Social News Website Moderation through Semi-supervised Troll User Filtering

Jorge de-la-Peña-Sordo, Igor Santos, Iker Pastor-López, and Pablo G. Bringas

S^{3}Lab, DeustoTech Computing, University of Deusto
Avenida de las Universidades 24, 48007, Bilbao, Spain
{jorge.delapenya, isantos, iker.pastor, pablo.garcia.bringas}@deusto.es

Abstract. Recently, Internet is changing to a more social space in which all users can provide their contributions and opinions to others via websites, social networks or blogs. Accordingly, content generation within social webs has also evolved. Users of social news sites make public links to news stories, so that every user can comment them or other users' comments related to the stories. In these sites, classifying users depending on how they behave, can be useful for web profiling, user moderation, etc. In this paper, we propose a new method for filtering trolling users. To this end, we extract several features from the public users' profiles and from their comments in order to predict whether a user is troll or not. These features are used to train several machine learning techniques. Since the number of users and their comments is very high and the labelling process is laborious, we use a semi-supervised approach known as collective learning to reduce the labelling efforts of supervised approaches. We validate our approach with data from 'Menéame', a popular Spanish social news site, showing that our method can achieve high accuracy rates whilst minimising the labelling task.

Keywords: User Profiling, Content Filtering, Web Mining, User Categorisation, Machine-learning.

1 Introduction

Multimedia content is widely spread in the web, thanks to the technological evolution, specially mobile devices; however, textual content has still a major role in web content. Regardless to the fact that most of the social websites have multimedia files, such video or images, a large section is still occupied by textual content published in ideas, opinions and beliefs. This kind of content can be comments or stories reflecting different users' behaviours. Therefore, in a certain way, the users' dynamic interaction and collaboration has been drastically enhanced.

In particular, social news websites such as Digg[1] or 'Menéame'[2] are very popular among users. These sites work in a very simple and intuitive way: users

[1] http://digg.com/
[2] http://meneame.net/

submit their links to stories online, and other users of these systems rate them by voting. The most voted stories appear, finally, in the front-page [1].

In the context of web categorisation, supervised machine-learning is a common approach. Specifically, in a similar domain as social news, such as filtering spam in reviews, supervised machine-learning techniques have also been applied [2]. However, supervised machine-learning classifiers require a high number of labelled data for each of the classes (i.e., troll user or normal user). Labelling this amount of information is quite arduous for a real-world problem such as web mining. To generate these data, a time-consuming process of analysis is mandatory and, in the process, some data may avoid filtering.

One type of machine-learning technique specially useful in this case, is semi-supervised learning. This technique is appropriate when a limited amount of labelled data exists for each class [3]. In particular, the approach of collective classification [4] employs the relational structure of labelled and unlabelled datasets combination, to increase the accuracy of the classification. With these relational models, the predicted label will be influenced by the labels of related samples. The techniques of collective and semi-supervised learning have been applied satisfactorily in several fields of computer science like text classification [4] or spam filtering [5].

In light of this background, we propose a novel user categorisation approach based on collective classification techniques to optimise the classification performance when filtering controversial users of social news website. This method employs a combination of several features from the public users' profiles and from their comments.

In summary, our main contributions are: (i) a new method to represent users in social news websites, (ii) an adaptation of the collective learning approach to filter troll users and (iii) an empirical validation which shows that our method can maintain high accuracy rates, while minimising the efforts of labelling.

The remainder of this paper is structured as follows. Section 2 describes the extracted features of the users. Section 3 describes the collective algorithms we applied. Section 4 describes the experimental procedure and discusses the obtained results. Finally, Section 5 concludes and outlines the avenues of the future work.

2 Method Description

'Menéame' is a Spanish social news website, in which news and stories are promoted. It was developed in later 2005 by Ricardo Galli and Benjamín Villoslada and it is currently licensed as free software. It ranks users using the 'karma' value whose boundaries are 1 and 20. When a new user is registered, a default value of 6 point of 'karma' is assigned. This value of 'karma' is computed and recomputed based on the activity of the user in the previous 2 days. The algorithm for the computation combines 4 different components: (i) received positive votes regarding the sent news, (ii) the number of positive votes the users made, (iii) negative votes made and (iv) the number of votes their comments have received.

Fig. 1. An example of a user profile in 'Menéame.net'

2.1 Extracted Features

In this sub-section, we describe the features that we extract from the users' profile and their comments. We divide these features into 2 different categories: profile and comment related.

