Weighted Aging Classifier Ensemble for the Incremental Drifted Data Streams

Michał Woźniak, Andrzej Kasprzak, and Piotr Cal

Department of Systems and Computer Networks
Wroclaw University of Technology
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
{michal.wozniak,andrzej.kasprzak,piotr.cal}@pwr.wroc.pl

Abstract. Evolving systems are recently focus of intense research because for most of the real problems we can observe that the parameters of the decision tasks should adapt to new conditions. In classification such a problem is usually called concept drift. The paper deals with the data stream classification where we assume that the concept drift is sudden but its rapidity is limited. To deal with this problem we propose a new algorithm called Weighted Aging Ensemble (WAE), which is able to adapt to changes of classification model parameters. The method is inspired by well-known algorithm Accuracy Weighted Ensemble (AWE) which allows to change the line-up of a classifier ensemble, but the proposed method incudes two important modifications: (i) classifier weights depend on the individual classifier accuracies and time they have been spending in the ensemble, (ii) individual classifier are chosen to the ensemble on the basis on the non-pairwise diversity measure. The proposed method was evaluated on the basis of computer experiments which were carried out on SEA dataset. The obtained results encourage us to continue the work on the proposed concept.

Keywords: machine learning, classifier ensemble, data stream, concept drift, incremental learning, forgetting.

1 Introduction

The market-leading companies realize that smart analytic tools which are capable to analyze collected, fast-growing data could lead to business success. Therefore they desire to exploit strength of machine learning techniques to extract hidden, valuable knowledge from the huge databases. One of the most promising directions of that research is classification task, which is widely used in computer security (e.g. designing intrusion detection/prevention systems IDS/IPS), medicine, finance (e.g., fraud detection or credit approval), or trade. Designing such solutions we should take into consideration that in the modern world the most of the data arrive continuously and it causes that smart analytic tools should respect this nature and be able to interpret so-called data streams. Unfortunately most of the traditional methods of classifier design do not take into consideration that:

H.L. Larsen et al. (Eds.): FQAS 2013, LNAI 8132, pp. 579–588, 2013.

- the statistical dependencies between the observations of a given objects and their classifications could change,
- data can come flooding in the analyzer what causes that it is impossible to label all records.

This work focuses on the first problem called *concept drift* [17] and it comes in many forms, depending on the type of change. In general, the following approaches can be considered to deal with the mentioned above problem

- Rebuilding a classification model if new data becomes available, which is very expensive and impossible from a practical point of view, especially if the concept drift occurs rapidly.
- Detecting concept changes in new data and if these changes are sufficiently "significant", then rebuilding the classifier.
- Adopting an incremental learning algorithm for the classification model.

We will concentrate on the last proposition. Adapting the learner is a part of an incremental learning [10]. The model is either updated (e.g., neural networks) or needs to be partially or completely rebuilt (as CVFDT algorithm [4]). Usually we assume that the data stream is given in a form of data chunks (windows). When dealing with the sliding window the main question is how to adjust the window size. On the one hand, a shorter window allows focusing on the emerging context, though data may not be representative for a longer lasting context. On the other hand, a wider window may result in mixing the instances representing different contexts. Therefore, certain advanced algorithms adjust the window size dynamically depending on the detected state (e.g., FLORA2 [17]) or algorithms can use multiple windows [9]. One of the important group of algorithms dedicated to stream classification exploits strength of ensemble systems, which work pretty well in static environments [8], because according to "no free lunch theorem" [18] there is not a single classifier that is suitable for all the tasks, since each of them has its own domain of competence. A strategy for generating the classifier ensemble should guarantee its diversity improvement therefore let us enumerate the main propositions how to get a desirable committee:

- The individual classifiers could be train on different datasets, because we hope that classifiers trained on different inputs would be complementary.
- The individual classifiers can use the selected features only.
- Usually it could be easy to decompose the classification problem into simpler ones solved by the individual classifier. The key problem of such approach is how to recover the whole set of possible classes.
- The last and intuitive method is to use individual classifiers trained on different models or different versions of models.

