Committees of Undemocratic Competent Models.

Włodzisław Duch
School of Computer Engineering
Nanyang Technological University, Singapore,
and Department of Informatics,
Nicholaus Copernicus University, Toruń, Poland
http://www.phys.uni.torun.pl/kmk

Łukasz Itert
Dept. of Pediatric Informatics
Children's Hospital Research Foundation
Cincinnati, USA,
and Department of Informatics,
Nicholaus Copernicus University, Toruń, Poland

***Abstract*— Committees of classification and approximation models are used to improve accuracy and decrease the variance of individual models. Each model has an equal right to vote (democratic procedure), despite obvious differences in model competence in different regions of the feature space. Adding competence factors to different models before calculation of the committee decision (undemocratic procedure) improves the quality of the committee. A method for creation of a committee of competent models is described and several real-life empirical tests performed. Significant improvement of results is observed.**

I. Introduction

Although brains are massively parallel computing devices attention mechanisms are used to inhibit parts of the neocortex that are not competent in analysis of a given type of signal. All sensory inputs (except olfactory) travel through the thalamus where their importance and rough category is estimated. Thalamic nuclei activate only those brain areas that may contribute useful information to the analysis of a given type of signals [1]. This may serve as an inspiration for construction of better algorithms for data analysis.

Combining information from different classifiers, called also ensemble learning, mixture of experts, voting classification algorithms, or committees of models [2], is an important and popular subject in machine learning. Conferences and special issues of journals are devoted to this subject (see references in [3]). In some real-life problems, such as predicting the glucose levels of diabetic patients, a large number of different classification algorithms have been applied [3]. The optimal way of combining results of many systems has not yet been found.

Committees of classification models have twofold advantage: they are less biased then individual models, providing flexibility to create more accurate data models, and they stabilize and improve generalization of the whole system, decreasing its variance [4]. Variability of indivdual models used in a committee comes from two sources: data and model construction. Many methods randomize training data, and use stochastic learning algorithms, creating different models at each run. In crossvalidation training, or using boosting, bagging or arcing [2], [4], models are trained on different data subsets. Construction of classification models is determined by many parameters, such as the pruning parameters in decision trees, number of neurons and topology of their connections in neural networks, or regularization parameters in other methods. Selection of features is another source of variability. Recently a framework for similarity based methods (SBM) has been developed [5] and used to create voting committees [6], obtaining for many datasets significant improvements of the accuracy of results. In this paper SBM, neural and decision tree are used with a new voting scheme.

Typical voting techniques follow the democratic majority decision, linear combination or selecting the most confident models. In the mixture of experts neural architecture Jacobs [7] has introduced a gating network to select the most competent model. Very recently Ortega et al [3] used similar idea, a "referee meta-model" deciding which model should contribute to the final decision. These undemocratic procedures exploit the fact that different models may have different areas of competence. The idea of competent voting was also mentioned in [8], but has not been developed further. Global selection of competent models has recently been introduced [9]. Instead of training a meta-model each area of the input space in which a given model makes a number of errors is identified and a penalty factor is used to decrease the influence of this model during the voting.

In the next section methods for model combination are briefly discussed and algorithms for creating committees of competent models are described. In the third section results of a numerical experiment are presented. Finally some conclusions and plans for further work are given.

II. Combining models.

Individual models are frequently unstable [4], i.e. quite different models are created as a result of repeated training (if learning algorithms are stochastic), or if the training set is slightly perturbed [10]. The mixture of models allows to approximate complicated probability distributions quite accurately. With $l = 1 \ldots m$ models providing estimation of probabilities $P(C_i|\mathbf{X};M_l)$ for $i = 1 \ldots K$ classes, one can use the majority voting, average results of all models, select one model that has highest confidence (i.e. gives the largest probability), or set a threshold to select a subset of models with highest confidence and use majority voting for these models.

