# A Statistical Unified Framework for Rank-Based Multiple Classifier Decision Combination

Afşar SARANLI* and Mübeccel DEMİREKLER
Speech Processing Laboratory
Department of Electrical and Electronics Engineering
Middle East Technical University, Ankara, Türkiye.
e-mail: {afsars,demirek}@metu.edu.tr

Revised, November 10, 1999

**Abstract**

This study presents a theoretical investigation of the rank-based multiple classifier decision combination problem, with the aim of providing a unified framework to understand a variety of such systems. The combination of the decisions of more than one classifiers with the aim of improving overall system performance is a concept of general interest in pattern recognition, as a viable alternative to designing a single sophisticated classifier. The problem of combining the classifier decisions in the raw form of candidate class rankings is formulated as a discrete optimization problem. The objective function to be maximized is selected as the overall probability of correct decision. This formulation introduces a set of observation statistics about the joint behavior of the classifiers which are to be estimated by observing the classifiers operated on a cross-validation test set. The resulting binary programming problem is shown to have a simple and global optimum solution but which also necessitates a prohibitive number of observation statistics. From the objective function expansion, the problem observation space is defined and a method based on partitioning is introduced to reduce its prohibitive dimensionality. Within this partitioning formalism called as the Partitioned Observation Space (POS) Theory, the number of behavior observation statistics can be reduced to levels which are feasible to estimate from the available cross-validation test data. It is shown by examples that such specific partitionings can be defined when reasonable assumptions or prior knowledge about the classifiers are incorporated into the problem domain. It is also demonstrated that certain specific partitionings of the classifier observation space effectively lead to the Highest Rank, Borda Count and Logistic Regression rank-based decision combination methods from the literature. The analysis presented is general and promises to lead to a class of algorithms for rank-based decision combination. The potential of the theory and practical issues in implementation are illustrated by applying it in a real-life phonetic discrimination problem from speech pattern classification with encouraging results.

**Keywords:** statistical classifier combination, statistical decision combination, statistical pattern recognition, multiple classifier systems, ranks, classifier observation space, event space partitioning, Bayesian formalism, phonetic discrimination, speech processing.

*Corresponding author. E-mail: afsars@metu.edu.tr

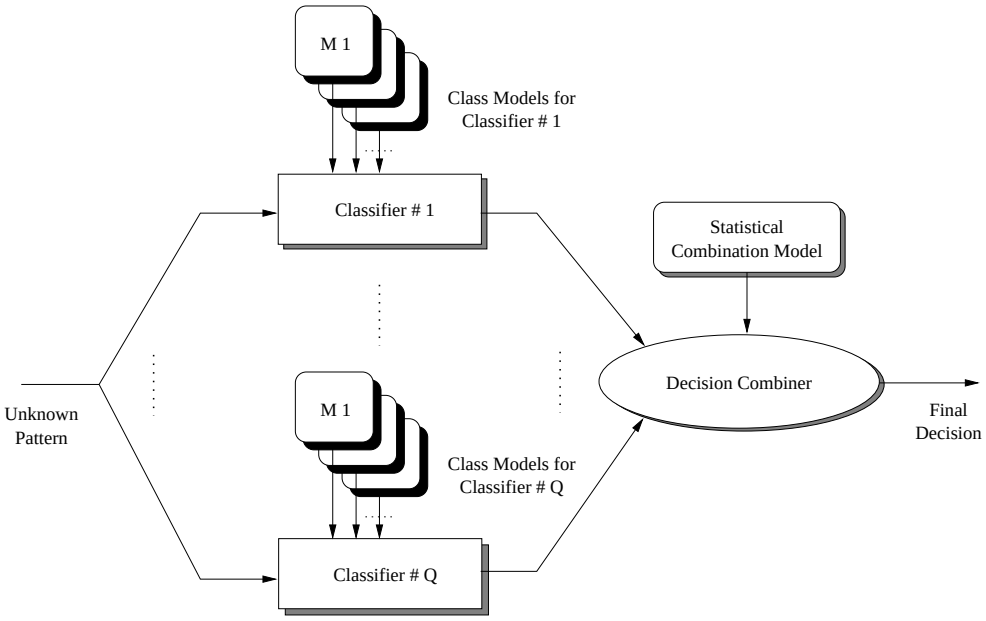


Figure 1: Multiple classifier decision combination process. The first layer consist of individual classifiers which may have different architectures but with outputs in a common form such as rankings of the candidate classes. The second layer is a *decision combination process* which operates on the raw decisions of the low level classifiers to generate the multiple classifier system overall decision.

# 1 Introduction

The last decade has witnessed extensive research on the problem of combining the classification data supplied by a multitude of classifiers with the aim of improving the generalization and hence the overall performance of the system. This approach, illustrated in Figure 1, is different from the extensively studied task of designing and fine-tuning a single sophisticated classifier and is a viable alternative as suggested in the literature with promising experimental results. The individual classifiers operating in the system are assumed to be *given* and they may be simple or sophisticated. It has also been shown that the combination of simple and fast classifiers of complementary nature can match a single sophisticated classifier in performance [1].

Contributions have been made or some form of decision combination system have been attempted in a variety of pattern recognition fields. These include machine printed word/character recognition [2], handwritten character recognition [3, 4, 5, 6, 7, 8, 9], speaker recognition,

[10, 11, 12, 13], face identification [14, 15], text to phoneme translation [16], remote sensing [17, 18], military target recognition [19] and biomedical signal processing [20, 21]. The neural networks community has also been active on this approach [22, 6, 23, 16, 24, 25, 26, 7].

In fact, the idea of combining multiple sources of evidence for decision making has been an extensive topic of research within the artificial intelligence literature for a long time. Two main approaches which still form the backbone of many approaches found in pattern recognition literature are the Bayesian Approach [27] and the Dempster-Shafer Approach [28]. The diversity of the fields in which the problem has been considered and encouraging results have been reported show that multiple classifier decision combination systems are of considerable interest to a large number of pattern recognition fields.

Xu and his colleagues have categorized multiple classifier decision combination systems with respect to the type of raw output information from each classifier [3], resulting in three categories: The classifier outputs may be single class labels (Type 1), rankings of a subset of candidate classes from highest to lowest "likelihood" (Type 2) or the complete set of *similarity score* values for the candidate classes leading to such rankings (Type 3).

There exist a variety of motivations which led to these different types of classifier combination systems. For example, it has been shown by the neural networks community that properly trained neural networks successfully estimated the class posterior probabilities [29, 30] which could be used for decision within the Bayesian formalism. This observation led to the idea of combining neural network outputs at the similarity score level, often by simple averaging, by weighted averaging, by order statistics or by some other techniques with the aim of reducing the variance of the estimation errors, hence improving the overall classification accuracy [23, 6, 25, 22, 17, 18, 8, 5]. It has been shown in [22] that this reduction in variance is indeed associated with a variance reduction in the *deviation* of the estimated decision boundary from the optimum Bayesian decision boundary. Expected reduction in classification errors is shown to decrease as the errors of the classifiers become more and more correlated [22].

Another motivation for combination arises when the individual classifier generalization performances (i.e., their true performance measure [16, 31]) are poor due to noisy inputs, insufficient and noisy training data. This motivation is the "integration" of different classification techniques,

perhaps based on different features, algorithms and architectures to achieve a better generalization performance for varying conditions [6] and is expressed by most of the researchers in the field. Although this is a generic motivation, the desire to integrate classifiers as diverse as possible, often with incompatible/incomparable outputs increased the attractiveness of especially Type 1 systems where only the final class labels are supplied by individual classifiers [3].

When a single classifier is considered, a final class label (the identified class) is obviously the only desired output. However, for combining multiple classifier outputs, using only this abstract level may lead to a loss of valuable information which could be harnessed to improve performance. The problems of output incompatibility, incomparability and scaling are of concern for using classifier outputs in the similarity score form but such problems do not occur when using classifier outputs in the form of candidate class rankings. It is an observed fact that when a pattern is misclassified, the true class of the pattern is often close to the top of the ranking [2]. A resulting ranking gives valuable information about classifier behavior for imperfect classifiers. Despite this fact and that there have been good theoretical attempts to analyze the properties and behavior of Type 1 and Type 3 decision combination systems [3, 18, 23, 22, 6, 16], there have been few attempts to develop an understanding of the rank-based decision combination systems [2, 24]. In [2], Ho proposes, without attempting an in-depth theoretical analysis, to generalize the *Borda Count* method by linearly weighting the individual classifiers while Al-Ghoneim and Kumar proposes in [24], a method to train individual classifiers exploiting the knowledge that they will be involved in combination.

Methods and motivation of using the rank-level information from a multitude of classifiers are extensively discussed in [2]. Of the two categories discussed, *class set reordering* category of methods are more interesting for a wider audience since these methods aim to reach a better ranking of candidate classes where the top ranking speaker is the consensus decision.[1] Methods belonging to this category are the *Highest Rank* method, the *Borda Count* method and the *Logistic Regression* method. The former two are simple approaches to the problem which do not use any statistical information about the observed behavior of the classifiers, but rely on assumptions

[1] The other category, namely class set reduction methods aim at reducing the set of candidate classes but cannot preserve the ordering in the final set. Therefore, these methods cannot be used effectively when one needs to arrive at a final decision. However, in some application areas, such as forensic pattern recognition, reaching a reduced set of candidates may be attractive as an aid to the human operators.

which may sometimes be unrealistic. The last method on the other hand, is introduced in [2] as a generalization of the Borda Count [32] method with linear weighting of the classifiers. This is an important attempt to incorporate a model of the classifier behavior into the rank-based decision combination problem. This is a promising approach but restricts itself to a simple weighting of the individual classifiers although the underlying ideas provide a fertile ground for further development.

The present work also considers the rank-based multiple classifier decision combination problem and introduces a unifying theoretical formulation treating the decision combination problem as a discrete optimization problem. A Partitioned Observation Space (POS) formalism is introduced to establish the relationship between *dimensionality reduction* and *decision combination* for rank-based systems, which is an important step toward understanding their behavior [8]. The three popular rank-based combination methods mentioned can be shown to be special cases within this unifying framework.

The paper is organized as follows: The fundemantals of the unifying framework are introduced in Section 2. In Section 3, the dimensionality reduction formalism which uses an observation space partitioning approach is discussed. Section 4 establishes the links with three existing approaches to this problem and show the generality of the presented theory. Finally, Section 5 presents the application of the theory in a real-life phonetic discrimination task from speech pattern classification.

