Prototype-based Committees

Karol Grudziński
E-mail: karol.grudzinski@wp.pl

Department of Physics, Kazimierz Wielki University
Plac Weyssenhoffa 11, 85-072 Bydgoszcz, Poland
and
Institute of Applied Informatics
University of Economy
Garbary 2, 85-229 Bydgoszcz, Poland

Affilation: Kazimierz Wielki University

Abstract. A prototype (case-based) way of data explanation is a powerful method for data analysis and understanding. This approach is the alternative to the rule induction techniques. Interesting instance vectors (prototypes) are usually generated by a training set pruning with various partial memory learners. In this paper prototype-based committees (PBC) are introduced, allowing for, which is believed, a better case-based analysis of any data. Due to the lack of space, only generalization abilities of PBC built on the SBL-PM-M partial memory learner are investigated. The study of suitability of PBC for a better case-based data interpretation is left for further papers.

1 Introduction.

Data mining is commonly used in many domains. A case-based way of data explanation, which is the alternative to the rule induction technique for data understanding, is very popular among researchers. Such an approach to data analysis is particularly often employed in medicine, where a medical doctor makes a diagnosis by referring to other similar cases in a database of patients.

Interesting instance vectors (prototypes) are usually generated by a training set pruning with various partial memory learners. Partial memory learning (PML) is a machine learning paradigm, in which only a small fraction of a usually much larger, original training set is used for a final classification of unseen samples. The term 'Partial Memory Learning' is most often reserved for on-line learning systems that select and store a portion of the past learning examples. In this paper we stick to the name PML, but this methodology is called also 'instance selection', 'training data compression, reduction or pruning' [1–8].

Protype-based Committees (PBC) have been introduced to permit a better case-based data understanding, however the study of suitability of them in this area is left for further papers. Because of the lack of space, only generalization abilities of PBC built on the SBL-PM-M system [9] are studied on four real-word datasets. Additionally to usually lowering the variance of the model on unseen cases, the size of the partial memory has been decreased significantly often without degradation or sometimes even with the improvement of the performance of the classifier on test sets in comparison to plain SBL-PM-M models (i.e. such ones, which are obtained without the application of PBC). The results obtained by us have been compared to those of Wilson and Martinez [4] and indicate superiority of SBL-PM-M and PBC over the algorithms described in [4] in the experiments that had been conducted by us on the studied datasets.

The paper consists of five sections. In the second section, the SBL-PM-M system is briefly described. In the third section, the idea of prototype-based committees is introduced. In the next section, our numerical results are presented. The last section concludes the paper.

2 The SBL-PM-M Algorithm

The acronym SBL-PM-M is short for **S**imilarity-**B**ased-**L**earner-**P**artial-**M**emory-**M**inimization. The SBL-PM-M system is based on the optimization of binary weights, which are assigned to samples in the training set. The weight with the value of 1 indicates, that the corresponding case takes part in the classification process and otherwise, it is not taken into account. The number of weights is equal to the number of samples in the training set. The cost function returns the number of errors the classifier makes. The target function is not continuous and therefore a non-gradient minimization routine must be used. So far we have tried only the simplex method for function minimization [10].

Our experiments indicate, that in order to arrive at a reasonable solution (i.e. good generalization with a small partial-memory size) only $250 - 300$ cost evaluations is needed for small training sets consisting of a couple of hundred cases. Similarly to SBL-PM-B [9, 11], in SBL-PM-M there is a way to roughly control the number of cases retained by initializing the simplex (randomly) with zeros and ones generated with a different probability.

The input parameters for the SBL-PM-M method are:

1. The probability of the occurrence of the values of 1 with which the simplex is initialized (the value of 0.5 indicates that we start the learning with approximately the half of the size of the training set).
2. The maximum number of permitted cost evaluations.

The output of the model is the generated prototype (partial memory) set with which the unseen cases are classified. The number of samples retained depends on the input parameters, the nature of the data under study and the classifier used. It is hard to estimate the number of samples retained on output.

Time requirements of the SBL-PM-M algorithm are proportional to the number of cost evaluations, the size of the original training set and the performance of the employed classifier. The larger is the training dataset, the more cost evaluations is needed to assure low variance of the model. To many cost evaluations may lead to over-fitting.

3 Prototype-based Committees

Let $M(\mathcal{I})$ be a partial memory method with a fixed input parameter set $\mathcal{I}$. Let $\mathcal{M}(\mathcal{I}, \mathcal{O})$ be a model with the fixed input and output parameters $\mathcal{O}$, which has been trained on the training data $\mathcal{T}$ and which corresponds to the method $M(\mathcal{I})$. Thus, the method is the algorithm and the model is a trained method.