An empirical comparison of voting algorithms, including bagging and boosting, has been published by Bauer and Kohavi [11]. Tests were made using decision trees and naive

Bayes method. The bagging algorithm uses classifiers trained on bootstrap samples, created by randomly drawing a fixed number of training data vectors from the pool which always contains all training vectors (i.e. drawing does not remove them from the pool). Results are aggregated by voting. AdaBoost (Adaptive Boosting) creates a sequence of training sets and determines weights of the training instances, with higher weights for those that are incorrectly classified. The arcing method uses a simplified procedure for weighting of the training vectors. Bauer and Kohavi [11] provided an interesting decomposition of bias and variance components of errors for these algorithms.

A linear meta-model

$$p(C_i|\mathbf{X};M) = \sum_{l=1}^{m} W_{i,l} P(C_i|\mathbf{X};M_l) \tag{1}$$

provides additional mK linear parameters for model combination, determined using the standard Least Mean Squares (LMS) procedure.

III. Committees of Undemocratic Competent (CUC) Models

In most approaches all models used in a committee are allowed to vote on the final result. Krogh and Vedelsby [12] showed that the committee generalization error is small if highly accurate classifiers disagreeing with each other are used. Xin Yao has used averaging of results with negative correlation between individual models to diversify their pool [13]. Each model does not need to be accurate for all data, but should account well for a different (overlapping) subset of data.

The Similarity Based Models [5] use reference vectors (selected from a training set) and it is relatively easy to determine the areas of the input space where a given model is competent (makes a few errors) and where it fails. Vectors that cannot be correctly classified show up as errors that all model make, but some vectors that are erroneously classified by one model may be correctly handled by another. Although in most methods large committees are prefered, here we shall create small committees, using explicit competence factor functions for each member of the committee. The algorithm proceeds as follows:

1) Preliminaries: Set the stopping criterion: maximum number of models L_{max}, or a minimum number of new vectors N_{min} correctly classified by the model to be added to the committee.
2) Start from a pool of $m > L_{max}$ models $M_l, l = 1 \ldots m$ and optimize their parameters on the training set using a cross-validation procedure;
3) Create an empty set for committee members; tag all training vectors.
4) Until the stopping criteria are true do:
   a) Select from the pool of available models model M_l that is most accurate on all tagged training vectors.
   b) Use this model for all training vectors $\mathbf{R}_i$ to predict classes $C_l(\mathbf{R}_i)$;
   c) if $C_l(\mathbf{R}_i) \neq C(\mathbf{R}_i)$, i.e. model M_l makes an error for vector R_i, determine the area of incompetence of the model, finding the distance $d_{i,j}$ to the nearest vector that model M_l has correctly classified;
   d) set parameters of the incompetence factor $F(||\mathbf{X}-\mathbf{R}_i||;M_l)$; their value should significantly decrease for $||\mathbf{X}-\mathbf{R}_i|| \geq d_{i,j}/2$.
   e) Create incompetence function for the model $F(\mathbf{X};M_l) = \prod_i F(||\mathbf{X}-\mathbf{R}_i||;M_l)$ for all training vectors that have been incorrectly handled.
   f) Untag all vectors that are correctly classified by this model and remove model M_l from the pool of available models.

The incompetence function $F(\mathbf{X};M_l) \approx 1$ in all areas where the model has worked well and $F(\mathbf{X};M_l) \approx 0$ near the training vectors where errors were made. A number of functions may be used for that purpose: a Gaussian function $F(||\mathbf{X}-\mathbf{R}_i||;M_l) = 1 - G(||\mathbf{X}-\mathbf{R}_i||^a;\sigma_i)$, where $a \geq 1$ coefficient is used to flatten the function, a simpler $F(||\mathbf{X}-\mathbf{R}_i||;M_l) = 1/(1+||\mathbf{X}-\mathbf{R}_i||^{-a})$ function or a sum of two logistic functions $\sigma(-||\mathbf{X}-\mathbf{R}_i|| - d_{i,j}/2) + \sigma(||\mathbf{X}-\mathbf{R}_i|| - d_{i,j}/2)$. Since a number of factors enters the incompetence function of the model each factor should quickly reach 1 outside the incompetence area. This is achieved either by using large a values, high slopes of sigmoids or defining a cut-off values where a value 1 is taken.

