Voting with a Parameterized Veto Strategy: Solving the KDD Cup 2006 Problem by means of a Classifier Committee

Domonkos Tikk, Zsolt T. Kardkovács[*] , Ferenc P. Szidarovszky[†]
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
H-1117 Budapest, Magyar Tudósok krt. 2., Hungary
Hungary
{tikk,kardkovacs,szidarovszky}@tmit.bme.hu

ABSTRACT

This paper presents our winner solution for the KDD Cup 2006 problem. It is based on the results of three different supervised learning techniques which are then combined in a classifier committee, and finally a single solution is obtained with a voting procedure. The voting procedure assigns weights to each member of the committee according to their average performance on a ten-fold cross-validation test and it also takes into account the confidence values returned by the three algorithms. The final decision of the committee is determined by means of a parameterized veto strategy, which takes into consideration the maximal allowed error rate beside the confidence values of the committee members. The solution presented here won Task 2 and became runner-up at Task 1 in the competition.

1. INTRODUCTION

1.1 Problem description

The object of KDD Cup 2006 was Computer Aided Detection (CAD) of a medical problem. The challenge was related to the automatic identification of a serious and often fatal lung disease, the so-called pulmonary embolism (PE), being the third most common cause of death in the US with more than 650.000 cases occurring yearly. The competitors had to build a series of automatic classifiers that were able to decide with reasonably accuracy and with well-estimated error rate the presence of PE at selected lung nodules.

The competitors were given pre-processed data of PE candidate regions. Each PE candidate corresponded to a voxel (3D pixel) represented by a 116 long vector of descriptive features. An additional data of patient ID (case identifier) was also given. Beside the first 3 features corresponding to the x, y z locations of the candidate, all other features were normalized to $[0, 1]$ range, but the range was arbitrarily shifted within the $[-1, 1]$ interval. The PE data was divided into training and test sets. The training set contained 3033 candidates from 46 positive cases and 20 negative cases; 1391 candidates of test set generated from 33 cases was held back by the organizers until the evaluation phase has been initiated by individual competitors. Candidates of training data was labelled by 0 in case of non-PE, and obtained a non-zero (> 0) identifier of PE region otherwise.

The challenge of KDD Cup 2006 is a supervised learning problem, which consists of three different classification tasks.

1. The first classification task is to label individual PE's.
2. The second classification task is to label each patient as having PE or not.
3. The third classification task is to identify healthy patient with perfect confidence.

The evaluation of the solutions is based on the following measures:

- Task 1: PE sensitivity – the number of PE's correctly identified. A PE is correctly identified if *at least one* of the candidates associated with that PE is correctly labelled as positive. Multiple correct identification of same PE does not increases sensitivity score.
- Task 2: Patient sensitivity – the number of patients for whom *at least one* true PE is correctly identified. The correct identification of PE is determined as above.
- Task 3: Negative prediction value – the value $TN/(TN+FN)$. To qualify this task, the classifier must not label any candidates of a healthy patient as PE.

As a system that often gives false positive prediction — "cries wolf" — cannot be accept for clinical use, therefore additional constraints are present concerning the false positive (FP) rate of solutions of Task 1 and Task 2. To test the scalability of the solution three sub-tasks were given, where organizers set up different limits for the maximal false positive rate per patient for Task 1 and Task 2. As a consequence, the ideal solution is the one which maximizes the sensitivity and also satisfies constraint on maximum allowable FP rate. Violation of maximal FP rate constraints in any sub-tasks resulted in disqualification of the entire solution from the given task. The following sub-tasks were defined for Task 1 and Task 2:

1. Build a classifier that has FP rate per patient not exceeding 2.

[*]Research was sponsored by the Mobile Innovation Centre

[†]F. P. Szidarovszky is also affiliated with Szidarovszky Ltd., Budapest, Hungary.
ferenc.szidarovszky@szidarovszky.com

2. Build a classifier that has FP rate per patient not exceeding 4.

3. Build a classifier that has FP rate per patient not exceeding 10.

The expected solution of Task 3 was identified by the organizer as the "Holy Grail" of the CAD systems. The number of available negative PE cases was very low in the training set (20 cases), and no additional training data had been made available during the competition – in contrast of the anticipation of the organizers. These facts motivated us to turn to the realistic problems of Task 1 and Task 2, and optimize our solutions for those.

1.2 Considerations about PE data set

The KDD Cup 2006 problem contains a number of peculiarities that makes a real challenge to find a good solution for it, and excludes the application of a single, off-the-shelf a classifier algorithm:

- The training data is noisy; this can be explained by the irregularly shaped PE regions. It also cannot be excluded that the training set contains incorrectly labelled data, since candidates are labelled based on proximity to a ground truth mark by an expert.
- The semantics of features is unknown; although a list of feature names is given there is no other information available. E.g., the spacial localization of neighbor voxels is unknown. This fact makes it impossible to exploit spatial correlation in multiple instances of a PE.
- Feature values may be imprecise; feature values may depend on different hardware settings and/or patients.
- The PE data set is quite small from machine learning point of view, and is very imbalanced for both training and evaluation. The positive candidates do not satisfy the IID assumption (independent and identically distributed).
- There is an *a priori* preference on labelling because the number of FP decisions are strictly limited in all tasks and subtasks.
- The evaluation metrics are problem-driven, traditional classifiers usually do not optimize for these.

As a consequence, we found that a good estimation on the number of false positives is the key to increase the sensitivity within the given limitations. In order to do so, we decided to solve the problem by three different classifiers, and then combine individual solutions into a single decision in a classifier committee manner. We expected that these three approaches either eliminate others' weaknesses (false positives) or validate real PEs.