# 2 A Binary Integer Programming Approach

## 2.1 Notation and Problem Formulation

We consider a closed-set pattern classification problem where patterns may belong to $P$ different classes $S_j, j = 1, 2, \cdots, P$ which we refer as the *candidate classes*. We assume there are $Q$ classifiers $X_q, q = 1, 2, \cdots, Q$ involved in the classification process. Furthermore, $x$ denotes a *pattern*, i.e., the smallest token composed of feature vectors processable inside the classifiers to generate candidate class rankings. Each classifier ranks all the candidate classes according to some internal measure and generate a *rank score* $r_{ji}$ for each such candidate. The rank score for a specific candidate class is defined as *the number of candidate classes placed after it by the*

*classifier in the generated ranking.* With this definition, as the class is placed close to the top of the classifiers' rankings of the candidates, it receives higher rank scores. The source class of an unknown pattern (i.e., the true class generating the pattern $x$) is represented as an integer valued random variable $\underline{s}_x$ taking index values of an ordered set of class labels $\mathcal{S} = \{\mathcal{S}_\infty, \mathcal{S}_\in, \cdots, \mathcal{S}_\mathcal{P}\}$. Hence the fact that a pattern comes from a generating class $S_j$ is denoted by a realization of this random variable as $(\underline{s}_x = j)$. [2] Our final decision at the end of the decision combination process is denoted by another integer valued random variable $\underline{d}$ with the same possible values as $\underline{s}_x$. When an unknown pattern $x$ arrives, the pattern is processed by all classifiers. Each classifier ranks all $P$ candidate classes and generates $P$ *rank scores*, namely one for each candidate class. The set of all rank scores generated by the classifiers for all candidate classes form a $P \times Q$ matrix $\mathbf{R}$ which we refer as the *rank score matrix.* Here, each column corresponds to scores by a single classifier while each row corresponds to scores by all classifiers for a single candidate class. For rank-based multiple classifier decision systems, the decision must be done solely on the rank score matrix.

Suppose that the ultimate objective in classifier combination is to achieve the maximum rate of correct classification. Although other objectives can also be defined, this is a meaningful and usual choice for closed-set pattern classification. If we define a binary valued random variable $\underline{y}$, as an *indicator* of correct classification,

$$\underline{y} = \begin{cases} 1 & \text{if } \underline{d} = j \text{ given } \underline{s}_x = j, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

then the problem of finding the optimum rank-based multiple classifier decision process can be expressed as an *optimization problem* with an objective function being *total probability of correct classification* as

$$\max P\{\underline{y} = 1\}. \tag{2}$$

[2] Throughout this paper, underscore notation denotes a random variable.

## 2.2 The Objective Function

The objective function implied by (2) is not useful in this form. It should contain free problem parameters and also statistics reflecting the joint ranking behavior of the classifiers. Consider that the probability of correct classification is expanded into a sum over the source class and rank score matrix indexes as

$$P\{\underline{y}=1\}=\sum_{j=1}^{P}\sum_{n=1}^{N}P\{\underline{d}=j,\underline{s}_x=j,\underline{r}=n\}, \tag{3}$$

where we have used the fact that $\underline{y}=1$ is equivalent to $\underline{d}=j$ once the source class is realized as $\underline{s}_x=j$. By using the Bayes rule, the joint probability inside the double summation can be decomposed to give

$$P\{\underline{y}=1\}=\sum_{j=1}^{P}\sum_{n=1}^{N}P\{\underline{d}=j|\underline{s}_x=j,\underline{r}=n\}P\{\underline{s}_x=j,\underline{r}=n\}. \tag{4}$$

By definition, the decision process to be found uses only the available information from individual classifiers, which is the rank score matrix. Therefore, the decision is a deterministic function of $\underline{r}$, as $\underline{d}=f(\underline{r})$. Using the dependence on the rank score matrix only, the first term of (4) can be simplified as $P\{\underline{d}=j|\underline{s}_x=j,\underline{r}=n\}=P\{\underline{d}=j|\underline{r}=n\}$. Hence the objective function expansion in (4) takes the final form

$$P\{\underline{y}=1\}=\sum_{j=1}^{P}\sum_{n=1}^{N}P\{\underline{d}=j|\underline{r}=n\}P\{\underline{s}_x=j,\underline{r}=n\}. \tag{5}$$

The two set of terms inside this expansion should be investigated. The first set of conditional terms $P\{\underline{d}=j|\underline{r}=n\}$ is directly linked with our decision process. When a specific multiple classifier decision scheme is specified, these terms can be determined. For the case of a deterministic decision process, they are binary valued with possible values 0 and 1. The joint probability terms $P\{\underline{s}_x=j,\underline{r}=n\}$ of the second set on the other hand are independent of the decision process and model the joint behavior of the classifier ensemble. Since they can be expressed as $P\{\underline{r}=n|\underline{s}_x=j\}P\{\underline{s}_x=j\}$, these joint probabilities can be estimated if the trained classifiers are

operated on a sufficient body of cross-validation data consisting of known source class identities coupled with the resulting rank score matrixes.

### 2.3 The Optimum Solution and The Optimum Decision Process

In order to develop a methodology for finding the optimum rank-based multiple classifier decision process[3] based on the observation of the classifier ensemble behavior on the cross validation data, we need an unambiguous interpretation of the results obtained so far with respect to finding an optimum decision process. From the expansion in (5), it is seen that the only terms dependent on the decision process are $P\{\underline{d} = j|\underline{r} = n\}$. Any specific decision process manifests itself as a large set of specific 0 and 1 values associated with these probabilities. *Conversely, any specific assignment to this set of probabilities constitutes a specific decision process.* Therefore, if these terms are treated as binary valued free problem parameters $b_{jn}$, the objective function in (5) becomes

$$P\{\underline{y} = 1\} = \sum_{j=1}^{P} \sum_{n=1}^{N} b_{jn} P\{\underline{s}_x = j, \underline{r} = n\}. \tag{6}$$

When this is combined with the optimization criterion, the problem can be expressed as

$$\max_{b_{jn}, j=1,2,\cdots,P; n=1,2,\cdots,N} \left\{ \sum_{j=1}^{P} \sum_{n=1}^{N} b_{jn} P\{\underline{s}_x = j, \underline{r} = n\} \right\}, \tag{7}$$

$$\text{Subject to} \quad \sum_{j=1}^{P} b_{jn} = 1 \quad \text{for} \quad n = 1, 2, \cdots, N. \tag{8}$$

where the set of constraints arises from the fact that the final output of the decision process should be a single class label. Since all $P\{\underline{s}_x = j, \underline{r} = n\}$ are non-negative, the obvious solution to this optimization problem is given by

$$b^*_{jn} = \begin{cases} 1 & \text{if } j = \underset{k=1,2,\cdots,P}{\text{argmax}} \; P\{\underline{s}_x = k, \underline{r} = n\}, \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

[3]Within this framework, an *optimal decision process* based on the outputs of multiple classifiers is equivalent to an *optimal classifier combination process.* Therefore, these two terms can be used interchangeably.

For $j = 1, 2, \cdots, P$ and $n = 1, 2, \cdots, N$. However, this global maximum may not be unique, specifically whenever the maximizing $j$ value in (9) is not unique for a fixed $\underline{r} = n$. For such cases, more than one set of $b^*_{jn}$ values constitute alternative solutions.

Each optimal solution set $\{b^*_{jn}\}$ corresponds to a unique global optimum *decision process.* To elaborate, consider the case where an unknown pattern $x$ is processed by an ensemble of classifiers and a rank score matrix $\mathbf{R}$ is generated. This specific rank score matrix is a realization ($\underline{r} = n$) where the value of $n$ is fixed. The optimal solution guarantees that there is a single variable $b^*_{jn}$ which is 1 for fixed $\underline{r} = n$ and the global optimum solution is a *decision process* with the decision criteria

$$\underline{d} = \underset{j=1,2,\cdots,P}{\operatorname{argmax}} P\{\underline{s}_x = j, \underline{r} = n\} \tag{10}$$

Denoting the global maximum value of the objective function by $P_{\max}$, the classification error rate of the optimal overall decision process is given by $P_E = (1 - P_{\max})$. The presence of multiple global optimum solutions signify that the overall system is unable to discriminate among certain classes when faced with specific classifier outputs. For closed-set problems, a random choice among these solutions has to be made.

## 2.4 The Curse of Dimensionality

In the previous section, it is shown that *a globally optimum method of combining any number of classifiers, based on their rank scores, can be found if we were able to obtain infinite number of observations of all the classifier rank-level outputs.* However, in practice, only a finite cross-validation data set is available, from which the values of $P\{\underline{s}_x = j, \underline{r} = n\}$ are to be estimated. The number of such probabilities for $P$ classes and $Q$ classifiers is $P(P!)^Q$, which is a prohibitive number for most practical applications. Therefore, although the solution to the problem is simple, it cannot be of practical value if a method of reducing the number of estimated statistics cannot be found. The next section attempts to formulate such a method.

## 3 Dimensionality Reduction by Partitioning

Consider the objective function expansion in (6). The problem domain is composed of two main parts. The first one is the space spanned by the free problem parameters $b_{jn}$ and will be called as the *Problem Parameter Space.* The second one is the space spanned by all the statistics about the joint behavior of the classifiers, i.e., all estimated parameters $P\{\underline{r} = n, \underline{s}_x = j\}$ and will be called as the *Classifier Observation Space. Classifier Observation Statistics* are the elements of this second space.

This prohibitive cardinality of the classifier observation space is the limiting factor for the usefulness of the resulting solution since it determines the number of statistics to be estimated from the available data. However, well behaving classifiers do not span the entire observation space. As the performance of the classifiers improve to acceptable levels, the cross-validation samples tend to be highly clustered. Therefore, enough data is accumulated for estimating certain statistics while there is no data available to estimate some very small probabilities. Consider the extreme case of ideal classifiers. Whenever a pattern is supplied, the classifiers will generate very similar rankings with the true source class being at the top of the ranking. For such an extreme behavior, all cross-validation data will be accumulated for a small number of statistics. As the classifiers deviate from ideal behavior, such clusters tend to spread. The implication is that by incorporating the expected or observed behavior of the classifiers for a certain task, we may introduce some assumptions about the possible distribution of the cross-validation data and arrive at considerable compressions of the classifier observation space by means of forming logical groups of statistics. This compression can also be interpreted as a *smoothing* operation on the estimates we are trying to find. Limited cross-validation data necessitates such a smoothing and one may argue that the available data determines the *resolution* with which the classifiers can be observed. However, one should bear in mind that with such a smoothing, the system becomes unable to model behavior data violating the assumptions made and the optimal solution to this smoothed problem is sub-optimal with respect to the original one.