- **Profile Related Features:** Several of the features have been gathered from the public profile of the user. In Figure 1, we can see an example of user profile. The different features used are: (i) from, the registered date; (ii) karma, a number between 1 and 20; (iii) ranking, user position in terms of 'karma'; (iv) the number of news submitted; (v) the number of published news; (vi) entropy, diversity ratio of the posts; (vii) the number of comments; (viii) the number of notes; (ix) number of votes and (x) text avatar.
  Using the new Google Images[3], we have utilised the avatar of the user to translate it into text. The avatar is used as a query of the system and the text of the most similar images is used as representation. For instance, we retrieve the URL of the user's avatar image and paste it as a query in the Google Images service. Hereafter, the service retrieves the most similar images and the most probable query for that image. If Google Images service does not find any possible query for the given image, our system leaves the avatar feature blank. However, there is a high number of users that did not change the default 'Menéame' avatar. To minimise the computing overhead, our system directly employs the string 'Meneame'.
- **Comment Related Features:** We have also analysed the comments using our comment categorisation approach [6] to count the number of comments in each category: (i) focus of the comment (focus on the news story or on other comments); (ii) type of information (contribution, irrelevant or opinion) and (iii) controversy level (normal, controversial, very controversial or joke).

The approach used different syntactic, opinion and statistical features to build a representation of the comments. In particular, the following features were used in order to categorise each comment of a particular user:

[3] `http://images.google.com`

- **Syntactic:** We count the number of words in the different syntactic categories. To this end, we performed a Part-of-Speech tagging using FreeLing[4]. The following features were used, all of them expressed in numerical values extracted from the comment body: (i) adjectives, (ii) numbers, (iii) dates, (iv) adverbs, (v) conjunctions, (vi) pronouns, (vii) punctuation marks, (viii) interjections, (ix) determinants, (x) abbreviations and (xi) verbs.
- **Opinion:** Specifically, we have used the following features: (i) number of positive votes of the comment, (ii) karma of the comment and (iii) number of positive and negative words. We employed an external opinion lexicon[5]. Since the words in that lexicon are in English and 'Menéame' is written in Spanish, we have translated them into Spanish.
- **Statistical:** We used: (i) the information contained in the comment using the Vector Space Model (VSM) [7] approach, which was configured using words or n-grams as tokens; (ii) the number of references to the comment (in-degree); (iii) the number of references from the comment (out-degree); (iv) the number of the comment in the news story; (v) the similarity of the comment with the description of the news story, using the similarity of the VSM of the comment with the model of the description; (vi) the number of coincidences between words in the comment and the tags of the commented news story and (vii) the number of URLs in the comment body.

Therefore, using our approach [6], we have categorised users' comments, generating 9 new features that count the number of comments in these comment categories: (i) referring to news stories, (ii) referring to other comments, (iii) contribution comments, (iv) irrelevant comments, (v) opinion comments, (vi) normal comments, (vii) controversial comments, (viii) very controversial comments and (ix) jokes comments.

3 Collective Classification

Collective classification is a combinatorial optimisation problem, in which we are given a set of users, or nodes, $\mathcal{U} = \{u_1, ..., u_n\}$ and a neighbourhood function N, where $N_i \subseteq \mathcal{U} \setminus \{\mathcal{U}_i\}$, which describes the underlying network structure [8]. Being $\mathcal{U}$ a random collection of users, it is divided into 2 sets $\mathcal{X}$ and $\mathcal{Y}$, where $\mathcal{X}$ corresponds to the users for which we know the correct values and $\mathcal{Y}$ are the users whose values need to be determined. Therefore, the task is to label the nodes $\mathcal{Y}_i \in \mathcal{Y}$ with one of a small number of labels, $\mathcal{L} = \{l_1, ..., l_q\}$. We employ the *Waikato Environment for Knowledge Analysis* (WEKA) [9] and its Semi-Supervised Learning and Collective Classification plugin[6].

- **Collective IBK:** Internally, it applies an implementation of the *K-Nearest Neighbour* (KNN), to determine the best k instances on the training set and

[4] Available in: `http://nlp.lsi.upc.edu/freeling/`
[5] Available in: `http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar`
[6] Available at: `http://www.cms.waikato.ac.nz/~fracpete/projects/collective-classification`

builds then, for all instances from the test set, a neighbourhood consisting of k instances from the training pool and test set (either a naïve search over the complete set of instances or a k-dimensional tree is used to determine neighbours) that are sorted according to their distance to the test instance they belong to. The neighbourhoods are ordered with respect to their 'rank'(the different occurrences of the two classes in the neighbourhood). The class label is determined by majority vote or, in tie case, by the first class for every unlabelled test instance with the highest rank value. This is implemented until no further test instances remain unlabelled. The classification ends by returning the class label of the instance that is about to be classified.