If N is the number of times the PML method $M(\mathcal{I})$ is trained on the training data $\mathcal{T} = \{\mathbf{X}_i\}_{i=1}^{n}$ with the same, fixed input parameters $\mathcal{I}$ and $\mathcal{M}_j(\mathcal{I}, \mathcal{O}_j)$ are the obtained models, then $\mathcal{R}_j = \{\mathbf{X}_{l_j}\}_{l_j=1}^{S_{\mathcal{R}_j}}$ are the obtained partial memory (prototype, reference) sets, where $S_{\mathcal{R}_j}$ is the cardinality of $\{\mathbf{X}_{l_j}\}$, $j = 1 \dots N$.

Prototype-based committees are similar to majority committees [12, 13]. In this paper only homogeneous prototype committees are studied, i.e. such ones, which are constructed out of the single SBL-PM-M method with fixed input parameters. PBC have three input parameters:

1. The method $M(\mathcal{I})$ (in our case SBL-PM-M),
2. The number N of models $\mathcal{M}_j(\mathcal{I}, \mathcal{O}_j)$ corresponding to the method $M(\mathcal{I})$ out of which the committee is constructed,
3. The confidence factor $CF(R, N)$ of the partial memory set R obtained as a result of application of the PBC.

The idea of PBC is very simple. Only these examples for the target partial memory set R are selected from $\mathcal{R}_j$, which occur at least CF times in $\bigcup \mathcal{R}_j$, $j = 1, \dots, N$. In other words the examples will be placed in R, if the probability of retaining them in $\mathcal{R}_j$ is greater or equal to $\frac{CF}{N}$.

We define the reliability $RL(R, N)$ of the partial memory set R by

$$RL(R, N) = \frac{CF(R, N)}{N} \cdot 100\%$$

4 Numerical experiments

The results of calculations on prototype-based committees of the SBL-PM-M models are denoted by EM. The parameters of the committee (the number of models and the confidence factor) follow the name of the method. Additionally, the SBL-PM-M, SBL-PM-B and k-NN calculations have been conducted and denoted by M and B respectively. The k-NN classifier has been used as a basis for the EM, M, and the B method. The results obtained by Wilson and Martinez with their implementation of k-NN, and their partial memory models: RT1, RT2, RT3 and H-IB3 are taken from [4] and are placed below ours. All our results have been obtained with our system SBL [14, 15]. In the all our experiments 10-fold stratified cross-validation test repeated 10 times has been used and every training partition has been pruned with EM, M or B models and with the partial memory set obtained in such a way, the unseen cases in the test partition have been classified. The identical test has been performed by Wilson and Martinez in [4]. As it can be concluded from their paper, the experiments have been performed without repetition and thus the average and variance is not provided. What concerns the adaptive parameters such as the the value of k and the similarity function used, we had no HVDM metric implemented in our system and thus our own metrics and different values for k have been used by us. Little effort has been made by us to optimize the parameters and the results given certainly could be better. The results in tables are sorted with respect to the accuracy attained by systems on test sets. The probability with which the weights of the cases taking part in the classification (i.e. the starting weights with the value of 1 which are inserted in the simplex) is given right after the method name, in the case of SBL-PM-M algorithm and after the committee parameters, in the case of committees of SBL-PM-M models. Following this number, in the square brackets, the restriction on the maximum number of the cost evaluations is given. The letter 'A' following B in the listing denotes that the parameter Δ of the SBL-PM-B method has been set to the leave-one-out value attained by k-NN on the training set. Otherwise, the preset value is given. In the tables, in the column 'Size', the average number of the retained examples and the corresponding percentage is given. In the case of the experiments carried out by Wilson and Martinez in the column 'Size' only the percentage of the samples retained in the partial memory is given. It was not clear for us when we read through [4] whether the authors of this paper have taken the one hundred percentage of the partial memory size for the plain k-NN as the size of the original dataset or rather as the size of the training partition. Thus we refrained from calculating the number of the instances retained in the partial memory in case of the results quoted from [4]. If the missing values occurred in the data, the corresponding 1-component distances which could not be calculated have not been included in the computation of the whole similarity function between query case and the samples from the training set. Std denotes that standardization of data has been performed. Wv is short for weighted-vote k-NN classification [17]. Following these abbreviations the number of nearest neighbors taken for classification and the metric is given.