Such committee of competent models may be used in several ways. In the voting phase nearest neighbor reference vectors should be determined and only those classifiers that are competent should be included in the voting procedure. If no competent models are found the vector given for classification is probably an outlier and should be left as 'rejected' or 'impossible to classify'. Sometimes it helps if all such vectors are removed from the training set, but this is achieved automatically by competent classifiers.

Even simpler way of creating competent committee is introduced if linear combinations are used instead of majority voting. For class C_i coefficients of linear combination are determined from the least-mean square solution of:

$$p(C_i|\mathbf{X};M) = \sum_{l=1}^{m} \sum_{m} W_{i,l} F(\mathbf{X};M_l) P(C_i|\mathbf{X};M_l) \tag{2}$$

The incompetence factors simply modify probabilities $F(\mathbf{X};M_l)P(C_i|\mathbf{X};M_l)$ that are used to set linear equations for all training vectors $\mathbf{X}$, therefore the solution is done in the same way as before. After renormalization $p(C_i|\mathbf{X};M)/\sum_j P(C_j|\mathbf{X};M)$ give final probability of classification. In contrast to AdaBoost and similar procedures [2] explicit information about competence, or quality of classifier performance in different feature space areas, is used here.

IV. Numerical experiments

Computer program implementing CUC has been tested on artificial data and applied to several complex datasets. Classification of 11 English vowels, searching for intron/exon coding areas in DNA, classification of hand-written letters,

and classification of satelite images (all data were taken from the UCI repository [14]). In each case several classification models have been included in the committee: kNN models with different number of neighbors and different distance functions, Feature Space Mapping (FSM) neurofuzzy network, Separability Split Value (SSV) decision tree, and the Incremental Network (IncNet) neural model. All calculations were done using the GhostMiner datamining software developed in the Department of Informatics[1]. Gaussian competence factors were used.

The Vowel dataset contains 528 training, and 462 test vectors, each with 10 continuous features describing one of the 11 vowels spoken several times by 14 people. FSM, IncNet and SSV contributed one model, 3 kNN with Euclidean distance and k=5, 7, 9, two kNN with Manhattan distance and k=7, 9, and one kNN model with Chebyshev distance, k=7, have been used. FSM achieved best training set result (98.7%), but was quite poor on the test set (50.9%). kNN with Euclidean distance and k=7 gave 92.6% on the training set but was most accurate (60.0%) on the test. Models selected for the committee could theoretically account correctly for 99.4% of all training and 88.5% of all test vectors in the sense that at least one model could correctly classify a given vector. Although CUC results on the training set are close to this maximum accuracy (99.2%) test set results are much worse, 62.2%. Majority voting gives 61.8% accuracy. Since the data is rather small (considering large number of classes) the gain due to the use of CUC committee is not significant.

The primate splice-junction gene sequences (DNA) data was used in the Statlog project [15]. It contains a set of 3190 sequences composed of 60 nucleotides. The task is to find if there is an "intron => exon", or "exon => intron" boundary in the string, or neither. 2000 strings are used for training and 1190 for testing. Best results obtained in the Statlog project are collected in Table I. Symbolic features $x \in \{a,c,t,g\}$ have been replaced by probabilities $p(C_j|x) = N_j(x)/N(x)$. Since there are 3 classes instead of 60-dimensional strings of symbols 180 real numbers are used.

6 models have been selected for the committee, two kNN, FSM, IncNet and two SSV models. For the kNN classifers the training accuracies reported in Table I refer to the leave-one-out calculations. FSM was the best single model on the training data (97.0%), with the test set accuracy of 94.5%, while kNN with k=7, Euclidean distance, reached 94.5% on training, but was most accurate (95.3%) on test. The majority voting committee gave 94.7% correct answers, a lower accuracy then obtained by the best model; this may happen since not all models are good in all feature space regions, they should rather specialize in correct classification of certain areas that other models do not handle well. At least one of the 6 models classifies correctly 98.5% of test cases. Although RBF result quoted in Statlog seem to be better than CUC result [15], the RBF model that has been used is quite complex (720 neurons) and should have a large variance, therefore this result may be fortuitous.