The remaining part of the paper is organized as follows: Section 2 presents the feature selection procedure we applied. Section 3 describes the three different classifier approaches that have been used to solve the problem, and Section 4 presents the voting mechanism by which the final judgement has been obtained. Finally, Section 5 analyzes the submitted solutions and Section 6 concludes the paper.

2. FEATURE SELECTION

Determination of the effective features in a classification problem is a very important issue. Decreasing the size of the feature set always improves the time requirement of the algorithm and can often increase the performance of the method as well. Surveys of feature selection algorithms are given by Kittler and Devijver [11; 5], Siedlecki and Sklansky [22].

In order to rank the elements of the feature set, one may apply e.g. one of the following well-known and simple search methods: Sequential Backward Selection (SBS) [16], or Sequential Forward Selection (SFS) [11] search method. SBS is a simple top-down search procedure where one feature at a time is deleted from the current feature set. At each stage, the attribute to be removed from the feature set is selected from among the elements of the feature set so that the new shrunk set of features yields a minimum value of the criterion function used. SFS is the bottom-up counterpart of SBS search method, where the one feature at a time, having the largest effect on the criterion function, is added to the current feature set.

We experimented with both selection technique on the PE data set in conjunction with a simple multi-layer perceptron algorithm. As criterion function we used the average of areas under ROC (receiver operating characteristic) curve for evaluation.

Though the SFS method is much less time consuming in the early phase, we found — in accordance with our previous experiments [26]; similar results have also been reported in the recent neural network related literature (see e.g. [7; 12]) — that SBS method produced better results. On the other hand, time requirement of SBS is quite significant, therefore in order to save time on performing SBS, we applied it by eliminating systematically two attributes at a time, which attributes' elimination degrades the sensitivity the most on ten-fold cross-validation tests. The optimal value of the criterion function has been attained at a feature set of 62 elements (see also Figure 1). Though, there is no drastic improvement in the performance with these *selected features* — the maximal and minimal values are 8. 315 and 8.0315 — in order to speed up some of the classifiers used in the committee, in 2 out of the 3 classifiers we only employed this feature set.

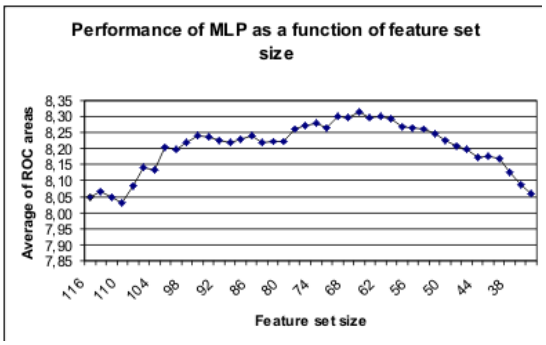

Figure 1: Determination of optimal feature set size by SBS algorithm

3. MEMBERS OF CLASSIFIER COMMITTEE

After the feature selection phase, we applied three different classifiers for prediction: a statistical one, a categorization algorithm, and a multi-layer artificial neural network. They served as members of the classifier committee. Now we describe them in detail.

In order to evaluate the three classifiers, we created randomly ten 90/10 train/test splits from the training data in such a way that the ratio of positive and negative samples equals to the one for the entire corpus. When testing the models, we applied ten-fold cross-validation based on the splits.

3.1 The statistical approach

The first member of the classifier committee has been a statistical approach that we developed for the very problem of KDD Cup 2006.

Despite of the fact that attribute values of data records could have been noisy, we still assumed they can be characterized by means of some interval based patterns, and these patterns might have been extracted by some kind of linear combinations or by a probabilistic model.

While our tests indicated no direct relevance between data records which can be produced by a linear combination, there were also indications that certain PE attribute values concentrate in very specific intervals. On the other hand, there can be found no value (nor interval) which could be assigned uniquely for PEs, moreover the non-PEs (NEs) usually outnumbered the PEs for any values but for intervals.

How to define intervals? Since neither the PEs, nor the NEs had no proportional distributions on any attributes it seemed to be better to define intervals with different sizes. For example, intervals can be defined by PE values with 0 and 1 for Task1, Task2, while NEs for Task3. In other words, if there are N PEs in the training set then the corresponding attribute values of PEs can divide the attribute's range into at most $N + 1$ intervals.

Let I be one of those intervals for an attribute A and $P(I)$, $N(I)$ be the number of PEs and NEs on I, respectively, and let $\mathcal{P}$ be a patient having an attribute value $i \in I$ on attribute A. We use the notation $P(N^+|A)$ and $P(N^-|A)$ for probabilities of PE and NE according to an attribute A, which are defined as follows:

$$P(N^+|A) = \frac{P(I)}{P(I) + N(I)} = 1 - P(N^-|A).$$

The average value (and the entropy) of $P(N^+|A)$ probabilities for all relevant attributes is the overall probability to have a PE (or NE depending on tasks). Since the training set is small, hence it may label PEs or NEs with high probability only because of lack of data. To avoid this problem we also calculated the probabilities for values on overlapping adjacent ($2 \sim 15$) intervals (see also Figure 2). If the probabilities remain high then it is treated as a valid PE judgement.