- **CollectiveForest:** The WEKA's implementation of RandomTree is utilised as base classifier to divide the test set into folds containing the same number of elements. The first repetition trains the model using the original training set and generates the distribution for all the instances in the test set. The best instances are then added to the original training set (choosing the same number of instances in a fold). The next repetitions train with the new training set and then produce the distributions for the remaining instances in the test set.
- **CollectiveWoods & CollectiveTree:** CollectiveWoods in association with the algorithm CollectiveTree operates like CollectiveForest by using the RandomTree algorithm. CollectiveTree is similar to WEKA's original version of RandomTree classifier.
- **RandomWoods:** This classifier works like WEKA's classic RandomForest but using CollectiveBagging (classic Bagging, a machine learning ensemble meta-algorithm to improve stability and classification accuracy, extended to make it available to collective classifiers) in combination with CollectiveTree algorithm instead of Bagging and RandomTree algorithms.

4 Empirical Validation

We have retrieved the information of the users' profiles from the users that appear in the downloaded comments [6], generating a combined dataset of 3,359 users' profiles and their comments. Afterwards, we labelled each user into *Normal* or *Controversial*. *Normal* means that the user is not hurtful or hurting, using in its argument a restrained tone. *Controversial* refers to a troll user which seeks to create polemic. To this end, we built a dataset, following the next distribution: 1,997 number of normal users and 1,362 of controversial users.

4.1 Methodology

We developed an application to obtain all the features described in Section 2. We implemented 2 different procedures to construct the VSM of the text avatar body: (i) VSM with words as tokens and (ii) n-grams with different values of n (n=1, n=2 and n=3) as tokens. Furthermore, we removed every word devoid of meaning in the text, called stop-words, (e.g., 'a','the','is') [10]. To this end,

we employed an external stop-word list of Spanish words[7]. Subsequently, we evaluated the precision of our proposed method. To this end, by means of the dataset, we conducted the following methodology:

- **Cross validation:** This method is generally applied in machine-learning evaluation [11]. In our experiments, we utilised $k = 10$ as a K-fold cross validation refers. As a consequence, our dataset is split 10 times into 10 different sets of learning and testing. We changed the number of labelled instances from 10% to 90% for each fold. Performing this operation, we measured the effect of the number of previously labelled instances on the final performance of collective classification.
- **Information Gain Attribute Selection:** For each training set, we extracted the most important features for each of the classification types using *Information Gain* [12], an algorithm that evaluates the relevance of an attribute by measuring the information gain with respect to the class. Using this measure, we removed any features in the training set that had a IG value of zero: (i) word VSM remove 99.01% of the features and (ii) n-gram VSM the 99.42%.
- **Learning the model:** We accomplished this step using different learning algorithms depending on the specific model, for each fold. As discussed above, we employed the implementations of the collective classification provided by the *Semi-Supervised Learning and Collective Classification* package for machine-learning tool WEKA. In our experiments, we used the following models: (i) *Collective IBK*, with k=10; (ii) *CollectiveForest*, with N=100; (iii) *CollectiveWoods*, with N=100 and (iv) *RandomWoods*, with N=100.
- **Testing the model:** We measured the *True Positive Rate* (TPR) to test our procedure; i.e., the number of the controversial users correctly detected divided by the total controversial users: $TPR = TP/(TP + FN)$; where TP is the number of controversial users correctly classified (true positives) and FN is the number of controversial users misclassified as normal users (false negatives). In the other hand, we also took into account the *False Positive Rate* (FPR); i.e., the number of normal users misclassified divided by the total normal users: $FPR = FP/(FP + TN)$; where FP is the number of normal instances incorrectly detected and TN is the number of normal users correctly classified. In addition, we obtained *Accuracy*; i.e., the total number of hits of the classifiers divided by the number of instances in the whole dataset: $Accuracy(\%) = (TP + TN)/(TP + FP + FN + TN)$. Finally, we recovered the *Area Under the ROC Curve* (AUC). This measure establishes the relation between false negatives and false positives [13]. By plotting the TPR against the FPR, we can obtain the ROC curve.

