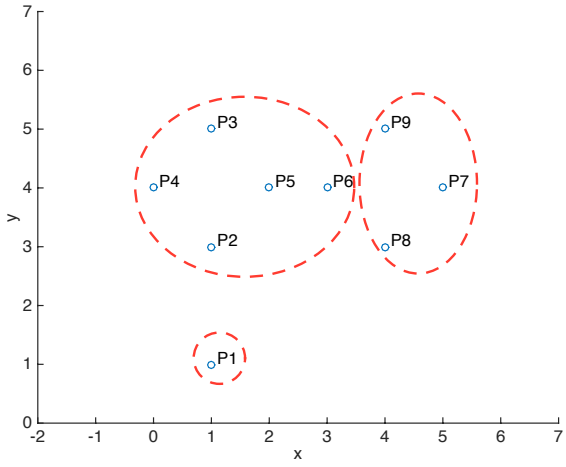

Figure 11: Output of a particular clustering algorithm

Solution.

Point i	1	2	3	4	5	6	7	8	9
P_i	1/1	4/5	4/5	4/5	4/5	1/5	1/1	1/1	1/1
R_i	1/1	4/4	4/4	4/4	4/4	1/4	3/4	3/4	3/4

$$Precision = (1/9) * \sum_{i=1}^{9} P_i = 37/45$$

$$Recall = (1/9) * \sum_{i=1}^{9} R_i = 5/6$$

8. Opinion [3'].

(a) I ☐ like ☐ dislike the exams in this style.

(b) In general, the exam questions are ☐ too hard ☐ too easy ☐ just right.

(c) I ☐ have plenty of time ☐ have just enough time ☐ do not have enough time to finish the exam questions.