Since these values are extremely low in general (0.3 at maximum) it was a problem to find a cutting ratio for PE and NE judgements. We ordered the records according to their probability values and we found reasonable on ten-fold cross-validation tests that the cutting ratio should be set to a value for which at most 3–10% of NEs of training data exceeds the limit for Task1 and Task2 depending on different subtasks, and 0% of PEs for Task3. According to cross-validation tests, this method discovers about 60% of PEs with a relatively high false positive rate (see also Table 1 for evaluation).

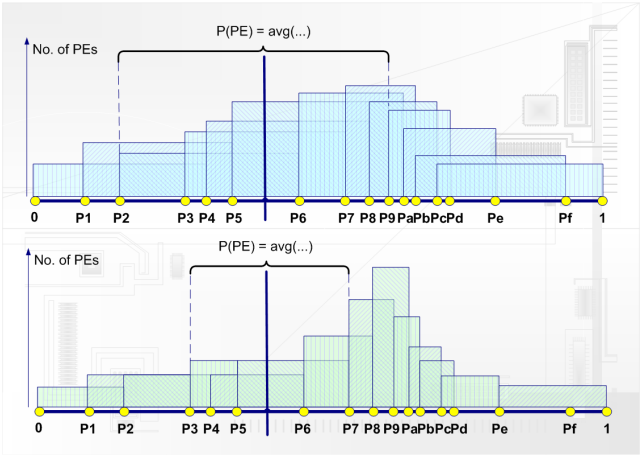

Figure 2: Histogram representation of probabilities calculated by the interval-based approach

3.2 HITEC: mistake-driven online classifier

The second member of the classifier committee was a variant of HITEC – the acronym stands for *hi*erarchical *te*xt *c*lassifier [23; 25].[1] The learning module of HITEC follows the methodology of so-called online (or incremental) mistake-driven algorithms.

3.2.1 *Learning model of online mistake-driven classifiers*

Online classifiers are considered to be one of the most effective learners that are particularly useful when the entire set of training document is not available at once, or when the semantic of the category may change in time after the arrival of new samples [19]. At the problem of KDD Cup 2006 the organizers anticipated that new training data might be available during the competition, hence the application of an online classifier was justified.[2]

The core idea of online mistake-driven (OMD) learning model was first proposed in the seminal paper of Littlestone [14]. It iteratively updates category weights corresponding to features for each category. In this manner it builds up *category profiles* represented as weighted feature vectors for each category in the taxonomy.

OMD algorithms have typically three parameters: a threshold θ, a promotion parameter $\alpha > 1$ and a demotion parameter $0 < \beta < 1$. After initializing the category weights, the algorithm assigns a record to a category if its similarity to the category profile exceeds θ. OMD algorithms may perform multiplicative or additive weight updating schema on active features[3]. They update the category weights in the following two cases of mistake (see also Figure 3):

[1] `http://categorizer.tmit.bme.hu`

[2] We note that despite the announcement no new training data had been published during the competition.

[3] A feature is active if it has non-zero value in the record. This is particularly important in text categorization context.

1. True label is not found: If the algorithm guesses 0 and the true label is 1 then all active weights are promoted by multiplying them with α (multiplicative update) or adding to them $\alpha > 0$ (additive update).
2. Misclassification: If the algorithm guesses 1 but the true label is 0 then all active weights are demoted by multiplying them with β (multiplicative update) or decreasing them by $\alpha > 0$ (additive update).

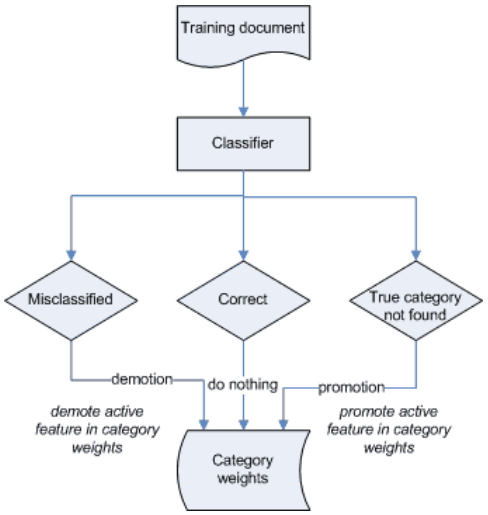

Figure 3: Learning of category profiles by means of weight updating; the schema of mistake-driven online classifier HITEC

In both cases non-active weights remain unchanged. A classifier may often commit both mistakes simultaneously.
Well-known instances of online mistake-driven algorithms are Positive and Balanced Winnow [14] and its variants [4], Perceptron [17], which has been applied to various problems and has been thoroughly investigated from theoretical point of view (see e.g. [14; 15; 1; 8]).

3.2.2 On the learning model of HITEC

HITEC has some key differences compared to classical OMD algorithms, which makes it very effective to work on large document corpora and huge topic taxonomies, the very arena where HITEC was originally designed to perform well (see also [23; 10; 24]). These are the followings:

- it is adapted to hierarchical category systems;
- it applies a novel weight updating method;
- it assigns each category a confidence value for each prediction;
- it allows relaxed greedy category selection strategy, in order to avoid high level mistake in the taxonomy.

Part of these features — see a detailed description in [24] — are also favorable when the learning model of HITEC is customized for the KDD Cup 2006 problem.[4] The sophisticated weight updating method improved the effectiveness of learning, the confidence values could be exploited at evaluation and the relaxed greedy strategy provided the necessary information for the parameterized veto strategy discussed in Section 4. This latter statement is based on the behavior that HITEC tends to simultaneously select both PE and non-PE categories with different confidence values for certain records, which provided us additional possibility to learn more about the problem's characteristics.

[4]At this process HITEC was tailored to comply with the simple input data format, and its architecture was simplified due to the binary categorization task.

