University of Ioannina,
Computer Science Department

Data Mining – CS 059

Assignment 2

The deadline for the Assignment is **Monday, May 14**. For late submissions the late policy on the page of the course will be applied. You can turn in code and files using the command: turnin assignment2_csXXXX@ple059 <your files>. Give self-explanatory names to your files. The date of the last turn-in determines the date of the turn-in. You can also submit your assignment via email.

Question 1 (15 units)

In the course textbook (Introduction to Data Mining by Tan, Steinbach, Kumar) except for the clustering algorithms that we described in class, there are also the algorithms CLARANS, BIRCH, ROCK, CHAMELEON, and DENCLUE. In the free online textbook Mining Massive Datasets by Rajaraman and Ullman the clustering algorithm CURE is described (in English). Select one of these algorithms, and describe the main idea in your own words in 2-3 paragraphs. You can read the corresponding paper if you need additional details.

Question 2 (20 units)

We have a collection D of N documents, where each document is “bag of words” drawn from a vocabulary W of m words. Document d_i can be represented as an m-dimensional vector where d_{ij} is the number of times that word w_j appears in document d_i. Consider a clustering $C = \{c_1 \dots, c_K\}$ of the documents in D where $1 \leq K \leq N$. When $K = N$ each document is in a cluster by itself (singleton clusters), while when $K = 1$, all documents are together in a single cluster. A cluster c_i contains the concatenation of all words of all the documents in the cluster. Therefore, it can also be represented as an m-dimensional vector over the set W of words: the sum of the vectors of all the documents in the cluster. For cluster c_i we use n_{ij} to denote the number of times that word w_j appears in the cluster. We use n_i to denote the number of words in cluster c_i, and n to denote the number of all words in all clusters. If we normalize the vector for cluster c_i we obtain a probability distribution $P(W|c_i)$, where $p_{ij} = p(w_j|c_i) = \frac{n_{ij}}{n_i}$ is the conditional probability of word w_j in cluster c_i. This is the probability that if we randomly select a word from the clusterc_i this word will be w_j. We use P_i to denote the distribution $P(W|c_i)$. We also define $p(c_i) = \frac{n_i}{n}$ to denote the probability of cluster c_i. This is the probability that if we pick a randomly a word among all the words in all documents, this will be from cluster c_i.

We can now define the conditional entropy of the words W given the clustering C. We have

$$H(W|C) = \sum_{c_i \in C} p(c_i) H(W|c_i) = -\sum_{c_i \in C} p(c_i) \sum_{w_j \in W} p\left(w_j|c_i\right) \log p(w_j|c_i)$$

The entropy of the words in cluster c_i is given by $H(W|c_i) = H(P_i)$, the entropy of the word distribution within the cluster. We call $H(P_i)$ the *entropy of the cluster* c_i, and $H(W|C)$, *the entropy of the clustering* C.

Consider now two clusters (e.g., cluster c_1 and cluster c_2), and suppose we merge them to create a new cluster c_*. We have that $p(c_*) = p(c_1) + p(c_2)$

1. Show that

$$P_* = P(W|c_*) = \frac{p(c_1)}{p(c_*)} P_1 + \frac{p(c_2)}{p(c_*)} P_2$$

We define a generalized version of Jenshen-Shannon divergence between distributions P_1 and P_2 as

$$D_{JS}(P_1, P_2) = \frac{p(c_1)}{p(c_*)} D_{KL}(P_1||P_*) + \frac{p(c_2)}{p(c_*)} D_{KL}(P_2||P_*)$$

$D_{KL}(P_1||P_*)$ is the KL-divergence between distributions P_1 and P_*. Now, let *C* be the clustering that has clusters c_1 and c_2, and *C** the clustering after merging the clusters c_1 and c_2 into cluster c_*.

2. Show that the increase in entropy is

$$H(C^*) - H(C) = p(c_*)\, D_{JS}(P_1, P_2)$$

   There are clustering approaches that aim at minimizing the entropy of a clustering and use the increase in entropy as a distance metric between two clusters. We can then apply an agglomerative algorithm, or a *k*-means like algorithm

Consider a toy example were the vocabulary consists of just two words W = {“sports”, “politics”}, and we have three documents d_1, d_2, d_3. Document $d_1$contains two occurrences of word “sports”, document d_2 contains three occurrences of word “sports”, and document $d_3$contains five occurrences of word “politics”.

3. What is the entropy of the documents, and what does this mean? What is the distance of the two closest documents and what does this mean? What is the entropy of a cluster that contains all three documents?

Question 3 (15 units)

The goal of this question is to experiments with clustering algorithms and the different validation criteria. You will use the “Iris” dataset from the UCI data repository (http://archive.ics.uci.edu/ml/datasets/Iris), which contains information for 200 different Iris flowers. The flowers are characterized by 4 different features and classified into 3 different classes. You will run the k-means algorithm and the four variations of the agglomerative hierarchical clustering algorithm (min, max, avg, ward). For your experiments you can use the implementations of the algorithms in WEKA, or MATLAB (information about the use of MATLAB for clustering can be found in the link on the class web page). You can find the ARFF file for the Iris dataset on the course web page, or download the data from the UCI repository.

To cluster the flowers you will use only the four features and not the class label. The class labels will be used as the ground truth clustering for the clustering validation. Compute the Precision, Recall, and F-Measure for each cluster and class pair, and Entropy and Purity for each cluster. For k-means take the average over 10 different runs. Submit the tables with the results, and your observations about the quality of the clusterings of the different algorithms.

Question 4 (50 units)

In class we described the Minimum Description Length principle and its application to the problem of co-clustering. In this question you will apply MDL to the problem of segmentation of a one-dimensional sequence. The input is a sequence of 0/1 numbers, and the goal is to segment the sequence into segments such that the total cost of encoding the sequence is minimized.

1. Define the encoding cost of the sequence. Clearly state what is the model cost and what is the data cost given the model. Hint: A segmentation into K segments is defined by K-1 breakpoints.
2. Give a heuristic algorithm for the problem of finding a segmentation that minimizes the encoding cost. Describe the algorithm using pseudocode.
3. The segmentation problem can be solved optimally in polynomial time using dynamic programming. Define the dynamic programming array and recurrence for the problem.
4. Implement an algorithm that solves the problem. Test your algorithm on a sequence of 1000 values that you will create as follows:
    a. Select two breakpoints randomly.
    b. Create random 0/1 for each segment with different probability: in the first segment the probability of value 1 is .7, in the second 0.3, and in the third 0.9
    c. Submit a file with the true breakpoints and one with the ones that your algorithm finds on 3 different random inputs.