It has been shown that a collective decision can increase classification accuracy because the knowledge that is distributed among the classifiers may be more comprehensive [14]. Usually, a diversity may refer to the classifier model, the feature set, or the instances used in training, but in a case of data stream classification diversity can also refer to the context, but the problem how the diversity of the classifier ensemble should be measured still remains.

Several strategies are possible for a data stream classification:

1. Dynamic combiners, where individual classifiers are trained in advance and their relevance to the current context is evaluated dynamically while processing subsequent data. The level of contribution to the final decision is directly proportional to the relevance [5]. The drawback of this approach is that all contexts must be available in advance; emergence of new unknown contexts may result in a lack of experts.
2. Updating the ensemble members, where each ensemble consists of a set of online classifiers that are updated incrementally based on the incoming data [2].
3. Dynamic changing line-up of ensemble e.g., individual classifiers are evaluated dynamically and the worst one is replaced by a new one trained on the most recent data.

Among the most popular ensemble approaches, the following are worth noting: the Streaming Ensemble Algorithm (SEA) [15] or the Accuracy Weighted Ensemble (AWE)[16]. Both algorithms keep a fixed-size set of classifiers. Incoming data are collected in data chunks, which are used to train new classifiers. All the classifiers are evaluated on the basis of their accuracy and the worst one in the committee is replaced by a new one if the latter has higher accuracy. The SEA uses a majority voting strategy, whereas the AWE uses the more advanced weighted voting strategy. A similar formula for decision making is implemented in the Dynamic Weighted Majority (DWM) algorithm [7].

In this work we propose the dynamic ensemble model called WAE (Weighted Aging Ensemble) which can modify line-up of the classifier committee on the basis of diversity measure. Additionally the decision about object's label is made according to weighted voting, where weight of a given classifier depends on its accuracy and time spending in an ensemble. The detailed description of WAE is presented in the next section. Then we present preliminary results of computer experiments which were carried out on SEA dataset and seem to confirm usefulness of proposed algorithm. The last section concludes our research.

2 Algorithm

We assume that the classified data stream is given in a form of data chunks denotes as $\mathcal{DS}_k$, where k is the chunk index. The concept drift could appear in the incoming data chunks. We do not detect it, but we try to construct self-adapting classifier ensemble. Therefore on the basis of the each chunk one individual is trained and we check if it could form valuable ensemble with the previously trained models. In our algorithm we propose to use the Generalized Diversity (denoted as $\mathcal{GD}$) proposed by Partridge and Krzanowski [11] to assess all possible ensembles and to choose the best one. $\mathcal{GD}$ returns the maximum values in the case of failure of one classifier is accompanied by correct classification by the other one and minimum diversity occurs when failure of one classifier is accompanied by failure of the other.

$$\mathcal{GD}(\Pi) = 1 - \frac{\sum_{i=1}^{L} \frac{i(i-1)p_i}{L(L-1)}}{\sum_{i=1}^{L} \frac{ip_i}{L}} \tag{1}$$

where L is the cardinality of the classifier pool (number of individual classifiers) and p_i stands for the probability that i randomly chosen classifiers from Π will fail on randomly chosen example.

Lets $P_a(\Psi_i)$ denotes frequency of correct classification of classifier Ψ_i and $itter(\Psi_i)$ stands for number of iterations which Ψ_i has been spent in the ensemble. We propose to establish the classifier's weight $w(\Psi_i)$ according to the following formulae

$$w(\Psi_i) = \frac{P_a(\Psi_i)}{\sqrt{itter(\Psi_i)}} \tag{2}$$

This proposition of classifier aging has its root in object weighting algorithms where an instance weight is usually inversely proportional to the time that has passed since the instance was read [6] and Accuracy Weighted Ensemble (AWE)[16], but the proposed method called Weighted Aging Ensemble (WAE) incudes two important modifications:

1. classifier weights depend on the individual classifier accuracies and time they have been spending in the ensemble,
2. individual classifier are chosen to the ensemble on the basis on the non-pairwise diversity measure.

The WAE pseudocode is presented in Alg.1.

