COS 435, Spring 2011 - Problem Set 5

Due at 1:30pm, Wednesday, April 6, 2011

Collaboration and Reference Policy

You may discuss the general methods of solving the problems with other students in the class. However, each student must work out the details and write up his or her own solution to each problem independently.

Some problems have been used in previous offerings of COS 435. You are NOT allowed to use any solutions posted for previous offerings of COS 435 or any solutions produced by anyone else for the assigned problems. You may use other reference materials; you must give citations to all reference materials that you use.

Lateness Policy

A late penalty will be applied, unless there are extraordinary circumstances and/or prior arrangements:

- No penalty if in Prof. LaPaugh's office or inbox by 5pm Wednesday (4/6/11).
- Penalized 10% of the earned score if submitted by 11:59 pm Wednesday (4/6/11).
- Penalized 25% of the earned score if submitted by 5pm Friday (4/8/11).
- Penalized 50% if submitted later than 5pm Friday (4/8/11).

Problem 1 (collaborative filtering):

For this problem you need the equations for ***user-based*** "memory-based" collaborative filtering . The equations were given in class, and the slides are posted under March 23.

The table below gives the ratings of six items by three users. The rating scale is 1 (poorest) through 5 (best) and "X" indicates not rated.

	User 1	**User 2**	**User 3**
Item 1	**3**	**X**	**2**
Item 2	**X**	**1**	**2**
Item 3	**4**	**2**	**X**
Item 4	**5**	**1**	**X**
Item 5	**3**	**5**	**5**
Item 6	**5**	**1**	**X**

Part a.
What is the similarity of User 1 and User 3? of User 2 and User 3?

Part b.
Compute the predicted ratings by User 3 for items 3, 4 and 6. For each item, calculate the prediction using the ratings of both User 1 and User 2.

Part c.
In the equations used in Part a and Part b, the average rating by a user is subtracted from that user's ratings to account for shifts in the way different users rate (think "grade inflation"). Different users also use different sub-ranges of values for their ratings. For example User 2 has used the full range 1 through 5 in her ratings, while User 1 has only used the range 3 through 5 (think "grade compression"). How might the collaborative filtering predictions take into account the differences in the range of values users use? You need not work out a full set of equations -- just give some thoughts.

Problem 2 (K-means clustering):

Consider the following set of 2-dimensional points in the plane as shown in Figure 1 below. Let p(x) and p(y) denote the x and y coordinates of point p. ***Measure the distance between two points using the L1 (also known as Manhattan) metric:***

