Uppsala University
Department of Information Technology
Kjell Orsborn

DATA MINING I - 1DL360

Assignment 1 - Classification using kNN

1 Classification using a k-Nearest Neighbours Algorithm

This assignment is taken from the field of forensic science. The goal is to classify a number of glass samples (building window glass, vehicle window glass, vehicle headlight glass, etc) using a k-Nearest Neighbors (kNN) algorithm. You will use a kNN-algorithm that is implemented in the AmosMiner application using the Amos II database management system. This means that you have to install Amos II and AmosMiner according to the instructions on the assignments home page.

The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence – if it is correctly identified.

The data in this assignment is taken from [ES87], from USA Forensic Science Service, and describes 6 types of glass defined in terms of their oxide content (i.e. Na, Fe, K, etc).

Attribute information:

1. RI: refractive index
2. Na: Sodium
3. Mg: Magnesium
4. Al: Aluminum
5. Si: Silicon

6. K: Potassium
7. Ca: Calcium
8. Ba: Barium
9. Fe: Iron
10. Type of glass: (class attribute)
    (a) 1: building_windows_float_processed
    (b) 2: building_windows_non_float_processed
    (c) 3: vehicle_windows_float_processed
    (d) 4: headlamps
    (e) 5: containers
    (f) 6: tableware

Attributes 2-9 are expressed as unit measurement, i.e. weight percent in corresponding oxide.

2 Preparation

You should prepare for this assignment by reading about the kNN-algorithm presented in Chapter 4 in Tan et al. [Tan06].

It is also advisible to read Chapters 2.3 and 2.4 that include sections on *sampling*, *normalization* and *proximity measures*.

You can find and download the AmosMiner and Amos II system from the assignments home page and you will also get a chance to familiarize yourself with those systems in two introductory tutorials.

The computer labs include a limited number of computers and, to avoid overcrowded labs, each group should sign up for one lab time.

Lab lists will be posted on the board outside 1346.

3 Assignment

You will find data for this assignment in data files available on the lab course home page. In the data, there are no missing values and you may assume that there is no noise

Data resides in the following files:

`glassdata.nt`: This file contain 163 data points of 10 dimensions. Each data point is one observation. Dimension 1...9 corresponds to a certain measurement of some feature (refractive index, magnesium concentration, etc.) for a data point. The 10th dimension is the class of each data point represented as an integer between 1 and 6.

`testdata.nt`: This file consists of 30 observations of unclassified glass samples, i.e. it includes values for the first 9 dimensions but the class value is unknown.

In the assignment, you should classify the 30 unclassified glass samples using the kNN algorithm in AmosMiner.

In the download of the AmosMiner system, you can also find the script file for the this assignment named 'a1.osql'. This file include some predefined functions that should be used for your kNN analysis.

You should do the following:

1. START-UP

   Once you have started the AmosMiner system (according to instructions on the assignments home page), you should load the script file for this assignment by:

   ```
   < 'a1.osql';
   ```

   When the script file has been loaded correctly, you have already been executing a kNN analysis that can be tracked by studying the content of the script file itself and the results that were generated during its execution. It is now your turn to modify these scripts by following the rest of these instructions.

2. LEAVE-ONE-OUT CROSS VALIDATION

You are supposed to repeat STEP 2, in the script file, with different values of k to find the best k.

Hint:

- It can be run automatically by implementing a loop.
- Amos II has a built-in function `iota(1, N)` which returns all incremental values from `1` to `N`.
- You should store value of k in the Amos II variable `:k`.

Questions:

- What is the best k?
- Will normalizing the data improve the classifier result?
- Can you explain how Leave-One-Out cross validation work?
- What can be the advantage or disadvantage with this approach?
- Can you think of an alternative method for partitioning your data into training data and test data?

