Automatically Generated Classifiers for Opinion Mining with Different Term Weighting Schemes

Shakhnaz Akhmedova[1], Eugene Semenkin[1] and Roman Sergienko[2]
[1]*Institute of Computer Science and Telecommunications, Siberian State Aerospace University, Krasnoyarsk, Russia*
[2]*Institute of Communications Engineering, Ulm University, Ulm, Germany*

Keywords: Opinion Mining, Support Vector Machine, Neural Networks, Bio-Inspired Algorithms.

Abstract: Automatically generated classifiers using different term weighting schemes for Opinion Mining are presented. New collective nature-inspired self-tuning meta-heuristic for solving unconstrained and constrained real- and binary-parameter optimization problems called Co-Operation of Biology Related Algorithms was developed and used for classifiers design. Three Opinion Mining problems from DEFT'07 competition were solved by proposed classifiers. Also different weighting schemes were used for data processing. Obtained results were compared between themselves and with results obtained by methods which were proposed by other researchers. As the result workability and usefulness of designed classifiers were established and best data processing approach for them was found.

1 INTRODUCTION

Opinion Mining (also called sentiment analysis) is the process of determining the attitude of a speaker or a writer with respect to some topic or the overall context of a document (Pang and Lee, 2008). The attitude may be some person's judgment or evaluation, affective or emotional state, or the intended emotional communication.

Most opinion mining algorithms use simple terms to express sentiment about a product or service (for example, a "positive" or "negative" review). However, cultural factors, linguistic nuances and differing contexts make it extremely difficult to turn a string of written text into a simple pro or con sentiment.

Nowadays various techniques are developed for solving opinion mining problems, for example latent semantic analysis, "bag of words" and other machine learning algorithms (Pang, Lee and Vaithyanathan, 2002). But it is well known that the way that documents are represented influences on the performance of the text classification (opinion mining problems can also be considered as text categorization problems) algorithms (Youngjoong Ko, 2012).

In this work artificial neural networks and support vector machines automatically generated by collective bionic algorithm and its modifications are decribed. They were used with eleven different text preprocessing techniques for solving three opinion mining problems which were taken from the DEFFT'07 competition. The best combinations (text preprocessing for problems) for classifiers were found.

In the second section proposed algorithms are described. Term weighting schemes are shown in the third section. Next experimntal results are presented and some conclusions are given.

2 AUTOMATICALLY GENERATED CLASSIFIERS

2.1 Co-Operation of Biology Related Algorithms

A new collective bionic method called Co-Operation of Biology Related Algorithms (COBRA) was introduced in (Akhmedova and Semenkin, 2013). It's based on the co-operative work of five well known heuristics: Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), Wolf Pack Search (WPS) (Yang, Tu and Chen, 2007), Firefly Algorithm (FFA) (Yang, 2009), Cuckoo Search Algorithm (CSA) (Yang and Deb, 2009) and Bat Algorithm (BA) (Yang, 2010).

Akhmedova S., Semenkin E. and Sergienko R..
Automatically Generated Classifiers for Opinion Mining with Different Term Weighting Schemes.
DOI: 10.5220/0005148508450850
In *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics* (ASAAHMI-2014), pages 845-850
ISBN: 978-989-758-040-6

The proposed approach is a self-tuning algorithm: the population size for each method can increase or decrease during the program run. Besides, populations communicate with each other: they exchange individuals after a given number of objective function evaluations.

The proposed heuristic's workability and usefulness were established by its testing on various optimization problems (Liang et al., 2012). Also it was established that COBRA outperforms its component-algorithms and exhibits a competitive level of performance compared to alternative optimization techniques.

The COBRA heuristic was originally developed for solving real-parameter unconstrained optimization problems. Later it was modified for solving not only real but also binary-parameter constrained and unconstrained optimization problems.

