Improving Web Learning through model Optimization using Bootstrap for a Tour-Guide Robot

Rafael León, J. Javier Rainer, José Manuel Rojo and Ramón Galán.
Intelligent Control Group
Polytechnic University of Madrid, Spain

Abstract **—We perform a review of Web Mining techniques and we describe a Bootstrap Statistics methodology applied to pattern model classifier optimization and verification for Supervised Learning for Tour-Guide Robot knowledge repository management. It is virtually impossible to test thoroughly Web Page Classifiers and many other Internet Applications with pure empirical data, due to the need for human intervention to generate training sets and test sets. We propose using the computer-based Bootstrap paradigm to design a test environment where they are checked with better reliability.**

Keywords **—Web Mining, Supervised Learning, Bootstrap, Patterns Mining, Web Classifiers, Knowledge Management.**

I. Introduction

THE Internet is an enormous information repository with spectacular growth and a high degree of updating. Using this exceptional database in an automatic way is a challenging field of research. Data mining has been extensively used by many organizations, large amount of data are processed to extract relevant information; applying these technologies to the Web it is possible to build systems that considerably improve the process of information gathering from the Internet. Web Mining has peculiarities that made it a subject of research in its own right. It can be applied to several aspects of the network like page content, user click stream, link structure of the web or social community opinion. Internet mining is performed in several steps: web pages covering a particular matter or belonging to a social community are searched and classified. Then, they are processed to remove all the words and tags that have no influence in the meaning of text and a mining algorithm is applied to harvest useful information from the pages. This knowledge can be used in several manners: analysis to study behavior patterns, social community opinions and product success or it can be incorporated into a knowledge repository that is the application for our robot.

In this article, we have dedicated our efforts to pattern classifiers that allow a binary classification of Web pages. Classification models that use patterns to define features or rules have been built for a long time. These models can be either more accurate or less precise, but they achieve more understandable results for humans. A lot of work has been performed on pattern finding and selection, algorithms and model building, but there is scarce work on model verification and comparison as stated in [1] by B. Bringmann. We have developed a test environment that implements Bootstrap resample strategy allowing to determine with more confidence how the model performs, so criteria can be clearly defined to compare between models performance, thus improving the whole learning process.

Bootstrap is a computer approach to get statistical accuracy. It is applied to a wide variety of statistical procedures like non parametric regressions, classification trees or density estimation. This technique requires fewer assumptions and offers greater accuracy and insight than other standard methods for many problems. Bootstrapping is an analogy in which the observed data assume the role of an underlying population: variances, distributions and confidence intervals are obtained by drawing samples from the empirical sample, as R. Stinewrotein [2]. A typical problem in applied statistics involves the estimation of an unknown parameter. The two main questions are: what estimator should be used? and having chosen a particular one, how accurate is the estimator? Bootstrap is a general methodology to answer the second question, as stated by Efron and Tibshiraniin [3].

This work is framed within the Intelligent Control Group, Universidad Politécnica de Madrid, whose members are carrying out research into robotics and intelligent control systems. Three robots have already been built which are designed to show visitors round museums and fairs described by Rodriguez-Losada in [4]. Research covers a wide number of areas: path finding, navigation, speaking, facial expression, mood and knowledge management.

II. Software Architecture for based on Intelligent Agents

We have developed our own interactive mobile robot called Urbano specially designed to be a tour guide in exhibitions. Urbano is a B21r platform from iRobot, equipped with a four

wheeled synchrodrive locomotion system, a SICK LMS200 laser scanner mounted horizontally on the top used for navigationas well as a mechatronic face and a robotic arm used to express emotions such as happiness, sadness, surprise or anger. The robot is also equipped with two sonar rings and one infrared ring, which allows detecting obstacles at different heights that can be used for obstruction avoidance and safety. The platform also has two onboard PCs and one touch screen, as explained by J. Rainer in [5].

