Nonlinguistic Information Extraction by Semi-Supervised Techniques

Maria Semenkina, Shakhnaz Akhmedova and Eugene Semenkin
Institute of Computer Sciences and Telecommunication, Siberian State Aerospace University,
Krasnoyarskiy Rabochiy ave., 31, Krasnoyarsk, 660014, Russia

Keywords: Nonlinguistic Information Extraction, Semi-supervised Learning, Bio-inspired Algorithms, Evolutionary Algorithms.

Abstract: The concept of nonlinguistic information includes all types of extra linguistic information such as factors of age, emotion and physical states, accent and others. Semi-supervised techniques based on using both labelled and unlabelled examples can be an efficient tool for solving nonlinguistic information extraction problems with large amounts of unlabelled data. In this paper a new cooperation of biology related algorithms (COBRA) for semi-supervised support vector machines (SVM) training and a new self-configuring genetic algorithm (SelfCGA) for the automated design of semi-supervised artificial neural networks (ANN) are presented. Firstly, the performance and behaviour of the proposed semi-supervised SVMs and semi-supervised ANNs were studied under common experimental settings; and their workability was established. Then their efficiency was estimated on a speech-based emotion recognition problem.

1 INTRODUCTION

Nowadays different types of information technologies that try to emulate human-human interaction are involved in different fields: decision support systems, distance higher education, monitoring of terrorist threats, call processing in call centres and others. Intelligent dialogue systems (IDS) must not only make some formulaic answers but use human-like behaviour, for example, they must take into account the user's emotions to adapt its answers for the particular speaker. This means IDS have to use not only linguistic information, but also nonlinguistic information (Yamashita, 2013). The concept of nonlinguistic information includes all types of extra linguistic information such as factors of age, emotion and physical states, accent and others (Campbell, 2005).

Different types of machine learning techniques can be used for the extraction of nonlinguistic information, for example, artificial neural networks (ANN) or Support Vector Machines (SVM). The usual method of such "machine" extraction demands the long work of human experts in its initial stages to prepare the learning data, a process which includes such complex tasks as the lablling of large numbers of examples. Semi-supervised techniques can use both labelled and unlabelled data to construct appropriate models (Zhu and Goldberg, 2009). In this case it is not nessecary to label all of this large number of examples, but just a few of them.

In this study we use several semi-supervised techniques, such as semi-supervised support vector machines (Bennett and Demiriz, 1999) and semi-supervised artificial neural networks trained by evolutionary algorithms

The rest of the paper is organized as follows: in Section 2 the problem description is given; in Section 3 we give some information on semi-supervised support vector machines tarained by the cooperation of biology related algorithms (COBRA); in Section 4 different variants of semi-supervised artificial neural networks trained by a self-configuring genetic algorithm (SelfCGA) are described; in Section 5 we consider the outcomes of numerical experiments; and in the last section some conclusions and directions of further investigations are presented.

2 PROBLEM DESCRIPTION

In the cases of both supervised and semi-supervised learning for speech-based nonlinguistic information extraction, some learning data are needed.

Semenkina, M., Akhmedova, S. and Semenkin, E.
Nonlinguistic Information Extraction by Semi-Supervised Techniques.
DOI: 10.5220/0006438703120317
In *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2017) - Volume 1*, pages 312-317
ISBN: 978-989-758-263-9

Generally, any approach applied to this recognition problem contains the step of acoustic characteristic extraction.

An appropriate set of acoustic characteristics representing any speech signal was introduced at the INTERSPEECH 2009 Emotion Challenge. This set of features comprises attributes such as power, mean, root mean square, jitter, shimmer, 12 MFCCs, 5 formants and the mean, minimum, maximum, range and deviation of the pitch, intensity and harmonicity. The number of characteristics is 384. To get the conventional feature set introduced at INTERSPEECH 2009, the Praat (Boersma, 2002) or OpenSMILE (Eyben, 2010) systems might be used.