4.2 Results

In our experiments, we examined various configurations of the collective algorithms with different sizes of the $\mathcal{X}$ set of known instances; the latter varied

[7] Available in: `http://paginaspersonales.deusto.es/isantos/resources/stopwords.txt`

Table 1. Results of the Controversy Level for Word VSM.

Dataset	Accuracy (%)	TPR	FPR	AUC
KNN K = 10	89.43 ± 4.93	0.80 ± 0.12	0.04 ± 0.01	0.97 ± 0.02
BN: Bayes K2	82.52 ± 2.00	0.82 ± 0.03	0.17 ± 0.02	0.87 ± 0.02
BN: Bayes TAN	90.47 ± 1.47	0.99 ± 0.02	0.15 ± 0.02	0.96 ± 0.01
Naïve Bayes	67.10 ± 4.33	0.28 ± 0.12	0.06 ± 0.02	0.82 ± 0.03
SVM: Polynomial Kernel	75.50 ± 4.46	0.43 ± 0.12	0.03 ± 0.01	0.70 ± 0.06
SVM: Normalise Polynomial	95.00 ± 1.81	0.93 ± 0.04	0.04 ± 0.01	0.95 ± 0.02
SVM: Pearson VII	95.47 ± 1.31	0.96 ± 0.03	0.05 ± 0.02	0.96 ± 0.01
SVM: Radial Basis Function	59.57 ± 0.31	0.00 ± 0.01	0.00 ± 0.00	0.50 ± 0.00
DT: J48	96.58 ± 0.86	0.97 ± 0.02	0.04 ± 0.01	0.97 ± 0.01
DT: Random Forest N = 100	96.43 ± 0.94	0.97 ± 0.02	0.04 ± 0.01	0.99 ± 0.00

Table 2. Results of the Controversy Level for N-gram VSM

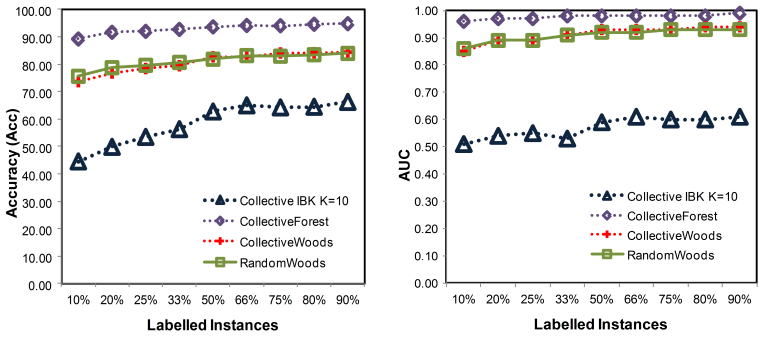
Dataset	Accuracy (%)	TPR	FPR	AUC
KNN K = 10	89.55 ± 4.81	0.80 ± 0.12	0.04 ± 0.01	0.97 ± 0.02
BN: Bayes K2	82.52 ± 2.00	0.82 ± 0.03	0.17 ± 0.02	0.87 ± 0.02
BN: Bayes TAN	90.47 ± 1.47	0.99 ± 0.02	0.15 ± 0.02	0.96 ± 0.01
Naïve Bayes	67.31 ± 4.63	0.27 ± 0.12	0.05 ± 0.02	0.82 ± 0.03
SVM: Polynomial Kernel	75.56 ± 4.43	0.44 ± 0.12	0.03 ± 0.01	0.70 ± 0.06
SVM: Normalise Polynomial	95.09 ± 1.59	0.94 ± 0.04	0.04 ± 0.01	0.95 ± 0.02
SVM: Pearson VII	95.68 ± 1.13	0.96 ± 0.03	0.05 ± 0.02	0.96 ± 0.01
SVM: Radial Basis Function	59.57 ± 0.31	0.00 ± 0.01	0.00 ± 0.00	0.50 ± 0.00
DT: J48	96.58 ± 0.86	0.97 ± 0.02	0.04 ± 0.01	0.97 ± 0.01
DT: Random Forest N = 100	96.49 ± 0.97	0.97 ± 0.02	0.04 ± 0.01	0.99 ± 0.00

from 10% to 90% of the instances utilised for training (i.e., instances known during the test). On the other hand, we compared the filtering capabilities of our method with some of the most used supervised machine-learning algorithms. Specifically, we use the following models:

- *Bayesian networks (BN):* We used different structural learning algorithms: K2 [14], Tree Augmented Naïve (TAN) [15] and Naïve Bayes Classifier [11].
- *Support Vector Machines (SVM):* We performed experiments with different kernels: (i) polynomial [16], (ii) normalised polynomial [17], (iii) Pearson VII function-based (PVK) [18] and (iv) radial basis function (RBF) [19].
- *K-nearest neighbour (KNN):* We launched experiments with $k = 10$.
- *Decision Trees (DT):* We executed experiments with J48 (the *Weka* implementation of the *C4.5* algorithm [20]) and Random Forest [21], an ensemble of randomly constructed decision trees. In particular, we employed $N = 100$.