The software is structured in several executable modules to allow a decoupled development by several teams of programmers, and they are connected via TCP/IP and CORBA. Most of these programs are conceived as servers or service providers, as the face control, the arm control, the navigation systems voice synthesis and recognition, and the web server. The client-server paradigm is used, the only client being a central module that we call the Urbano Kernel. This kernel is responsible for managing the whole system, as illustrated in [4].

Modules with a more advanced implementation are: Decision Making, Knowledge Server, Automatic Presentation Generator and Acquisition of Information, as can be seen at Fig. 1.

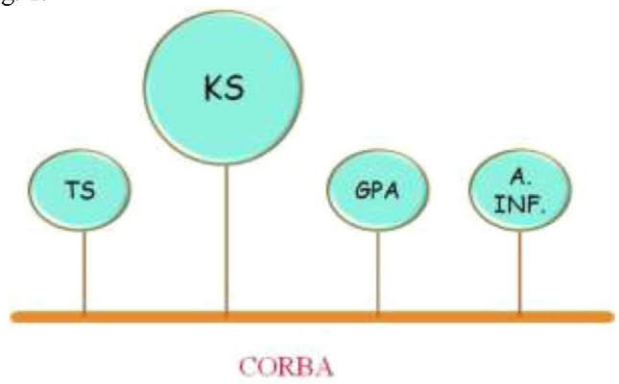

Fig. 1. Modular Architecture based on Intelligent Agents: TS, Decision Making, KS, Knowledge Server, GPA, Automatic Presentation Generator, and A. Inf., Acquisition of Information

We have built the Acquisition of Information Agent which aims to automate as much as possible the incorporation of information to the Knowledge Server, using the Internet the as primary source of Information.

A. KS: Knowledge Server, Urbano Ontology Implementation

Knowledge Server is at the center of the architecture, providing data and intelligence to the behavior of the rest of the components. It incorporates cognitive inspired ontologies that store the information and concepts. Feeding these ontologies in an automated way is a challenge. URBANOntology consists of a foundational ontology (DOLCE) plus different domain specific ontologies, like art, history etc. The robot is able to give presentations about different topics as domain ontologies that are mapped to DOLCE as described in [6]-[8]. It is not only a classification; it also provides the tools needed to conceptualize the world and describes how the different objects relate to each other. DOLCE is made up of categories based on perception and human common sense, cultural details and social conventions.

Using DOLCE as a fundamental ontology, we are setting out a general framework that can be tailored to any specific domain; in this way the URBANOntology can serve as a reliable tool to potentially generate presentations in all possible areas. Every component in the Museum Ontology must be mapped to its respective fundamental concept in DOLCE. The use of a knowledge server means having a useful tool with which to meet the needs of handling the knowledge. By abstraction of knowledge we understand a learning process that involves the formation of new concepts or categories based on information available about the world. The knowledge server consists of a Java application developed using the libraries of Protégé-OWL API. In [5], J. Rainer y R. Galán explain it with more detail.

The robot changes between museums and exhibition fairs makes it necessary to update its knowledge database for each location. We have optimized the information gathering process, including the option of Web Mining from the Internet, as shown in Fig. 2.

Web Learning is performed in several steps, in this paper we suggest an improvement for the Page Selection phase, proposing a test environment that increases the reliability of

Fig. 2. Information extraction process from the Internet

the results obtained by the Selection Models. The robot's environment has been enriched with the application of these techniques.

III. A Review on Mining Techniques Applied to Web Learning

We present in this section a review of mining techniques that have been developed and theoretical support for Web Mining. Many approaches have been proposed to extract information from raw data, ranging from those involving more human intervention, supervised methods, to more unattended systems, unsupervised ones. The selection depends on several factors like the type, heterogeneity and volume of data.

Supervised learning is also known as classification or inductive learning. It is similar to the human behavior of

learning from past experiences thus gaining new knowledge and abilities. The experiences are represented as past data, so there are the following sets of data: training data, that allow model training, test data, that allow the verification of the classification and the real data that are processed by the model. The Accuracy of the classifier is evaluated in terms of the number of correct classifications versus the total number of cases. Decision tree is one of the most popular methods of classification; it is efficient and can compete with other classification techniques, like Bayesian Classification, D. Hand in [9], or Support Vector Machines, V. Vapnik in [10].

