INTRODUCTION TO ARTIFICIAL INTELLIGENCE

TASK 4: TEXT CLASSIFICATION

Misrraim Suárez Pérez - misrraimsp@gmail.com

1. ARFF Builder

The very first step in this lab assignment was to build up a program to transform a bunch of labelled documents into proper *arff* files readable for WEKA environment. This program was actually implemented as a set of tools, that works in sequence to reach the goal. The next diagram show the main idea.

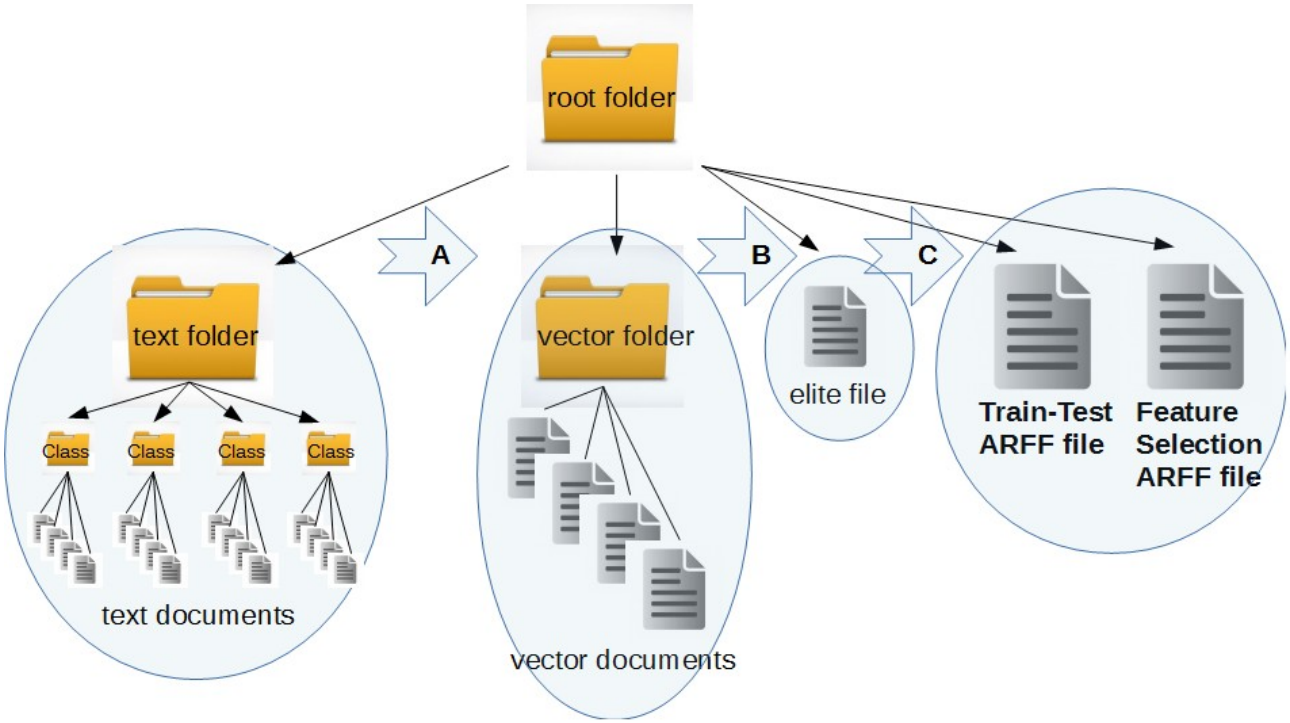

Figure 1. arff builder system diagram

There are three main tools, labelled in above figure as A, B, and C:

- A – take as input a set of raw text documents and convert each of them into a vector of word-frequency elements.
- B – take as input a set of word-frequency vector files and builds up an especial vector format file, in which there are the 10k (by default, but it can be edited) most frequent words. This document is called *elite*.
- C – takes an *elite* document and builds up two ARFF files, one with instances for feature selection and other for train and testing purposes. The proportion by default is 1/3 feature selection and 2/3 train and testing, but it is an editable parameter.

2. Feature Selection Methods

2.1 Correlation

From WEKA docs:

"Evaluates the worth of an attribute by measuring the correlation (Pearson's) between it and the class."

This coefficient ranges from −1 to 1, and give information about to what extent there exist a **linear** dependence between both variables. Values far from 0 means strong linear relation, while values close to 0 means poor linear relation.