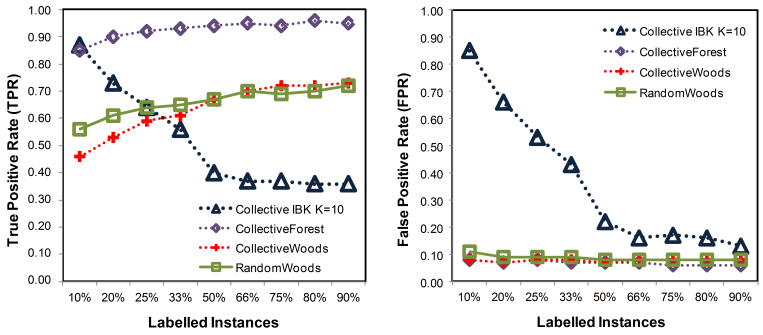
Table 1 shows the results with words as tokens and supervised learning, Table 2 shows the results with n-grams as tokens and supervised learning. These two tables contain the supervised results about both VSM approaches. Figure 2 shows the results with word-based VSM and collective learning, Figure 3 shows the results with VSM generated with n-grams and collective learning.

Regarding the results obtained both word and n-gram VSM applying supervised algorithms and collective classifications, Random Forest with 100 trees using n-grams was the best classifier. It obtained similar results in Accuracy (96.49% and 96.58%), TPR (0.97) and FPR (0.04) than J48, but in AUC terms achieved the highest value: 0.99. CollectiveForest using n-grams as tokens with



(a) **Accuracy results.** CollectiveForest was the best classifier with 90% of labelling and achieving a value of 94.76%.

(b) **AUC results.** CollectiveForest achieved of 0.98, with a 33% of labelling instances.



(c) **TPR results.** The best classifier was CollectiveForest, with an accuracy of 96%, labelling the 80%.

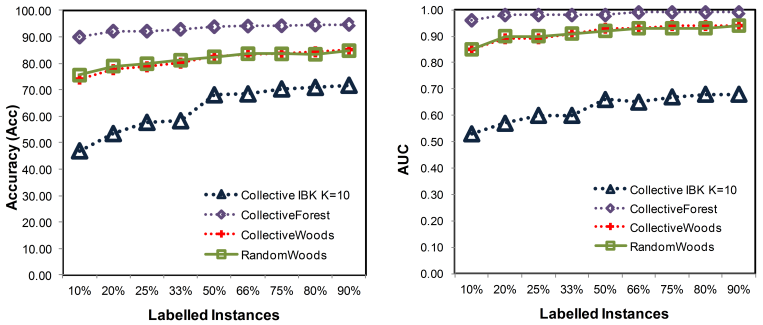
(d) **FPR results.** CollectiveWoods, obtained values close to zero and with 20% labelling effort.

Fig. 2. Results of our collective-classification-based for users categorisation using words as tokens. In resume, *CollectiveForest* was the best classifier.

the 66% of known instances, obtained 94.08% of accuracy, 0.99 of AUC, 0.94 of TPR and 0.06 of FPR. With regards to the use of collective classification, comparing with the supervised approaches, it achieved close results. We can maintain the results of the best supervised learning algorithm whilst the labelling efforts are reduced significantly, in this case a 33% of the dataset.