3 Experimental Investigations

The aims of the experiment were to assess if the proposed method of weighting and aging individual classifiers in the ensemble is valuable proposition compared with the methods which do not include aging or weighting techniques.

3.1 Set-Up

All experiments were carried out on the SEA dataset describes in [15]. Each object belongs to the on of two classes and is described by 3 numeric attributes with value between 0 and 10, but only two of them are relevant. Object belongs to class 1 (TRUE) if $arg_1 + arg_2 < \phi$ and to class 2 (FALSE) if $arg_1 + arg_2 \geq \phi$. ϕ is a threshold between two classes, so different thresholds correspond to different concepts (models).Thus, all generated dataset is linearly separable, but we add 5% noise, which means that class label for some samples is changed, with expected value equal to 0. The number of objects, noise and the set of concepts are set by user. We simulated drift by instant random model change.

Algorithm 1. Weighted Aging Ensemble (WAE)

Require: input data stream, data chunk size, classifier training procedure, ensemble size L

1: $i := 1$
2: **repeat**
3: collect new data chunk DS_i
4: train classifier Ψ_i on the basis of DS_i
5: add Ψ_i to the classifier ensemble Π
6: **if** $i > L$ **then**
7: $\Psi_{k+1} = \Psi_i$
8: $\Pi_t = \emptyset$
9: $GD_t = 0$
10: **for** $j = 1$ **to** $L + 1$ **do**
11: **if** $\mathcal{GD}(\Pi \backslash \Psi_i)$ (calculated according to (1)) $> GD_t$ **then**
12: $\Pi_t = \Pi \backslash \Psi_i$
13: **end if**
14: **end for**
15: $\Pi = \Pi_t$
16: **end if**
17: $w := 0$
18: **for** $j = 1$ **to** L **do**
19: calculate $w(\Psi_i)$ according to (2)
20: $w := w + w(\Psi_i)$
21: **end for**
22: **for** $j = 1$ **to** L **do**
23: $w(\Psi_i) := \frac{w(\Psi_i)}{w}$
24: **end for**
25: $i := i + 1$
26: **until** end of the input data stream

For each of the experiments we decided to form homogenous ensemble i.e., ensemble which consists of the classifier using the same model. We repeated experiments for Naive Bayes, decision tree trained by C4.5 [13], and SVM with polynomial kernel trained by the sequential minimal optimization method (SMO) [12].

During each of the experiment we tried to evaluate dependency between data chunk sizes (which were fixed on 50, 100, 150, 200) and overall classifier quality (accuracy and standard deviation) for the following ensembles:

1. $w0a0$ - an ensemble using majority voting without aging.
2. $w1a0$ - an ensemble using weighted voting without aging, where weight assigned to a given classifier is inversely proportional to its accuracy.
3. $w1a1$ - an ensemble using weighted voting with aging, where weight assigned to a given classifier is calculated according to (2).

Method of ensemble pruning was the same for each ensemble and presented in Alg.1. The only difference was line 19 of the pseudocode what was previously

described. All experiments were carried out in the Java environment using Weka classifiers [3].

3.2 Results

The results of experiments are presented in Fig.1-6. Fig. 1-3 show the accuracies of the tested ensembles for a chosen experiment. Unfortunately, because of the space limit we are not able to presents all extensive results, but they are available on demand from corresponding author. Fig.4-6 present overall accuracy and standard deviation for the tested methods and how they depend on data chunk size.

3.3 Discussion

On the basis of presented results we can formulate several observations. It does not surprise us that quality improvements for all tested method according to increasing data chunk size. Usually the WAE outperformed others, but the differences are quite small and only in the case of ensemble built on the basis of Naive Bayes classifiers the differences are statistical significant (t-test) [1] i.e., differences among different chunk sizes. The observation is useful because the bigger size of data chunk means that effort dedicated to building new models is smaller because they are being built rarely.