Another approach is Unsupervised Learning that discovers patterns in the attributes of the data, that are used to predict the value of class attribute of future instances. The classes are used to classify items, for example decide if a web text is a social science article or if it is about microelectronics. When the data have no class attributes, clustering techniques are applied to find similarity groups. Clustering makes partitions of data or can have a hierarchical approach. A distance function is chosen between data points, and a set of centroids are calculated and recalculated recursively until similar groups are found. K-means Algorithm is an example of this strategy, as explained by MacQueen in [11].

There is an intermediate approach that is Partially Supervised Learning. Supervised Learning requires a lot of human effort and a large set of labeled data, therefore an alternative was proposed. To minimize the tedious task of labeling data, the model is trained with labeled and unlabeled examples, also known as LU learning, an example of the algorithm used is EM, Expectation and Maximization, A. Dempster in [12]. There is a lot of classified data and a large set of unlabelled data that are used to improve learning of the model. Subsequently the learning process uses positive and unlabeled data, PU learning, assuming a two classes set of data. This method can be applied successfully for classification of web pages.

In this section a number of general mining techniques have been described. As stated before, we have focused our efforts on mining patterns for Supervised Learning and their use to build a Web page classifier.

A. Peculiarities of Web Mining

Web mining is an activity that discovers useful information from the Web. It can target data or hyperlinks and can be classified into three kinds: Web Structure Mining, Web Content Mining and Web Usage Mining. Web Structure Mining discovers useful knowledge from hyperlinks structure. It is used to find important sites and communities and gives an image of the structure of the Internet itself. Web Content Mining extracts useful information from page contents. Web Usage Mining discovers how the users make use of a Web Server. Analyzing the Web logs it is possible to review the click stream and determine the user behavior and what he likes and what he dislikes, as stated by Bing Liu in [13].

One key step for Web mining is the pre-processing of the Web pages. To begin with HTML information is identified to classify the importance of the different paragraphs of the page. HTML codes allow recognizing titles, main content blocks or anchor text; this information can be used to speed up the identification and processing of the page.

Subsequently HTML tags are removed in order to extract the information in a more efficient way. When the page text has no tags it is clean from words that form syntactic constructions that have little influence on the meaning of the phrases, stop words like prepositions, conjunctions and articles are removed. The rest of the words are converted to their roots in a procedure called stemming. Verbs are transformed to the infinitive form and suffixes are stripped from words to get the roots which are easier to recall. A good example of stemming algorithm is from M. Porter in [14]. After the page is pre-processed a better precision for classification is obtained, getting improved results using distance functions like cosine similarity.

Parts of Speech, POS, can be used at a later stage in order to get the sentiments and semantic meaning attached to the text. With these techniques a word is identified in its category: noun, verb, adjective, adverb, pronoun, etc. Knowing the type of word it is possible for a machine to perform further processing to extract information from a text, being able to identify pictures by a painter, dates of birth or companies working in a particular sector. An algorithm implementing this approach was proposed by P. Turney in [15]. Using methodologies like Latent Semantic Analysis, was proposed by S. Dreewester [16], it is possible to apply statistical analysis to find the Singular Value Decomposition of a Web page, discerning when several texts have the same semantic meaning expressed with different words.