The coefficient for a sample is defined as:

$$r_{wc} = \frac{\sum_{i=1}^{n} (w_i - \bar{w})(c_i - \bar{c})}{(n-1) s_w s_c}$$

where $\bar{w}$ and $\bar{c}$ are the sample means of attribute *w* and class *c*, and s_w and s_c are the corrected sample standard deviations of *w* and *c*.

Note that it only detects linear dependences (following picture is quite illustrative about this)

2.2 Information Gain

From WEKA docs:

"Evaluates the worth of an attribute by measuring the information gain with respect to the class."

The formula used to obtain this parameter is:

$$\text{InfoGain(Class,Attribute)} = \text{H(Class)} - \text{H(Class} \mid \text{Attribute)}$$

Intuitively, this parameter tries to measure how many evidence, or *information*, a particular attribute contributes to determine if class *c* is the proper one. In this context we can take information as the opposite of uncertainty, or *entropy*.

Lets give an example: I am wondering if tomorrow we'll win the lottery. The fact that the sun will rise up does not say nothing new, and so that it doesn't contribute with any additional information for our purpose: the uncertainty (*entropy*) remains constant. On the other hand, if I figure out that tomorrow a big meteorite will reach the Earth, which has very low probability, it adds so many information, and our uncertainty goes to zero: no lottery in a destructed world. More general, the higher the probability of a particular events, the lower the reduction of uncertainty such an event produce.

The degree of uncertainty is measured with *entropy* parameter: *H*. In our calculation, as shown above, we measure the entropy difference with and without the particular attribute in

predicting if a document *d* belongs to class *c*. For example, the attribute “be”, or the attribute”you”, does not add any additional information in determining the class of the document. These attributes doesn’t reduce the entropy, so the InfoGain parameter is close to zero. On the other hand, the attribute “hockey” will reduce the entropy, and the InfoGain will reflects this fact.

3. Classifiers

3.1 Multinomial Naïve Bayes

This is a statistical method that finds the proper class using Bayes conditional probability principles. It calculates the probability of a document *d* being in each class *c,* and then assign the document to the most likely class. The probability of a document *d* being in class *c* is computed using the Bayes rule as:

$$P(c \mid d) = P(c)\ P(d \mid c)$$

It is possible to drop the denominator in the last step in above Bayes rule because $P(d)$ is the same for all classes and does not affect the final result. As we represent documents as vectors of word-freq, the Bayes rule can be written as:

$$P(c \mid d) = P(c)\ P(d \mid c) = P(c)\ P(w1, w2, \ldots, wN \mid c)$$

Within this method there are two main assumptions: c*onditional independence assumption* (the words 'Hong' and 'Kong' are independent, i.e. the presence of 'Hong' does not affect for the probability of presence of 'Kong'...) and *positional independence assumption* (the conditional probabilities for a term does not depend of position in the document). With these two assumptions it is possible to rewrite the main formula:

$$P(c \mid d) = P(c)\ P(w1, w2, \ldots, wN \mid c) = P(c)\ P(w1 \mid c)\ P(w2 \mid c) \ldots P(wN \mid c)$$

In this way, it is possible to interpret $P(wi \mid c)$ as a measure of how much evidence *wi* contributes that *c* is the correct class. $P(c)$ is the prior probability of a document occurring in class *c*. If a document's words don't provide clear evidence for one class versus another, it is reasonable to choose the one that has a higher prior probability.

But so far we was talking about *real* values of $P(c)$ and $P(wi \mid c)$. These *real* probabilities are not possible to obtain, but estimate them. For the prior this estimate is $P(c)=\frac{N_c}{N}$ where *Nc* is the number of documents in class *c* and *N* is the total number of documents. The conditional probability $P(wi \mid c)$ is calculated as the relative frequency of word *wi* in documents belonging to class *c*: $P(wi|c)=\frac{count(wi,c)}{\sum count(w,c)}$

Finally, it is worth to point out that [1]:

> "The winning class in Naïve Bayes (NB) classification usually has a much larger probability than the other classes and the estimates diverge very significantly from the true probabilities. But the classification decision is based on which class gets the highest score. It does not matter how accurate the estimates are. *Correct estimation implies accurate prediction, but accurate prediction does not imply correct estimation.* **NB classifiers estimate badly, but often classify well**."