### 3.1 An Observation Event Space

Define an *event space* $\mathcal{F}$ where the realizations of the source class label and rank score matrix indexes are combined into compound events. The most basic events (*event atoms*) in this space are defined as $(\underline{s}_x = j; \underline{r} = n)$. Here the event atom specifies occurrence of the joint event "The source class for the pattern $x$ was $S_j$ and the set of classifiers generated the rank score matrix $\mathbf{R}_n$". This event space is clearly finite and its cardinality is $P \times (P!)^Q$ since we have $P$ source classes and $Q$ classifiers and hence a total of $(P!)^Q$ possible rank score matrixes [33].

Now assume that a *mapping* $\mathcal{W}$ partitions this event space into disjoint sets of event atoms. Also assume that the mapping (or partitioning)[4] $\mathcal{W}$ results in $M_{\mathcal{W}}$ such sets, i.e., $M_{\mathcal{W}}$ partitions $W_1, W_2, \cdots, W_{M_{\mathcal{W}}}$ which satisfy the standard properties,

$$\begin{aligned} &\text{(a)} \quad W_i \cap W_j = \emptyset \qquad\qquad \forall i, j \in \{1, 2, \cdots, M_{\mathcal{W}}\}, \\ &\text{(b)} \quad W_1 \cup W_2 \cup \cdots \cup W_{M_{\mathcal{W}}} = \mathcal{F}. \end{aligned} \tag{11}$$

Such a partitioning defines a new event space in which each resulting partition defines a *new basic event*. Composed of a set of original event atoms, these are also *compound events* in the original event space $\mathcal{F}$. If such partitions(or events) $W_i$ form an ordered set $G_{\mathcal{W}} = \{W_1, W_2, \cdots, W_{M_{\mathcal{W}}}\}$ which is our new event space, the partitioning $\mathcal{W}$ effectively defines a new random variable,

$$\underline{g}_{\mathcal{W}} : \mathcal{S} \times \mathcal{R} \longmapsto \{1, 2, \cdots, M_{\mathcal{W}}\}. \tag{12}$$

whose values are the index values of the ordered set $G_{\mathcal{W}}$. Here $\mathcal{S}$ is the set of possible source classes and $\mathcal{R}$ is the set of possible rank score matrixes.

At this point, a new expansion for our objective function in (2), can be introduced. First, the objective function is expanded over the source class labels and rank score matrixes to obtain (3). Following (12), the random variable $\underline{g}_{\mathcal{W}}$ can be expressed as a $\underline{g}_{\mathcal{W}} = \mathcal{W}(j, n)$. Therefore, the double sum can also be written by introducing the new random variable $\underline{g}_{\mathcal{W}}$ as

[4]Since each such mapping defines a new partitioning of this space, within this context a mapping $\mathcal{W}$ is synonymous with a partitioning $\mathcal{W}$ and the terms *partitioning* and *mapping* will be used interchangeably.

$$P\{\underline{y}=1\}=\sum_{j=1}^{P}\sum_{n=1}^{N}P\{\underline{d}=j,\underline{s}_x=j,\underline{r}=n,\underline{g}_{\mathcal{W}}=\mathcal{W}(j,n)\} \tag{13}$$

This is possible since the value of $\underline{g}_{\mathcal{W}}$ is known once the values of $\underline{s}_x$ and $\underline{r}$ are known and no new probabilistic event is introduced. By successively using the Bayes rule, the joint probability inside the double summation can be put into the form

$$\begin{gathered}P\{\underline{d}=j,\underline{s}_x=j,\underline{r}=n,\underline{g}_{\mathcal{W}}=\mathcal{W}(j,n)\}=\\ P\{\underline{d}=j|\underline{s}_x=j,\underline{r}=n,\underline{g}_{\mathcal{W}}=\mathcal{W}(j,n)\}P\{\underline{s}_x=j,\underline{r}=n|\underline{g}_{\mathcal{W}}=\mathcal{W}(j,n)\}P\{\underline{g}_{\mathcal{W}}=\mathcal{W}(j,n)\}.\end{gathered} \tag{14}$$

Again, using the fact that the decision should, by definition, be based on the rank score matrix alone, and inserting (14) into (13) we obtain the final expression for the objective function as

$$P\{\underline{y}=1\}=\sum_{j=1}^{P}\sum_{n=1}^{N}P\{\underline{d}=j|\underline{r}=n\}P\{\underline{s}_x=j,\underline{r}=n|\underline{g}_{\mathcal{W}}=\mathcal{W}(j,n)\}P\{\underline{g}_{\mathcal{W}}=\mathcal{W}(j,n)\}. \tag{15}$$

This time there are three sets of terms inside the expansion. The first set of terms is, as before, the one directly associated with the decision process, yet unknown. The last set of terms on the other hand, is again the statistics about the behavior of the classifiers. However, this time, the observation space is the result of the partitioning $\mathcal{W}$ and the observable events for modeling the classifier behavior are the resulting partitions $W_m, m=1,2,\cdots,M_{\mathcal{W}}$ represented by the possible values of the random variable $\underline{g}_{\mathcal{W}}$. This is a coarser resolution for the classifier observation space where the actual rank score matrixes are hidden inside the observable partitions. The middle set of terms, which is new as compared with the previous expansion, is a set of *transition terms* defining the relation between the coarser resolution of the partitions and the finer resolution of the class label, rank score matrix pairs. Since a deliberate decision is made to set the observation resolution to the coarser one, by definition, there is no cross-validation data left to estimate these transition terms. In fact, due to this choice, one is *ignorant* about this finer detail. These terms allow us to formally introduce our ignorance within the Bayesian formalism [27]. as a uniform distribution among the individual elements of any any partition $W_m$. I.e., we have

$$P\{\underline{s}_x = j, \underline{r} = n | \underline{g}_{\mathcal{W}} = m\} = \begin{cases} 0, & \text{if event atom } \{\underline{s}_x = j, \underline{r} = n\} \notin W_m, \\ \frac{1}{|W_m|}, & \text{if event atom } \{\underline{s}_x = j, \underline{r} = n\} \in W_m, \end{cases} \tag{16}$$

where $|W_m|$ is the cardinality of the partition $W_m$.

The previous discussion makes it clear that the first set of terms will again be labelled as the problem parameters to find an optimum decision process. The last terms, on the other hand, will again be estimated from the classifier behavior on the cross-validation data.

With this new expansion, a controlled tool to selectively decrease resolution on the observation and modeling of the classifier ensemble behavior is introduced. By the selection of the partitioning, it is possible to reduce the number of partitions, hence the events of the observation space. Specifically, for the above expansion we have $M_{\mathcal{W}}$ statistics to estimate. For limited cross-validation data, a reduction in the number of statistics to estimate corresponds to an increase in the number of observations about each individual statistic, hence to an increase in the reliability of the estimate. It is well known that the reliability of the estimates is crucial to the generalization performance, hence to the classification performance of the system [16, 34].

Although we have mentioned that the number of observable events can be arbitrarily reduced, the nature of the partitioning is crucial for the usefulness of the resulting solution. The objective should be to maintain the maximum observation resolution which is reasonable for the fixed amount of data available, and not a finer one. It is also illogical to use a very coarse resolution while enough data for a finer one is available since this over-smoothing, being unable to properly model the collective behavior of the set of classifiers, will limit the usefulness of the approach.

The motivation in compressing the observation space is not only the reduction in the number of statistics to estimate. For some cases, one may have an intuition about the pattern recognition task at hand, or prior knowledge about the dynamics of individual classifiers which may suggest that some higher level of resolution in the observation space may in fact be *irrelevant* or *unreliable*. This leads to the intuitive formulation of suitable partitionings, for which a clear example is presented in Section 3.3.

In the present work, we consider three specific partitionings to unify three methods from the literature and two other intuitive methods based solely on assumptions about classifier behavior

to use on the BDEV task in Section 5. However, a challenging and yet open problem is the selection of a partitioning in an optimal manner based on the actual distribution of the data. E.g., using Genetic Algorithms to determine an optimal partitioning rule may be a promising direction for future research.

### 3.2 A Computational Model to Implement the Optimal Solution

The terms $P\{\underline{g}_{\mathcal{W}} = \mathcal{W}(j,n)\}$ in the new in (15) makes it clear that the number of unique statistics is $M_{\mathcal{W}}$, a number necessarily lower than the original $P \times (P!)^Q$. However, the number of terms inside the double summation is still $P \times (P!)^Q$. Even with this huge number of terms inside the expansion, the optimum solution presented in Section 2.3 may be converted into an algorithmic form so that only a small number of computations is necessary for making the optimum decision based on the estimated statistics. Considering the optimum solution given in (9), this algorithmic form can be summarized as follows: For each pattern, process the pattern by all classifiers and obtain the rank score matrix. For the fixed index ($\underline{r} = n$), compute exactly $P$ objective function coefficients $P\{\underline{d} = j|\underline{r} = n\}P\{\underline{s}_x = j, \underline{r} = n|\underline{g}_{\mathcal{W}} = \mathcal{W}(j,n)\}P\{\underline{g}_{\mathcal{W}} = \mathcal{W}(j,n)\}$. I.e., one coefficient is computed for each candidate class, using the corresponding estimates and the transition terms determined by the partitioning rule. The final step is to decide on the class with the maximum coefficient. A total of at most $P$ multiplications and a search is involved. A final random choice is necessary for the closed-set problem if there is more than one maximum coefficient for this $\underline{r}$ value. Note that the determination of the transition terms is only possible if the partitioning is based on a rule which can be easily applied when the rank score matrix is given. For an arbitrary partitioning, a look-up table of the original size would be required, which is infeasible.

### 3.3 A Sensible Partitioning: First Two Ranks

The implications of a partitioning choice are discussed in Section 3.1. One may often have prior insight into the task and classifiers involved before the observation statistics are collected. This may be incorporated into the solution by means of a partitioning of the observation space.