Our aim is to illustrate the influence of the reduction of the training set size and the effect of application of PBC on the results of the classification. The hepatitis data [16] consists of two classes, there are nineteen attributes. The class random choice for this data is 50% and the base rate is 79%. There are 155 cases. The best results have been obtained by application of the SBL-PM-M PBC. The variance on unseen samples is in this case significantly lower than the one, obtained in all the other experiments. The results achieved with the plain SBL-PM-M method are slightly better than those obtained with SBL-PM-B, and are comparable to those obtained by Wilson and Martinez. For this data the selection of prototypes improves the results on the unseen set in comparison to those obtained by taking the entire training set for classification. The summary of the results is given in Table 1.

The second dataset used for our study is the well known iris dataset [16]. It consists of four attributes, 150 cases belonging to three classes, there are 50 cases per class. The EM and SBL-PM-M results are again on the top of the list.

Low variance of the results obtained with the classifiers used in the study of this dataset is typical for this database. The summary of the results obtained for this data is given in Table 2.

The third dataset which has been taken for our experiments is the ionosphere data [16]. There are 34 attributes in this data. The first experiment has been performed by us on the dataset, which consisted of 200 samples. The second calculation has been done on the joint training and test set which contained 350 cases. The class distribution in the case of the data that has been taken for the first experiment is (50.5%, 49.5%) and (64%, 36%) in the second case. It is not clear for us on which data Wilson and Martinez performed their experiments. In the case of our cross-validation experiments, on both versions of the ionosphere data the SBL-PM-M and EM models performed excellent and the results attained with the help of these systems are statistically indistinguishable from the ones obtained with the plain k-NN. Both our PML systems indicated a significant reduction of the demand for the partial memory size in order to perform well on unseen data in comparison to the size of the whole training partition required by k-NN. However is is hard to compare our results to those obtained in [4], we can be sure that our systems are competitive to those of Wilson and Martinez. The results for the ionosphere data have been collected in Table 3.

The last data taken for our experiments is the sonar database [16]. It consists of 60 attributes, there are two classes and the majority class in the joint training and test set which has been taken for our cross-validation experiments constitutes 53.4% of the size of the whole data. The dataset consists of 208 samples. Again, as in the case of the previous dataset that has been studied in this paper, it is not clear which data Wilson and Martinez have taken for their experimentation. The results of SBL-PM-M and EM are almost statistically indistinguishable from those obtained with the plain k-NN, however the first two algorithms require a small fraction of the original training partition data to operate. It seems that our systems perform better than those of Wilson and Martinez. The summary of the results for the sonar dataset is given in Table 4.

5 Conclusions and Further Research

The prototype-based committees have been introduced in this paper in order to improve the case-based data analysis however the study of suitability of PBC in this area will be left for further papers. Therefore, due to the lack of space, in this paper only the generalization abilities of the homogeneous committees constructed out of the SBL-PM-M models with the same input parameters have been investigated. The preliminary numerical experiments have been done with the k-nearest neighbors algorithm which has been used as a classification engine. Experiments with other classifiers are in preparation and will be a subject of a separate paper. Also the performance of heterogeneous committees of different PM models will be examined.

The algorithms described in this paper are designed to provide the alternative, case-based way of data explanation to the rule discovery analysis. They can also serve as a preprocessing step for large datasets in order to speed up classification and learning of other models. In the last case a sufficiently large number of samples should be retained to avoid a 'curse of dimensionality' problem.

SBL-PM-M can give comparable results to other methods and is a valuable PML model. The application of PBC allowed sometimes to increase the classification accuracy and lower the variance making the classifier more stable. In the all experiments in which the PBC were used, one could observe a significant reduction of the partial memory size in comparison to the requirements of the SBL-PM-M model. A very interesting trend that can be noted in the observation of the results of the experiments is the lack of degradation of the classification accuracy of our PML systems under study in comparison to the one obtained with the plain k-NN method. This is surprising as most of the experiments with PML methods known in the literature often indicate a significant degradation of the generalization of the PML systems in comparison to models they are built on.

It would be also interesting to check the sensitivity of SBL-PM-M with and without the application of PBC to noise and study how these methods handle concept drift.

What requires further investigation, is the way how to automatically optimize the adaptive parameters for PBC and SBL-PM-M. The results in this paper have been obtained by the interactive experimentation. The experience gained during conducting calculations for this paper indicates, that these algorithms are very sensitive to the selection of the input parameters.

Acknowledgments: Support by the Department of Physics, Kazimierz Wielki University, which made possible by the author of this paper to take part in this conference is gratefully acknowledged.