3.2 kNN

kNN stands for *k Nearest Neighbours.* This is a simple but good method for classifying. The main idea is to predict a document *d* being part of the class of the nearest neighbour document in the training set. This actually holds for 1NN, but for any *k*, kNN predict a document *d* being part of the most frequent class of the *k* nearest neighbours documents in the training set.

Of course, one key element in this kind of classifiers is the way a *distance* is defined. The document representation we use, vectors of words frequencies, forms a mathematical space in which it is possible to define many types of *distances*. Maybe the most popular one is the *Euclidean* distance, which is the way our brains most frequent understand the concept of space.

There exists a variant of the method, which it is also used in the experiments. It consists on weighting the importance of a neighbour by an inverse factor of its distance.

So, summarizing, with this method a document is predicted to be part of a class of the *nearest* document in the training set (1NN), or be part of the most frequent (with or without weighting) class within the *k nearest* documents in the training set (kNN).

4. Experiments

Classification experiments were carried out using 7926 documents from 8 different classes. For feature selection were used 1/3 documents and rest for train and testing purposes.

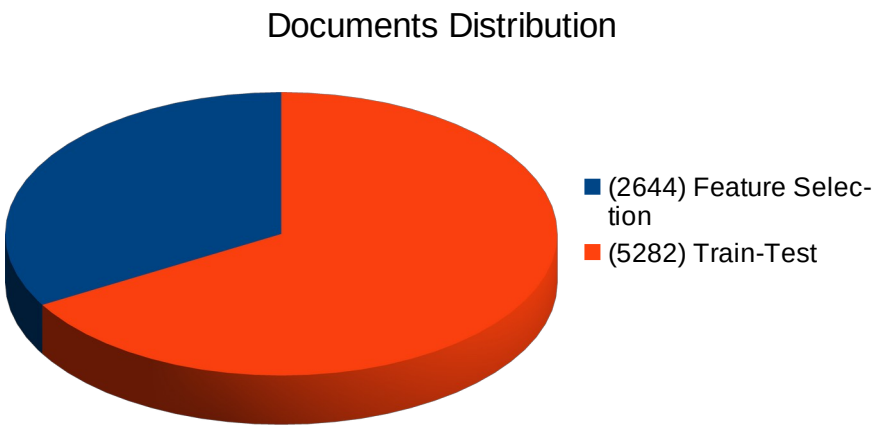

Figure 2. Documents distribution: 1/3 feature selection, 2/3 train and test

The different classes has been chosen in order to not make the decision task so evident. In following picture all classes are detailed.