Using the technique described in (Kennedy and Eberhart, 1997), the COBRA binary modification (COBRA-b) was developed. COBRA was adapted to search in binary spaces by applying a sigmoid transformation which also was taken from (Kennedy and Eberhart, 1997) to the velocity (PSO, BA) and coordinates (FFA, CSA, WPS) to compress them into a range [0, 1] and force the component values of the individual position to be 0's or 1's.

Then COBRA's modification for solving constrained optimization problems was also developed. Three constraints handling methods were used for this purpose: dynamic penalties (Eiben and Smith, 2003), Deb's rule (Deb, 2000) and the technique described in (Liang, Shang and Li, 2010). Method proposed in (Liang, Shang and Li, 2010) was implemented to PSO-component of COBRA; at the same time other components were modified by implementing Deb's rule followed by calculating function values using dynamic penalties. The new algorithm was called COBRA-c (Akhmedova and Semenkin, 2013).

The performance of both modifications was evaluated with a set of various test functions. For example, 18 scalable benchmark functions provided by the CEC 2010 competition and a special session on single objective constrained real-parameter optimization (Mallipeddi, Suganthan, 2009) were used for heuristic COBRA-c testing. The constrained modification of COBRA was compared with algorithms that took part in the CEC'2010 competition. It was established that COBRA-b and COBRA-c work successfully and are sufficiently reliable. Besides, the proposed approach for constrained optimization problems is superior to 3-4 of the 14 winning methods from this competition. And finally, COBRA's modifications outperform all of its component algorithms.

2.2 ANN-Based Classifiers

The feed-forward artificial neural network (ANN) models have three primary components: the input data layer, the hidden layer(s) and the output layer. Each of these layers contains nodes and these nodes are connected to nodes at adjacent layer(s). Also each node has its own activation function. So the number of hidden layers, the number of nodes (neurons), and the type of activation function on each node will be denoted as "ANN's structure". Before solving a classification problem by neural network its structure should be chosen.

Besides, nodes in network are interconnected and each connection has a weight coefficient; the number of these coefficients depends on the solving problem (number of inputs) and the number of hidden layers and nodes. Thus, networks with a more or less complex structure usually have many weight coefficients which should be adjusted.

The neural networks' structure design and the tuning of their weight coefficients are considered as the solving two unconstrained optimization problems: the first one with binary variables and the second one with real-valued variables. The type of variables depends on the representation of the ANN's structure and coefficients.

For this work we set a maximum number of hidden layers and a maximum number of neurons on each hidden layer (both equal to 5), so that the maximum number of hidden neurons is equal to 25. We could have chosen a larger number of layers and nodes, but our aim was to show that an even network with a relatively small structure can show good results if it is tuned with effective optimization techniques.

Each node was represented by a binary string of length 4. If the string consisted only of zeros ("0000") then this node wouldn't exist in ANN. So, the whole structure of the neural network was represented by a binary string of length 100 (25x4), and each 20 variables represented one hidden layer. The number of input layers depended on the problem in hand.

Also we used 15 different activation functions for all nodes, for example, sigmoidal function, linear function, hyperbolic tangent function and so on. For determining which activation function would be used on a given node, the integer corresponding to its binary string was calculated. Namely, if some

neuron had the binary string "0110", then the integer was calculated in the following way: $0\times2^0 + 1\times2^1 + 1\times2^2 + 0\times2^3 = 6$. So for this neuron we used the sixth activation function from the 15 mentioned above.

Thus we used the optimization method COBRA-b for finding the best ANN's structure and the optimization method COBRA for the weight coefficients adjustment of every structure.

2.3 SVM-based Classifiers

Support vector machines are linear classification mechanisms, which represent examples from a training set as points in space mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible (Vapnik and Chervonenkis, 1974). New examples (from a test set) are then mapped into the same space and predicted to belong to a category based on which side of the gap they fall. So, SVM-based classifiers linearly divide examples from different classes.

SVM is based on the maximization of the distance between the discriminating hyper-plane and the closest examples. This maximization reduces the so-called structural risk, which is related to the quality of the decision function. The most discriminating hyper-plane can be computed by solving constrained the real-parameter optimization problem described in (Vapnik and Chervonenkis, 1974).