3.2.3 Settings

There are two important parameters of training which allows to make a trade-off between recall and precision. By max-variance ($v_{\max}$) one can specify the minimal deviation (ratio) of relevance score of a category w.r.t. the best relevance score ($\Phi_{\max}$) to be considered as a solution. Setting this value low (around 0.2), one can raise recall and lower precision. By threshold (ϑ), one can specify the value of minimal acceptable relevance score. Setting this value low ($0.01 \sim 0.1$ results in better recall values. HITEC selects a category c if its evaluation function, Φ, satisfies

$$\Phi(c) \geq \max\{\vartheta, \Phi_{\max} \cdot v_{\max}\}.$$

At the KDD Cup 2006 problem, where only very few positive training samples were available and false positive decisions were penalized strictly, our goal was to stimulate HITEC for longer learning phase, and to avoid early — and possible — wrong decision during training. To fulfill this requirement, we learned experimentally by the ten-fold cross-validation that $v_{\max} \approx 0.2$ and $\vartheta \approx 0.1$ allowed longer learning phase, consequently, cautious decisions, which finally gave better results.
Using the committee analogy, HITEC had two votes in the classifier committee for a decision, because we created two classifiers, one with the complete feature set (116 features) and one with the selected feature set (62 features), termed as HITEC-116 and HITEC-62 We decided to use HITEC in two work points because this approach achieved the best performance on the training set on most splits of ten-fold cross-validation, and the two versions had different characteristics. For more details see the evaluation in Section 5.

3.3 The neural network classifier

We decided to use feed-forward multi-layer artificial neural networks (ANN) with two output neurons, one for PE and one non-PE decision, but there were many open choices about the particular neural network to be used. To find the best network possible, we created a set of candidate networks, and selected the best performing one to be a member in the classifier committee.

3.3.1 Collecting candidate neural networks

We collected the candidate neural networks with the following steps:

1. Identification of open choices.
2. Defining set of options as answers to each identified choice.
3. Generating candidate neural network for each combination of answers for open choices.

The identified open choices and their defined options were the following:

- **Choice 1. Set of features**
  We had to determine the set of features to be used as inputs. Our selected options were to use all features or the selected features (see section 2) as inputs.
- **Choice 2. Hidden layers**
  We had to determine the number of hidden layers to use. Our selected options were to use no or one hidden layer.
- **Choice 3. Number of hidden neurons**
  We had to determine the number of hidden neurons. Our selected options were to use the same, the half and the quarter of the number of input neurons. (These ratios are not exact. We used 66 and 33 for the half and quarter of all features, and 33 and 16 for the half and quarter of selected features.)
- **Choice 4. Characteristic function of hidden neurons**
  We had to determine the characteristic function to use for the hidden neurons. Our selected options were the linear and the logistic characteristic functions.
- **Choice 5. Characteristic function of output neurons**
  We had to determine the characteristic function to use for the output neurons. Our selected options were the linear and the logistic characteristic functions.
- **Choice 6. Initial connection weights**
  We had to determine the initial connection weights. Our selected options were to use all zeroes and random weights with random seeds 1 to 5. (We used the pseudo-random number generator of the development platform in question).

3.3.2 Training issues

We used the standard gradient descent method for error propagation with the same constant learning rate for all candidates.
We used a non-standard way of modifying the learning rate during training. After each modification of the weights, we checked, if the error rate has decreased. If it didn't, we undid the changes and halved the learning rate. We repeated this until either the error rate decreased (in which case the training step was successful) or the learning rate became practically zero (in which case the training stopped). After each successful training step, we increased the learning rate by half.

3.3.3 Reducing the number of candidates

We didn't have the computing resources to train all candidates on all cross-validation splits. We assumed, that the ten splits don't differ enormously in their usefulness for training neural networks. So we trained all candidates on our first cross-validation split, and dropped the candidates that fulfilled at least one of the following criteria:

1. It had a lot worse validation error rate than some other candidate.
2. It had a similar error rate than some other candidate, but had more hidden neurons or more input neurons. (We introduced that criterion due to our lack of computing resources.)

By this procedure, we managed to drop almost ninety percent of the candidates.

3.3.4 Evaluating candidates' performance

The first step was to choose the cross-validation split to use for candidate evaluation. Since the training set is small and noisy, we aimed at choosing the split on which the candidates tended to "overlearn" the least. We represented overlearning with the difference between the final training error rate and the validation error rate (i.e. overlearning rate).
For each split we generated the averages of the overlearning rates of a small number of top performing candidates (that is, the ones with the smallest validation error rates). The split we considered to be the best for candidate evaluation (i.e. evaluation split) was the one with the smallest average overlearning rate.
The second step was to find the best candidate based on their performance on the evaluation split.
We chose the candidate for which the following two conditions held:

1. Its overlearning rate was in the bottom part of the candidates.
2. Its validation error rate was the smallest of those satisfying condition 1.

3.3.5 Chosen candidate

The neural network we chose had the following choice options:

- **Choice 1.**: It had input neurons for the selected features.
- **Choice 2.**: It had one hidden layer.
- **Choice 3.**: It had 33 hidden neurons.
- **Choice 4.**: Hidden neurons had the logistic characteristic function.
- **Choice 5.**: Output neurons had the logistic characteristic function.
- **Choice 6.**: We used one of the random initial weight sets.

3.3.6 Final neural network preparation and decision

The neural network with the above choices was then trained with the complete evaluation cross-validation split.
On the ten-fold cross-validation tests, we found that the network is quite accurate for labelling PEs, however, it finds only a minor part (30–50%) of all possible true positives.