Another interesting observation is that the standard deviation is smaller for bigger data chunk and usually standard deviation of WAE is smallest among all

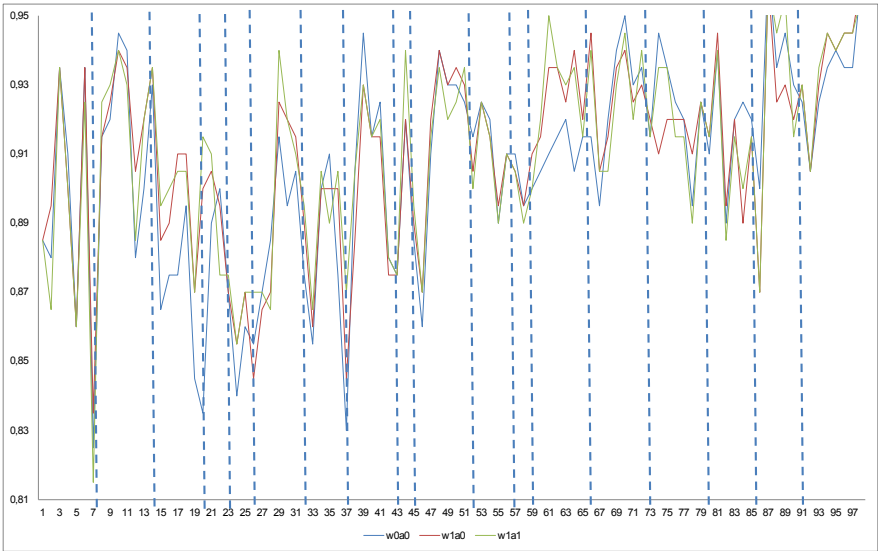

Fig. 1. Classification accuracy of the ensembles consist of Naive Bayes classifiers for the chunk size = 200. Vertical dotted lines indicate concept drift appearances.

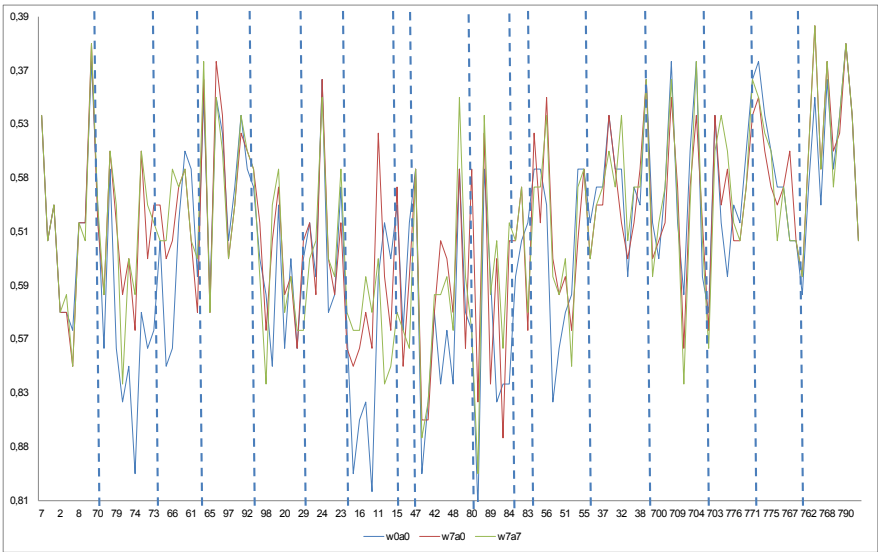

Fig. 2. Classification accuracy of the ensembles consist of C4.5 (decision tree) classifiers for the chunk size = 150. Vertical dotted lines indicate concept drift appearances.