However, instances from different categories are not always linearly separable, and in this case SVM (as a linear classifier) does not provide satisfying classification results. One way to resolve this problem is to map the data onto a higher dimension space and then to use a linear classifier in that space. The general idea is to map the original feature space to a higher-dimensional feature space where the training set is linearly separable. SVM provides an easy and efficient way of performing this mapping to a higher dimensional space, which is referred to as the kernel trick (Boser, Guyon and Vapnik, 1992).

In this study we use COBRA and COBRA-c to automatically design appropriate SVM-based classifiers for the problem in hand.

3 TERM WEIGHTING SCHEMES

Generally, text documents are not classified as sequences of symbols. They are usually transformed into vector representation in so-called feature space because most machine learning algorithms are designed for vector space models. The document mapping into the feature space remains a complex non-trivial task.

Text pre-processing techniques can be considered as term weighting schemes: calculating the weight of each word. The term weighting methods can be roughly divided into two groups: supervised and unsupervised methods. Almost all of them use the frequency of the term occurring.

Two of the most popular text pre-processing approaches among researchers are the TF-IDF technique (Salton and Buckley, 1988) and the ConfWeight method (Soucy and Mineau, 2005). In this study we used the above schemes and also the term weighting method proposed in (Gasanova et al., 2013) called C-values. It has some similarities with the ConfWeight method, but has improved computational efficiency: no morphological or stop-word filtering before text pre-processing was used. It means that the text pre-processing can be performed without a human expert or linguistic knowledge and that the text pre-processing technique doesn't depend on the language. All term weighting schemes (binary pre-processing, 8 TF-IDF techniques, ConfWeights, C-values) which were used are briefly described in (Gasanova et al., 2014).

Thus we tried to determine which of the term weighting schemes mentioned is the most useful for solving Opinion Mining problems.

4 EXPERIMENTAL RESULTS

The DEFT07 ("Défi Fouille de Texte") Evaluation Package (Plate-forme AFIA 2007) has been used for the application of algorithms and the comparison of results. For the testing of the proposed approach three corpora were used: "Books", "Video games" (later just "Games") and "Debates in Parliament" (later just "Debates"). Descriptions of corpora are given in Table 1.

These corpora are divided into train (60%) and test (40%) sets by the organizers of the DEFT'07 competition and this partition has been retained in our study to be able to directly compare the performance achieved using the methods developed in this work with the algorithms of participants. The train and test set parameters of all corpora are shown in Table 2.

In order to apply the classification algorithms, all words which appear in the train set have been extracted. Then words have been brought to the same letter case: dots, commas and other punctuation marks have been removed. It should be

mentioned that no other information related to the language or domain (no stop or ignore word lists) has been used in the pre-processing.

Table 1: Test corpora.

Corpus	Description	Marking scale
Books	3000 commentaries about books, films and shows	0:unfavorable, 1:neutral, 2:favourable
Games	4000 commentaries about video games	0:unfavorable, 1:neutral, 2:favourable
Debates	28800 interventions by Representatives in the French Assembly	0:against the proposed law, 1:for it

Table 2: Corpora sizes.

Corpus	Data set sizes	Classes (train set)	Classes (test set)
Books	Train size = 2074 Test size = 1386 Vocabulary = 52507	0:309, 1:615, 2: 1150	0:207, 1:411, 2: 768
Games	Train size = 2537 Test size = 1694 Vocabulary = 63144	0:497, 1:1166, 2:874	0:332, 1:583, 2:779
Debates	Train size = 17299 Test size = 11533 Vocabulary = 59615	0:10400, 1:6899	0:6572, 1:4961

The F-score value (Van Rijsbergen, 1979) was used for evaluating the obtained results. The F-score depends on the "precision" and "recall" of each criterion. The classification "precision" for each class is calculated as the number of correctly classified instances for a given class divided by the number of all instances which the algorithm has assigned for this class. "Recall" is the number of correctly classified instances for a given class divided by the number of instances that should have been in this class.