4. DECISION OF THE CLASSIFIER COMMITTEE

The original idea of classifier committees (a.k.a. ensembles) is that, given a task that requires expert knowledge to perform, several experts may be better than one if their individual judgments are appropriately combined [19]. The different classifiers are applied to the same task independently, and their outcome are combined into a single decision. Classifier committees have been applied successfully to all kind of classification problems. Numerous papers have reported

that classifier committees have substantial advantages on single-classifier approaches in the field of text categorization [29; 18; 3], mining concept-drifting data streams [28], gene data classification [2], etc. An ensemble based method has also been used in the winner solution of KDD cup 2005 [20; 21].

A classifier committee can be characterized by the committee members (with possible multiplicity), and the choice of a combination function. In [27] it was shown that classifier members should be as independent as possible to form an ideal committee.

Concerning the combination function several rules have been investigated. The simplest one is the *majority voting*, where the binary outputs of the k (odd number) classifiers are put together, and the decision is taken that reaches the majority of $\frac{k+1}{2}$ votes. Another policy is the *weighted linear combination* of committee members, where weights represent the effectiveness of members. Certain combination functions selects dynamically some classifiers at the classification of each record, those ones that produced the best results at the validation set on records being similar to the actual one. Example of such rules are called *dynamic classifier selection* and *adaptive classifier combination* [13].

In our experiments, three approaches formed the classifier committee, where the statistical and neural network based approaches had one vote each, and HITEC had two votes (HITEC-62 and HITEC-116). We applied a novel strategy to combine the binary decisions of the classifier to a single one, that we call *voting with parameterized veto strategy*. Intuitionally, a positive decision can be accepted, if there are more pro-votes than contra-votes, allowing also neutral or close-to-neutral votes. Confidence values of pro and contra votes are cumulated and the decision is made based on the margin between them.

The main points of our decision strategy were as follows. In order to solve successfully the KDD Cup 2006 problem, one had to simultaneously optimize for high sensitivity and low FP rates. The different FP rates required different decision strategy for each task and subtask. With the concept of *veto* we could control the number of strength of PE decisions: if any of the committee members voted for non-PE we considered its vote as a possible veto of PE decision. Obviously, the strength of veto should depend on the effectiveness of the classifier and the confidence value of non-PE vote. Therefore we applied the parameterized veto strategy where this factors were taken into consideration and we applied different veto thresholds for different subtasks according to the allowed FP rate.

As a consequence, for Task1 and Task2 we defined the following rules.

- **Rule 1**. If all committee members label a record as PE then the committee's decision is PE.
- **Rule 2**. If only three or two members label a record as PE, and there is no veto (high confidence non-PE vote) method(s) then we label it as PE.
- **Rule 3**. If only one classifier labels a record as PE and there is not even a weak veto then the decision is PE.
- **Rule 4**. It is non-PE otherwise.

Alternatively, the following equation can be viewed as an approximation of these rules:

$$\text{Decision}(\vec{X}) = \left(\frac{\sum_i \eta_i P_i^+(\vec{X})}{\sum_i \eta_i P_i^-(\vec{X})} > 1\right)$$

where $\vec{X}$ is the examined record, and η_i is a parameterized weight of a certain approach which was calculated according to its overall error rate and the maximum number of allowed FP rate. ANN made a simple binary decision without confidence value therefore probabilities were treated as 1 for the given decision value while 0 for the other.

Note that, $P_i^+ + P_i^- \neq 1$ in general in classifier members (except ANN), hence e.g. statistical model combines cutting ratio with calculations of entropies of several different settings, and HITEC evaluates PE and non-PE cases independently. In other words it means that the same classifier can simultaneously vote for PE and non-PE for the very same record.

As a consequence, the veto parameter depends on

1. the allowed FP rates: 2, 4, 10;
2. the effectiveness of the classifier members that is calculated based on the *worst-case error rate* experienced on the ten-fold cross-validation tests to increase the reliability of the individual member and the entire committee.
3. the confidence values of the PE and non-PE votes (if such information were produced by the classifier)

As we had no time to adjust all veto parameters according to FP rates for each subtasks during the 24 hours submission period, we optimized our solution for subtasks 1b and 2b.

5. EVALUATION

We performed tests on the ten-fold cross-validation splits described in the beginning of Section 3.

We decided to apply HITEC at two settings because we found that HITEC-116 and HITEC-62 have somewhat different characteristics, and these models performed the best among the three approaches. HITEC-116 tended to give more balanced prediction for all test splits, its recall was ranged from 52% to 60% depending on the split and on the settings of training. It often produced balanced (neutral or close-to-neutral) decisions which could be advantageous at the voting strategy. On the other hand, HITEC-62 was likely to produce sharper decisions, it tended to overlearn the actual split since its recall was ranged on a wider interval from 50% to 71%. It very seldom produced balanced decisions where both of PE's and non-PE's confidence value was non-zero.

The test set originally contained 1391 records (156 true PEs) from 23 patients, but during the evaluation data of two patients with identifiers 3111 and 3126 were removed, because their data might have been corrupted, and organizer did not intend to skew the results by using them. Therefore there were 1297 records in the test set (137 true PEs), and the FP rates were calculated based on 21 patients. Because many competitors protested against the change in the test set, we deem it worthwhile to investigate and publish our results for both 21 and 23-patient cases.