IV. USING STATISTICAL TECHNIQUES TO TEST AND DEVELOP

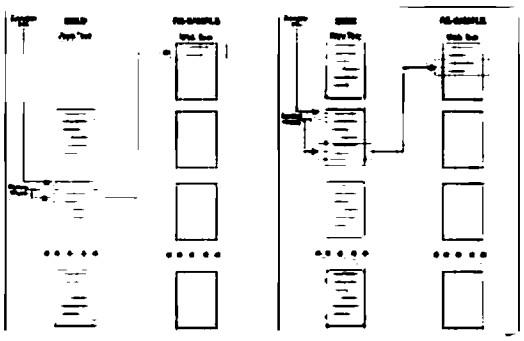

Fig. 3. Generation process of the new sample

SUPERVISED LEARNING MODELS

One of the main challenges of Supervised Learning Models like classifiers for Web Pages is the sheer number of pages that has to be processed, but the training and test pages set have a limited size due to the need for a human operator that classifies pages manually. There is a plethora of algorithms and literature about pattern classifiers, but fewer studies have

been carried out so far on verification and comparison between models. To solve this problem, we have developed a methodology based on the statistical paradigm called Bootstrap that allows one to synthesize re-samples automatically with replacement therefore greatly improving the accuracy of the verification and refining phase of the models, strengthening the reliability of their results and giving a clear idea about their optimum working conditions. Bootstrap is a computer approach to get statistical accuracy. It is applied to a wide variety of statistical procedures like non parametric regressions, classification trees or density estimation. We go a little further, iteratively varying the attributes of the population and applying Bootstrap for each situation.

A. The Bootstrap estimate

In our case we used Bootstrap to estimate parameters for a classifier. From an equally distributed empirical population, a collection of samples are constructed replacing randomly the original dataset. Given a set of independent and identically distributed observations, web pages in our case:

$$x_i,\ i = 1,2,\ldots,n \quad (1)$$

from an unknown probability distribution F has been observed. To estimate a parameter of interest $\theta = t(F)$ on the basis of x. For this purpose we calculate estimate $\hat{\theta} = s(x)$ from x. To know how accurate $\hat{\theta}$ is, Bootstrap was introduced as a computer based method. It is completely automatic no matter how complicated the estimator is from a mathematical point of view. Let $\hat{F}$ be the empirical distribution with a probability of 1/n on each of the observed values:

$$x^*(x_1^*, x_2^*, \ldots\ldots, x_n^*) \quad (2)$$

A bootstrap sample is defined as a random sample of size n drawn from $\hat{F}$, say

$$x^* = (x_1^*, x_2^*, \ldots\ldots, x_n^*) \quad (3)$$

$$\hat{F} \rightarrow x^* = (x_1^*, x_2^*, \ldots\ldots, x_n^*) \quad (4)$$

Corresponding to bootstrap dataset x^*, the sample of pages that are generated, is a Bootstrap replication of $\hat{\theta}$:

$$\hat{\theta} = s(x^*) \quad (5)$$

It is necessary to evaluate the bootstrap replication corresponding to each bootstrap sample:

$$\hat{\theta} = s(x^{*b})_{b=1,2,\ldots,B} \quad (6)$$

As an example, to estimate an estimator as the standard error:

$$se_F = (\hat{\theta}) \quad (7)$$

The sample standard deviation of the B replications:

$$\widehat{se}_B = \left\{\sum_{b=1}^{B}\left[\hat{\theta}^*(b) - \hat{\theta}^*(.)\right]^2/(B-1)\right\}^{1/2} \quad (8)$$

Where:

$$\hat{\theta}^*(.) = \sum_{b=1}^{B}\hat{\theta}^*(b)/B \quad (9)$$

As stated by Efron and Tibshiraniin [17].

V. TEST ENVIRONMENT

As explained above, the test environment is based on Bootstrap statistical estimation and it is applied to evaluate pattern classifier models and how they perform when some features of the pages change. The generation of Web page re-samples with replacement is as follows: from a set of thoroughly classified web pages used as seeds, a page is selected at random and inside this page, a phrase is randomly chosen and is written in the new sample, as can be seen in Fig. 3.

