Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 102 (2016) 611 – 616

Procedia Computer Science

12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016, 29-30 August 2016, Vienna, Austria

Anticipation of political party voting using artificial intelligence

Zeliha Khashman[a,b], Adnan Khashman[b,c,]*

[a] *International Relations Department, Near East University, ,P.O.BOX:99138,Nicosia,North Cyprus, Mersin 10, Turkey.*
[b] *European Centre for Research and Academic Affairs (ECRAA), Lefkosa, Mersin 10, Turkey .*
[c] *University of Kyrenia, Girne, Mersin 10, Turkey*

Abstract

This paper presents a novel application in the emerging field of computational politics. Here, we anticipate American congressional voting outcome using support vector machines (SVM) and back propagation learning algorithm neural network models. Our proposed method successfully associates opinions of American congress members on defined national issues with their political party affiliation as republican or democrat, thus providing a artificially intelligent anticipation system of the congressional voting outcome based on previous knowledge of how the congress members perceive national issues. Knowledge is obtained from existing congressional records which are unclassified and available online for research purposes. The obtained experimental results suggest that our novel method and application can be further applied to similar voting polls in order to anticipate the party members voting inclination.

Peer-review under responsibility of the Organizing Committee of ICAFS 2016

Keywords: Computational politics; vote prediction; american congress; support vector machine; radial base function; neural networks.

1. Introduction

The congress of the United States of America (USA) has an efficient system that records the votes of congress members on various national issues for posterity purposes. The two major political parties in the USA are the Democrats and the Republicans. Both parties have different political ideologies and policies on various issues. It is anticipated that party members would vote along those ideological lines; thus making it possible to use anonymous

* Corresponding author. Tel.: +90 548 824 2 824.
E-mail address: adnan.khashman@ecraa.com; http://www.ecraa.com/

Peer-review under responsibility of the Organizing Committee of ICAFS 2016
doi:10.1016/j.procs.2016.09.450

opinions or responses of party members to predict their voting outcome and association with a party. A freely available record of the votes in the year 1984 provides the political party affiliation and the responses of 435 anonymous congressmen and congresswomen to 16 different topics[1]. This congressional data record forms the dataset used in our investigation.

The task here is to associate responses of a sample of congress members to their political party vote which, once established, enables us to generalize this association in order to predict the congress political party voting outcome based simply on members' opinions or responses to similar national issues.

Previous research works that address this task are scarce; but few existing works[2-11, 21-22] addressed a similar problem. We believe that there are two main reasons for the scarcity of research works on this particular task: firstly, the application is amongst those emerging topics which combine political science and computational intelligence. Secondly, the problem of finding database records for developing a prediction or association model; and in the case where database is available such as the one we use in this work, often the dataset contains missing data that requires compensation or imputation, thus, complicating the prediction task even further. However, some recent works[10,11] suggested successful solutions to missing data imputation and investigated different artificially intelligent methods for prediction in computational politics.

As an example of the more recent investigations, authors[22] used support vector machines (SVM), to analyze the ideological content of legislative speech in order to provide an additional insight into the content of the political ideology. Their approach uses a group of reference texts to train the classifier, which is then tested on new text not contained in the training corpus. In another recent work[10], the authors suggested a solution for missing data imputation and applied their method to predict voting intention polls. Their classifier is a supervised learner based on a fuzzy min–max neural network model. Their dataset contained a mixture of categorical and numerical attributes and stems from voting opinion polls conducted in Spain prior to the 2004 and 2008 general elections.

Lately[11], four methods for missing data imputation were presented and utilized for training a supervised artificial neural network (ANN) model to predict congress members party affiliations based on information retrieved from the available USA congressional voting records.

In this work, we continue our investigation into the use of intelligent methods in computational politics, and we focus in particular on the use of ANNs and SVMs for such applications. Our hypothesis is that because of the nonlinearity of the relationship between congress members' responses to the questions and their party affiliation; then ANN models can be used to approximate this relationship. Once trained, the neural system may be used to predict other peoples' potential voting outcome (voting polls) to one party or another based on their responses to similar questions.