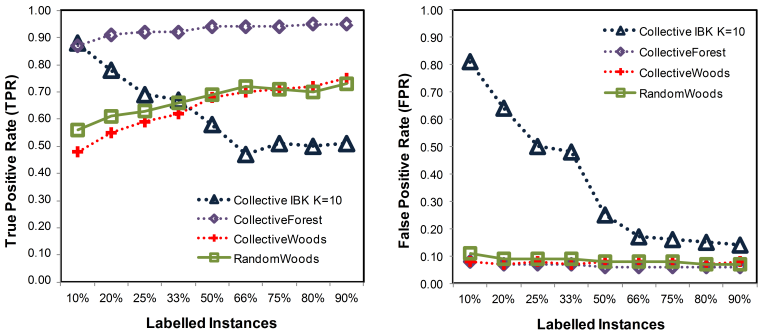
5 Conclusions

One problem about supervised learning algorithms is that a preceding work is required to label the training dataset. When it comes to web mining, this process may originate a high performance overhead because of the number of users that post comments any time and the persons who create new accounts everyday. In this paper, we have proposed the first trolling users filtering method system



(a) **Accuracy results.** CollectiveForest, using a 90% of labelled instances, achieved a 94.70% of accuracy.

(b) **AUC results.** The best algorithm was CollectiveForest. This classifier achieved 0.99, labelling only 66% of the dataset.



(c) **TPR results.** CollectiveForest was the best classifier obtaining 0.95 labelling 80% of the dataset.

(d) **FPR results.** CollectiveForest achieved results close to zero, being the best classifier.

Fig. 3. Results of our collective-classification-based for users categorisation using n-grams as tokens. Summarising, *CollectiveForest* obtained the best results.

based on collective learning. This approach is able to determine when a user is controversial or not, employing users' profile features and statistical, syntactic and opinion features from their comments. We have empirically validated our method using a dataset from 'Menéame', showing that our technique obtains the same accuracy than supervised learning, despite having less labelling requirements.

The avenues of future work are oriented in three main ways. Firstly, we project to extend the study of our semi-supervised classification by applying additional algorithms to the problem of filter trolling users in social news websites. Secondly, incorporating new different extracted features from the user dataset to train the models. And finally, we will focus on executing an extended analysis of the effects of the labelled dataset dimension.

References

1. Lerman, K.: User participation in social media: Digg study. In: Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops, IEEE Computer Society (2007) 255–258
2. Jindal, N., Liu, B.: Review spam detection. In: Proceedings of the 16th international conference on World Wide Web, ACM (2007) 1189–1190
3. Chapelle, O., Schölkopf, B., Zien, A., et al.: Semi-supervised learning. Volume 2. MIT press Cambridge, MA: (2006)
4. Neville, J., Jensen, D.: Collective classification with relational dependency networks. In: Proceedings of the Second International Workshop on Multi-Relational Data Mining. (2003) 77–91
5. Laorden, C., Sanz, B., Santos, I., Galán-García, P., Bringas, P.G.: Collective classification for spam filtering. In: Computational Intelligence in Security for Information Systems. Springer (2011) 1–8
6. Santos, I., de-la Peña-Sordo, J., Pastor-López, I., Galán-García, P., Bringas, P.: Automatic categorisation of comments in social news websites. Expert Systems with Applications (2012)
7. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)
8. Namata, G., Sen, P., Bilgic, M., Getoor, L.: Collective classification for text classification. Text Mining (2009) 51–69
9. Garner, S.: Weka: The waikato environment for knowledge analysis. In: Proceedings of the 1995 New Zealand Computer Science Research Students Conference. (1995) 57–64
10. Salton, G., McGill, M.: Introduction to modern information retrieval. McGraw-Hill New York (1983)
11. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press (1995)
12. Kent, J.: Information gain and a general measure of correlation. Biometrika **70**(1) (1983) 163–173
13. Singh, Y., Kaur, A., Malhotra, R.: Comparative analysis of regression and machine learning methods for predicting fault proneness models. International Journal of Computer Applications in Technology **35**(2) (2009) 183–193
14. Cooper, G.F., Herskovits, E.: A bayesian method for constructing bayesian belief networks from databases. In: Proceedings of the 1991 conference on Uncertainty in artificial intelligence. (1991)
15. Geiger, D., Goldszmidt, M., Provan, G., Langley, P., Smyth, P.: Bayesian network classifiers. In: Machine Learning. (1997) 131–163
16. Amari, S., Wu, S.: Improving support vector machine classifiers by modifying kernel functions. Neural Networks **12**(6) (1999) 783–789
17. Maji, S., Berg, A., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2008) 1–8
18. Üstün, B., Melssen, W., Buydens, L.: Visualisation and interpretation of support vector regression models. Analytica chimica acta **595**(1-2) (2007) 299–309
19. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural computation **3**(2) (1991) 246–257
20. Quinlan, J.: C4. 5 programs for machine learning. Morgan Kaufmann Publishers (1993)
21. Breiman, L.: Random forests. Machine learning **45**(1) (2001) 5–32