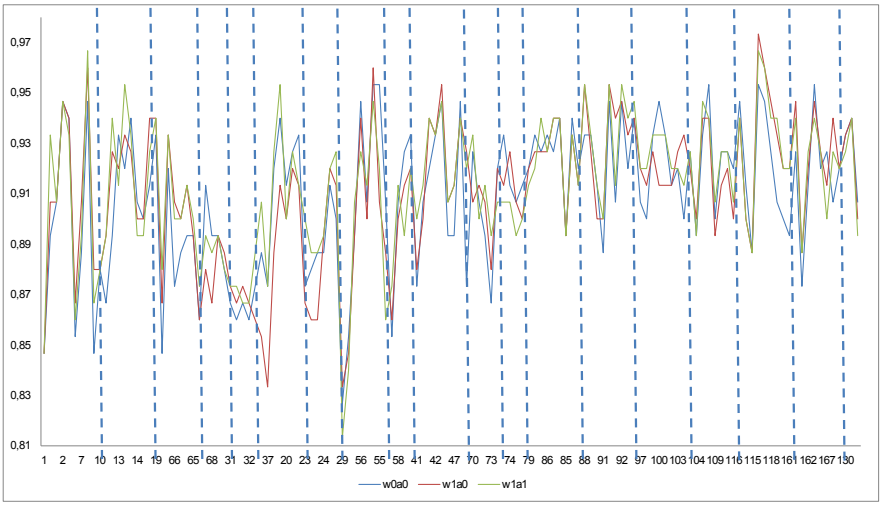

Fig. 3. Classification accuracy of the ensembles consist of SVM classifiers for the chunk size = 150. Vertical dotted lines indicate concept drift appearances.

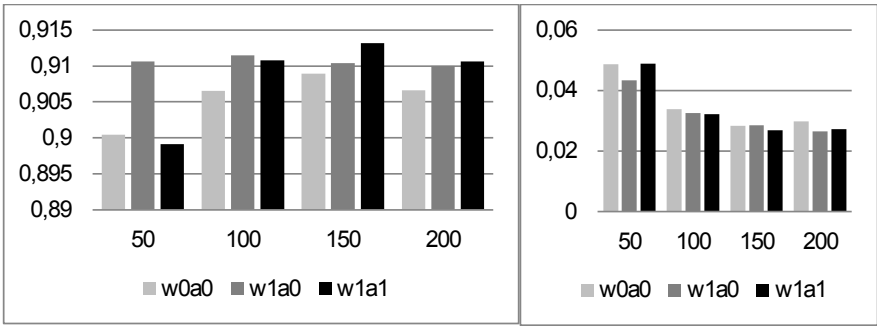

Fig. 4. Classification accuracy (left) and standard deviation (right) of Naive Bayes classifier for different data chunk sizes

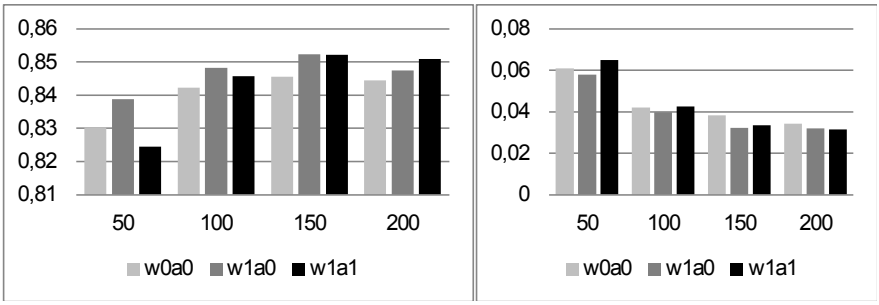

Fig. 5. Classification accuracy (left) and standard deviation (right) of C4.5 classifier for different data chunk sizes

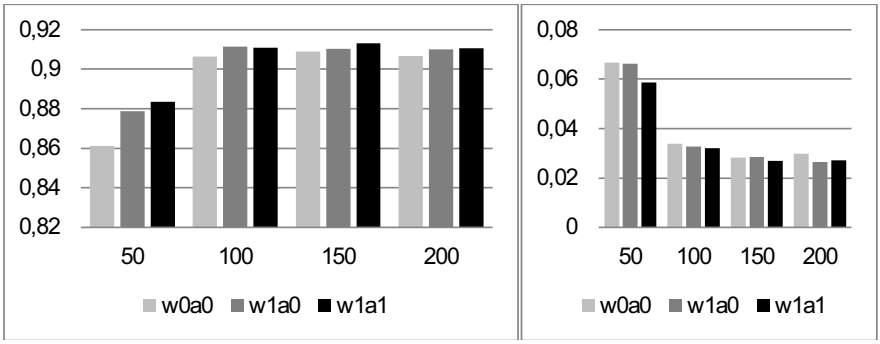

Fig. 6. Classification accuracy (left) and standard deviation (right) of SVM classifier for different data chunk sizes

tested methods. It means that the concept drift appearances have the weakest impact on the WAE accuracy.