References

1. Maloof, M. A., Michalski, R. S.: AQ-PM: A System for Partial Memory Learning. Intelligent Information Systems, Ustroń, Poland (1999) 70-79
2. Maloof, M., Michalski, R. S.: Selecting Examples for Partial Memory Learning. Machine Learning, **41**, (2000) 27-52
3. Maloof, M., Michalski, R. S.: Incremental Learning with Partial Instance Memory. Proceedings of the Thirteenth International Symposium on Methodologies for Intelligent Systems, Lyon, France, (2002), In Foundations of Intelligent Systems, Lecture Notes in Artificial Intelligence, (2366), Berlin:Springer-Verlag, 16-27
4. Wilson, D. R., Martinez, T. R.: Instance Pruning Techniques. In Fisher, D.: Machine Learning: Proceedings of the Fourteenth International Conference. Morgan Kaufmann Publishers, San Francisco, CA, (1997), 404-417.
5. Wilson, D. R., Martinez, T. R.: Reduction Techniques for Instance-Based Learning Algorithms. Machine Learning, **38**, (2000), 257-286
6. Grochowski, M.: Selecting Reference Vectors in Selected Methods for Classification. MSc. Thesis, Nicholaus Copernicus University, Department of Applied Informatics, Toruń, Poland, (2003) (In Polish)

7. Jankowski N., Grochowski, M.: Comparison of Instances Selection Algorithms I: Algorithms Survey. Artificial Intelligence and Soft Computing ICAISC 2004, in Lecture Notes in Artificial Intelligence (LNAI 3070), (Springer), 598-603.
8. Grochowski, M., Jankowski N.: Comparison of Instance Selection Algorithms II: Results and Comments. Artificial Intelligence and Soft Computing ICAISC 2004, in Lecture Notes in Artificial Intelligence (LNAI 3070), (Springer), 580-585.
9. Grudziński, K.: SBL-PM-M: A System for Partial Memory Learning. Artificial Intelligence and Soft Computing ICAISC 2004, in Lecture Notes in Artificial Intelligence (LNAI 3070), (Springer), 586-591.
10. Nelder, J. A., Mead, R.: A simplex method for function minimization. Computer Journal **7** (1965), 308-313
11. Grudziński, K., Duch, W.: SBL-PM: A Simple Algorithm for Selection of Reference Instances for Similarity-Based Methods. Intelligent Information Systems, Bystra, Poland (2000), in Advances in Soft Computing, Physica-Verlag (Springer), 99-108
12. Ortega J., Koppel M., Argamon S.: Arbitrating Among Competing Classifiers Using Learned Referees. Knowledge and Information Systems 3 (2001), 470-490.
13. Bauer E., Kohavi R.: An empirical comparison of voting classification algorithms: bagging, boosting and variants. Machine Learning 36 (1999), 105-142
14. Duch, W.: Similarity Based Methods: a general framework for classification, approximation and association. Control and Cybernetics 29 **4**, (2000), 1-30
15. SBL, Similarity Based Learner, Software Developed by Karol Grudziński. Nicholaus Copernicus University: 1997-2002, Academy of Bydgoszcz: 2002-2006, University of Economy: 2005-2006.
16. Mertz, C. J., Murphy, P. M.: UCI repository of machine learning databases. http://www.ics.uci.edu/pub/machine-learning-data-bases.
17. Dudani S. A.: The distance-weighted k-nearest-neighbor-rule. IEEE Transactions on Systems, Man and Cybernetics 6 (4) 1975, 325-327.

Tables

Table 1. Results for the 10-fold CV Test on Hepatitis Data.

System	**Other parameters**	**Size (%)**	**Test %**
EM-2-2-0.1-[300]	wv, std, $k = 9$, Manhattan	11.9 (8.5)	84.4 ± 0.1
EM-5-5-0.1-[300]	wv, std, $k = 9$, Manhattan	9.1 (6.5)	83.9 ± 0.0
EM-2-2-0.2-[300]	wv, std, $k = 9$, Manhattan	17.2 (12.3)	83.7 ± 0.1
k-NN,	wv, std, $k = 9$, Manhattan	140 (100)	82.9 ± 1.4
M-0.1-[300]	wv, std, $k = 9$, Manhattan	26.5 (18.9)	82.3 ± 1.3
B-82	wv, std, $k = 1$, Manhattan	4.9 (3.5)	81.3 ± 3.7
B-82	wv, std, $k = 9$, Manhattan	11.2 (8.0)	80.3 ± 1.7
k-NN	wv, std, $k = 1$, Manhattan	140 (100)	79.6 ± 1.0
M-0.1-[300]	wv, std, $k = 1$, Manhattan	20.7 (14.8)	79.5 ± 1.0
RT2	$k = 3$, HVDM	? (14.0)	82.0 ± ?
RT3	$k = 3$, HVDM	? (7.8)	81.9 ± ?
k-NN	$k = 3$, HVDM	? (100)	80.6 ± ?
RT1	$k = 3$, HVDM	? (8.7)	76.2 ± ?
H-IB3	$k = 3$, HVDM	? (8.0)	72.8 ± ?