From the viewpoint of optimization, the design of ANN-based classifiers for the mentioned corpora requires solving optimisation problems having from 115 to 120 real-valued variables for ANN weight coefficients and 100 binary variables for the ANN structure. For example, here is the structure of the neural network with the best obtained result using the C-values pre-processing scheme for the problem "Books" which has five hidden layers, three neurons on the first layer, four neurons on the third layer and five neurons on the other layers:

- The first layer is (0000 0000 0011 1100 1100), i.e. neurons with the 3rd and 12th activation functions;
- The second layer is (0001 0111 1100 0111 1111), i.e., neurons with the 1st, 7th, 12th, and 15th activation functions;
- The third layer is (1011 0111 1110 1111 0000), i.e., neurons with the 11th, 7th, 14th, and 15th activation functions
- The fourth layer is (0001 1001 0100 1101 1111), i.e., neurons with the 1st, 9th, 4th, 13th, and 15th activation functions;
- The fifth layer is (0011 0110 1011 0101 1110), i.e., neurons with the 3rd, 6th, 11th, 5th and 15th activation functions.

The results for all text categorization problems obtained by generated SVM-based and ANN-based classifiers with different term weighting schemes are presented in Tables 3, 4 and 5.

Table 3: Results obtained for corpus "Books".

Books	SVM	ANN
C-values	0.619	0.585137
ConfWeight	0.588023	0.613048
Binary_SUM	0.558442	0.566378
TF-IDF_1	0.580087	0.554113
TF-IDF_2	0.563492	0.533189
TF-IDF_3_0.1	0.577201	0.518038
TF-IDF_3_0.5	0.576479	0.460317
TF-IDF_3_0.9	0.55772	0.505051
TF-IDF_4_0.1	0.549784	0.553719
TF-IDF_4_0.5	0.559163	0.550767
TF-IDF_4_0.9	0.561328	0.541913

Table 4: Results obtained for corpus "Games".

Games	SVM	ANN
C-values	0.695772	0.691919
ConfWeight	0.645218	0.726551
Binary_SUM	0.681818	0.65368
TF-IDF_1	0.668831	0.642136
TF-IDF_2	0.661157	0.649351
TF-IDF_3_0.1	0.686541	0.678932
TF-IDF_3_0.5	0.65987	0.643579
TF-IDF_3_0.9	0.603896	0.627706
TF-IDF_4_0.1	0.691263	0.701299
TF-IDF_4_0.5	0.645218	0.678932
TF-IDF_4_0.9	0.657096	0.551948

So the best classification quality for the problem "Books" is provided with the C-values approach for text pre-processing and Support Vector Machine generated with COBRA as a classification method.

This result is better than the one obtained by the DEFT'07 participants although no term filtering has been used in the text pre-processing. The second best result for the "Books" corpora was obtained by Artificial Neural Network generated by COBRA but with the ConfWeight term weighting scheme.

Table 5: Results obtained for corpus "Debates".

Debates	SVM	ANN
C-values	0.699905	0.704587
ConfWeight	0.714211	0.709269
Binary_SUM	0.641984	0.6471
TF-IDF_1	0.638429	0.640336
TF-IDF_2	0.64129	0.640683
TF-IDF_3_0.1	0.661406	0.645712
TF-IDF_3_0.5	0.668659	0.611983
TF-IDF_3_0.9	0.669323	0.6436
TF-IDF_4_0.1	0.663314	0.643284
TF-IDF_4_0.5	0.665135	0.601231
TF-IDF_4_0.9	0.660827	0.566375

For the problem "Games" ANN-based classifiers showed better results than Support Vector Machines. And again the best result achieved by SVM-based classifiers was provided with the C-values pre-processing technique, while the best result achieved by ANN-based classifiers was with the ConfWeight approach.

There is no significant difference between the results obtained by neural networks with the C-values term weighting scheme and the ConfWeight method for the problem "Debates". However ANN-based classifiers showed the best results with the text pre-processing technique ConfWeight for this corpus. Support Vector Machines were more successful while solving the given problem. But that time the best result was achieved by SVM-based classifiers also with the ConfWeight approach.

