# Web Services for Bayesian Learning

**Muntsa Padró**
Universitat Pompeu Fabra
Barcelona, Spain
muntsa.padro@upf.edu

**Núria Bel**
Universitat Pompeu Fabra
Barcelona, Spain
nuria.bel@upf.edu

## Abstract

In this demonstration we present our web services to perform Bayesian learning for classification tasks.

## 1 Introduction

The Bayesian framework for probabilistic inference has been proposed (for instance, Griffiths et al., 2008 and a survey in Chater and Manning, 2006 for language related topics) as a general approach to understanding how problems of induction can be solved given only the sparse and noisy data that humans observe. In particular, how human acquire words if the available data severely limit the possibility of making inferences. Bayesian framework has been proposed as way to introduce a priori knowledge to guide the inference process. In particular for Lexical Acquisition, Xu and Tenembaum (2007) proposed that given a hypothesis space (all what a word can be, according to a set of existing classes) and one or more examples of a new word, the learner evaluates all hypotheses for candidate word classes by computing their posterior probabilities, proportional to the product of prior probabilities and likelihood. The prior probabilities are the learner's beliefs about which hypotheses are more or less plausible. The likelihood reflects the learner's expectations about which examples are likely to be observed given a particular hypothesis about a word class. And the decision on new words is determined by averaging the predictions of all hypothesis weighted by their posterior probabilities.

The hypothesis behind is that natural language characteristics, such as the Zipfian distribution of words (Zipf, 1935) and considerations as the classic argument on sparse data (Chomsky, 1980), make it necessary to postulate that the learning of words must be guided by the knowledge of the lexical system itself, information about abstracted, not directly observable categories (Goldberg, 2006; Bybee, 1998).

In order to test this hypothesis we developed a series of tools for the task of noun classification into lexical semantic classes (such as EVENT, HUMAN, LOCATION, etc.). The tools perform Bayesian parameter estimation where prior knowledge is included into the parameters as virtual evidence (following Griffiths et al. 2008) and a Naive Bayes based classification. Our assumption is that, if introducing prior knowledge improves the classification results, it may give some insights about the way humans learn lexical classes.

The developed tools have been deployed as web services (following web-based architecture of the PANACEA project[1]) in order to make them easily available to the community. They can be used in the task just mentioned but also in other tasks that may profit from a Bayesian approach.

## 2 Web Services for Bayesian modeling

In this demonstration, we present two web services that can be used for Bayesian inference of parameters and classification with the aim that they may be useful to other researchers willing to use Bayesian methods in their research.

### 2.1 Naive Bayes Classifier

A first web service performs a traditional Naive Bayes classification. The input is the observed data from a given instance encoded as cue vectors, this is, the number of times we have seen each cue in the context of the studied instance. Then, the web service computes how likely is that this instance belongs to a particular class. The input needed by the classifier is the set of probabilities of seeing each cue given each class $P(cue_i|k)$. Those parameters should have

[1] http://panacea-lr.eu/



*Proceedings of the EACL 2012 Workshop on Computational Models of Language Acquisition and Loss*, pages 29–31,
Avignon, France, April 24 2012. 

been previously induced (using Maximum Likelihood Estimation (MLE), a Bayesian approach, etc.).

The classifier web service reads those probabilities from a coma separated file and the cue vectors of the instances we want to classify in Weka format (Quinlan, 1993). In our implementation, we work with binary classification, i.e. we want to decide whether the noun belongs or does not belong to a given class. Thus, the service returns the most likely class for each instance given the parameters and a score for this classification (i.e. how different was the probability of being and not being a member of the class).

### 2.2 Bayesian Estimation of Probabilities

A second web service performs parameter inference for the Naive Bayes classifier using Bayesian methods.

Bayesian methods (Griffiths et al., 2008; Mackay, 2003) are a formal framework to introduce prior knowledge when estimating the parameters (probabilities) of a given system. The main difference between those methods and MLE is that the latter use only data to estimate parameters, while the former use both data and prior knowledge.