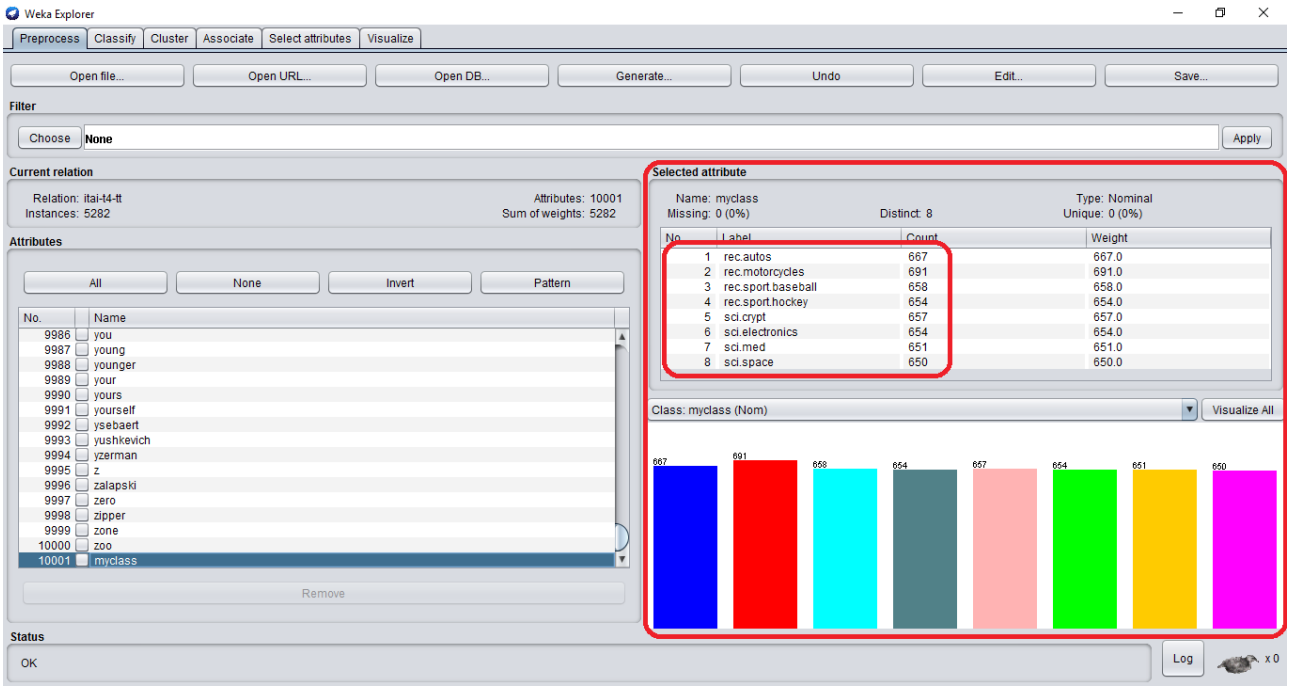

Figure 3. Classes detail

In the following figures it is represented the classifier performance against number of features (NB) and against value *k* (kNN). In total 6 different classify systems were tested.

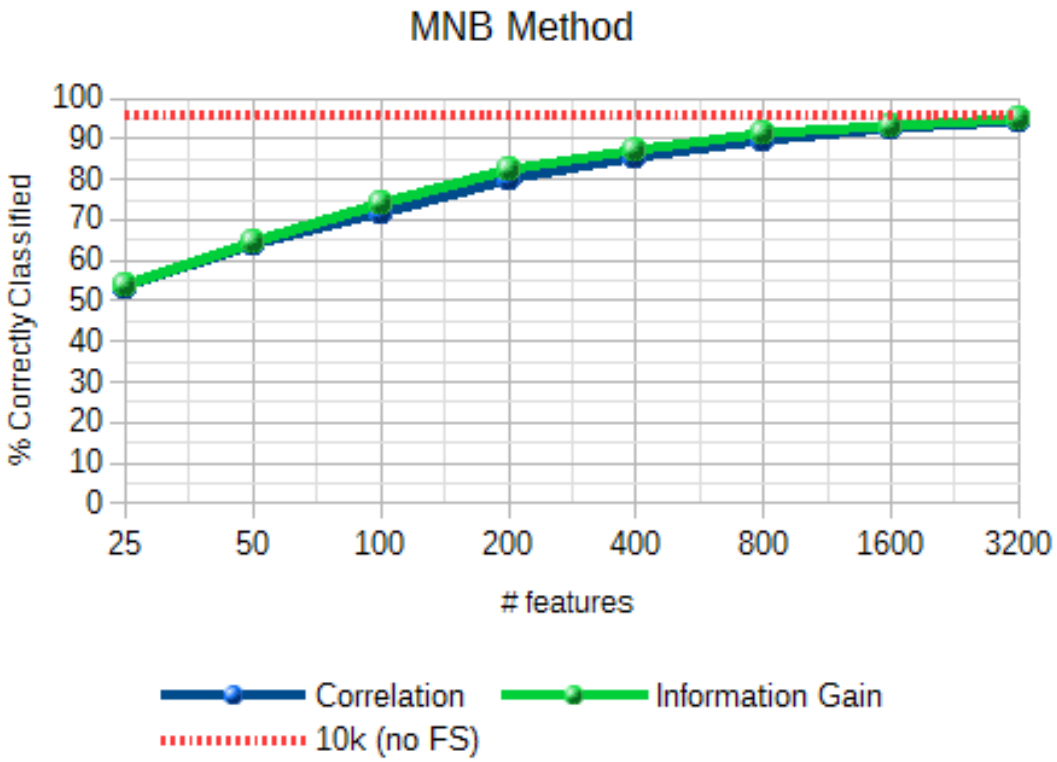

Figure 4. MNB performance with two different feature selection method for different number of features being selected.