It was established that generally the best term relevance scheme for SVM-based classifiers is the C-values technique and the ConfWeight method was more useful for neural networks.

Table 6 contains the best results obtained with SVM-based and ANN-based classifiers automatically generated by developed bionic algorithms. There are also results obtained for the best submission of other researchers for each corpus. The results for each corpus were ranked and then the total rank was evaluated as a mean value. So the best results were obtained by the method with the smallest total rank, and vice versa, and the worst results were obtained by the method with the largest total rank value.

It should be noted that in Table 6 only the best combinations of text pre-processing methods for our algorithms are presented. Thus the proposed classification methods outperformed almost all alternative approaches. In (Gasanova et al., 2014) results obtained by *k*-nearest neighbours algorithm (*k* varied from 1 to 15) were presented. The best F-score achieved by k-NN classifiers (with *k* equal to 15) for corpus "Books" was equal to 0.5593 and for corpus "Debates" it was equal to 0.695. But for problem "Games" k-NN algorithm outperformed SVM-based classifiers generated by COBRA, its result was 0.7203.

Table 6: Results obtained for all corpora, comparison of the performance.

Team or method	Books (rank1)	Games (rank2)	Debates (rank3)	Rank
J.-M. Torres-Moreno (LIA)	0.603 (3)	0.784 (1)	0.720 (1)	1
G. Denhiere (EPHE)	0.599 (4)	0.699 (6)	0.681 (6)	4
S. Maurel (CELI France)	0.519 (8)	0.706 (5)	0.697 (5)	5
M. Vernier (GREYC)	0.577 (5)	0.761 (3)	0.673 (7)	6
E. Crestan (Yahoo ! Inc.)	0.529 (7)	0.673 (8)	0.703 (4)	7
M. Plantie (LGI2P)	0.472 (10)	0.783 (2)	0.671 (9)	8
A.-P. Trinh (LIP6)	0.542 (6)	0.659 (9)	0.676 (8)	9
M. Genereux (NLTG)	0.464 (11)	0.626 (10)	0.569 (12)	11
E. Charton (LIA)	0.504 (9)	0.619 (11)	0.616 (10)	10
A. Acosta (Lattice)	0.392 (12)	0.536 (12)	0.582 (11)	12
SVM+COBRA	0.619 (1)	0.696 (7)	0.714 (2)	3
ANN+COBRA	0.613 (2)	0.727 (4)	0.709 (3)	2

5 CONCLUSIONS

In this paper we have described a new meta-heuristic, called Co-Operation of Biology Related Algorithms, and introduced its modification for solving unconstrained optimization problems with binary variables (COBRA-b) and constrained optimization problems with real-valued variables (COBRA-c). The proposed algorithms' usefulness and workability were established by their testing on sets of benchmark problems.

Then we used described optimization methods for the automated design of ANN-based and SVM-based classifiers. These approaches were applied to three Opinion Mining problems which were taken from the DEFT'07 competition. For that purpose different text pre-processing techniques were used.

Solving these problems is equivalent to solving big and hard optimization problems where objective functions have many variables and are given in the form of a computational program. The suggested algorithms successfully solved all the problems of designing classifiers with competitive performance.

A comparison with alternative classification methods showed that SVM-based and ANN-based classifiers designed by COBRA outperformed almost all of them. This fact allows us to consider the study results as confirmation of the reliability, workability and usefulness of the algorithms in solving real world optimization problems.

Having these appropriate tools for data mining, we consider the following directions for the approach development: the design of other types of neural network models, the design of Support Vector Machines with alternative kinds of kernel function, the application to the design of fuzzy systems, the improvement in optimization performance of developed algorithms COBRA, COBRA-b and COBRA-c.

ACKNOWLEDGEMENTS

Research is performed with the financial support of the Ministry of Education and Science of the Russian Federation within the federal R&D programme (project RFMEFI57414X0037).

