Data Mining

Spring 2013

Mandatory Assignment

March 22

Jens A. Grøn (jang)

IT Univerity of Copenhagen 2013

Introduction

In this small report the answers to the mandatory assignment of the course will be given.

Methods

In this project I used:

- 2-3 Preprocessing methods (Normalization, Discretization, and either a unknown value or a minimum value for missing or unknown values). Some reduction decisions were also taken
- a Supervised Learning method (ID3 with Information Gain)
- a Frequent Pattern method (Apriori) for the Programming Languages
- a Clustering method (k-Mediods) on the date-of-births

Each subject will be discussed further in the next four sections.

Preprocessing methods

All attributes were roughly handled by either **Normalization** or **Discretization**.

Normalization:

- Date of Birth
- Years of Study
- English Level
- Random 1
- Random 2
- Random 3

Discretization:

- Programming Skills
- an array of Programming Languages
- Animal
- Danish Mountains (yes + no)
- Winter Tired (yes + no)
- Canteen Food
- Favourite Color
- Knowledge of Neural Network (yes + no)
- Knowledge of Support Vector Machine (yes + no)
- Knowledge of SQL (yes + no)
- SQL Server
- Knowledge of Apriori (yes + no)
- Noor Shaker's hometown
- Knowledge of the Number of Planets in the Solar System (yes + no)
- Knowledge of the Next Number (yes + no)
- Knowledge of Fibonacci (yes + no)

Two input attributes, ”How old are you” and ”date of birth” was combined into one atribute. The three first normalized attributes could also have been discretized, but they weren't. Many of the discretized attributes are only binary values while the rest have some number of values including an unknown value used, when the input was not that clear. Some attributes were omitted if they were either not relevant (the optional “username”) or the information value was not that clear (“square root” or “therb fortt glag”).

Supervised Learning method

To analyse what kind of people are winter tired or not, the **ID3 method**, together with the **Information Gain**, was used to create such a decision tree. The code from Lab 2 was used with the extension to support numerical values. This case showed to be useless when the date-of-birth had the highest information gain and splitted all people into their own partition and hence does not tell us anything.
When excluding all the normalised attributes a technical issue appeared – Java throwed a *ConcurrentModificationException*, which then has not yet been resolved. Funny enough the code from Lab 2 worked without any of such kind of problems, but in this case... it should not be the case?

Frequent Pattern method

To find some frequent patterns among the data, the **Apriori method** was used to mining often-used-programming languages together. With the support of 2 the following three tables were created.

Keys with a single attribute: 9

Language	Support
java	25
csharp	24
python	11
cpp	9
fsharp	7
ruby	3
vb	3
ml	2
php	2

Keys with two attributes: 13

Language	Support
csharp - java	18
cpp - java	7
csharp - fsharp	6
python - csharp	5
python - java	5
csharp - cpp	4
fsharp - java	2
fsharp - python	2
csharp - php	2
ruby - java	2
ruby - csharp	2
python - cpp	2
vb - java	2

Keys with three attribute: 4

Language	Support
csharp – java - python	3
csharp – java - cpp	3
csharp – java - ruby	2
csharp – java - fsharp	2

There's no doubt that *csharp* and *java* are heavy used tools influencing hte mass. They each have, individualy, more than the doubled amount of users than number three. As a pair they 2,5 times in front of number two, and they influences all the triangles.

Clustering method

To cluster all the date-of-births (dob) the **k-Mediods method** is used. Some k random dobs is selected and the rest of the dobs are joined with the nearest one. Then a new random dob in each cluster is selected and compared with the current mediod (representative for that cluster) to see whether to choose a better mediod or contain the current.

The status of the implementation so far is not very good, because the total amount of objects in the clusters (above 50) is larger than than the beginning amount of persons (36), which means the clusters contain duplicates. This is obviously a bug in the implementation somewhere. The clustering of the objects is good because their is no overlap between the clusters.

A better choice would, probably, have been AGNES because it does not start with a predifined k. When choosing randomly which object to try to be the next mediod for a cluster, the outliers might not be choosen. But, of course, running all possibilities will not be effecient for larger data sets.

Conclusion

A very good assignment though I had some unresolved bugs. Mining data is a funny task because you try to uncover some ”hidden” knowledge, and when preprocessing you become aware of the difficulty of processing non-multiple-choice answers. And the methods Apriori, ID3 and k-Mediods are interesting to learn, though they seem rather intuitively.