Bootstrap demonstrates that the new sample has the same underlying conditions than the original. We create two sets of pages: one from pages referring to Francisco de Goya as a painter and other one from pages that have to be discarded by the model. The two sets are evaluated and the confusion matrix is built:

		Actual value	
		p	N
PredictedValue	p'	A	B
	n'	C	D

Then we calculate the statistical tests Sensibility and Specificity for the model according to the following equations:

$$sensibility^* = \frac{A}{A+C} \quad (10)$$

$$sensibility^* = \frac{D}{B+D} \quad (11)$$

We greatly improve the accuracy of the verification and refining phase of the models, strengthening the confidence in their results and giving a clear idea about their optimum working conditions. With this information for example, we are able to adjust the models more precisely, improving their performance and enabling the building of dynamic strategies that obtain better results.

In our case we have tested how the models are affected by the size of the pages. The sample generation process can be easily modified to configure different page sizes for the generated sample. It is very difficult to implement this test

with pure empirical data, selecting a big enough set of pages manually according to size and accuracy.

We have implemented classifiers based on text patterns that recognize if a Web page is about the Spanish painter Francisco de Goya, discarding pages about Goya Street, Goya Awards, Goya Train, etc. Search engines return links to a collection of Web pages that have to be filtered. In a mining process of information about Goya, is necessary to discard all the pages that, while referring to the painter's name, are about other matters. To perform the page filtering, models implement a variety of approaches: from a simple static configuration to more advanced and dynamic methods.

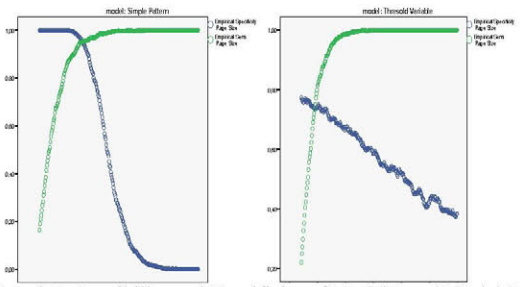

Fig. 4 and 5. Sensibility and Specificity of Model 1 and Model 2

A. Implementation details

The implementation is written in Python language and is modular, using a collection of classes that provide a flexible test environment to analyze new models. These classes are:

The control program: that implements the global loop, and calls the other classes.

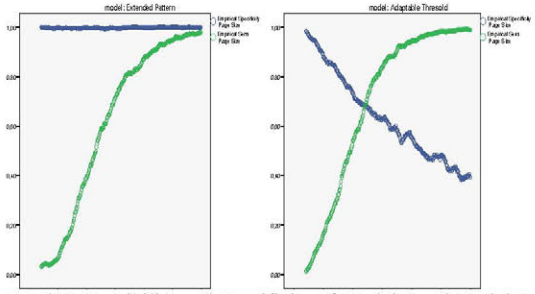

Fig. 6 and 7. Sensibility and Specificity of Model 3 and Model 4

The extraction class: that extracts text from web pages. Dynamic content adds noise to the text data of web pages and is hard to remove completely. Fortunately, this noise has very little effect on pattern mining that precisely is a good tool to filter it.

The generation class: that is responsible for generating the samples. It uses a method that generates the pages and another that is responsible for cleaning the samples in preparation for the next iteration.

The model classes: they implement a collection of models. In addition they use two of methods: one to calculate the sensibility and the other the specificity.

A random set of 200 files of re-sampled pages are created, increasing the page size with one phrase at each iteration. They are built from positive and negative seeds and placed in a directory where the model being assessed is applied. Then, the corresponding sensitivity and specificity are calculated. The results are rendered in the next section.

VI. RESULTS ANALYSIS

As described above, we have run our models over samples of 200 correct pages and 200 bad pages, with increasing size from 10 to 300 phrases. We have chosen to test the influence of the size of the page on the performance of the classifier.

We were able to verify that the size of the pages is strongly related to the performance of the models. As can be seen on the graphics, when the pages are small, models have low Sensibility: a lot of good pages are classified as bad pages. But when the pages are larger, the classifiers fail to recognize bad pages, so the Specificity decreases.