3. NORMALIZATION

You should experiment with different normalization methods.

```
gaussian(mean + stdev)
minmax(min + max)
maxNorm(maxNorm, 0)
```

Hint: It is useful to use the built-in aggregate functions: mean, standard deviation and maximum and minimum object in a bag of objects respectively (see Amos II manual).

```
average(Bag of Number x) -> Real
stdev(Bag of Number x) -> Real
maxagg(Bag of Object x) -> Object
minagg(Bag of Object x) -> Object
```

This is an example of normalization on identified data using min-max normalization.

```
set :substract = aggv(values_of_known_data(), #'minagg');
set :divide = aggv(values_of_known_data(), #'maxagg')
                  - :substract;

normalize_iddata(#'known_data', :substract, :divide,
                 #'normalized_known_data');
```

Questions:

- Which normalization gives the best classifier accuracy (a number of points which have been classified correctly)?
- Why?

4. DISTANCE FUNCTION

   Try the Minkowski distance function (`#'minkowski'`) with some different r factors.

   Questions:

   - What r gives the best classifier accuracy?
   - Why?

5. DISTANCE WEIGHTED VOTING

   You should write a function to weight the impact of the nearest neighbour according to its distance to the query point.

   Hint:

   - Call your defined function in *k_votes* (file: knnclassifier.osql).
   - Since you have changed the system code, you have to try to reinstall (*install.cmd*) and start the system again (*amosMiner.cmd*).

   Question :

   - What is the new optimal value of k ?
   - Does distance weighted voting give you a better classification ?
   - Why ?

6. ATTRIBUTE WEIGHT (Optional)

   In kNN, all attributes of an object equally contribute to the distance between the object and another.

   However, in reality, some attributes might be less important than others in order to measure the difference (distance) betwen two objects. That means the less important attributes should have less influence on the distance function compared to important ones.

   This problem can be treated by weighting the contribution of each attribute so that some attributes will have a higher weight than others, when calculating the distance.

   Hint :

   - Defining a vector weight:

     ```
     create function weights() -> vector of number;
     ```

   - Assigning your own weights:

     ```
     set weights() = {1,1,1,1,1,1,1,1,1};
     ```

   - Creating your own weighted_euclid function:

     ```
     create function weighted_euclid(Vector of Number p1,
                                     Vector of Number p2)
         -> Real d as euclid(p1.*weights(), p2.*weights());
     ```

   - Passing your own `#'weighted_euclid'` function instead of the built-in one `#'euclid'`
   - You can experiment with a weighted Minkowski function

   Questions:

   - If all the values of a dimension are the same, how valuable is that dimension as a classifier?
   - Discuss what weights give the best classification?

7. CLASSIFY THE UNKNOWN DATA

   Classify the unknown data with the best settings found in Excercises 2-6 above.

The stored function `classified_data` should contain all classified data.

You can also find these instructions and hints in the TODO list of the `a1.osql` script file.

You are also referred to the Amos II manual for further reading, which is available through the lab course home page. We suggest that you read section 2.6 about Collections. You might find the following sections especially useful: `average`, `stdev`, and `count` in 2.6.1, `groupby` in 2.6.2, and `aggv`.

4 Examination

At the examination, your implementation will be executed in AmosMiner by executing the script file `a1.osql`. When the execution is done, we expect that :

`:k` (an interface variable containing the best number of neighbors, and)

`classified_data(vector of number x) -> bag of Vector of Number` (a stored function containing the result of classifying data in `testdata.nt`).

Be prepared to answer questions regarding topics such as:

- A brief description of your design decisions in the kNN implementation. For example, how the data was preprocessed and issues in the partitioning of data into training and test sets.
- The reasoning behind your choice of k.
- If it is known a priori (beforehand) that each feature has different importance, how can that knowledge be used in the pre-processing to improve the classication performance?
- How did you choose your distance measure? Why did you do it that way?

- What would be the complexity of the current algorithm for chosing k? What is the complexity for the algorithm to classify one data point?

Acknowledgements

Contributions to the material for this assignment has been supplied by several of the former and current colleagues at the department, in alphabetical order, including Per Gustavsson, Tobias Lindahl, Tore Risch, Thanh Truong and Erik Zeitler.

References

[ES87] Ian W. Evett and Ernest J. Spiehler. Rule Induction in Forensic Science. Central Research Establishment. Home Office Forensic Science Service. Aldermaston, Reading, Berkshire RG7 4PN.

[Tan06] Tan, P-N, Steinbach, M. and Kumar, V.: Introduction to Data Mining, Addison-Wesley, 2006.