The partitioning we call as the *First Two Ranks* is based on such a general observation about the behavior of the classifiers. Assume that *the resolution below the topmost two ranks (largest*

$$\left\{\left(S_{1,2,3,4},\begin{bmatrix}3&3\\2&2\\1&1\\0&0\end{bmatrix}\right);\left(S_{1,2,3,4},\begin{bmatrix}3&3\\2&2\\1&0\\0&1\end{bmatrix}\right);\left(S_{1,2,3,4},\begin{bmatrix}3&3\\2&2\\0&1\\1&0\end{bmatrix}\right);\left(S_{1,2,3,4},\begin{bmatrix}3&3\\2&2\\0&0\\1&1\end{bmatrix}\right)\right\}\mapsto\left(S_{1,2,3,4},\begin{bmatrix}2&2\\1&1\\0&0\\0&0\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=1,2,3,4$$

$$\left\{\left(S_{1,2,3,4},\begin{bmatrix}3&3\\2&1\\1&2\\0&0\end{bmatrix}\right);\left(S_{1,2,3,4},\begin{bmatrix}3&3\\2&0\\1&2\\0&1\end{bmatrix}\right);\left(S_{1,2,3,4},\begin{bmatrix}3&3\\2&1\\0&2\\1&0\end{bmatrix}\right);\left(S_{1,2,3,4},\begin{bmatrix}3&3\\2&0\\0&2\\1&1\end{bmatrix}\right)\right\}\mapsto\left(S_{1,2,3,4},\begin{bmatrix}2&2\\1&0\\0&1\\0&0\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=5,6,7,8$$

$$\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots$$

$$\left\{\left(S_{1,2,3,4},\begin{bmatrix}1&1\\0&0\\2&2\\3&3\end{bmatrix}\right);\left(S_{1,2,3,4},\begin{bmatrix}1&0\\0&1\\2&2\\3&3\end{bmatrix}\right);\left(S_{1,2,3,4},\begin{bmatrix}0&1\\1&0\\2&2\\3&3\end{bmatrix}\right);\left(S_{1,2,3,4},\begin{bmatrix}0&0\\1&1\\2&2\\3&3\end{bmatrix}\right)\right\}\mapsto\left(S_{1,2,3,4},\begin{bmatrix}0&0\\0&0\\1&1\\2&2\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=573,574,575,576$$

Table 1: Illustration of First Two Ranks partitioning. The first column is the actual partition contents, the second column is a label for the partition and the last column is the partition random variable. Actual class labels and rank score matrixes are used instead of their corresponding random variables for clarity.

*two rank scores) is unreliable.* This is a reasonable assumption, e.g., for distance classifiers, since the separation between class models becomes less significant as the models become more and more distant from the pattern being classified in the feature space. Hence, noise on the feature vectors has a greater chance of disturb the lower ranks. Therefore, we decide not to discriminate among the ranks lower than the second and group them to represent the *last rank*. The resulting new score assignment (hence partitioning) can be expressed as

$$\hat{r}_{jq}=\begin{cases} r_{jq}-P+3, & \text{if } \; r_{jq}>P-3,\\ 0, & \text{if } \; r_{jq}\leq P-3. \end{cases}$$

An illustrative example for $P=4$ classes and $Q=2$ classifiers is considered in Table 1, where there is a total of $M_{\mathcal{W}}=576$ partitions instead of the original 2304 event atoms. A shortcut notation is used as follows: Each row is the summary of 4 actual rows which correspond to the source classes $S_1, S_2, S_3, S_4$ summarized as $S_{1,2,3,4}$ in the table.

# 4 Special Cases by Means of Specific Partitionings

In this section, the theory will be used to unify three existing rank-based decision combination methods discussed in the literature. It will be shown that they indeed correspond to specific

partitionings, hence for a given data their optimality can be analyzed by the introduced theory.

### 4.1 The Highest Rank Method

Highest Rank method is discussed in [2] and is a simple technique of rank-based multiple classifier decision. Since this technique does not use any model of the observed classifier behavior it is not optimum for the general case. In this section, the conditions under which the Highest Rank solution coincides with the optimum solution will be established by means of a specific partitioning. The Highest Rank method may be described as follows: "For each source class, select the highest of the rank scores assigned by all classifiers for that class as a new score. These new scores constitute a *max score* vector. The class with the maximum *max score* is selected." The status of this method with respect to our formalism can be summarized by two facts.

**Fact 1** *As a predefined decision process, the Highest Rank method coincides with a specific fixed set of values for the problem variables $b_{jn}$. The cases where the Highest Rank method is unable to make a decision because of more than one maximum in the sum score matrix correspond to more than one fixed sets of $b_{jn}$ values.*

**Fact 2** *The Highest Rank method does not make use of any observation on classifier ensemble behavior and hence is not in general optimum for the combination of non-ideal classifiers.*

For the Highest Rank method, a partitioning $\mathcal{W}$ can be defined where each partition corresponds to a possible *max score vector*. Consider the illustrative case of $P = 3$ source classes and $Q = 2$ classifiers. There are a total of $M_{\mathcal{W}} = 42$ partitions some of which are illustrated in Table 2 where the summary notation introduced in Section 3.3 is used.

The equivalence relation between the optimum solution resulting from the objective function expansion with respect to the partitioning $\mathcal{W}$ and the Highest Rank solution can be stated as a theorem.

**Theorem 1** *The Highest Rank method and the optimum solution to the classifier combination problem coincides in the context of maximizing the probability of correct decision expressed as in (15) only for a set of classifier observation statistics $P\{\underline{g}_{\mathcal{W}} = \mathcal{W}(j,n)\}$ which satisfy a fixed set of constraints.*

**Proof 1** We will prove this theorem by directly specifying the set of constraints that should be satisfied by the statistics estimated from the joint behavior of the classifiers. The Highest Rank method inherently specifies a partitioning on the event space. For each max-score vector (which

$$
\begin{array}{ll}
\left\{\left(S_{1,2,3};\begin{bmatrix}2&2\\1&1\\0&0\end{bmatrix}\right)\right\}\mapsto & \left(S_{1,2,3};\begin{bmatrix}2\\1\\0\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=1,2,3\\
\left\{\left(S_{1,2,3};\begin{bmatrix}2&2\\1&0\\0&1\end{bmatrix}\right);\left(S_{1,2,3};\begin{bmatrix}2&2\\0&1\\1&0\end{bmatrix}\right)\right\}\mapsto & \left(S_{1,2,3};\begin{bmatrix}2\\1\\1\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=4,5,6\\
\vdots & \vdots\qquad\qquad\vdots\\
\left\{\left(S_{1,2,3};\begin{bmatrix}2&0\\1&0\\0&2\end{bmatrix}\right);\left(S_{1,2,3};\begin{bmatrix}2&0\\1&1\\0&2\end{bmatrix}\right);\left(S_{1,2,3};\begin{bmatrix}2&0\\0&1\\1&2\end{bmatrix}\right);\right. & \\
\left.\left(S_{1,2,3};\begin{bmatrix}1&2\\0&1\\2&0\end{bmatrix}\right);\left(S_{1,2,3};\begin{bmatrix}0&2\\1&1\\2&0\end{bmatrix}\right);\left(S_{1,2,3};\begin{bmatrix}0&2\\1&0\\2&1\end{bmatrix}\right)\right\}\mapsto & \left(S_{1,2,3};\begin{bmatrix}2\\1\\2\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=10,11,12\\
\vdots & \vdots\qquad\qquad\vdots\\
\left\{\left(S_{1,2,3};\begin{bmatrix}0&0\\2&1\\1&2\end{bmatrix}\right);\left(S_{1,2,3};\begin{bmatrix}0&0\\1&2\\2&1\end{bmatrix}\right)\right\}\mapsto & \left(S_{1,2,3};\begin{bmatrix}0\\2\\2\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=40,41,42
\end{array}
$$

Table 2: Illustration of the partitioning establishing the relation with the Highest Rank method.

correspond to a set of $(\underline{r}=n)$ values), Highest Rank method decides on a unique class $S_k$ except for the cases with a max-score collision. In order for this decision method to be optimal, for each specific $(\underline{r}=n)$, a condition of the form

$$
\begin{aligned}
&P\{\underline{r}=n,\underline{s}_x=k|\underline{g}_{\mathcal{W}}=\mathcal{W}(k,n)\}P\{\underline{g}_{\mathcal{W}}=\mathcal{W}(k,n)\}\geq\\
&\qquad P\{\underline{r}=n,\underline{s}_x=j|\underline{g}_{\mathcal{W}}=\mathcal{W}(j,n)\}P\{\underline{g}_{\mathcal{W}}=\mathcal{W}(j,n)\},\ \text{for } j=1,2,\cdots,P,
\end{aligned}
\tag{17}
$$

should be satisfied. If this is the case, then based on its inherent partitioning, the Highest Rank decision method is optimal in the sense of the POS theory.

### 4.2 The Borda Count Method

This is yet another popular method of rank-based multiple classifier decision. It is a slightly generalized majority voting technique from Group Decision Theory [2, 32]. Since it is simple to implement, it has been used as a popular rank-based technique. The Borda Count method uses the full rank score matrix $\mathbf{R}$ such that rank scores for each class are summed up across different classifiers. Therefore a *sum score* is obtained for each candidate class and the decision is made by choosing the candidate class having the maximum sum score. The partitioning $\mathcal{W}$ is again illustrated in Table 3 for an example case of $P=3$ classes and $Q=2$ classifiers. There are a total of $M_{\mathcal{W}}=57$ partitions instead of the original 108. The arguments of Fact 1, Fact 2 and Theorem 1 are again valid for this method which does not use any observation about classifier

$$
\begin{array}{ll}
\left\{\left(S_{1,2,3};\begin{bmatrix}2&2\\1&1\\0&0\end{bmatrix}\right)\right\}\mapsto & \left(S_{1,2,3};\begin{bmatrix}4\\2\\0\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=1,2,3\\
\left\{\left(S_{1,2,3};\begin{bmatrix}2&2\\1&0\\0&1\end{bmatrix}\right);\left(S_{1,2,3};\begin{bmatrix}2&2\\0&1\\1&0\end{bmatrix}\right)\right\}\mapsto & \left(S_{1,2,3};\begin{bmatrix}4\\1\\1\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=4,5,6\\
\vdots & \vdots\qquad\qquad\vdots\\
\left\{\left(S_{1,2,3};\begin{bmatrix}2&0\\1&1\\0&2\end{bmatrix}\right);\left(S_{1,2,3};\begin{bmatrix}2&0\\0&2\\1&1\end{bmatrix}\right);\left(S_{1,2,3};\begin{bmatrix}1&1\\2&0\\0&2\end{bmatrix}\right);\right. & \\
\left.\left(S_{1,2,3};\begin{bmatrix}1&1\\0&2\\2&0\end{bmatrix}\right);\left(S_{1,2,3};\begin{bmatrix}0&2\\2&0\\1&1\end{bmatrix}\right);\left(S_{1,2,3},\begin{bmatrix}0&2\\1&1\\2&0\end{bmatrix}\right)\right\}\mapsto & \left(S_{1,2,3};\begin{bmatrix}2\\2\\2\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=16,17,18\\
\vdots & \vdots\qquad\qquad\vdots\\
\left\{\left(S_{1,2,3};\begin{bmatrix}0&0\\1&1\\2&2\end{bmatrix}\right)\right\}\mapsto & \left(S_{1,2,3};\begin{bmatrix}0\\2\\4\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=55,56,57
\end{array}
$$

Table 3: Illustration of the partitioning establishing the relation with the Borda Count method.

behavior. It coincides with the optimum decision process only for a set of classifier observation statistics, satisfying a set of conditions similar to (17).