We realize that the scope of the experiments we carried out is limited and derived remarks are limited to the tested methods and one dataset only. In this case formulating general conclusions is very risky, but the preliminary results are quite promising, therefore we would like to continue the work on WAE in the future.

4 Conclusions

The paper presented the original classifier for data stream classification tasks. Proposed WAE algorithm uses dynamic classifier ensemble i.e., its line-up is formed when new data chunk is come and the decision which classifier is chosen to the ensemble is made on the basis of General Diversity (diversity measure). The decision about object's label is made according to weighted voting where weight assigned to a given classifier depends on its accuracy (proportional) and how long the classifier participates in the ensemble (inversely proportional). The experiments conformed that proposed method can adapt to changing concept returning stable classifier. We would like to emphasize that we presented preliminary study on WAE which is a starting point for the future research. In the near future we are going to:

- carry out experiments on the wider number of datasets,
- evaluate WAE's behavior for more sharp sudden concept drift,
- evaluate usefulness of the other diversity measures for WAE's classifier ensemble pruning,
- assess more sophisticated combination rules based on support functions of individual classifiers,
- check if training set of different classifier model on the basis of new data chunk could have an impact on WAE's quality, because such an approach will lead to the more diverse heterogenous classifier ensemble.

Acknowledgment. The work was supported by the statutory funds of the Department of Systems and Computer Networks, Wroclaw University of Technology and by the Polish National Science Center under a grant N N519 650440 for the period 2011-2014.

References

1. Alpaydin, E.: Introduction to Machine Learning, 2nd edn. The MIT Press (2010)
2. Bifet, A., Holmes, G., Pfahringer, B., Read, J., Kranen, P., Kremer, H., Jansen, T., Seidl, T.: Moa: a real-time analytics open source framework. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part III. LNCS, vol. 6913, pp. 617–620. Springer, Heidelberg (2011)
3. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)

4. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 97–106 (2001)
5. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. 3, 79–87 (1991)
6. Klinkenberg, R., Renz, I.: Adaptive information filtering: Learning in the presence of concept drifts, pp. 33–40 (1998)
7. Kolter, J.Z., Maloof, M.A.: Dynamic weighted majority: a new ensemble method for tracking concept drift. In: Third IEEE International Conference on Data Mining, ICDM 2003, pp. 123–130 (November 2003)
8. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
9. Lazarescu, M.M., Venkatesh, S., Bui, H.H.: Using multiple windows to track concept drift. Intell. Data Anal. 8(1), 29–59 (2004)
10. Muhlbaier, M.D., Topalis, A., Polikar, R.: Learn^{++}.nc: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes. IEEE Transactions on Neural Networks 20(1), 152–168 (2009)
11. Partridge, D., Krzanowski, W.: Software diversity: practical statistics for its measurement and exploitation. Information and Software Technology 39(10), 707–717 (1997)
12. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods, pp. 185–208. MIT Press, Cambridge (1999)
13. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann Publishers (1993)
14. Shipp, C.A., Kuncheva, L.: Relationships between combination methods and measures of diversity in combining classifiers. Information Fusion 3(2), 135–148 (2002)
15. Nick Street, W., Kim, Y.: A streaming ensemble algorithm (sea) for large-scale classification. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2001, pp. 377–382. ACM, New York (2001)
16. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2003, pp. 226–235. ACM, New York (2003)
17. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Mach. Learn. 23(1), 69–101 (1996)
18. Wolpert, D.H.: The supervised learning no-free-lunch theorems. In: Proc. 6th Online World Conference on Soft Computing in Industrial Applications, pp. 25–42 (2001)