REFERENCES

Actes de l'atelier DEFT'07. *Plate-forme AFIA 2007*. Grenoble, Juillet. http://deft07.limsi.fr/actes.php

Akhmedova, Sh., Semenkin, E., 2013. Co-Operation of Biology related Algorithms. In *IEEE Congress on Evolutionary Computations*. IEEE Publications.

Akhmedova, Sh., Semenkin, E., 2013. *New optimization metaheuristic based on co-operation of biology related algorithms*, Vestnik. Bulletine of Siberian State Aerospace University. Vol. 4 (50).

Boser, B., Guyon, I., Vapnik, V., 1992. A training algorithm for optimal margin classifiers. In *The 5th Annual ACM Workshop on COLT*. ACM.

Deb, K., 2000. *An efficient constraint handling method for genetic algorithms*, Computer methods in applied mechanics and engineering. Vol. 186(2-4).

Eiben, A.E., Smith, J.E., 2003. *Introduction to evolutionary computation*, Springer. Berlin.

Gasanova, T., Sergienko, R., Minker, W., Semenkin, E., Zhukov, E., 2013. A Semi-supervised Approach for Natural Language Call Routing. In *SIGDIAL 2013 Conference*.

Gasanova, T., Sergienko, R., Akhmedova, Sh., Semenkin, E., Minker, W., 2014. Opinion Mining and Topic Categorization with Novel Term Weighting. In *5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Association for Computational Linguistics*.

Kennedy, J., Eberhart, R., 1995. Particle Swarm Optimization. In *IEEE International Conference on Neural Networks*.

Kennedy, J., Eberhart, R., 1997. A discrete binary version of the particle swarm algorithm. In *World Multiconference on Systemics, Cybernetics and Informatics*.

Liang, J.J., Qu, B.Y., Suganthan, P.N., Hernandez-Diaz, A.G., 2012. *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization*. Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China, and Technical Report, Nanyang Technological University, Singapore.

Liang, J.J., Shang Z., Li, Z., 2010. Coevolutionary Comprehensive Learning Particle Swarm Optimizer. In *CEC'2010, Congress on Evolutionary Computation*. IEEE Publications.

Mallipeddi, R., Suganthan, P.N., 2009. *Problem Definitions and Evaluation Criteria for the CEC 2010 Competition on Constrained Real-Parameter Optimization*. Technical report, Nanyang Technological University, Singapore.

Pang, B., Lee, L, 2008. *Opinion Mining and Sentiment Analysis,* Now Publishers Inc. New-York.

Pang, B., Lee, L., Vaithyanathan, Sh., 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *EMNLP, Conference on Empirical Methods in Natural Language Processing*.

Salton, G., Buckley, C., 1988. *Term-weighting approaches in automatic text retrieval*, Information Processing and Management. Vol. 24 (5).

Soucy, P., Mineau, G.W., 2005. Beyond TFIDF Weighting for Text Categorization in the Vector Space Model. In *IJCAI'2005, The 19th International Joint Conference on Artificial Intelligence*.

Van Rijsbergen, C.J., 1979. *Information Retrieval*. Butterworth, 2nd edition.

Vapnik, V., Chervonenkis, A., 1974. *Theory of Pattern Recognition*, Nauka. Moscow.

Yang, Ch., Tu, X., Chen, J., 2007. Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search. In *International Conference on Intelligent Pervasive Computing*.

Yang, X.S., 2009 Firefly algorithms for multimodal optimization. In *The 5th Symposium on Stochastic Algorithms, Foundations and Applications*.

Yang, X.S., 2010. A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies for Optimization*, Studies in Computational Intelligence. Vol. 284.

Yang, X.S., Deb, S., 2009. Cuckoo Search via Levy flights. In *World Congress on Nature & Biologically Inspired Computing*. IEEE Publications.

Youngjoong Ko, 2012. A study of term weighting schemes using class information for text classification. In *SIGIR'12, The 35th Annual SIGIR Conference*. ACM.