### 4.3 The Logistic Regression Method

Unlike the previous two approaches, the logistic regression method attempts to capture and model the joint behavior of the classifiers and is therefore a much more important and interesting rank-based method to investigate. The method is introduced in [2], where the fundamental motivation is expressed as *to obtain a linearly weighted Borda Count method where the weights reflect the relative significance of the classifiers.* This method can be interpreted as follows: While the system computes the similarity of a pattern to a class model, only rank scores assigned to that class by all classifiers are used. Let $S_k$ be the class model considered. The next step of this method is to find the probability of correct decision if the decision is $S_k$ given the rank scores for $S_k$. This probability should be estimated based on the experiments done on the cross-validation data. Finally, the class leading to the highest probability of correct decision is selected. In [2], these probabilities are approximated by first counting frequencies of the related events and then further *smoothing* them by the best hyper-plane approximation. This hyper-plane approximation

is determined by formulating and solving a prediction problem. The parameters of the hyper-plane determine the bias and the optimum weighting factors for the classifier outputs, effectively resulting in a *linearly-weighted* Borda Count method extension.

Before introducing the partitioning and the resulting statistics with coincide with this specific method, a number of observations should be discussed. The prediction problem in [2] is first introduced as a class dependent problem, i.e., *to predict the probability of correct decision for a class based on the rank scores generated for that class.* Then, the class dependence is dropped for a *simplicity in notation.* However, this is not the case. Dropping the class index conveniently reduces the problem into predicting a single variable instead of $P$ variables, and at the same time, introduces the important assumption that *the joint behavior of the classifiers is independent of the true source class involved.* This difference will be illustrated by two corresponding partitionings.

Another point is the use of a hyper-plane fit to the classifier observation statistics. Such an hyper-plane fit is probably the result of the initial motivation in [2], namely to obtain a linearly weighted Borda Count. This is a parametric model which effectively smoothes the estimated statistics, with the assumption that their reliability are uniformly poor. For this discrete case however, the reliability of each statistic should be considered separately, since the number of data points leading to such estimates may be different. The hyper-plane fit treats all such statistics uniformly and may be an over-smoothing for the statistics which are more reliable than others. In this sense, the partitioning approach presented in the present study may be considered as a more controlled way of "smoothing" unreliable estimates, especially if this partitioning is done by considering structure of the training data. As discussed in Section 3.1, such an *automatic partitioning* may be done by means of optimization methods such as Genetic Algorithms or by some other clustering techniques. One may even use the estimates from a coarser partitioning to smooth the estimates of a finer partitioning.

Now we will relate the unifying theory presented to the Logistic Regression Method by a specific partitioning of the event space which leads to the statistics described and used in [2]. Motivated by the Borda Count method, in [2], for each source class, only the scores of the classifiers for that class are considered for observation. If source class dependence were kept, the resulting partitioning for an example case of $P = 3$ classes and $Q = 2$ classifiers would have been

$$\left\{\left(S_1,\begin{bmatrix}2&2\\1&1\\0&0\end{bmatrix}\right);\left(S_1,\begin{bmatrix}2&2\\1&0\\0&1\end{bmatrix}\right);\left(S_1,\begin{bmatrix}2&2\\0&1\\1&0\end{bmatrix}\right);\left(S_1,\begin{bmatrix}2&2\\0&0\\1&1\end{bmatrix}\right)\right\}\mapsto\left(S_1;\begin{bmatrix}2\\2\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=1$$

$$\left\{\left(S_2,\begin{bmatrix}1&1\\2&2\\0&0\end{bmatrix}\right);\left(S_2,\begin{bmatrix}1&0\\2&2\\0&1\end{bmatrix}\right);\left(S_2,\begin{bmatrix}0&1\\2&2\\1&0\end{bmatrix}\right);\left(S_2,\begin{bmatrix}0&0\\2&2\\1&1\end{bmatrix}\right)\right\}\mapsto\left(S_2;\begin{bmatrix}2\\2\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=2$$

$$\left\{\left(S_3,\begin{bmatrix}0&0\\1&1\\2&2\end{bmatrix}\right);\left(S_3,\begin{bmatrix}0&1\\1&0\\2&2\end{bmatrix}\right);\left(S_3,\begin{bmatrix}1&0\\0&1\\2&2\end{bmatrix}\right);\left(S_3,\begin{bmatrix}0&0\\0&0\\2&2\end{bmatrix}\right)\right\}\mapsto\left(S_3;\begin{bmatrix}2\\2\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=3$$

$$\left\{\left(S_1,\begin{bmatrix}2&1\\1&2\\0&0\end{bmatrix}\right);\left(S_1,\begin{bmatrix}2&1\\1&0\\0&2\end{bmatrix}\right);\left(S_1,\begin{bmatrix}2&1\\0&2\\1&0\end{bmatrix}\right);\left(S_1,\begin{bmatrix}2&1\\0&0\\1&2\end{bmatrix}\right)\right\}\mapsto\left(S_1;\begin{bmatrix}2\\1\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=4$$

$$\vdots \qquad\qquad \vdots \qquad \vdots$$

$$\left\{\left(S_2,\begin{bmatrix}2&2\\0&0\\1&1\end{bmatrix}\right);\left(S_2,\begin{bmatrix}2&1\\0&0\\1&2\end{bmatrix}\right);\left(S_2,\begin{bmatrix}1&2\\0&0\\2&1\end{bmatrix}\right);\left(S_2,\begin{bmatrix}1&1\\0&0\\2&2\end{bmatrix}\right)\right\}\mapsto\left(S_2;\begin{bmatrix}0\\0\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=26$$

$$\left\{\left(S_3,\begin{bmatrix}2&2\\1&1\\0&0\end{bmatrix}\right);\left(S_3,\begin{bmatrix}2&1\\1&2\\0&0\end{bmatrix}\right);\left(S_3,\begin{bmatrix}1&2\\2&1\\0&0\end{bmatrix}\right);\left(S_3,\begin{bmatrix}1&1\\2&2\\0&0\end{bmatrix}\right)\right\}\mapsto\left(S_3;\begin{bmatrix}0\\0\end{bmatrix}\right)\mapsto \underline{g}_{\mathcal{W}}=27$$

Table 4: Illustration of a generalization of the partitioning done in the Logistic Regression method by means of preserving the class dependence.

as given in Table 4. The number of partitions for this case is $P^{(Q+1)}$ in general and the number of elements in each partition is $[(P-1)!]^Q$. However, since the source class dependence is dropped in [2], the actual partitioning done corresponds to the one illustrated in Table 5. For this case, the number of partitions is reduced to $P^Q$ for the general case and the number of elements inside each partition is increased to $P[(P-1)!]^Q$.

The logistic regression method, in its original form, does not use the event counts directly to estimate the statistics resulting from the partitioning illustrated in Table 5. It further applies a smoothing by using an hyper-plane fit to the count estimates, in addition to the partitioning involved. Considering the promising results discussed in [2] the smoothing used seems to be justified for the machine printed character recognition task.

Apart from the restrictions imposed by this specific partitioning and the additional linear smoothing, the Logistic Regression method is optimal in the sense of the POS theory introduced in the present paper. However, using the partitioning interpretation instead of viewing the method as a simple Borda Count generalization shows that many other partitionings (hence new

$$\left\{\left(S_1,\begin{bmatrix}2&2\\1&1\\0&0\end{bmatrix}\right);\left(S_1,\begin{bmatrix}2&2\\1&0\\0&1\end{bmatrix}\right);\left(S_1,\begin{bmatrix}2&2\\0&1\\1&0\end{bmatrix}\right);\left(S_1,\begin{bmatrix}2&2\\0&0\\1&1\end{bmatrix}\right);\left(S_2,\begin{bmatrix}1&1\\2&2\\0&0\end{bmatrix}\right);\left(S_2,\begin{bmatrix}1&0\\2&2\\0&1\end{bmatrix}\right);\right.$$
$$\left.\left(S_2,\begin{bmatrix}0&1\\2&2\\1&0\end{bmatrix}\right);\left(S_2,\begin{bmatrix}0&0\\2&2\\1&1\end{bmatrix}\right);\left(S_3,\begin{bmatrix}0&0\\1&1\\2&2\end{bmatrix}\right);\left(S_3,\begin{bmatrix}0&1\\1&0\\2&2\end{bmatrix}\right);\left(S_3,\begin{bmatrix}1&0\\0&1\\2&2\end{bmatrix}\right);\left(S_3,\begin{bmatrix}0&0\\0&0\\2&2\end{bmatrix}\right)\right\}\mapsto\left(\begin{bmatrix}2\\2\end{bmatrix}\right)\mapsto\underline{g}_{\mathcal{W}}=1$$