An example of Bayesian learning is determining the probability of a coin producing heads in a short throw series. A MLE approach will determine this probability as $p(head) \approx \frac{N_{heads}}{N}$. Thus, after observing a sequence of 5 heads in a row, MLE would assess that the probability of the coin producing heads is 1. Nevertheless, because of our knowledge, we would rather say that a tail is more than possible, and that the coin probability can still be close to 0.5. Bayesian models allow us to formally introduce this knowledge when estimating the probabilities.

In the case of Naive Bayes classification using cue vectors, we need to estimate $P(cue_i|k)$ for each cue and $k$ (for binary classification this would be $k$=1 for being a member of the class and $k$=0 for not being a member of the class).

Bayesian modelling computes these parameters approximating them by their Maximum a Posteriori (MAP) estimator. The canonical approach introduces the prior probabilities as a Beta distribution, and leads to the following MAP estimator (see Griffiths et al. (2008) and Mackay (2003) for details):

$$MAP = \hat{P}(cue_i|k) = \frac{N^i_{yes}(k) + V^i_{yes}(k)}{N^i_{yes}(k) + V^i_{yes}(k) + N^i_{no}(k) + V^i_{no}(k)}$$

Where $N^i_{yes}(k)$ and $N^i_{no}(k)$ are the observed occurrences in real data ($N^i_{yes}(k)$ is the number of times we have seen $cue_i$ with class $k$ and $N^i_{no}(k)$ is the number of times we have not seen it, and $V^i_{yes}(k)$ and $V^i_{no}(k)$ represent what is called *virtual data*, this is, the data we expect to observe a priori. Thus, it can be seen from the MAP estimator that Bayesian inference allows us to add virtual data to actual evidence.

The web service we want to show in this demonstration implements the estimation of $P(cue_i|k)$ combining the data and the priors supplied by the user. The service reads labelled data in Weka format and the priors for each cue and class and computes $P(cue_i|k)$. The output of this web service can be directly used to classify new instances with the first one.

## 3 Test case: Lexical Acquisition

As a showcase, we will show our work in cue-based noun classification. The aim is the automatic acquisition of lexical semantic information by building classifiers for a number of lexical semantic classes.

### 3.1 Demonstration Outline

In our demonstration, we will show how we can use the web services to learn, tune and test Bayesian models for different lexical classes. We will compare our results with a Naive Bayes approach, which can also be learned with our system, using null virtual data.

First of all, we will get noun occurrences from a corpus and encode these occurrences as cue vectors applying a set of regular expressions. This will be done with another web service that directly outputs a Weka file. This Weka file will be divided into train and test data.

Secondly, the obtained training data will be used as input in the Bayesian learner web service, obtaining the values for $P(cue_i|k)$ for each cue and class. We will perform two calls: one using prior knowledge and one without it (MLE approach).

Finally, these two sets of parameters will be used to annotate the test data and we will compare the performance of the Bayesian model with the performance of the MLE model.

**Acknowledgments**

This work was funded by the EU 7FP project 248064 PANACEA.

## References


J. Bybee. 1998. The emergent lexicon. CLS 34: The panels. *Chicago Linguistics Society*. 421-435.

N. Chater,and C.D. Manning. 2006. Probabilistic models of language processing and acquisition.*Trends in Cognitive Sciences*, 10, 335-344.

N. Chomsky.1980. Rules and representations. Oxford: Basil Blackwell.

A. E. Goldberg. 2006. Constructions at work. Oxford University Press.

T. L. Griffiths, C. Kemp, and J.B. Tenenbaum. 2008. Bayesian models of cognition. In Ron Sun (ed.), *Cambridge Handbook of Computational Cognitive Modeling*.Cambridge University Press.

D. J. C. MacKay. 2003. Information Theory, Inference, and Learning Algorithms. *Cambridge University Press*, 2003. ISBN 0-521-64298-1

R.J. Quinlan. 1993. C4.5: Programs for Machine Learning. *Series in Machine Learning*.Morgan Kaufman, San Mateo, CA.

Xu, F. and Tenenbaum, J. B. (2007).Word learning as Bayesian inference.*Psychological Review 114*(2).

G.K. Zipf. 1935. *The Psycho-Biology of Language*, Houghton Mifflin, Boston.