The motivation of using ANNs has in general been attributed to their capabilities in effectively associating or approximating nonlinear paradigms[12]. Furthermore, the successful use of ANNs in different application fields has been demonstrated in many previous recent works[13-20]. Therefore, in this paper we design, implement and compare the performances of two neural network models based on the back propagation learning algorithm, and a C-SVM model with an RBF kernel. The task of both intelligent models would be to associate and predict the party affiliation (Republican or Democrat) of a congress member based on his/her responses to a number of questions.

The paper is organized as follows: in section 2 we briefly describe the congressional voting records dataset and explain our method for input data and output data coding. Section 3 presents the design of the neural network and the SVM models. In section 4, the experimental results of training and testing the support vector learner and the neural model are presented, and a performance comparison is provided for each computational model. Finally, section 5 concludes this work.

2. Congressional Vote Dataset and Data Coding

This dataset[1] consists of 435 instances (267 Democrats, 168 Republicans) and 16 attributes or questions (see Table 1). The records contain also the party affiliation as Republican or Democrat for each congress member. There are basically nine different types of votes (responses) which are labelled as *yes*, *no* or *missing value* responses. These three labels correspond to the nine types of vote as follows:

- *yes*: voted for, paired for, and announced for,
- *no*: voted against, paired against, and announced against,

- *missing value*: voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known.

In total, there are 6960 values (435 instances x 16 attributes) in the dataset. A total of 288 attributes; or 4.13% of the data, have missing values. A missing value does not mean that the value of the attribute is unknown; it rather means that the value is not a *yes* or a *no*. It could also mean that there was abstaining from voting due to conflict of interest. Table 2 shows the party affiliation (output) and the responses (input) of the first 10 congress members in its original form with "*y*" representing *yes*, "*n*" representing *no*, and "*?*" representing *missing value*. This data cannot of course be fed into the neural network as it is; thus, the need for pre-processing the data, which involves two main tasks: firstly, converting the *yes* and *no* responses to Boolean or binary values, and secondly, imputing or filling in the missing values.

Table 1. Issues or questions and corresponding types of votes: *y*: yes, *n*: no, *?*: missing value.

Issues / Questions	Votes
Handicapped-infants	(y,n,?)
Water-project-cost-sharing	(y,n,?)
Adoption-of-the-budget-resolution	(y,n,?)
Physician-fee-freeze	(y,n,?)
El-Salvador-aid	(y,n,?)
Religious-groups-in-schools	(y,n,?)
Anti-satellite-test-ban	(y,n,?)
Aid-to-Nicaraguan-contras	(y,n,?)
MX-missile	(y,n,?)
Immigration	(y,n,?)
Synfuels-corporation-cutback	(y,n,?)
Education-spending	(y,n,?)
Superfund-right-to-sue	(y,n,?)
Crime	(y,n,?)
Duty-free-exports	(y,n,?)
Export-administration-act-South-Africa	(y,n,?)

Table 2. Original input attributes for the first 10 cases.

Input attributes (responses to questions)																Output
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Vote
n	y	n	y	y	y	n	n	n	y	?	y	y	y	n	y	Re.
n	y	n	y	y	y	n	n	n	n	n	y	y	y	n	?	Re.
?	y	y	?	y	y	n	n	n	n	y	n	y	y	n	n	De.
n	y	y	n	?	y	n	n	n	n	y	n	y	y	n	y	De.
y	y	y	n	y	y	n	n	n	n	y	?	y	y	y	y	De.
n	y	y	n	y	y	n	n	n	n	n	n	y	y	y	y	De.
n	y	n	y	y	y	n	n	n	n	n	n	?	?	y	y	De.
n	y	n	y	y	y	n	n	n	n	n	n	y	y	?	y	Re.
n	y	n	y	y	y	n	n	n	n	n	y	y	y	n	y	Re.
y	y	y	n	n	n	y	y	y	n	n	n	n	n	?	?	De.

y: Yes, n: No, ?: Missing value, Re.: Republican De.: Democrat

Table 3. Final input attributes for the first 10 cases of imputed *Dataset I*.