In this study the emotional database was considered. It consists of labelled emotional utterances which were spoken by actors. Each utterance has one of the emotional labels, neutral or strong. The average time of one record is 2.7 seconds. It contains 3210 examples, 426 of them belong to a neutral class. We used this dataset for the preliminary testing of semi-supervised techniques before the implementation in a real problem with unlabelled data.

So during the algorithm run only 10% of the data set will be used as labelled data (321 examples). The rest will be considered as unlabelled.

3 SEMI-SUPERVISED SUPPORT VECTOR MACHINES

In Support Vector Machines (SVM), the aim is to try to create a separating hyperplane between the instances from different classes (Vapnik and Chervonenkis, 1974). SVM is based on the maximization of the distance between the discriminating hyperplane and the closest examples. In other words since many choices could exist for the separating hyperplane, in order to generalize well on test data, the hyperplane with the largest margin has to be found.

Suppose $L = \{(x_1, y_1), \dots, (x_l, y_l)\}$, is a training set with l examples (instances), each instance has m attributes and x_i is labelled as y_i, where $i = \overline{1, l}$. Let v be a hyperplane going through the origin, δ be the margin and $w = \frac{v}{\delta}$. The margin maximizing hyperplane can be formulated as a constrained optimization problem in the following manner:

$$\frac{1}{2}\|w\|^2 \to min$$
$$y_i(w \cdot x_i) \geq 1$$

To solve the mentioned optimization problem the proposed cooperation of biology related algorithms or COBRA was used (Akhmedova and Semenkin, 2013).

However, in this study semi-supervised SVMs were considered. Thus, given the additional set $U = \{x_{l+1}, \dots, x_{l+u}\}$ of unlabelled training patterns, semi-supervised support vector machines aim at finding an optimal prediction function for unseen data based on both the labelled and the unlabelled part of the data (Joachims, 1999). For unlabelled data, it is assumed that the true label is the one predicted by the model based on what side of the hyperplane the unlabelled point ends up being.

In this study, self-training was used to learn from the unlabelled data. Namely, the idea is to design the model with labelled data and then use the model's own predictions as labels for the unlabelled data to retrain a new model with the original labelled data and the newly labelled data and then iteratively repeat this process.

The problem with this method is that it can suffer from "semantic drift", where considering its own predictions as true labels can cause the model to drift away from the correct model. The model would then continue to mislabel data and use it again and continue to drift farther and farther away from where it should be. To prevent this problem, in (Ravi, 2014) the model's predictions to label the data were used only when there was a high level of confidence about the predictions.

The notion of confidence used for the SVM model is the distance from the found hyperplane. The larger the distance from the hyperplane, the more confident we can be because this means the item is deeper in the space of the class the SVM thinks the item belongs to and thus it is likely it should be on the other side of the SVM.

So, the following basic steps were carried out:

- Train SVM on the labelled set L by the proposed meta-heuristic approach COBRA;
- Use the obtained SVM to classify all unlabelled instances from U by checking the confidence criteria from (Ravi, 2014);
- Label instances from the set U if this is possible;
- Repeat from the first step.

4 SEMI-SUPERVISED ANN AUTOMATED DESIGN

The appropriate structure of ANN must be chosen for the effective solving of the problem. Below we

consider a genetic algorithm (GA) for the choice of the number of layers, the number of neurons in each layer and the type of the activation function of each neuron for the multi-layered perceptron in the case of semi-supervised learning.

4.1 ANN in Binary String

First of all, we choose the perceptron with 5 hidden layers and 5 neurons in each hidden layer as the maximum size of the structure for ANN. Each node is represented by a binary string of length 4. If the string consists of zeros ("0000") then this node does not exist in ANN. So, the whole structure of the neural network is represented by a binary string of length 100 (25x4); each 20 variables represent one hidden layer. The number of input neurons depends on the problem in hand. ANN has one output layer.

We use 15 activation functions such as a bipolar sigmoid, a unipolar sigmoid, Gaussian, a threshold function and a linear function. For determining which activation function will be used on a given node, the integer that corresponds to its binary string is calculated.

Thus, we use optimization methods for problems with binary variables for finding the best structure and the optimization method for problems with real-valued variables for the weight coefficient adjustment of each structure.

