**PANACEA Project**
**P**latform for **A**utomatic, **N**ormalized **A**nnotation and
**C**ost-**E**ffective **A**cquisition
*Grant Agreement no. 248064*

# Documentation of Naive Bayes classifier Web Service[1]

Author: Muntsa Padró. Barcelona, 2012.

Contact: muntsa.padro@upf.edu

In this document, first of all we present the inputs needed by the web service and their format and the ouput it produces. Some theoretical details can be found in section 2.

## 1 Overview

This webservice performs traditional Naive Bayes classification of instances given in a weka file. It ouputs the predictied classification for each instance and some statistics about the performance of the classification. The parameters needed as input can be learnt using estimate_bayesian_parameters webservice.

## 2 Inputs, outputs and formats

**Inputs:**

- ***test_data*** (test data): instances encoded as cue vectors that need to be classified. Each slot of the vector contains the number of times each feature has been observed for that instance. Also, we add special slots: total number of occurrences in the first slot and correct class (1 or 0) and lemma (or any identifier of the instance) in the two last slots. If the correct class is unknown, substitute this slot by "?".

  **Format:** The vectors should be encoded in a weka file. For example, for the class of eventive nouns in English we would have some examples of eventive nouns and some of non-eventive nouns:

```
@relation eventive.arff'
@attribute "[total_occurences]" numeric
@attribute "[cue_1]" numeric
@attribute "[cue_2]" numeric
…
@attribute "[cue_n]" numeric
@attribute "[eventive]" {0,1}
@attribute "[lemma]" string

@data
5,0,0,…,3,0,visa
386,0,1,…,162,0,characteristic
23,1,0,… ,0,1,ceremony
270,0,2,…,0,1,assembly
```

  **Important:** the cue counts must be encoded as integers, this is, no relative frequency needs to be given but the number of times each cue has been seen.

- ***Parameters***: parameters to perform a Naïve Bayes classification, this is basically the probability of seeing each cue for members of non-members of the class.

  **Format:** The format of this input is equivalent to the output of the Bayesian parameter estimation Web Service, which contains some extra information. It consists of data separated by ";", each line containing the

information of each cue. The first line explains the contents of each column. There are some fields that are specific for a Bayesian inference of parameters and of the Bayesian parameter estimation Web Service, but they can be left blank if parameters have been estimated in another way. In fact, the only fields used by the classifier are the last two (log quotient ratios). The fields of the input file are:

- *cue*: label of the cue
- *data size class*: number of tokens in the training data that belong to the class
- *virtual size class*: virtual size given to the class (input parameter)
- *data p class*: probability of seeing this cue for members of the class estimated only from data (MLE)
- *prior p class*: prior *virtual relative frequency* given in the input file for members of the class
- *theta class*: probability of seeing this cue for members of the class estimated using a Bayesian approach, i.e. combining observed data and virtual frequencies.
- *data size no class, virtual size no class, data p no class, prior p no class, theta no class*: same fields presented above but for non-class members.
- ***log(theta[class]/theta[noclass]), log((1-theta[class])/(1-theta[noclass]))*:** pre-computed log-quotients to be used in the Naïve Bayes classifier (see equation 2).

*Example*:

#cue;data size class;virtual size class;data p class;prior p class;theta class;data size no class;virtual size no class;data p no class;prior p no class;theta no class;log(theta[class]/theta[noclass]);log((1-theta[class])/(1-theta[noclass]))
cue_1;55262;522008;1.80956172415e-05;1e-05;1.07749926378e-05;36745;15474000;2.72145870186e-05;0.0;6.44714357692e-08;5.11876096228;-1.07105792506e-05
…
cue_n;55262;522008;0.00152003184829;0.01;0.00918821348762;36745;15474000;0.00144237311199;0.0001;0.000103180085805;4.4892009316;-0.00912750007329

- ***cues_to_ignore*** (optional): list of cues to be ignored when performing classification. They must be identified by their position, counting the first cue as 1, and separated with comma. E.g: 1,4,17. By default none cue is ignored.
- ***prior_probability*** (optional): prior probability given to the the class (*P(k=1)* in equation 2). By default it is set to 0.5.

**Outputs:**

- ***classification:*** all instances given in *test_data* classified as belonging/not belonging to the class.

  **Format:** each line contains the information about one classified instance. It contains different fields separated by ";". The fields are:

  - original signature as given in the weka file
  - score for the classification: if it is bigger than 0, the element is considered to belong to the class.
  - the classification prediction: 1 for members of the class, 0 for non members
  - the correct class (given in test file). If the correct class is not given, this field will be "?".

  Note that comparing two last fields we can get the accuracy of the classification.

  *Output example:*

2149,0,0,0,0,7,33,9,41,0,2,145,0,0,11,1,accident;669.8618994;1;1
2357,8,0,0,0,0,1,0,19,4,9,94,0,1,585,1,speech;-986.238226929;0;1
2428,0,0,0,0,0,1,1,0,0,66,357,1,1,40,0,organism;-554.668468551;0;0

- ***Performance:*** the web service also outputs some statistics about the performance of the system (accuracy, false positives, false negatives, etc).

# 3 Theoretical background

## 3.1 Cue-based classification of nouns

As an example of use of this web service, we will present the cue-based classification of nouns, but this can be extended to any task that performs classification based on studying the number of times each feature has been observed.

Traditional Naive Bayes approach to cue-based noun classification computes the probability of a given noun belonging to a particular class *k* given a vector *w* that represents the number of times the word to classify has been seen with each cue ($n_i$). This probability is calculated by Bayesian inversion as:

Where *P(k)* is the prior probability of the class *k*, and *P(w|k)* is the likelihood of vector *w* given a class *k*. This likelihood is estimated (assuming independence of cues) as the product over all components $n_i$ of the vector.

In our approach, we built the classifier as a comparison of two hypotheses: the word belongs to the class (*k=1*), or it does not (*k=0*). The hypothesis with higher probability (assessed using the log quotient in 2) is considered the correct one.

Due to the Bernoulli distribution of cue occurrences, we can calculate the probability of a particular vector component given the class we need for 2 using equation 3:

Where *N* is the total number of occurrences of the word and $P(cue_i|k)$ refers to the probability of the cue given the class for a single occurrence.

Thus, in order to classify a noun using a Naive Bayes classifier (equation 2), we estimated $P(cue_i|k)$ for each cue and class and the prior probabilities of the classes *P(k=0)* and *P(k=1)*. Usually, these prior probabilities are considered equal to 0.5.

## Related Web Service:

Naive Bayes classifier Web Service: http://lod.iula.upf.edu/resources/229