Table 2. Results for the 10-fold CV Test on Iris Data.

System	Other parameters	Size	(%)	Test %
EM-2-2-0.1-[300]	wv, $k = 10$, Euclidean	11.3	(8.4)	96.3 ± 0.1
M-0.1-[300]	wv, $k = 10$, Euclidean	22.8	(16.9)	96.1 ± 0.0
M-0.1-[300]	wv, $k = 1$, Euclidean	21.7	(16.1)	96.0 ± 0.0
EM-2-2-0.2-[300]	wv, $k = 1$, Euclidean	11.2	(8.3)	95.9 ± 0.0
k-NN	wv, $k = 1$, Euclidean	135.0	(100)	95.8 ± 0.1
M-0.1-[300]	$k = 1$, Euclidean	16.9	(12.5)	95.7 ± 0.4
k-NN	$k = 1$, Euclidean	135.0	(100)	95.6 ± 0.3
B-A	$k = 1$, Euclidean	4.9	(3.6)	94.7 ± 1.7
EM-2-2-0.5-[300]	$k = 1$, Euclidean	28.1	(20.8)	92.4 ± 0.6
RT3	$k = 3$, HVDM	?	(14.8)	95.3 ± ?
RT2	$k = 3$, HVDM	?	(16.9)	95.3 ± ?
k-NN	$k = 3$, HVDM	?	(100)	94.0 ± ?
H-IB3	$k = 3$, HVDM	?	(11.0)	92.0 ± ?
RT1	$k = 3$, HVDM	?	(11.7)	89.3 ± ?

Table 3. Results for the 10-Fold CV Test on Ionosphere Data.

System	Other parameters	Size	(%)	Test %
k-NN	wv, $k = 1$, Euclidean	180	(100)	84.5 ± 0.7
M-0.1-[300]	wv, $k = 1$, Euclidean	25.1	(13.9)	84.5 ± 0.3
EM-2-2-0.1-[300]	wv, $k = 1$, Euclidean	5.9	(3.3)	84.1 ± 0.4
EM-2-2-0.1-[300]	wv, $k = 1$, Manhattan	4.6	(1.5)	90.7 ± 0.1
k-NN	$k = 1$, Manhattan	315	(100)	90.5 ± 0.1
M-0.1-[300]	wv, $k = 1$, Manhattan	33.9	(10.8)	90.4 ± 0.2
B-A	$k = 1$, Manhattan	17.8	(5.7)	87.9 ± 1.8
M-0.1-[300]	$k = 1$, Manhattan	34.8	(11.0)	87.2 ± 2.1
RT2	$k = 3$, HVDM	?	(12.1)	88.3 ± ?
H-IB3	$k = 3$, HVDM	?	(13.6)	88.3 ± ?
RT3	$k = 3$, HVDM	?	(7.1)	87.8 ± ?
RT1	$k = 3$, HVDM	?	(5.7)	84.9 ± ?
k-NN	$k = 3$, HVDM	?	(100)	84.6 ± ?

Table 4. Results for the 10-Fold CV Test on the Sonar Data.

System	Other parameters	Size	(%)	Test %
k-NN	wv, $k = 1$, Camberra	187	(100)	86.8 ± 0.3
EM-2-2-0.2-[300]	wv, $k = 1$, Camberra	10.5	(5.6)	86.7 ± 0.9
M-0.1-[300]	wv, $k = 1$, Camberra	21.4	(11.4)	86.6 ± 0.3
EM-2-2-0.175-[300]	wv, $k = 1$, Camberra	8.0	(4.3)	86.5 ± 1.2
M-0.1-[300]	wv, std, $k = 8$, Jacard	16.9	(8.1)	85.5 ± 1.0
k-NN	std, $k = 8$, Jacard	187	(100)	80.0 ± 0.5
B-A	$k = 1$, Camberra	23.7	(12.7)	76.8 ± 7.5
k-NN	$k = 3$, HVDM	?	(100)	87.6 ± ?
RT2	$k = 3$, HVDM	?	(30.8)	81.9 ± ?
RT3	$k = 3$, HVDM	?	(26.9)	78.0 ± ?
H-IB3	$k = 3$, HVDM	?	(15.2)	71.7 ± ?
RT1	$k = 3$, HVDM	?	(24.0)	69.8 ± ?