Although the automated design of the ANN structure by self-adapting optimization techniques improves their efficiency, it can work unsatisfactorily with large real-world problems. Therefore, the automation of the most important input selection can have a significant impact on the efficiency of neural networks. In this paper, we use additional bits in every string for the choice of relevant variables to put them in model. The number of these bits equals the number of input variables. If this bit is equal to '0' then the corresponding input variable is not used in the model and is removed from the sample. During initialization, the probability for a variable to be significant will be equal to 1/3. This idea can help end users to avoid the significant and complicated procedure of choosing the appropriate set of input variables with the necessary impact on the model performance.

For the choice of more flexible models, more sophisticated tools must be used.

4.2 Self-configuring Genetic Algorithm

If the decision is made to use evolutionary algorithms for solving real world optimization problems, it will be necessary to choose an effective variant of algorithm parameters such as the kind of selection, recombination and mutation operators. Choosing the right EA setting for each problem is a difficult task even for experts in the field of evolutionary computation. It is the main problem in effectively implementing evolutionary algorithms for end users. We can conclude that it is necessary to find the solution for the main problem of evolutionary algorithms before suggesting for end users any EA application for the automated design of tools for solving real world problems.

We propose using the self-configuring evolutionary algorithms (SelfCEA) which do not need any end user efforts as the algorithm itself adjusts automatically to the given problem. In these algorithms (Semenkin, 2012), the dynamic adaptation of operators' probabilistic rates on the level of the population with centralized control techniques is applied.

Instead of adjusting real parameters, setting variants were used, namely the types of selection (fitness proportional, rank-based, and tournament-based with three tournament sizes), crossover (one-point, two-point, as well as equiprobable, fitness proportional, rank-based, and tournament-based uniform crossovers (Semenkin, 2012)), population control and level of mutation (medium, low, high for two mutation types). Each of these has its own initial probability distribution which is changed as the algorithm executes.

This self-configuring technique can be used both for the genetic algorithm (SelfCGA). In (Semenkin, 2012) SelfCGA performance was estimated on 14 test problems from (Finck, 2009). The statistical significance was estimated with ANOVA.

Analysing the results related to SelfCGA (Semenkin, 2012), it can be seen that self-configuring evolutionary algorithms demonstrate higher reliability than the average reliability of the corresponding single best algorithm but sometimes worse than the best reliability of this algorithm.

SelfCGA can be used for the automated choice of effective structures and weight tuning of ANN-based predictors. For such purposes, classification accuracy can be used as a fitness function.

4.3 Semi-Supervised ANN Design by Evolutionary Algorithms

Generally, any supervised techniques contain two stages:

1. extracted attributes or the most relevant of them

should be involved in the supervised learning process to adjust a classifier;
2. and then the trained classification model receives an unlabelled feature vector to make a prediction.

The method of genetic algorithm implementation in such a case was described above.

However, in the case of semi-supervised techniques, the following basic steps have to be implemented (Chapelle, 2006):

1. Train ANN on the labelled set;
2. Use the obtained ANN to classify all unlabelled instances from *U* by checking the confidence criteria;
3. Label instances from the set *U* if this is possible;
4. Repeat from the first step.

The main question is: "Which ANN from the population of ANNs will be making the decision about labelling some example?". There are two possible answers:

1. The best individual in the generation will be used for labelling examples if the confidence criterion is met (SelfCGA-ANN-Elitism);
2. All population members will vote and if the majority of them will be confident in one decision, the example will be labelled (SelfCGA-ANN-Ensemble).

The second important question is: "Do we have to train just ANN weights or automatically design the ANN structure?". There are two possible answers:

1. Only weight coefficients of ANN will be adjusted (SelfCGA-ANN-w);
2. The complete ANN will be designed, including both the ANN structure design and the adjusting of weights (SelfCGA-ANN).

And the last question is: "How often should we stop the evaluation process and begin the process of labelling for test (unlabelled) data?".