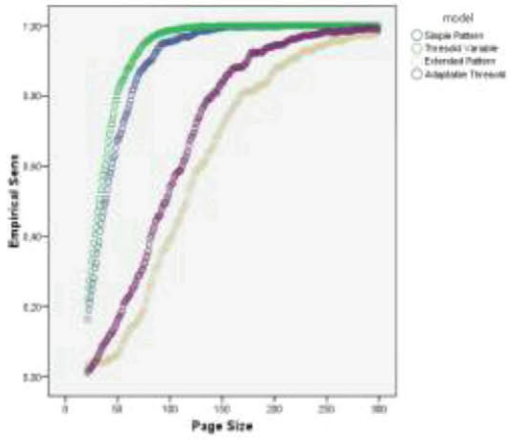

Fig. 8. Sensibility comparison for all Models

The result obtained Model 1 is shown in Fig. 4. It is the simplest approach, being static and using a short set of patterns. It has its best performance with pages of around 90 phrases where sensibility and specificity curves crossover.

We can see in Fig. 5 the results for Model 2, it has a similar pattern set to Model 1, but dynamically adjusted to the size of the page. Its performance is better than Model 1, having linear decrease of its specificity.

Model 3 and Model 4 use a larger set of patterns and both are dynamic but with different thresholds. We can see in Fig. 6 and 7 that their results are better than Models 1 and 2.

In Fig. 8 we can see the comparison of sensibilities of all the models and in Fig. 9 we see the comparison of specificities.

As we have stated, our test environment provides us with a clear comparison between models and how they perform with respect to the feature that we are analyzing. With this information, we can adapt the models and we can determine the best working conditions for them or design a strategy

where models are dynamically selected based on the size of the page.

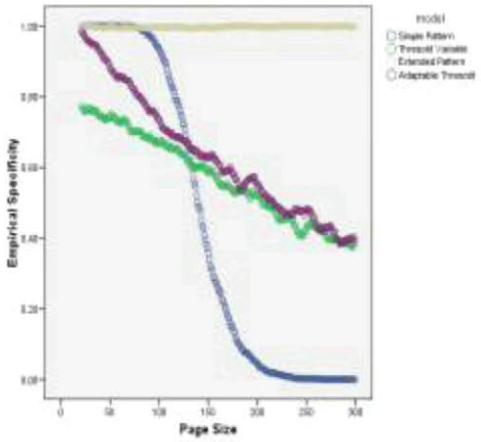

Fig. 9. Specificity comparison for all Models

VII. Conclusions

It is possible to improve the Web Learning process by refining page selection using Bootstrap technique to evaluate, refine and compare models based on patterns implemented for binary classification. A pure empirical sample for training and testing is limited, because the need for human intervention and the difficulty in finding pages with the desired qualities. Bootstrap provides a computer-based methodology that helps to have a wider dataset, where specific page features can be tested to determine how they really affect model performance and its outcome when real data are processed. We go further, by varying the characteristics of the sample and applying Bootstrap for each case analyzing model performance.

In our case, we were able to test how a pattern classifier is affected by the size of the page. We observed that if the page size was too small, the number of false negatives was excessively high and if the page size was big enough, the model performed better. This is an intrinsic problem for pattern models. Bootstrap technique provides excellent support for building dynamic models and their evaluation.

In addition, Bootstrap technique is a powerful tool for all related works with the Internet. It allows creating test environments that can simulate real conditions involving less human effort. Further work can be accomplished including new page and model features on the test environment as well as more advanced statistical techniques related with Bootstrap.

Web Learning improvement is used in the Urbano Robot environment. Information management is a key aspect of the robot software architecture; it allows a higher level of control providing intelligence to all the agents that comprise the system.

