CS6220/DS5230 Unsupervised Data Mining
HW1: Data Features, Similarity, KNN

General Instructions

- **Due Date**: Refer to the syllabus for the due date.
- **Notations**: Use the notations adopted in class, even if the problem is stated differently in the book.
- **Response Length**: Keep answers concise. Aim for one or two pages of typed text per problem.
- **Focus**: Emphasize good ideas and explanations over exact details.

Datasets

1. **Kosarak**: Click-stream data of a Hungarian online news portal
   `http://fimi.uantwerpen.be/data/kosarak.dat`
2. **Aminer**: Public citation dataset
   `https://lfs.aminer.cn/lab-datasets/citation/acm.v9.zip`
3. **20 NewsGroups**: News articles
   `http://qwone.com/~jason/20Newsgroups/`
4. **MNIST**: Digit images
   `http://yann.lecun.com/exdb/mnist/`

Problem 1: Aminer – Basic Dataset Analysis

Tasks

A. Compute the number of distinct authors, publication venues, publications, and citations/references.

B. Evaluate the accuracy of these numbers. Analyze the publication venue names associated with *Principles and Practice of Knowledge Discovery in Databases* and discuss your observations.

C. For each author, construct the list of publications. Plot a histogram of the number of publications per author (logarithmic scale on the y-axis).

D. Calculate the mean, standard deviation, Q1 (1st quartile), Q2 (median), and Q3 (3rd quartile) for the number of publications per author. Compare the median to the mean and explain differences.

E. Plot a histogram of the number of publications per venue and calculate the mean, standard deviation, median, Q1, and Q3. Identify the venue with the most publications.

F. Plot histograms for the number of references (publications cited by a publication) and citations (publications citing a publication). Identify the publication with the most references and citations and evaluate the results.

G. Calculate the "impact factor" for each venue as the total citations divided by the number of publications. Plot a histogram of impact factors.

H. Identify the venue with the highest impact factor. Assess whether this value is reasonable.

I. Repeat the impact factor calculation for venues with at least 10 publications. Compare histograms and analyze citation distributions for the venue with the highest impact factor.

J. Construct a list of publications by year. Plot the average number of references and citations per publication over time. Discuss observed trends.

Problem 2: Kosarak Association Rules

Tasks

A. Write a Python program to convert the dataset from itemset format to a sparse ARFF file.

B. Use your program to convert the `kosarak.dat` file to a sparse `kosarak.arff`. Measure and report the runtime.

C. Load the resulting file into Weka. Ensure it has 41,270 attributes and 990,002 instances. Measure and report the runtime.

D. Use Weka's FP-Growth implementation to find association rules with a minimum support count of 49,500 and confidence of at least 99%. Record the resulting two rules.

E. Run the algorithm five times, record runtimes, and calculate the average. Compare the runtime to the dataset conversion and loading times.

Problem 3: MNIST and 20NG Preprocessing

Tasks

Parsing: Write or use a library to parse the datasets.
Normalization:

- Determine and apply appropriate normalization for each dataset.
- Common methods: Shift-and-scale, zero mean/unit variance, term frequency (TF) weighting.
- Retain sparsity for text datasets.

Pairwise Similarities:

- Compute pairwise similarity or distance matrices for:
  - Euclidean distance (library and custom implementation).
  - Edit distance (for text) or cosine similarity (for vectors).
  - Optional: Jaccard similarity, Manhattan distance.

Problem 4: MNIST and 20NG – Train and Test KNN Classification

Tasks

1. Implement a custom K-nearest neighbor (KNN) classifier.
2. Partition datasets into 80% training, 10% testing, and 10% validation.
3. Train and test the KNN classifier for both datasets:
   - Report training and testing performance.
   - Optionally implement a scikit-learn compatible estimator class supporting `.fit()`, `.predict()`, and `.transform()` methods.

Resources:

- `https://scikit-learn.org/stable/developers/develop.html`
- `https://en.wikipedia.org/wiki/Category:Similarity_and_distance_measures`