1. The SelfCGA for the automated ANN structure design has to make a pause every 5 generations, try to label the data and after this continue its work with new a learning set (additional examples that took label).
2. The SelfCGA for ANN weights training have to make a pause every 10 generations, try to label data and after that continue its work with a new learning set (additional examples that took the label).

5 EXPERIMENTAL RESULTS

At the first stage of experiments, we tested all algorithm variants on one artificial and two real-world problems that will be described in Table 1. All these data sets are classification problems and for the testing of semi-supervised techniques, each data set instance was randomly split into two parts: one labelled and one unlabelled – and different ratios for the particular settings were used.

First of all, one well-known artificial problem was considered, namely the two-dimensional "Moons" data set (Jain, 2005). This problem is known to be a complex problem for semi-supervised techniques and a very simple problem for humans. This is why it is often used as a test problem for different machine learning algorithms and became a classical test problem for them. It consists of two groups of moon-like sets of points and it has a separating hyperplane between them. So it has a non-linear structure that makes it difficult for semi-supervised support vector machines. In this experiment, the starting learning set contains only 4 labelled examples, 2 from one class and 2 from another one that were randomly chosen. All other examples must be labelled during the run.

The usual results obtained on the "Moons" problem are shown in Figure 1 (COBRA-SVM) and Figure 2 (SelfCGA-ANN). As can be seen, the algorithms do not recognize all the points correctly. However, most of the points are in the right class. COBRA-SVM builds an almost linear classification, SelfCGA-ANN-Ensemble builds a more complex separating hyperplane. The best result was shown by SelfCGA-ANN-Elitism, it usually made mistakes only on 1-2 points. It is probable that SelfCGA-ANN-Ensemble excessively averaged single ANN results.

Table 1: Data sets, considered in the experimental evaluation, each consisting of n patterns having d features.

Data Set Name	Example's number	Input number
Moons	200	2
Breast Cancer Wisconsin	699	9
Pima Indians Diabetes	768	8

Then two medical diagnostic problems, namely Breast Cancer Wisconsin and Pima Indian Diabetes (Frank and Asuncion, 2010), were solved. Both problems are binary classification tasks. For these data sets, 10 examples were randomly selected to be used as labelled examples, and the remaining instances were used as unlabelled data. The

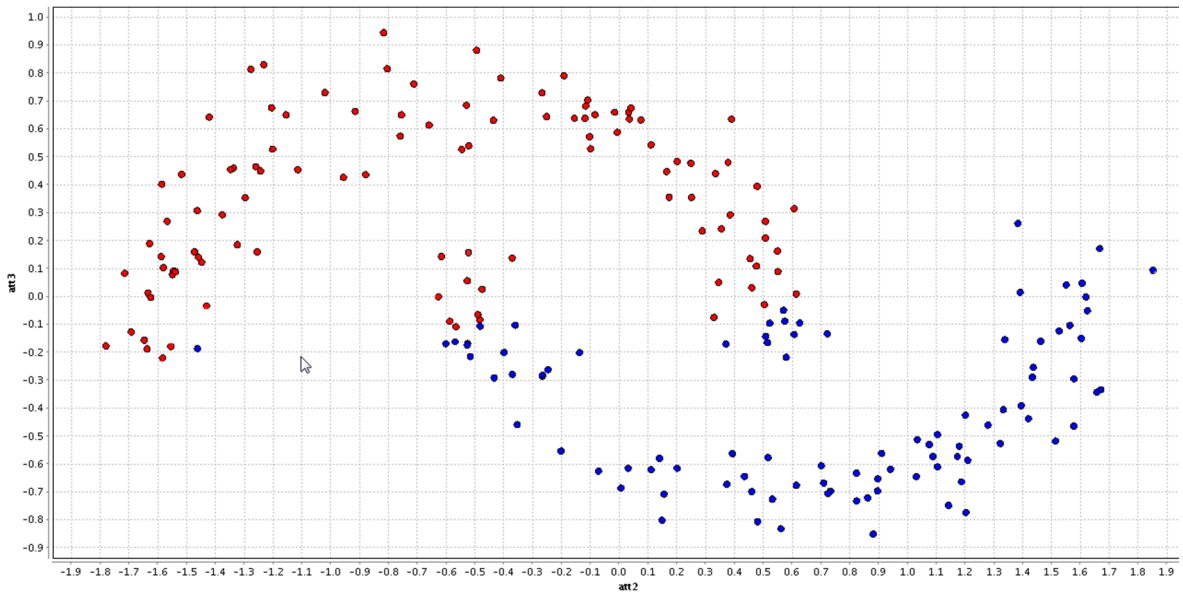

Figure 1: Semi-supervised classification of "Moons" by COBRA-SVM.