References

[1] Bringmann, B., Nijssen, S., & Zimmermann, A. Pattern-Based classification: a Unifying perspective. 2009.ISSN 0718-3305

[2] Stine, R. An Introduction to Bootstrap. Sociological Methods and Research, Vol. 18, Nos. 2&3, November 1989/February 1990 243-291. 1990.ISSN: 1552-8294

[3] Efron, B., &Tibshirani, R. J..Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy. Statistical Science, 1986, Vol. 1, 54-77. 1986.ISSN 1726-3328

[4] Rodriguez-Losada, D., Matia, F., Galán, R. Hernando, M., Montero, J.M., & Lucas, M. .Service Robotics. Urbano, an Iteractive Mobile Tour-Gide Robot. pp. 229-252. I-Tech Education and Publishing Alekasandar Lazinica, Viena, Austria. 2008.ISBN978-3-902613-24-0

[5] Rainer, J.J, Gómez, J., &Galán, R.An automatic Generator of Presentations for Gide-Tour Robot.Mathware&Soft Computing. Vol. 16, No 2. 2009.ISSN 1134-5632

[6] Gruber, T.R..Ontolingua: a mechanism to support portable ontologies, KSL-91-66, Knowledge System Laboratory, Stanford University, Stanford, CA, USA, 1991.

[7] Guarino, N..Understanding building and using ontologies, Int. J. Human–Computer Stud. 46 (2/3) 293–310, 1997. ISSN 1095-9300.

[8] Sowa J.F..Ontology, metadata, and semiotics, Lecture Notes in Artificial Intelligence, Vol. 1867, pp. 56–83 Springer, Berlin, 2000.ISSN 0302-9743.

[9] Hand, D. J., and Yu, Y. . Idiots Bayes - not so stupid after all? International Statistical Review 69:385–389. 2001.

[10] Vapnik, V. The Nature of Statistical Learning Theory. Heidelberg, Germany. Springer.1995. ISBN 0-387-98780-0

[11] MacQueen, J. B.. Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297. 1967.ISSN 0097-0433

[12] Dempster, A. P., Laird, N. M., & Rubin, D. R. . Maximum Likelihood from Incomplete Data via the EM algorithm. Journal of the Royal Satistical Society, Series B, 39 (1). pp. 1-38. 1997. ISSN 1467-985X

[13] Liu, B. . Web Data Mining. Berlin, Heidelberg, Germany. Springer-Verlag. 2007. ISBN10 3-540-37881-2

[14] Porter, M. F. . An Algorithm for Suffix Stripping. Program, 14(3), pp. 130-137. 1980.ISBN:1-55860-454-5

[15] Turney, P..Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proc. of the Meeting of Association for computational Linguistics (ACL 02), pp. 417-424. 2002.DOI 10.3115/1073083.1073153

[16] Dreewester, S., Dumais, S. T., Furnas, G. W., Launderer, T. K., &Harshman. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science. 41, pp. 391-407. 1990.ISSN: 1532-2890

[17] Efron, B., &Tibshirani, R. J..An Introduction to the Bootstrap, Chapman & Hall, Boca Ratón. 1993.ISBN 0-412-04231-2

Rafael Léon is Telecommunications Engineer and PhD in Industrial Engineering with 20 years experience in the IT industry, having worked in multinational companies with long teaching experience as Associate Professor.

J. Javier Rainer (PhD). Received a PhD in Robotics and Automation from the Universidad Politécnica de Madrid, UPM (Spain) in 2011. He is Director of Research and Director of Engineering Area at Bureau Veritas Business School, and researcher at the Intelligent Control Group of the UPM (http://intelligentcontrol.es/). He is Industrial Engineer, from Málaga University. He has been co-author of published papers in several journals and recognized international conferences and symposiums. He received the best paper Award in IARIA Cognitive 2010.

José Manuel Rojo is Permanent Specialized Staff in the CSIC, he is charge of the Statistical Analysis Unit of CCHS data. Bachelor of Science and Technology Statistics, Faculty of Mathematics, Complutense University.

Ramón Galán is Professor in the area of Systems Engineering at the Department of Automatic Control, Electronic Engineering and Industrial Computing of the UPM. His research activity has targeted in the area of process control, artificial intelligence, expert systems and computer science. He has leaded several EU funded projects.