Table 1: Overall contributions of approaches for KDD Cup result

	Approaches				Combinations						Task1			Task2		
	Statistical	ANN	Hitec-62	Hitec-116	Stat & ANN	Stat & Hitec-62	Stat & Hitec-116	ANN & Hitec-62	ANN & Hitec-116	Hitec-62 & Hitec-116	Voting for 2FP	Voting for 4FP	Voting for 10FP	Voting for 2FP	Voting for 4FP	Voting for 10FP
Number of true PEs (23)	43	42	32	77	22	22	36	24	33	27	51	80	90	49	64	98
Number of false PEs (23)	62	12	18	91	8	5	19	3	5	8	21	57	109	18	58	144
Uniquely found PEs (23)	6	7	3	27												
Number of true PEs (21)	38	35	27	71	19	19	34	21	29	24	45	72	80	43	56	87
Number of false PEs (21)	58	12	13	89	8	4	19	3	5	7	20	54	102	17	54	135
Uniquely found PEs (21)	4	5	2	26												

Table 1 summarizes the performances of the committee members, their combinations and our submissions for Task 1 and Task 2. It can be seen that HITEC-116 found in overall and uniquely the most true PE (77 and 27, resp.), and produced the highest recall. On the other hand, its precision was the second worst among the four classifiers. In this regard, ANN gave the least incorrect predictions (12 out of 52), but its recall was low. The statistical approach and HITEC-62 produced in-between solutions, the former one had better recall, while the latter one better precision, and importantly they uniquely identified 6 and 3 true PEs, respectively. As it will be shown next, the combination of the approaches gave considerably better result than the individual ones. Precision, recall and FP rate values are tabulated in Table 2. For HITEC-116 and HITEC-62 we applied threshold 0.48 and 0.55, respectively, on the confidence value of PE decision as the condition of PE.

Analyzing the differences of our submissions for the 21 and 23 patients test sets we can state the following. The FP rates of our submissions were very low, they reached maximally the 63% (23) and 64% (21) of the limits, and the precision values of both cases are about the same. By observing the differences between recalls of our submission for 21 and 23 patients cases, we can see it is slightly higher at 4 of out the 6 subtasks for the 21 patients set, and slightly lower otherwise. The maximal increase was 2.48%. As a conclusion, we can state that our solution performed equally well on both test sets.

The members of the classifier completed each others' prediction in the sense that 20, resp. 18 PEs were discovered by all methods (23, resp. 21 patients cases, only 2 FPs among them!), and about 61% of true PEs were labelled thus by at least one of these approaches. One can also observe that 30–50% of the FPs have been eliminated by the combined decision of the classifier committee. The strength of veto algorithm can be illustrated by the fact that in case of Task 1a 112 PE votes promoted by at least two members of the committee were filtered by this strategy, 77 of the 112 were FP, and only 35 were TP. In case of Task 2a the these numbers are 81 FP and 38 TP out of 119 filtered PE votes.

At submission, we had assumed for qualification purposes that the combined algorithm might have an 50% worst-case

Table 2: Precision, recall and FP rate values of individual committee members and combined submissions (A: 23 patients case; B: 21 patients case)

Method/submission	Precision (%)	Recall (%)	FP rate
Statistical	40.95	27.56	2.70
ANN	77.77	26.92	0.52
HITEC-116	45.83	49.35	3.96
HITEC-62	64.00	20.51	0.78
Task 1a	70.83	32.69	0.91
Task 1b	58.39	51.28	2.48
Task 1c	43.06	57.69	4.74
Task 2a	73.13	31.41	0.78
Task 2b	52.46	41.02	2.52
Task 2c	40.50	62.82	6.26

A: 23 patients case

Method/submission	Precision (%)	Recall (%)	FP rate
Statistical	39.58	27.74	2.76
ANN	74.47	25.55	0.57
HITEC-116	44.38	51.82	4.23
HITEC-62	67.50	19.71	0.62
Task 1a	69.23	32.85	0.95
Task 1b	57.14	52.55	2.57
Task 1c	43.96	58.39	4.85
Task 2a	71.67	31.39	0.81
Task 2b	50.91	40.88	2.57
Task 2c	39.19	63.50	6.42

B: 21 patients case

error rate, however in the reality, it was significantly lower. We investigated this factor, i.e. how we underestimated — due to the lack of time for optimization — the performance of the committee and particularly its parameterizable members after the true labels of test data had become available (as mentioned before we optimized HITEC and the committee for 4FP subtasks). It has been reasonable to check how the selection of threshold parameters influences HITEC's performance. As it can be observed on Figure 4 we could select these thresholds more efficiently. It is noteworthy to remark that the high FP rate of HITEC's solutions at low

threshold can be balanced by the more conservative decisions of the ANN and statistical method in the committee's decision. We remark that with the combination of HITEC-116 and HITEC-62 and with very low threshold we could achieve recall values around 92% (144/156 and 125/137 in case of the two test sets) at about 23% precision.

The Task 1 and Task 2 (PE and patient sensitivity) of submissions were evaluated according to their raw score on the complete test set, as well as on a bootstrap estimate of their mean performance and confidence interval on that mean [6; 9]. As noted by the organizers, "the bootstrap estimate penalizes some competitors because they fail some rounds. But this competition was *designed* to have very stringent criteria and to force competitors to be fairly conservative."

Our conservative solutions achieved scores 1.27 (as average of: 2FP – 0.93, 4FP – 1.36, 10FP – 1.51) and 13.58 (as average of 2FP – 11.50, 4FP – 14.34, 10FP – 14.90) for Task 1 and Task 2, respectively.