Dataset I: Coded and imputed input attributes (responses to questions)																Coded Output
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
0	1	0	1	1	1	0	0	0	1	1	1	1	1	0	1	1 0
0	1	0	1	1	1	0	0	0	0	0	1	1	1	0	1	1 0
1	1	1	1	1	1	0	0	0	0	1	0	1	1	0	0	0 1
0	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	0 1
1	1	1	0	1	1	0	0	0	0	1	1	1	1	1	1	0 1
0	1	1	0	1	1	0	0	0	0	0	0	1	1	1	1	0 1
0	1	0	1	1	1	0	0	0	0	0	0	1	1	1	1	0 1
0	1	0	1	1	1	0	0	0	0	0	0	1	1	1	1	1 0
0	1	0	1	1	1	0	0	0	0	0	1	1	1	0	1	1 0
1	1	1	0	0	0	1	1	1	0	0	0	0	0	1	1	0 1

The first task is performed using a simple binary representation scheme where *yes* votes are represented by binary value '1' and *no* votes are represented by binary value '0' for all the 435 observations. A similar scheme is also used to represent the output classes with the *Democrats* being represented by a two-digit binary value '0 1' and the *Republicans* by '1 0'.

The second task is data imputation which involves replacing the missing values with either *yes* or *no* votes. In a previous work[11], we investigated four methods for data imputation which resulted in four different imputed congressional vote datasets. It was then concluded that the arbitration of a neural model when using *Dataset I;* where all missing values are replaced by binary values of '1', yielded the highest correct prediction and accuracy rates. Therefore, it will be used in this work as our imputed congressional vote training and testing dataset for the ANN and SVM neural models. Table 3 lists the processed input attributes of *Dataset I* showing the first 10 cases.

3. Arbitration of the Neural Anticipation Models

Our artificially intelligent anticipation system is implemented using two neural models; namely, back propagation learning algorithm and SVM.

The back propagation learning algorithm uses a supervised multilayer neural network and is known for its simplicity and efficiency when implemented using large databases. For our task in this paper, this neural model has one input layer with 16 neurons corresponding to the 16 input attributes (responses to questions) in the dataset. There is one hidden layer with 10 hidden neurons; which was determined after several experiments using different values of hidden neurons. There is also one output layer with two output neurons which deliver the affiliation of the congress members as *Republican* (1 0) or *Democrat* (0 1).

The ANN model was implemented using the MATLAB software environment. For training purposes, a maximum number of iterations was set to a value of 4000, while the minimum required error value was set to 0.005. During the experiments several values of learning coefficient and momentum rate were also altered, and the final training parameters of the ANN were recorded when the neural network converged; i.e. achieved the required minimum error or the maximum number of iterations.

For SVM learning, the C-SVM model with an RBF kernel was used. In order to search for suitable parameters (C and γ) for the RBF kernel, a parameter search was performed using cross validation; specifically the v-fold cross validation method. The cross-validation procedure is a technique used to avoid the over fitting problem. In v-fold cross-validation, we first divide the training set into v subsets of equal size. Sequentially one subset is tested using the SVM classifier trained on the remaining ($v - 1$) subsets. Cross-validation accuracy is the percentage of data which are correctly classified.

The parameters which produce the best cross validation accuracy are saved and then used to train the SVM learner. The saved model is then used on the out-of-sample data (testing set). In this work v=5. The parameter search range for C was conducted from ($2^{-10} - 2^{10}$) while for γ it was ($2^{-15} - 2^{15}$). The best obtained C and γ are the values used to train the SVM learner for the dataset.

4. Experimental Results and Discussion

This section presents and discusses the obtained results when implementing both the SVM and the Back Propagation ANN models using the imputed congressional vote dataset. Tables 4 and 5 list in details the obtained results and optimized final parameters of the trained SVM and ANN models, respectively.