$$\left\{\left(S_1,\begin{bmatrix}2&1\\1&2\\0&0\end{bmatrix}\right);\left(S_1,\begin{bmatrix}2&1\\1&0\\0&2\end{bmatrix}\right);\left(S_1,\begin{bmatrix}2&1\\0&2\\1&0\end{bmatrix}\right);\left(S_1,\begin{bmatrix}2&1\\0&0\\1&2\end{bmatrix}\right);\left(S_2,\begin{bmatrix}1&2\\2&1\\0&0\end{bmatrix}\right);\left(S_2,\begin{bmatrix}1&0\\2&1\\0&2\end{bmatrix}\right);\right.$$
$$\left.\left(S_2,\begin{bmatrix}0&2\\2&1\\1&0\end{bmatrix}\right);\left(S_2,\begin{bmatrix}0&0\\2&1\\1&2\end{bmatrix}\right);\left(S_3,\begin{bmatrix}0&0\\1&2\\2&1\end{bmatrix}\right);\left(S_3,\begin{bmatrix}0&2\\1&0\\2&1\end{bmatrix}\right);\left(S_3,\begin{bmatrix}1&0\\0&2\\2&1\end{bmatrix}\right);\left(S_3,\begin{bmatrix}1&2\\0&0\\2&1\end{bmatrix}\right)\right\}\mapsto\left(\begin{bmatrix}2\\1\end{bmatrix}\right)\mapsto\underline{g}_{\mathcal{W}}=2$$

$$\vdots\qquad\qquad\vdots\qquad\vdots$$

$$\left\{\left(S_1,\begin{bmatrix}0&0\\2&2\\1&1\end{bmatrix}\right);\left(S_1,\begin{bmatrix}0&0\\2&1\\1&2\end{bmatrix}\right);\left(S_1,\begin{bmatrix}0&0\\1&2\\2&1\end{bmatrix}\right);\left(S_1,\begin{bmatrix}0&0\\1&1\\2&2\end{bmatrix}\right);\left(S_2,\begin{bmatrix}2&2\\0&0\\1&1\end{bmatrix}\right);\left(S_2,\begin{bmatrix}2&1\\0&0\\1&2\end{bmatrix}\right);\right.$$
$$\left.\left(S_2,\begin{bmatrix}1&2\\0&0\\2&1\end{bmatrix}\right);\left(S_2,\begin{bmatrix}1&1\\0&0\\2&2\end{bmatrix}\right);\left(S_3,\begin{bmatrix}2&2\\1&1\\0&0\end{bmatrix}\right);\left(S_3,\begin{bmatrix}2&1\\1&2\\0&0\end{bmatrix}\right);\left(S_3,\begin{bmatrix}1&2\\2&1\\0&0\end{bmatrix}\right);\left(S_3,\begin{bmatrix}1&1\\2&2\\0&0\end{bmatrix}\right)\right\}\mapsto\left(\begin{bmatrix}0\\0\end{bmatrix}\right)\mapsto\underline{g}_{\mathcal{W}}=9$$

Table 5: Illustration of the partitioning done for the Logistic Regression method.

methods) are possible some of which may be better suited for a particular task. The motivation of finding better ones opens new direction for future research and the introduced formalism can be used as a tool to analyze them.

# 5 Application on The BDEV Discrimination Task

In this section, we consider the application of the theory in a real-life problem from speech pattern classification, namely the discrimination of the highly confusable four short letter-words: The names of the letters b,d,e and v.

## 5.1 Task Description

The task of discriminating the BDEV words has been first discussed by Lang to introduce the time-delayed neural network architecture [35]. It is an interesting and difficult real-life task having constrained size, which enables us a closer examination of the statistical behavior of the classifiers. The task aim is to classify an unknown speech signal to be an utterance of one of the 4 letter-words with the highest rate of correctness. Actual recordings from this database are

illustrated in Figure 2 with their corresponding time-frequency evolutions (spectrograms).

A multi-speaker database of 5 speakers is collected in common office environment for the experiments. One training and one testing session are recorded with a 2 to 3 days time separation. The speakers are asked to read a random sequence of isolated letter-words with approximately 60 utterances/letter-word/session which result in approximately 1200 training and 1200 testing patterns. The recordings have *telephone quality* with 8 kHz sampling rate and 8 bits/sample quantization and are made in a common office environment with computer cooling fans as the dominant background noise.

The main difficulty of the task is that the discriminating consonant sound at the beginning of the word is of very short duration and is followed by a high energy vowel 'e' sound which is common to all 4 classes and hence non-discriminative. Also, the SNR is quite low due to both the slow sampling rate/coarse quantization and the presence of background noise, resulting in low performance for all individual classifiers.

### 5.2 Speech Feature Extraction and the Individual Classifiers

A speech pattern is often the sampled speech waveform of an utterance which is for this case an utterance of a letter-word. Since the speech signal has time-varying behavior, the descriptive *features* of the speech signal are often sequences of feature vectors which are extracted from fixed length segments of the sampled speech signal, called *frames*, within which the signal may be assumed to be stationary [36]. In our experiments, three different feature extraction methods are used on speech frames of length 256 samples with an overlap of 128 samples, windowed by an Hamming window. For the first method, the frame spectrum is computed by an FFT analysis and this is transformed into FT-Derived Cepstral Coefficients. The coefficients 2-13 are retained as the frame feature vector. For the second method, the frame is subjected to a Linear Predictive (LPC) analysis followed by the recursive computation of the LP-Derived Cepstral Coefficients [36]. Coefficients 2-13 are again retained as the feature vector. For the last method, the frame is passed through the LPC inverse filter and an LPC residual signal is obtained. Then the residual signal is used for the computation of the FT-Derived Cepstral Coefficients.

Each of the three feature extraction methods is used in combination with a common model-

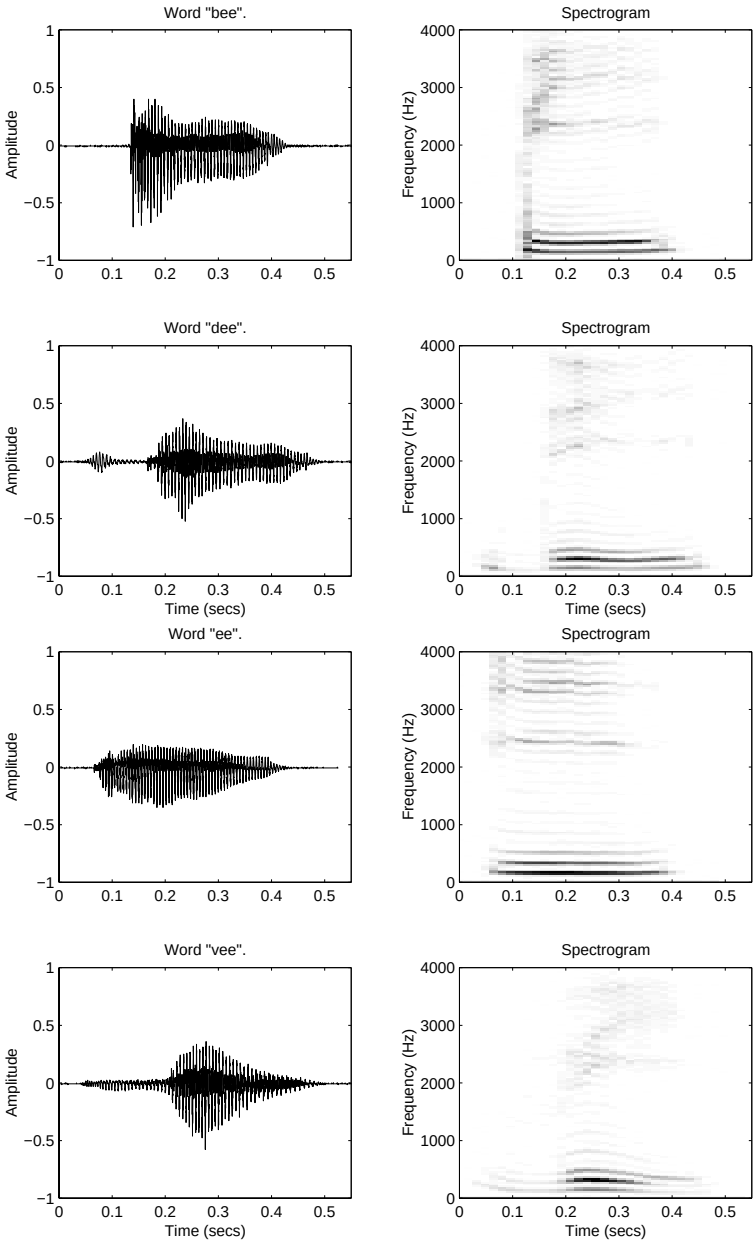


Figure 2: Examples for the 4 letter-word classes. Each time waveform is associated with the corresponding spectrogram which illustrate the time-frequency behavior of the speech waveform. One can note that the dominant high energy formants corresponding to the vowel sound (near black lines close to the bottom) are very similar for each letter-word while the differences arise mainly in the low energy (near white) parts.

ing/similarity scoring method based on Vector Quantization(VQ) and LBG training algorithm [37] to form a complete classifier. The three resulting classifiers are denoted by *FFTCep*, *LPC-Cep* and *LPCResCep*. During the classifier training phase, each of the three individual classifiers generates a VQ-codebook model for each class to be recognized. This is done as follows: The training speech utterances are processed by the classifier specific feature extraction method in combination with the LBG training algorithm to generate a VQ-codebook model for each class and to be used by that specific classifier. This process is repeated for all three classifiers. During the testing phase, the test speech utterance is supplied to all three classifiers. Again, classifier specific feature extraction is performed and each classifier matches its extracted set of unknown features against its stored VQ-codebook models by computing a *cumulative Euclidean distance measure*. The model distances are ranked from closest match downwards and is transformed into rank scores which constitute the classifier raw outputs. This entire process is illustrated in Figure 3.

### 5.3 Combination based on the POS Formalism

Two new rank-based decision combination strategies are proposed based on the POS formalism. One is based directly on the *First Two Ranks Partitioning* described in Section 3.3 where the classifier behavior observation space is partitioned to discriminate only among the top two positions of the candidate class ranking from each classifier. We call this method as *Rank2*. This intuitive data-independent partitioning is based on an expectation that the lower ranks in a classifier output may be unreliable due to noisy estimates. The other method is based on an extension of this idea by a partitioning which discriminates only the top ranking of candidate class rankings and hence degenerates to the case where only the final decisions of each classifier is considered for joint classifier behavior modeling. We call this method as *Rank1*.

