Data Mining I: Example Exam Questions

Example 1:

Question:

Briefly explain the K-Means clustering algorithm and discuss the problem that can arise if numeric attributes have different value ranges within the dataset as well as a way to deal with this problem.

Answer:

k-Means is a partitional clustering algorithm which assigns the examples of a dataset to k (manual input) different clusters based on a given distance function. The algorithm works as follows:

1. Select k points as initial centroids
2. Assign each data point to the closest centroid using the distance function
3. Re-compute the centroids of each cluster
4. Return to Step 2 until the centroids do not change any more

Attributes with larger value ranges (e.g. age 1-99 has a larger range than grades 1-5) will dominate the distance calculation and screw up the results. This can be overcome by normalizing the attribute value ranges to the same range (e.g. values between 0 and 1) before the distance is calculated.

Example 2:

Question:

Describe the characteristics of a Rule-Based Classifier generated from an unpruned decision tree. Briefly explain the problems that can arise from simplifying of such a classifier in respect to the just named characteristics and explain how these problems can be addressed.

Answer:

A rule-based classifier that is derived from a decision tree is (1) mutual exclusive, meaning every record is covered by at most one rule and (2) exhaustive, meaning every record is covered by at least one rule. Simplification of the derived rules reduces the complexity of rules, which might result in (a) a single record to trigger more than one rule (not mutual exclusive any more) and (b) a record might not trigger any rule (no longer exhaustive). Ordering the rules can overcome (a) and including a default rule can overcome (b).

Example 3:

Question:

Given the following three documents (d1, d2, and d3) explain the effect of stop word removal on calculating the similarities between the three documents. In order to do so, create for all three documents the corresponding binary term occurrence word vector and calculate the similarities between the documents using a similarity measure of your choice. (Hint: The choice of the similarity function has an influence on the complexity of the computation you have to perform.)

d1: berlin is a big city
d2: garda lake is a lake
d3: rome was the largest city

Note: Stop words are is, was, a and the in this example.

Answer:

We use the Jaccard Coefficient as distance measure.

Binary term vectors with stop words:

Doc	berlin	is	a	big	city	garda	lake	rome	was	the	largest
d1	1	1	1	1	1	0	0	0	0	0	0
d2	0	1	1	0	0	1	1	0	0	0	0
d3	0	0	0	0	1	0	0	1	1	1	1

d(1,2) = 2 / (3 + 2 + 2) = 2 / 7
d(1,3) = 1 / (4 + 1 +4) = 1 / 9
d(2,3) = 0 / (4 + 5 + 0) = 0

Based on these values d1 and d2 are most similar, which is not correct as the d1 talks about a city and d2 about a lake.

Binary term vectors without stop words:

Doc	berlin	big	city	garda	lake	rome	largest
d1	1	1	1	0	0	0	0
d2	0	0	0	1	1	0	0
d3	0	0	1	0	0	1	1

d(1,2) = 0 / (3 + 2 + 0) = 0
d(1,3) = 1 / (2 + 2 + 1) = 1 / 5
d(2,3) = 0 / (2 + 3) = 0

Based on these vectors d1 and d3 are most similar, which is a better result as with stop words as both talk about cities.