6. CONCLUSIONS AND FUTURE RESEARCH

In this paper we presented our classifier committee based method that has been applied for the KDD Cup 2006 problem. We applied a novel parameterized veto strategy to determine the decision of the committee. We identify three key points why our submission was successful. First, we applied four different approaches to tackle the problem. Second, all these algorithms are general in the sense that they have no pre-assumptions on semantics of attributes. Third, some of the algorithms allowed us to apply thresholding based on confidence level. And finally, we optimized our solution for 4 false positive rate (subtasks 1b and 2b).

As we indicated at the evaluation, our submission was quite timid in order to stay under FP rate limits. Plenty of room left for the optimization of the method by tuning the members of the classifier committee, increasing the number of members in the committee, and creating better heuristics for the parameterized veto strategy. In the future we intend to investigate these issues.

Acknowledgement

Authors thank Gábor Takács and Bottyán Németh to help in executing the feature selection with SBS method. We also thank György Biró for his advices concerning the internal set-up of HITEC classifier and Zoltán Bálint for his valuable help during our discussion. Domonkos Tikk was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Science.

7. REFERENCES

[1] A. Blum. Empirical support for Winnow and weighted-majority based algorithms: results on a calendar scheduling domain. In *Proc. of 12th Int. Conf. on Machine Learning*, pages 64–72, San Francisco, CA, 1995. Morgan Kaufmann.

[2] S.-B. Cho and J. Ryu. Classifying gene expression data of cancer using classifier ensemble with mutually exclusive features. *Proc. IEEE*, 90:1744–1753, 2002.

[3] W. J. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. In *Proc. of the 19th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, pages 307–315, 1999.

[4] I. Dagan, Y. Karov, and D. Roth. Mistake-driven learning in text categorization. In C. Cardie and R. Weischedel, editors, *Proc. of the 2nd Conf. on Empirical Methods in Natural Language Processing (EMNLP 97)*, pages 55–63. Association for Computational Linguistics, Somerset, New Jersey, 1997.

[5] P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall, London, 1982.

[6] R. O. Duda, P. Hart, and D. G. Stork. *Pattern classification*. Wiley, New York, 2001.

[7] A. A. Ghaibeh, S. Kuroyanagi, and A. Iwata. Efficient subspace learning using a large scale neural network Combnet-II. In *Proc. of the 9th Int. Conf. on Neural Information Processing (ICONIP'02)*, volume 1, pages 447–451, Singapore, 2002.

[8] A. R. Golding and D. Roth. Applying Winnow to context-sensitive spelling correction. In *Proc. of 13th Int. Conf. on Machine Learning*, pages 182–190, Bari, Italy, 1996. Morgan Kaufmann.

[9] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2001.

[10] Z. T. Kardkovács, D. Tikk, and Z. Bánsághi. The ferrety algorithm for the KDD cup 2005 problem. *ACM SIGKDD Explorations Newsletter*, 7(2):111–116, 2005.

[11] J. Kittler. Feature set search algorithms. In C. H. Chen, editor, *Pattern Recognition and Signal Processing*, pages 41–60. Sijthoff & Noordhoff, Alphen aan den Rijn, The Netherlands, 1978.

[12] M. Kugler, K. Aoki, S. Kuroyanagi, A. Iwata, and A. S. Nugroho. Feature subset selection for support vector machines using confident margin. In *Proc. of the IEEE Int. Joint Conf. on Neural Networks (IJCNN'05)*, volume 2, pages 907–912, Montréal, Canada, 2005.

[13] Y. H. Li and A. K. Jain. Classification of text documents. *The Computer Journal*, 41(8):537–546, 1998.

[14] N. Littlestone. Learning quickly when irrelevant attributes around: A new linear-threshold algorithm. *Machine Learning*, pages 285–318, 1988.

[15] N. Littlestone. Comparing sereval linear-threshold learning algorithm on tasks involving superfluous attributes. In *Proc. of 12th Int. Conf. on Machine Learning*, pages 353–361, San Francisco, CA, 1995. Morgan Kaufmann.

[16] T. Marill and D. Green. On the effectiveness of receptors in recognition systems. *IEEE Trans. on Information Theory*, 9:11–17, 1963.

[17] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in brain. *Psychological Review*, pages 386–407, 1958. (Reprinted in *Neurocomputing*, MIT Press, 1988).