In this work 217 instances were used for training the ANN and SVM models while 218 instances were used for testing or validating the trained models, thus, the training-to-validation ratio was 49.88%:50.12%, which is considered as non-biased due to almost equal distribution of training and testing data.

The performance evaluation of both neural models is based on the *correct anticipation rate* (CAR); i.e. the capability of the neural model to anticipate the responses of the congress members with their political party affiliation. The obtained CAR values (see Table 6) include those obtained using training instances, testing or validation instances and the overall rates which are calculated by averaging training and testing rates. The highest obtained overall-CAR value was 96.33% using the ANN model. This highly successful result was achieved with a fast anticipation decision time of 2.4×10^{-8} seconds. The SVM model performed also well, albeit slower than the ANN model at 2.0×10^{-4} decision time and with a good overall-CAR value of 95.29%. In all cases, the obtained overall correct anticipation rate (CAR) values were higher than those obtained in previous parallel works[2,9,11,22].

Table 4. SVM neural model final parameters

No. of features	Number of classes	SVM Type	Kernel Type	C parameter search range	γ parameter search range	C	γ	ν
16	2	C-SVM	RBF	2-10 - 210	2-15 - 215	1024	0.000488281	5

Table 5. Back propagation neural model final parameters

Input neurons	Hidden neurons	Output neurons	Learning rate	Momentum rate	Minimum error	Obtained error	Iterations	Training time(s)[1]
16	10	2	0.00845	0.61	0.005	0.004991	196	28.17

[1] Using a 2.2 GHz PC with 2 GB of RAM, Windows XP OS and MATLAB.

Table 6. Experimental result of neural models implementation for correctly anticipating congress voting results.

	Testing time(s)[1]	TR-CAR[2]	TS-CAR[3]	Overall CAR[4]
SVM	2.0×10^{-4}	95.63%	94.95%	95.29%
Back Propagation ANN	2.4×10^{-8}	98.62%	94.04%	96.33%

[1] Using a 2.2 GHz PC with 2 GB of RAM, Windows XP OS and MATLAB, [2] TR-CAR: training set correct association rate, [3] TS-CAR: testing set correct association rate, [4] CAR: correct association rate.

5. Conclusions

This paper presented a novel and successful application of artificially intelligent neural models in the emerging field of computational politics. The task of a neural model for such ‘social science’ applications would be more like association, anticipation or prediction. While this emerging field of application is still in its fancy, we believe that many more research works would be exploring this area of application within the next decade. The work in this paper demonstrated the potential of effectively using artificial intelligence based on neural network arbitration in such emerging applications.

The paper investigated the use of a simple back propagation neural network (ANN) and a support vector machine (SVM) model to associate American congress members’ responses (or views) on certain national issues with the members’ political party affiliations; i.e. whether they would vote Republican or Democrat. A potential real life application of a trained neural model would be to predict the voting outcome of a larger sample of voters based on their responses or views on national issues. We consider this type of association as nonlinear, thus in our hypothesis, it is suitable to simulate using artificial neural network models.

The training and implementation of both neural models in this work was based on using a freely available congressional record[1] of the votes in the year 1984. This record provides the political party affiliation and the responses of 435 anonymous congressmen and congresswomen to 16 different topics. However, the available online dataset has a problem of missing values which amount to 4.13% of the total values. As a solution to this problem, we adopted an imputed dataset with compensated missing values that we published in a previous work[11].

In conclusion, we suggest that the applications of neural network models have no boundaries, and we anticipate more emerging application fields to emerge. The work in this paper demonstrated one such application with a high degree of success. The back propagation learning algorithm ANN model which we used in this work, is chosen as the superior model for this particular application; as its obtained correct anticipation rate as well as its decision time were higher than those of the SVM model under same circumstances. This suggests we could successfully implenet such trained models for anticipate political polls on many national issues in any country. Future work will focus on exploring recently emerging neural models with applications in computational politics.