These two methods, are used by the procedure illustrated in Figure 4 as follows: 1.) First, an appropriate partitioning $\mathcal{W}$ is chosen for the classifier observation space. This determines the transition terms $P\{\underline{s}_x = j, \underline{r} = n | \underline{g}_{\mathcal{W}} = \mathcal{W}(j,n)\}$. 2.) The labelled training data used to train the individual classifiers is transformed into a cross-validation data set by a variant of the leave-one-out method [38]. Each pattern is left out of the training set during classifier training and

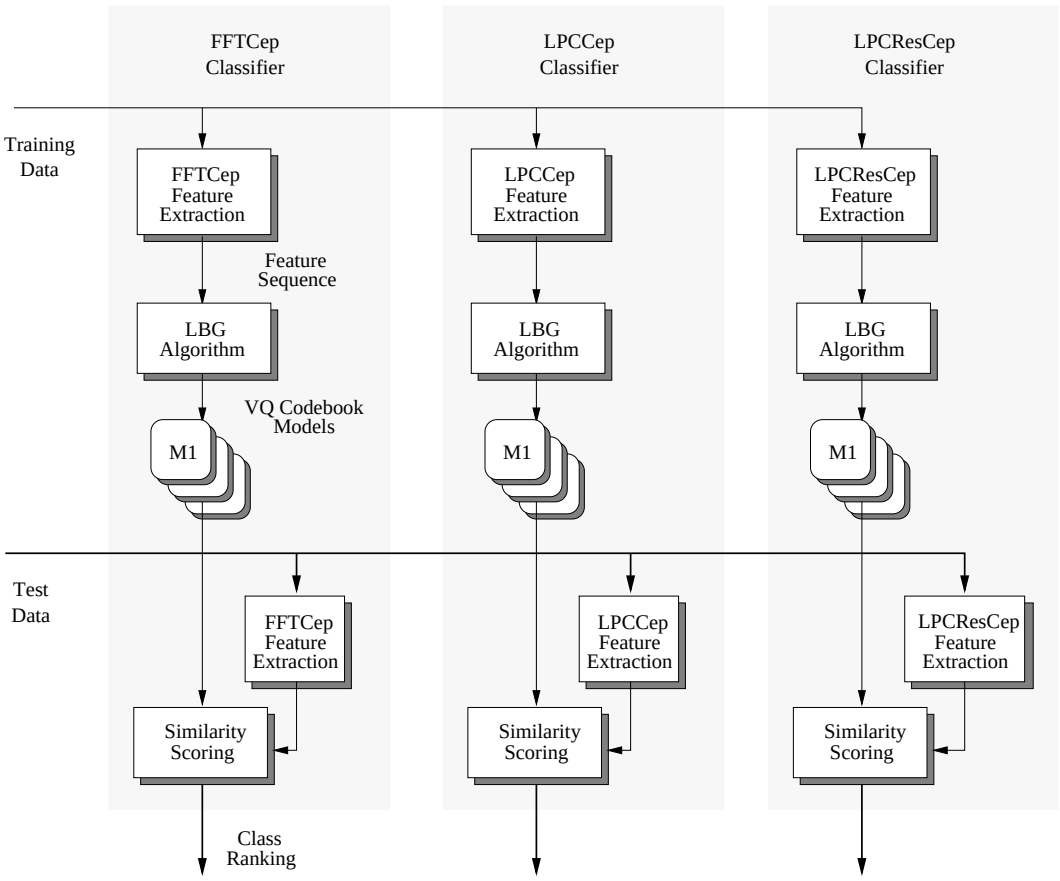


Figure 3: Illustration of the three classifiers *FFTCep*, *LPCCep* and *LPCResCep*. During training, VQ-codebook class models are generated by the classifiers while during testing, classifiers compute the similarity of the unknown speech pattern to the stored models to generate candidate class rankings as their individual outputs.

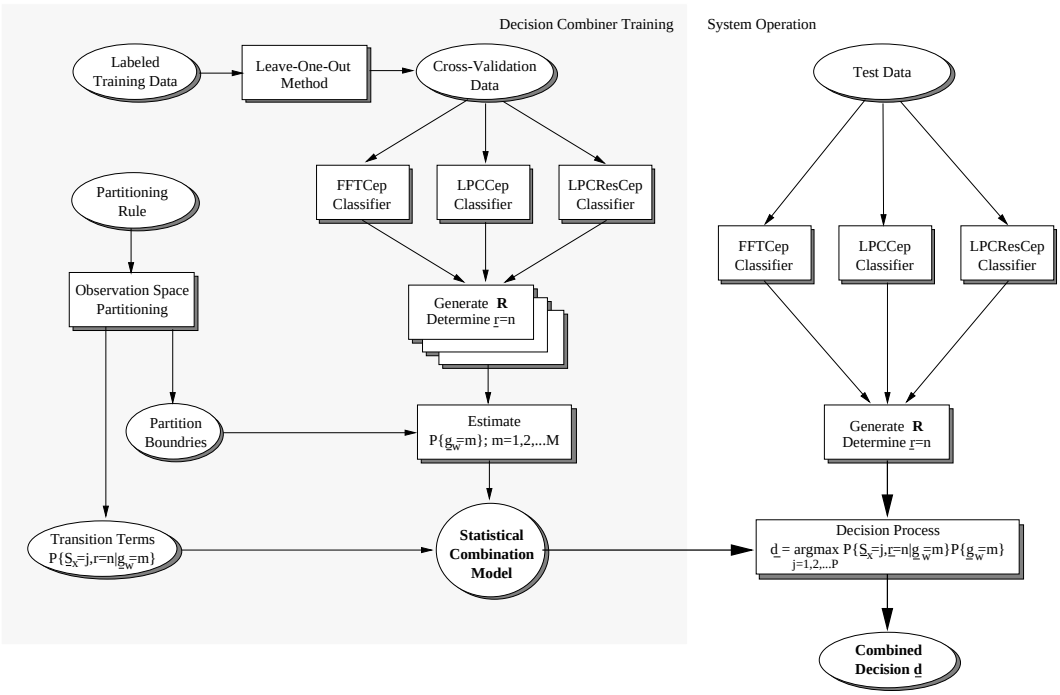


Figure 4: Block diagram of Multiple Classifier Decision Combiner System training and operation.

used as a *test pattern* to test the trained classifiers. Hence, $L$ cross-validation tests outputs are obtained using the classifiers (where $L$ is the number of patterns in the database) with each test pattern being previously unseen by the classifiers. 3.) The results of these cross-validation tests are used to determine the distribution of the source class-rank score pairs among the partitions by means of *partition accumulators*. Hence, partition occurrence statistics $P\{\underline{g}_{\mathcal{W}} = m\}$ are estimated for $m = 1, 2, \cdots, M$. This forms the statistical combination model. 4.) Given an unknown test pattern, the classifiers are operated on the pattern and the rank score matrix $\mathbf{R}$ is obtained. The computational model of the optimum solution (Section 3.2) is applied for the given $\mathbf{R}$ and the coefficients $P\{\underline{s}_x = j, \underline{r} = n|\underline{g}_{\mathcal{W}} = \mathcal{W}(j,n)\}P\{\underline{g}_{\mathcal{W}} = \mathcal{W}(j,n)\}$ for all candidate classes $j = 1, 2, \cdots, P$ are computed. The candidate class with the maximum coefficient is selected by the rule

$$\underline{d} = \underset{j=1,2,\cdots,P}{\text{argmax}}\ P\{\underline{s}_x = j, \underline{r} = n|\underline{g}_{\mathcal{W}} = \mathcal{W}(j,n)\}P\{\underline{g}_{\mathcal{W}} = \mathcal{W}(j,n)\}$$

| Classifier | FFTCep | LPCCep | LPCResCep |
|---|---|---|---|
| Cross-Validation Data | 66.18 | 63.77 | 37.73 |
| Test Data | 65.83 | 61.83 | 35.00 |

Table 6: Individual classifier percent classification performances on the BDEV task.

| Combined Classifiers | FFTCep LPCCep | FFTCep LPCResCep | LPCCep LPCResCep | FFTCep LPCCep LPCResCep |
|---|---|---|---|---|
| Highest Rank | 65.83 | 65.83 | 61.83 | 35.00 |
| Borda Count | 67.58 | 58.17 | 55.58 | 64.67 |
| Logistic Regression | 67.50 | 65.83 | 61.83 | 66.25 |
| Rank 1 | 68.83 | 65.83 | 61.83 | 69.20 |
| Rank 2 | 69.22 | 66.75 | 62.28 | 68.83 |

Table 7: Percent classification performances of existing and proposed combination methods on the BDEV task.

### 5.4 Experimental Results

This computational model is applied to the BDEV letter-word discrimination task. As the first step, the individual classifiers are tuned for their best performance. The classification performances of these three individual classifiers are given in Table 6 both for the cross-validation data and for the test data. The comparative results of pairwise and collective combination of these individual classifiers by the rank-based decision combination methods are given in Table 7. A serious problem for the Highest Rank method is the excessive score collisions(more than one class with the same max score) when small number of classes are involved. Highest Rank and Borda Count methods are supplemented by resolving score collisions with the decision of the best performing classifier instead of a random decision between colliding classifiers. The last three combination methods show consistent classification improvement over the individual classifiers with the best performance achieved by the Rank2 method. However, the performance seems the drop when all three classifiers are combined with the Rank2 method. It can be argued that the reason for this drop is the increasing number of statistics to estimate for Rank2 with three classifiers which degrades the reliability of the estimates.

For the methods *Logistic Regression*, *Rank 1* and *Rank 2*, the statistics about the joint

| Combined Classifiers | FFTCep LPCCep | FFTCep LPCResCep | LPCCep LPCResCep | FFTCep LPCCep LPCResCep |
|---|---|---|---|---|
| Logistic Regression | 67.50 | 65.83 | 61.83 | 66.75 |
| Rank 1 | 69.00 | 65.83 | 61.83 | 70.92 |
| Rank 2 | 73.92 | 70.25 | 66.92 | 85.83 |

Table 8: Percent classification performance of statistical combination methods on the BDEV task based on statistics derived from the test data, showing upper bounds in performance possible for an exact statistical match between cross-validation and test.

behavior of the classifiers are extracted from the cross-validation tests. The maximum gain from these methods can be achieved when these statistics exactly reflect the behavior on the actual test data. To see this upper bound on performance, the behavior statistics are also extracted from the actual test data and the results for using this data for the combination model are given in Table 8. It can be observed that the potential improvement in combined results is much larger than the actual one achieved which suggests that there is a mismatch in the classifier behavior between cross-validation and testing. This is an expected behavior but shows that there is still a margin for improvement if the behavior statistics can be more reliably estimated. The distribution of the cross-validation test and actual test samples for the original event space and for the partitions resulting from Rank1 and Rank2 methods is illustrated in Figures 5 and 6 where *white* signifies the lack of any samples to estimate a statistic. From the two figures, it can be observed that there is an observable mismatch in the behavior statistics between cross-validation and test which is more apparent when there is no partitioning at all. The mismatch is partially smoothed with the introduction of the two partitionings.