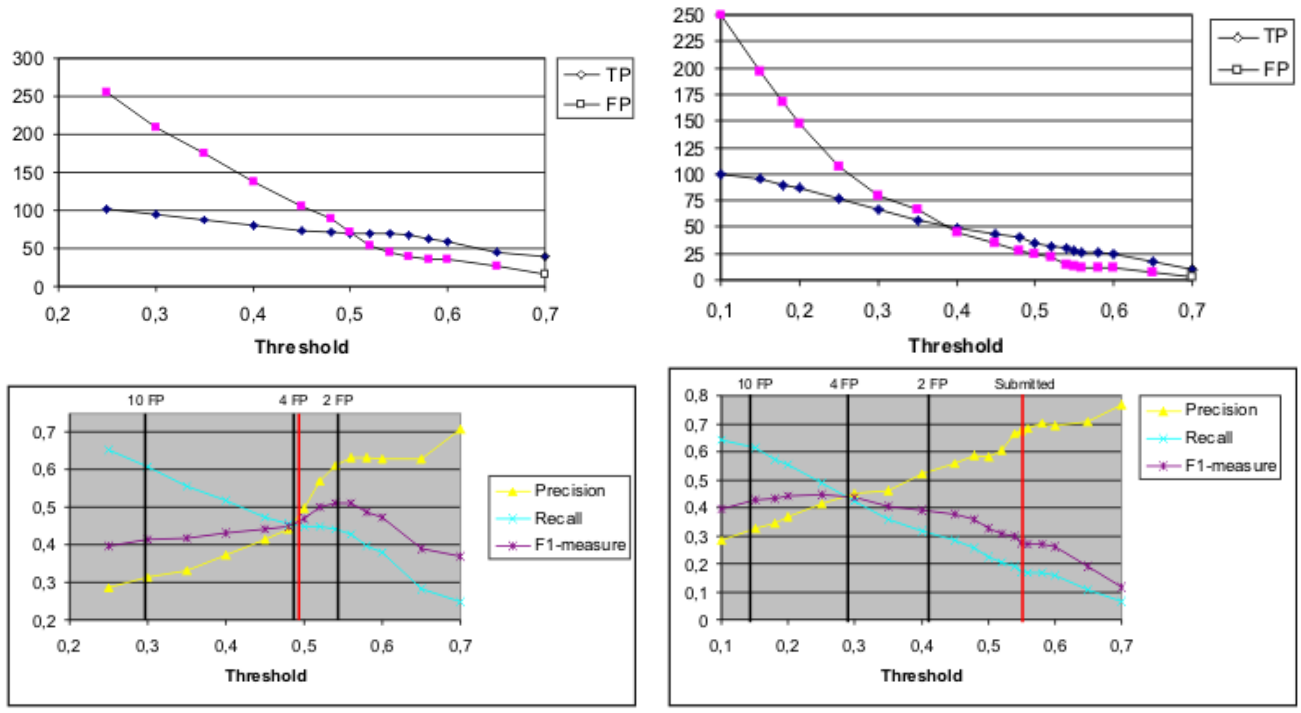

Figure 4: Performance of HITEC as the function of threshold. Upper diagrams shows number of TP and FP decisions depending on the threshold, lower diagrams shows the traditional precision, recall, F_1-measure values vs. threshold (left side HITEC-116, right side HITEC-62). We indicated by black vertical bars the threshold values where the allowed FP rate, red bar locates the FP rate of the solution used in our submissions. Tests are from 21 patients case.

[18] R. E. Schapire and Y. Singer. Boostexter: a boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[19] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, March 2002.

[20] D. Shen, R. Pan, J.-T. Sun, J. J. Pan, K. Wu, J. Yin, and Q. Yang. $Q^2C@UST$: Our winning solution to query classification in KDDCUP 2005. *ACM SIGKDD Explorations Newsletter*, 7(2):100–110, 2005.

[21] D. Shen, J. T. Sun, Q. Yang, and Z. Chen. Building bridges for web query classification. In *Proc. of SIGIR'06, 29th ACM Int. Conf. on Research and Development in Information Retrieval*, pages 131–138, Seattle, Washington, USA, 6–11 August, 2006.

[22] W. Siedlecki and J. Sklansky. On automatic feature selection. *Int. J. of Pattern Recognition and Artificial Intelligence*, 2(2):197–220, 1988.

[23] D. Tikk and G. Biró. Experiment with a hierarchical text categorization method on the WIPO patent collection. In *Proc. of the 4th Int. Symp. on Uncertainty Modeling and Analysis (ISUMA'03)*, University of Maryland, USA, September 21–24, 2003.

[24] D. Tikk, G. Biró, and A. Törcsvári. A hierarchical online classifier for patent categorization. In H. A. do Prado and E. Ferneda, editors, *Emerging Technologies of Text Mining: Techniques and Applications*. Idea Group Inc., 2006. (in press).

[25] D. Tikk, G. Biró, and J. D. Yang. A hierarchical text categorization approach and its application to FRT expansion. *Australian Journal of Intelligent Information Processing Systems*, 8(3):123–131, 2004.

[26] D. Tikk, T. D. Gedeon, and K. W. Wong. A feature ranking algorithm for fuzzy modelling problems. In J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, editors, *Interpretability Issues in Fuzzy Modeling*, number 128 in Studies in Fuzziness and Soft Computing, pages 176–192. Springer-Verlag, Heidelberg, 2003.

[27] K. Tumer and J. Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Sci.*, 8(3–4):385–403, 1996.

[28] H. Wang, W. Fan, P. S. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 226–235, Washington D.C., USA, 2003.

[29] S. M. Weiss, C. Apte, F. J. Damerau, D. E. Johnson, F. J. Oles, T. Goetz, and T. Hampp. Maximizing text-mining performance. *IEEE Intelligent Systems*, 14(4):2–8, July/August 1999.

About the authors:

Domonkos Tikk is a senior researcher at Department of Telecommunications and Media Informatics (DTMI) at Budapest University of Technology and Economics (BUTE). He has received his Ph.D. in 2000 from the same institution in fuzzy systems. His research covers text and data mining, natural language processing (NLP), internet search engines, pattern recognition, fuzzy systems and soft computing techniques. Recently he is focusing on classification problems, particularly hierarchical text categorization, and NLP related problems. He was the team leader of this KDD Cup team.

Zsolt T. Kardkovács is an assistant lecturer at DTMI, BUTE. His research interests include (deductive and relational) databases, logic, knowledge representation, natural language processing, and human computer interaction. He had the most contribution in the veto strategy.

Ferenc P. Szidarovszky is a postgraduate student at BUTE, and owner/CEO of Szidarovszky Ltd. His research interests are data mining, text categorization, information security and natural language processing.