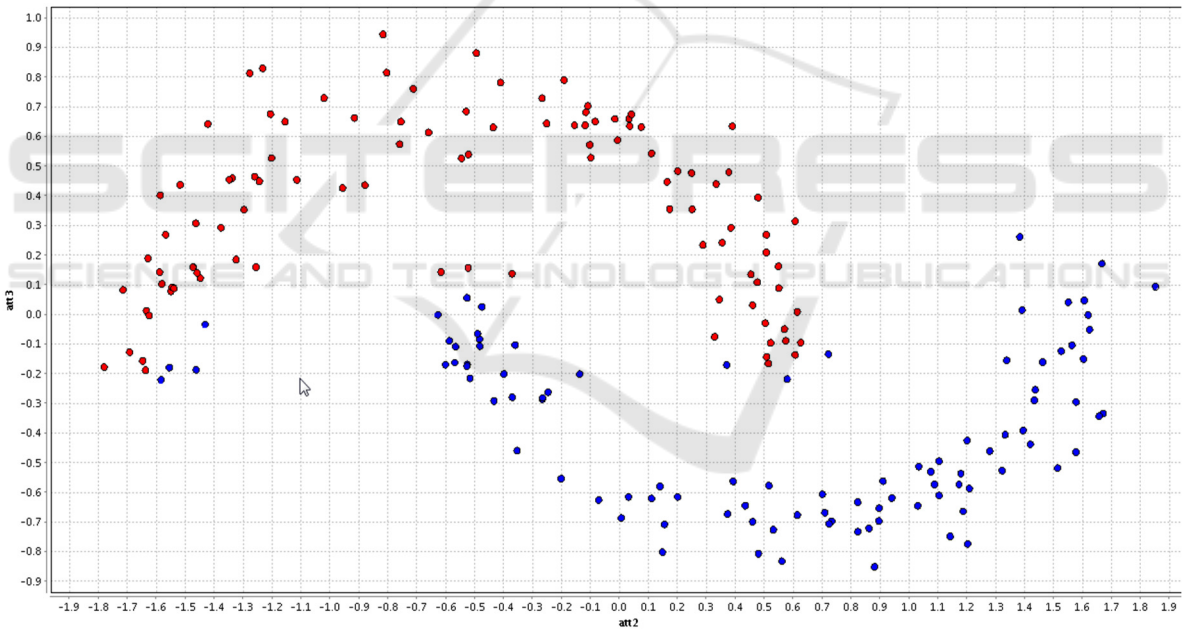

Figure 2: Semi-supervised classification of "Moons" by SelfCGA-ANN-Ensemble.

experiments are repeated 50 times and the average accuracies and standard deviations are recorded. Alternative algorithms (linear SVMs) for comparison are taken from (Li and Zhou, 2011). The results are shown in Table 2.

As can be seen, COBRA-SVM and SelfCGA-ANN are sufficiently effective for solving semi-supervised problems.

At the second stage of experiments, we tested all algorithm variants on speech-based emotion recognition problems that had 384 features and only 321 randomly selected instances in the initial learning set (stratified sampling) and 2889 instances which were used as unlabelled ones. The experiments are repeated 50 times and the average accuracies and range of variation are recorded in Table 3. In all experiments, weighted accuracy was assessed to compare the quality of classification. The statistical robustness of the results obtained was confirmed by ANOVA tests, which were used for

processing the received evaluations of our algorithms' performance.