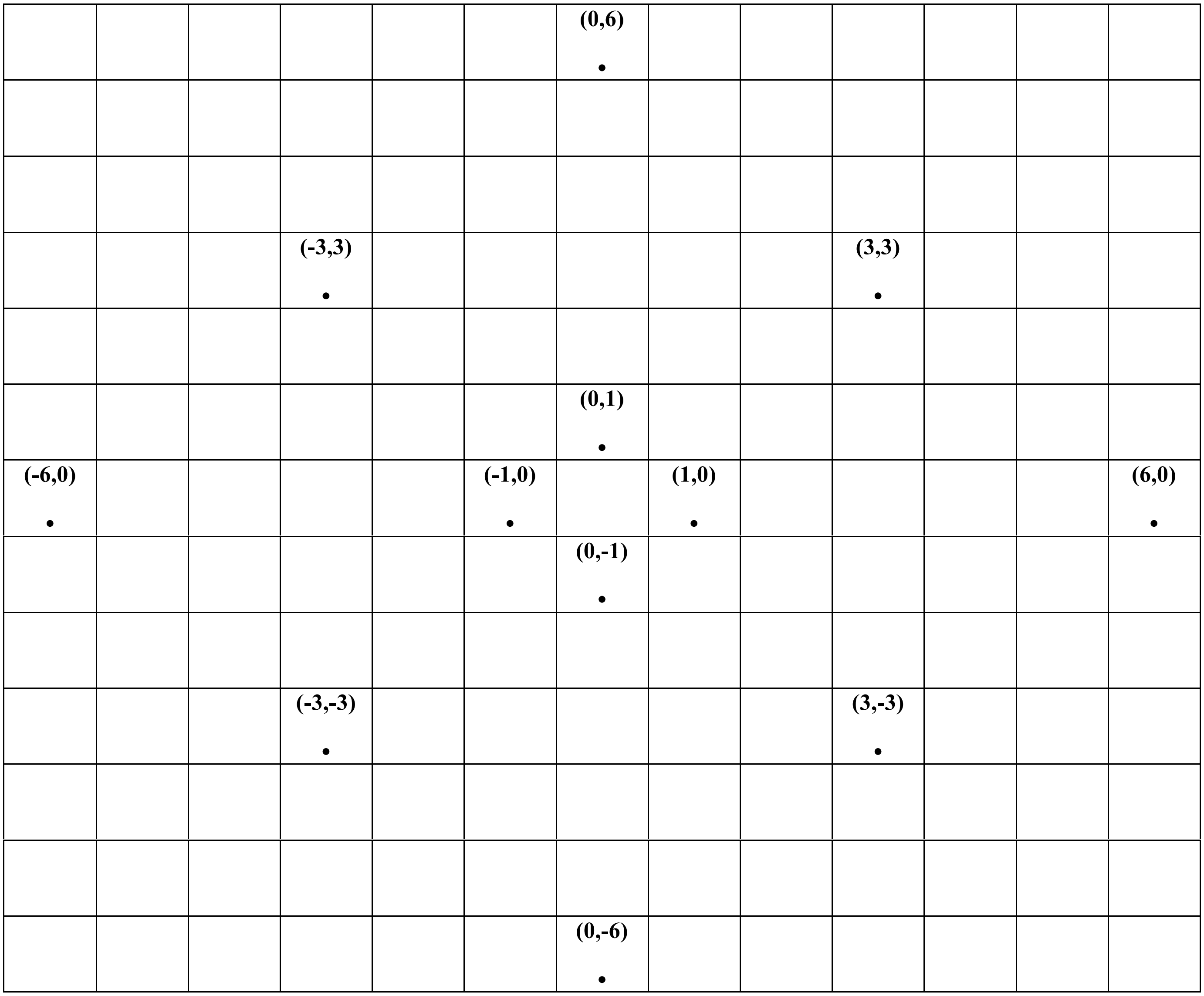
$$\text{Dist\_L1}(p,q) = |p(x) - q(x)| + |p(y)-q(y)|$$

Similarity between two points is determined by their distance: more distant is less similar.

Part A: Do ***three iterations*** of the k-means clustering algorithm for k=2 given initial centroids (0,1) and (0,6). Remember to use the L1 metric for distances. Give the clusters and new centroids obtained after each iteration. Has the calculation converged? If not, can you tell what it will converge to?

Part B: Is there any initial pair of centroids that would result in k-means yielding the ***final*** 2-clustering in which one cluster contains all points within L1-distance one from the origin (0,0) and the other cluster contains all points within L1-distance six from the origin? Justify your answer.

Figure 1



List of points:
(1,0), (0,-1), (-1,0), (0,1), (6,0), (3,-3), (0,-6), (-3,-3), (-6,0), (-3,3), (0,6), (3,3)

Problem 3 (agglomerative clustering):

The algorithm for hierarchical agglomerative clustering giving in Figure 17.8 of *Introduction to Information Retrieval* uses one priority queue for each cluster to efficiently find the most similar pair of clusters to merge. The priority queues are updated for each merge step by deleting the two clusters that have been merged and inserting the new combined cluster. Consider ***breaking ties*** when selecting the pair of clusters to merge by choosing the pair that results in the smallest combined cluster. What modifications would be needed in the algorithm and data structures of Figure 17.8? Would the running time be affected? Explain.