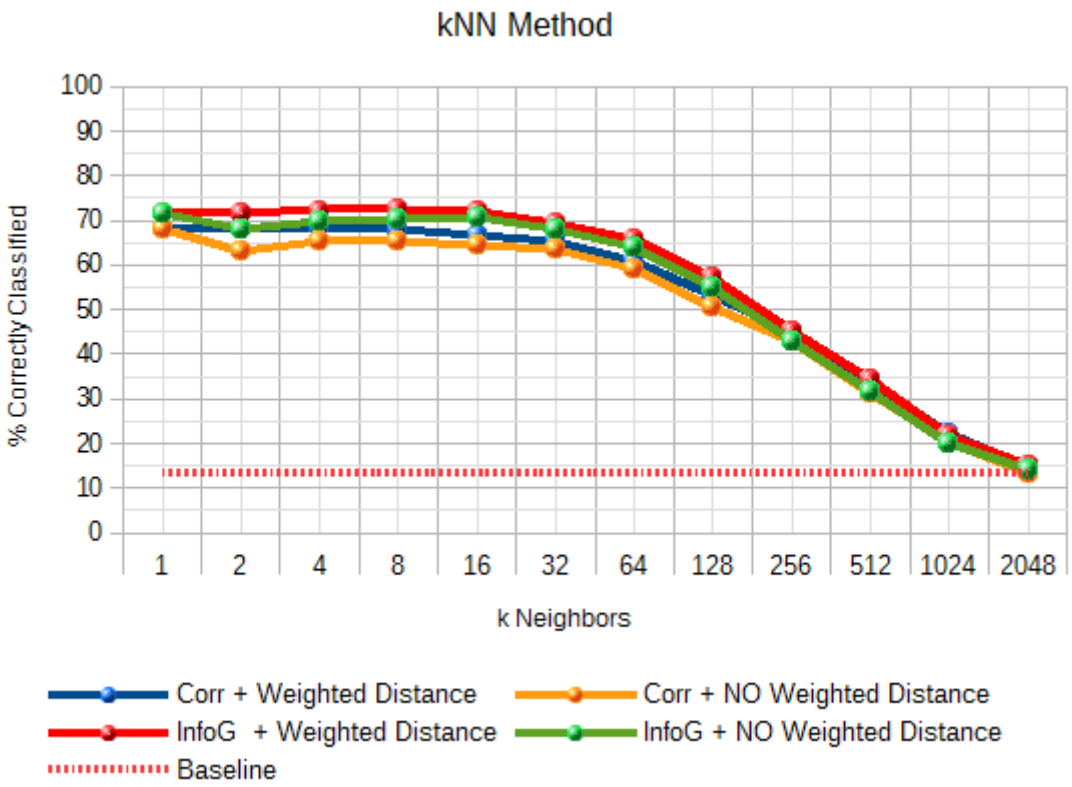

Figure 5. kNN performance with two different feature selection method for different values of parameter k, with and without weighting distances.

Several evidences can be pointed out from the experiments. In further lines it is exposed the principal and more obvious ones.

- In general, the correlation technique for feature selection performs worst than the information gain method. It is possible to see that on Figure 4 and Figure 5. With Naïve Bayes the difference is almost imperceptible, but in kNN the difference is significant.

- kNN method, see Figure 5, performs better with weighting technique. It is consistent with intuition: documents *d* closer to test document might have higher probability to belongs to the same class.

- Joining above two paragraphs it is possible to stablish an order among the four different kNN based classifiers that was tested:

IG+WD *better than* **IG+nWD** *better than* **Corr+WD** *better than* **Corr+nWD**

- In kNN systems, the optimal *k* value needs to be tuned. Experiments in above four different kNN systems showed that the optimal value is somewhere between 4 and 16 in all cases.
- For *k* above 16 to the whole train-test set size, the performance decrease severely, meeting the baseline for large *k* values. This is a reasonable behaviour, provided that kNN tries to take advance of *local* smooth continuity by choosing the most frequent class within its neighbourhood. In the limit, such neighbourhood is the whole set, and by definition performs like a classifiers that just predict a document to being part of global most frequent class: the baseline.

- In the experiments carried out NB systems performs better than kNN systems. Also, the higher number of features the higher results obtained. But, as we can see in Figure 4, the number of features tested increase exponentially while the relative performance improvement decrease significantly. This means that the decision of how many features to be selected needs a trade off between cost and precision.

5. References

Manning, Christopher D., Prabhakar Raghavan & Hinrich Schütze. *Introduction* to. *Information*. *Retrieval*. Cambridge University Press, 2008.

Data Mining with Weka. MOOC:

https://www.youtube.com/playlist?list=PLm4W7_iX_v4NqPUjceOGd-OKNVO4c_cPD