The classification quality is relatively high even with only 10% of labelled examples in the training set. This result gives the possibility to use a small amount of data labelled by experts with a huge amount of available unlabelled data for nonlinguistic information extraction in the future.

Table 2: Performance comparison for medical diagnostics problems.

Algorithm's Name	Breast Cancer Wisconsin	Pima Indians Diabetes
TSVM	89.2±8.6	63.4±7.6
S3VM-c	94.2±4.9	63.2±6.8
S3VM-p	93.9±4.9	65.6±4.8
S3VM-us	93.6±5.4	65.2±5.0
COBRA-SVM	95.5±1.8	69.3±1.5
SelfCGA-ANN-w	94.8±2.1	66.7±2.3
SelfCGA-ANN-Elit.	96.5±1.9	69.4±1.8
SelfCGA-ANN-Ens.	95.6±1.3	68.7±1.5

Table 3: Performance comparison for emotion recognition problem.

Algorithm's Name	F-score
COBRA-SVM	0.8799 [0.8763; 0.8832]
SelfCGA-ANN-Elit.	0.8864 [0.8794; 0.8901]
SelfCGA-ANN-Ens.	0.8807 [0.8775; 0.8849]
SelfCGA-ANN-w	0.8582 [0.8534; 0.8623]

6 CONCLUSIONS

The possibility to use semi-supervised classification for nonlinguistic information extraction is important due to the fact that getting labelled examples is often very expensive and sometimes must be repeated for any new person. However, using unlabelled data during classification may be helpful. In this paper, the semi-supervised SVM was trained using a cooperative algorithm and semi-supervised ANNs were automatically designed by SelfCGA for solving semi-supervised classification problems in the field of speech-based emotion recognition. The results show that the proposed approaches are sufficiently effective for solving this kind of problems. The comparison of their results show that models with a more complex structure, for example, ANNs with a more flexible structure, can give better results.

ACKNOWLEDGEMENTS

This research is partially supported by Grant of the President of the Russian Federation for state support of young Russian scientists (MK- 3378.2017.9).

REFERENCES

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

Bennett, K.P., Demiriz, A., 1999. Semi-supervised support vector machines. *Advances in Neural Information Processing Systems 11*.

Boersma P., 2002. Praat, a system for doing phonetics by computer. *Glot international*, vol. 5, no. 9/10, pp. 341–345.

Campbell N., 2005. Developments in corpus-based speech synthesis: Approaching natural conversational speech, *IEICE Trans. Inf. Syst.,* E88-D, 376–383.

Chapelle O., Zien A., Schoelkopf B. (Eds.), 2006. *Semi-supervised learning.* MIT Press.

Eyben F., Wllmer M., and Schuller B., 2010. Opensmile: the Munich versatile and fast opensource audio feature extractor. *Proceedings of the international conference on Multimedia,. ACM*, pp. 1459–1462.

Finck, S. et al., 2009. Real-parameter black-box optimization benchmarking 2009. *In: Presentation of the noiseless functions. Technical Report Researh Center PPE.*

Jain A. and Law M., 2005. Data clustering: A user's dilemma. *Lecture Notes in Computer Science. 3776: p. 1-10.*

Joachims, T., 1999. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning.*

Ravi, S., 2014. *Semi-supervised Learning in Support Vector Machines.* Project Report COS 521.

Semenkin, E.S., Semenkina, M.E., 2012. Self-configuring Genetic Algorithm with Modified Uniform Crossover Operator. *Advances in Swarm Intelligence, Lecture Notes in Computer Science 7331*, Springer-Verlag, Berlin Heidelberg, pp. 414-421.

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

Zhu, X., Goldberg, A.B., 2009. *Introduction to Semi-Supervised Learning*. Morgan and Claypool.

Li, Y.F., Zhou, Z.H., 2011. Improving Semi-Supervised Support Vector Machines through Unlabeled Instances Selection. In *The Twenty Fifth AAAI Conference on Artificial Intelligence*.

Yamashita Y., 2013. A review of paralinguistic information processing for natural speech communication. Acoust. Sci. & Tech. 34, 2, pp. 73-79.