[1]http://www.fqspl.com.pl/ghostminer/

TABLE I
COMPARISON OF RESULTS ON THE DNA DATA. RESULTS ARE FROM THE STATLOG BOOK OR OUR OWN CALCULATIONS.

System	Train %	Test %	Remarks
CUC committee	98.1	95.7	
Majority committee	96.6	94.7	
RBF (720 neurons)	98.5	95.9	Statlog
kNN, k=7, Euclidean	94.9	95.3	best single CUC model
Dipol92	98.3	95.2	Statlog
Alloc80	93.7	94.3	Statlog
Quadratic DA	100	94.1	Statlog
LDA	96.6	94.1	Statlog

TABLE II
COMPARISON OF RESULTS ON THE LETTER DATASET. RESULTS ARE FROM THE STATLOG BOOK OR OUR OWN CALCULATIONS.

System	Train %	Test %	Remarks
CUC committee	98.5	96.5	
Majority committee	95.8	95.4	
kNN, k=5, Euclidean	94.8	95.4	best single CUC model
Alloc80	93.5	93.6	Statlog
kNN, k=1, Euclidean	100	93.2	Statlog
LVQ	94.3	92.1	Statlog
Quadratic DA	89.9	88.7	Statlog

The letter dataset contains 16 features derived from OCR images of 26 letters written using more than 20 different fonts. these images were randomly distorted to provide 500-600 training samples for each letter (a total of 15000 training samples), and about 200 samples per letter for testing (5000 test samples). This dataset was used in the Statlog project and the best results are presented in Table II.

The committee included 7 models, five kNN (k=5, 7, 9, 11 Euclidean, and k=5, Manhattan), one FSM and one SSV model. These 7 models can theoreticaly account for 98.7% test samples correctly. The worst training results were obtained by the SSV decision tree (81.2%), and the best one by FSM (97.5%). The worst test result was still obtained by SSV (77.4%), while the best by kNN, k=5, with Euclidean distance function (95.4%). The majority voting committee has improved upon the best kNN result only on the training set, while CUC committee gave significantly better result (the two-tailed t-test with p_{max} as high as 0.9998 still finds it significantly better over other results).

The satimage dataset contains intensities of pixels derived from Landsat satellite images that have been segmented into areas corresponding to 6 different types of surface: red soil, grey soil, damp grey soil, very damp grey soil, cotton crop and vegatation. 4 spectral bands were used and the feature vector contains intensities of the central and 8 surrounding pixels, altogether 36 features quantized from 0 to 255. The training set contains 4435 vectors and the test set 2000 vectors. Best

TABLE III
COMPARISON OF RESULTS ON THE SATIMAGE DATASET. RESULTS ARE FROM THE STATLOG BOOK OR OUR OWN CALCULATIONS.

System	Train %	Test %	Remarks
CUC committee	95.0	91.1	
Majority committee	93.3	89.6	
kNN, k=5, Euclidean	90.8	90.4	best single CUC model
kNN	91.1	90.6	Statlog
LVQ	95.2	89.5	Statlog
Dipol92	94.9	88.9	Statlog
RBF	88.9	87.9	Statlog
Alloc80	96.4	86.8	Statlog

results from the Statlog project are reported in the Table III.

The committee included again 7 models, five kNN (k=5, 7, 9 Euclidean, and k=5, 9 Manhattan), one FSM and one SSV model. These 7 models can theoreticaly account for 95.6% test samples correctly. Again the decision tree was worst both on training (83.7%), and test (81.8%), while FSM was best on training (93.9%) and kNN, k=5, with Euclidean distance function gave the best results on the test set (90.4%). The majority voting committee has improved upon the best kNN result only on the training set, while CUC committee gave significantly better result (the two-tailed t-test with p_{max} as high as 0.97 still finds it significantly better over other results). The majority voting improves the result only for the Vowel database, while CUC results were better comparing to the best model and to majority voting in all cases.