References

1. UCI Machine Learning Repository. Irvine: CA. *University of California*, School of Information and Computer Science. Retrieved April,
2. 2016 from the World Wide Web: https://archive.ics.uci.edu/ml/index.html.
3. Hadzikadic M, Bohren BF. Learning to predict: INC2.5. *IEEE Trans. on Knowledge and Data Engineering*.1997; 9(1): 168–173.
4. Eisinga R, Franses PH, van Dijk D. Timing of vote decision in first and second order Dutch elections 1978–1995: Evidence from artificial neural networks. *Political Analysis*. 1998;7(1): 117-142.
5. Monterola C, Lim M, Garcia J, Saloma C. Feasibility of a neural network as classifier of undecided respondents in a public opinion survey. *International Journal of Public Opinion Research*. 2002; 14(2): 222-229.
6. Hruschka ER, Hruschka ER, Ebecken NFF. Bayesian networks for imputation in classification problems. *Journal of Intelligent Information Systems* 2007; 29(3):231-252.
7. Tasoulis DK, Vrahatis MN. Generalizing the k-windows clustering algorithm in metric spaces. *Mathematical and Computer Modelling* 2007; 46(1-2): 268–277.
8. Jill GS. Election result forecasting using two layer perceptron network. *Journal of Theoretical and Applied Information Technology* 2008; 4(11): 1019-1024.
9. Yu B, Kaufmann S, Diermeier D. Classifying party affiliation from political speech. *Journal of Information Technology & Politics* 2008;5(1): 33-48.
10. Rey-del-Castillo P, Cardenosa J. Categorical missing data imputation using fuzzy neural networks with categorical and numerical inputs. *World Academy of Sciences Engineering and Technology* 2009;**55**: 622-629.
11. Rey-del-Castillo P, Cardenosa J. Fuzzy min–max neural networks for categorical data application to missing data imputation. *Neural Computing & Applications* 2012; 21(6):1349-1362.
12. Khashman A, Nwulu N.I. Missing data imputation and neural network application to computational politics. In: *Proceedings of the 10th IEEE Region 8 AFRICON International Symposium* 2011. Livingstone: Zambia.
13. Khashman A. Modeling cognitive and emotional processes:A novel neural network architecture. *Neural Networks*. 2010; 23(10): 1155-1163.
14. Khashman A, Dimililer K. Comparison criteria for optimum image compression. *Proceedings of the IEEE International Conference on Computer as a Tool* EUROCON 2005; Serbia & Montenegro.
15. Khashman A, IBCIS: Intelligent blood cell identification system. *Progress in Natural Science*. 2008; 18(10): 1309-1314.
16. Khashman A. Application of an emotional neural network to facial recognition. *Neural Computing and Applications*. 2009; 18(4): 309-320.
17. Khashman A. Blood Cell Identification using Emotional Neural Networks. J. Inf. Sci. Eng. 2009; 25(6): 1737-1751.
18. Khashman A, Sekeroglu B. A novel thresholding method for text separation and document enhancement. *Proceedings of the 11th Panhellenic Conference on Informatics.*PCI 2007; Patras, Greece.
19. Khashman A. Face recognition using neural networks and pattern averaging. *Advances in Neural Networks;* 2006: 98-103.
20. Khashman A, Dimililer K. Neural networks arbitration for optimum DCT image compression. *Proceedings of the IEEE International Conference on Computer as a Tool.* EUROCON 2007; Warsaw, Poland.
21. Khashman A, Nwulu NI. Support vector machines versus back propagation algorithm for oil price prediction. *Advances in Neural Networks* 2011: 530-538.
22. Borisyuk R, Borisyuk G, Rallings C, Thrasher M. Forecasting the 2005 general election: A neural network approach. *British Journal of Politics and International Relations* 2005; 7(2): 199-209.
23. Diermeier D, Godbout JF, Yu B, Kaufmann S. Language and ideology in congress. *British Journal of Political Science*. 2012; 42(1): 31-55.