# 6 Conclusion

We have considered the closed-set pattern classification and formulated rank-based multiple classifier decision combination as a binary integer programming problem. The optimization has been based on maximizing the total probability of correct decision and has led to an objective function expansion including decision related terms as well as statistics which can be estimated from joint classifier behavior on the cross-validation data. Although the existence of a simple optimum solu-

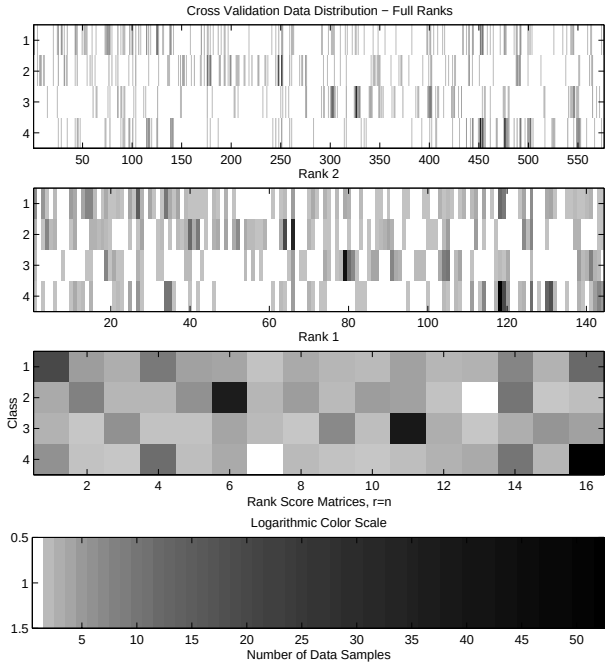


Figure 5: The distribution of the *cross-validation test patterns* among the observation space partitions for 2 classifiers case: No partitioning, Rank1 and Rank2. The nonlinear gray color scale is illustrated at the bottom.

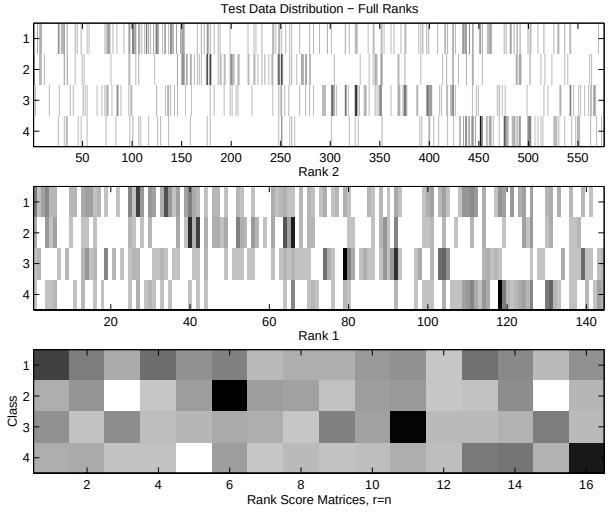


Figure 6: The distribution of the *actual test patterns* among the observation space partitions for 2 classifiers case: No partitioning, Rank1 and Rank2.

tion is shown, the dimensionality of the problem necessitated techniques of reducing the number of statistics to be estimated. We have proposed the partitioning approach as a controlled tool to selectively achieve dimensionality reduction and provided an illustrative example. The full implications of different partitionings are yet to be explored. However, we have shown that a number of such partitionings establish the relations of the presented theory with three popular rank-based multiple classifier decision methods. Particular emphasis has been on the *Logistic Regression method* since it was the only such method attempting to capture knowledge about classifier joint behavior.

We believe that the theory presented is a promising direction for understanding the rank-based multiple classifier decision systems. During our discussion, we have provided a clear methodology of obtaining a family of methods by means of different partitionings of the observation space. Specific partitionings are introduced to establish the links with some existing approaches and decision combination is applied to the BDEV discrimination task from speech pattern recognition by means of two simple partitionings. We argued that it is important to account for the structure and amount of cross-validation data available as well as for an understanding of the task at hand. We believe that methods of achieving such partitionings, preferably in an automatic manner, are promising directions for future research. These may be the tools to tailor and analyze specific rank-based decision (combination) methods for specific pattern classification tasks under the introduced formalism.

## 7 Acknowledgements

The authors would like to express their gratitude to Dr. Buyurman Baykal, Dr. Kemal Leblebicioğlu, Dr. Tankut Özgen and the Referees, whose comments, suggestions and criticism have contributed to this study.

## References

[1] Chuanyi Ji and Sheng Ma. Combination of weak classifiers. *IEEE Transactions on Neural Networks*, 8(1):32–42, January 1997.

[2] Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75, January 1994.

[3] Lei Xu, Adam Krzyżak, and Ching Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 22(3):418–435, May/June 1992.

[4] Y.S. Huang and C.Y. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):90–94, January 1995.

[5] Tin Kam Ho. Recognition of handwritten digits by combining independent learning vector quantizations. In *Proceedings of 2nd International Conference on Document Analysis and Recognition*, pages 818–822, Tsukuba Science City, Japan, October 1993.

[6] Roberto Battiti and Anna Maria Colla. Democracy in neural nets: Voting schemes for classification. *Neural Networks*, 7(4):691–707, 1994.

[7] Galina Rogova. Combining the results of several neural network classifiers. *Neural Networks*, 7(5):777–781, 1994.

[8] Josef Kittler, Mohamad Hatef, Rober P. W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, March 1998.

[9] F. Kimura and M. Shridhar. Handwritten numerical recognition based on multiple algorithms. *Pattern Recognition*, 24(10):969–983, 1991.

[10] Kevin R. Farell and Richard J. Mammone. Data fusion techniques in speaker recognition. In R.V. Ramachandran and Richard J. Mammone, editors, *Modern Methods of Speech Processing*, chapter 12, pages 279–297. Kluwer Academic Publishers, Boston, Massachusetts, 1995.

[11] Younnès Bennani and Patrick Gallinari. Neural networks for discrimination and modelization of speakers. *Speech Communication*, 17:159–175, 1995.

[12] Mübeccel Demirekler and Afşar Saranlı. A study on improving decisions in closed set speaker identification. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1127–1130, Munich, Germany, April 1997.

[13] Vlasta Radová and Joseph Psutka. An approach to speaker identification using multiple classifiers. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1135–1138, Munich, Germany, April 1997.

[14] Bernard Achermann and Horst Bunke. Combination of face classifiers for person identification. In *Proceedings of IAPR International Conference on Pattern Recognition*, pages 416–420, Vienna, Austria, 1996.

[15] R. Brunelli and D. Falavigna. Person identification using multiple cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):955–966, October 1995.

[16] David H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[17] Jon Atli Benediktsson, Johannes R. Sveinsson, Okan K. Ersoy, and Philip H. Swain. Parallel consensual neural networks. *IEEE Transactions on Neural Networks*, 8(1):54–64, January 1997.

[18] Jon Atli Benediktsson and Philip H. Swain. Consensus theoretic classification methods. *IEEE Transactions on Systems, Man and Cybernetics*, 22(4):688–704, July/August 1992.

[19] Belur V. Dasarathy. *Decision Fusion*. IEEE Computer Society Press, Los Alamitos, 1994.

[20] James Shih-Jong Lee, Jenq-Neng Hwang, Daniel T. Davis, and Alan C. Nelson. Integration of neural networks and decision tree classifiers for automated cytology screening. In *Proceedings of the International Joint Conference on Neural Networks*, volume 1, pages 257–262, Seattle, July 1991.

[21] Yu Hen Hu, Surekha Palreddy, and Willis J. Tompkins. A patient-adaptable ECG beat classifier using a mixture of experts approach. *IEEE Transactions on Biomedical Engineering*, 44(9):891–900, September 1997.

[22] Kagan Tumer and Joydeep Ghosh. Analysis of decision boundries in linearly combined neural classifiers. *Pattern Recognition*, 29(2):341–348, February 1996.

[23] Michael P. Perrone and Leon N. Cooper. When networks disagree: Ensemble methods for hybrid neural networks. In Richard J. Mammone, editor, *Artificial Neural Networks for Speech and Vision*, pages 127–142. Chapman & Hall, London, UK, 1993.

[24] Khaled Al-Ghoneim and B.V.K.Vijaya Kumar. Learning ranks with neural networks. In *Proceedings of SPIE*, pages 446–464, 1995.

[25] G. Mani. Lowering variance of decisions using artificial neural networks portfolios. *Neural Computation*, 3:484–486, 1991.

[26] L.K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:993–1001, 1990.

[27] Judea Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* Morgan Kaufmann Publishers, Palo Alto, CA, USA, 1988.

[28] Glenn Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, New Jersey, USA, 1976.

[29] M.D. Richard and R.P. Lippmann. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Computation*, 3(4):461–483, 1991.

[30] D.W. Ruck et al. The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE Transactions on Neural Networks*, 1(4):296–298, 1990.

[31] David H. Wolpert. A mathematical theory of generalization: Part 1. *Complex Systems*, 4:151–200, 1990.

[32] D. Black. *The Theory of Commitees and Elections.* Cambridge University Press, London, 2nd edition, 1963.

[33] Afşar Saranlı and Mübeccel Demirekler. Rank-based multiple classifier decision combination: A theoretical study. In *Proceedings of the IEEE International Workshop on Intelligent Signal Processing*, pages 51–56, Budapest, Hungary, September 1999.

[34] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, Boston, 2 edition, 1990.

[35] Kevin J. Lang. *A Time-Delay Neural Network Architecture for Speech Recognition.* PhD thesis, Carnegie Mellon University, School of Computer Science, Pittsburg, PA 15213, July 1989.

[36] Joseph W. Picone. Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, 79(4):1214–1247, April 1994.

[37] R.M. Gray Y. Linde, A. Buzo. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 20(1):84–95, January 1980.

[38] Hermann Ney, Ute Essen, and Reinhard Kneser. On the estimation of 'small' probabilities by leaving-one-out. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1202–1212, December 1995.