V. CONCLUSIONS

Assigning incompetence factors in various voting procedures, including linear combination of models, is an attractive idea that may significantly improve analysis of difficult problems. Since there is no need to create a single model that handles all data correctly learning may become modular, with each model specializing in different subproblems. A constructive approach to committee growth may be used: after creating initial committee by combining competent models created so far new models should be searched that classify correctly just those vectors, that the committee has still problems with. Significant improvements have been achieved over individual classifiers and over committee based on majority voting. Although more empirical tests are needed to evaluate CUC performance the approach seems to be promissing and may be developed in a number of directions.

So far we have tried to aggregate only a few models generated with different parameters and the selection process has not yet been systematic. Diversification of models by adding explicit negative correlation is also worth considering [13]. CUC voting may be applied to models generated using adaptive boosting or similar algorithms [2]. The competence factors may be calculated during classification, using the training data results as the reference, in the spirit of the nearest neighbor methods. Receiver Operator Characteristic (ROC) may be used instead of accuracies for evaluation of results. A combination of classifiers gives ROC curves that cover a convex combination of all individual ROC curves, allowing to reach better operating points, i.e. detection rates for a given false alarm rate [16]. Boosting schemes may also benefit from adding local competence factors [10]. A number of other options remains to be investigated.

REFERENCES

[1] Thompson R.F. (1993): *The Brain. The Neuroscience Primer.* W.H. Freeman and Co, New York.

[2] Bauer E, Kohavi R. (1999): An empirical comparison of voting classification algorithms: bagging, boosting and variants. *Machine learning* **36**, 105-142

[3] Ortega J, Koppel M, Argamon S. (2001): Arbitrating Among Competing Classifiers Using Learned Referees. Knowledge and Information Systems **3**, 470-490

[4] Breiman, L. (1998): Bias-Variance, regularization, instability and stabilization. In: Bishop, C. (Ed.) Neural Networks and Machine Learning. Springer, Berlin, Heidelberg, New York

[5] Duch W. (2000): Similarity based methods: a general framework for classification, approximation and association, Control and Cybernetics **29**, 937-968

[6] Duch W, Grudziński K. (2001): Ensembles of Similarity-Based Models. Advances in Soft Computing, Physica Verlag (Springer), pp. 75-85

[7] Jacobs R. A. (1997): Bias/Variance Analyses of Mixtures-of-Experts Architectures. Neural Computation **9**, 369-383

[8] Duch W, Adamczak R, Diercksen G.H.F. (2000): Classification, Association and Pattern Completion using Neural Similarity Based Methods. Applied Mathematics and Computer Science **10**, 101-120

[9] Giacinto G, Roli F. (2001): Dynamic Classifier Selection Based on Multiple Classifier Behaviour. Pattern Recognition **34**, 179-181

[10] Avnimelech R, Intrator N. (1999): Boosted Mixture of Experts: An Ensemble Learning Scheme. Neural Computation **11**, 483-497

[11] Bauer E, Kohavi R. (1999): An empirical comparison of voting classification algorithms: Bagging, Boosting and variants. Machine Learning **36**, 105-139

[12] Krogh A, Vedelsby J. (1995): Neural Network Ensembles, Cross Validation, and Active Learning. Advances in Neural Information Processing Systems, MIT Press, **7**, 231–238.

[13] Yao, X., Liu, Y. (1997): A New Evolutionary System for Evolving Artificial Neural Networks. IEEE Transaction on Neural Networks **8**, 694-713

[14] Blake, C.L, Merz, C.J. (1998): UCI Repository of machine learning databases http://www.ics.uci.edu/ mlearn/MLRepository.html. Irvine, CA: University of California, Department of Information and Computer Science.

[15] Michie D, Spiegelhalter D.J. and Taylor C.C. (1994): *Machine learning, neural and statistical classification.* Elis Horwood, London.

[16] Swets J.A. (1988): Measuring the accuracy of diagnostic systems. Science **240**, 1285-93