Generating Support Set Classifiers

Levon Aslanyan[0000-0002-5354-2730] and Hasmik Sahakyan[0000-0002-8449-6845]

Institute for Informatics and Automation Problems, Yerevan 0014, 1 P. Sevak str.
lasl@sci.am hsahakyan@sci.am

Abstract. Experimental data, as well as their structure, volume, and properties, have undergone substantial changes in recent decades. A wide range of big-data scenarios has emerged, among which high-dimensional low-sample-size (HDLSS) data constitute a particularly challenging class. Such data are characterized by an extremely large number of features relative to the number of available observations, which significantly complicates the application of conventional classification methods. Consequently, new concepts and models are required. A central problem is the identification of relatively small subsets of features that are capable of achieving the desired classification accuracy. In pattern recognition and machine learning, the idea of using special subsets of features, referred to as *support sets*, as classifiers was originally proposed by Yu. Zhuravlev. The objective of the present work is to develop algorithms for the efficient generation of all such support sets in HDLSS classification problems. The mathematical framework underlying this generation process is based on the concepts of monotonicity and Boolean functions. The support set classifier (SSC) model developed in this paper can be viewed, in a certain sense, as an inverse counterpart of the classical approach to association rule mining in relational databases. Indeed, support sets, which correspond to minimal classifiers, are closely related to frequent itemsets in association rule mining: when considered solely as collections of features, minimal classifiers reduce to frequent itemsets satisfying additional discriminative properties. This connection establishes a bridge between classification theory and data mining and provides new opportunities for analyzes of HDLSS data.

Keywords: HDLSS · Gene Expression · Classification · Rule mining.

Generating Support Set Classifiers
L. Aslanyan, H. Sahakyan
Institute for Informatics and Automation Problems
lasl@sci.am

1 The roots of the problem

Automatic object classification algorithms rely on a small set of training examples characterized by a limited number of attributes [19,25], although in other

cases classification data can be large, for example, in data mining tasks [1-3]. In this, the sample descriptions are small, but their quantity is quite large. The data is also specific in multivariate models, with small sample sizes, for which classification algorithms are less studied. Here, the samples are a few, but their feature descriptions are significantly large. In examining these three cases, in the following sections, we will demonstrate their interrelation: namely, that high-dimensional low-sample-size (HDLSS) data analysis [22] can be transformed to the problem of association rule mining [1-3], which has a very large set of solution algorithms.

1.1 Pattern Recognition (PR).

Classification, or pattern recognition is one of the fundamental components of classical machine learning (ML) [19]. A typical pattern recognition (PR) model is defined on a set M of objects, referred to as the pattern recognition data space. Each object $\alpha \in M$ is represented by a feature vector $(\alpha_1, \alpha_2, \ldots, \alpha_n)$, where α_i denotes the value of the i-th feature associated with α. In most formulations, M is assumed to be a metric space, or a subset of a metric space.

A learning set $L \subseteq M$ is given, consisting of objects α_i together with their classification labels $y(\alpha_i)$. The classes $C_1, C_2, \ldots, C_l$ are subsets of M, typically assumed to be pairwise disjoint, and the labels belong to the set $Y = \{1, 2, \ldots l\}$. Within this framework, the objective of pattern recognition is to construct a classification rule or decision function $A(\alpha) = y$ such that $A(\alpha) = y$ approximates the unknown labeling function on the training set L and, more importantly, generalizes accurately to previously unseen objects in $M \backslash L$.

To assess and control the generalization performance of a classification algorithm A, one typically employs cross-validation procedures or introduces an additional test set $P \subseteq M$, constructed according to the same principles as the training set L. The predictive performance of A on P provides an estimate of the generalization error, that is, the error expected on previously unseen objects.

A wide variety of classification algorithms have been developed within the pattern recognition framework. Classical examples include ensemble methods such as Boosting, Bagging, and Voting schemes, as well as Support Vector Machines (SVMs), decision trees, nearest-neighbor methods [8-9], and many others.

1.2 Association rule mining (ARM).

Besides the classical PR model described above, classification is both – a concept and a computational procedure that appears in many data-analysis frameworks. Examples include data mining, association rule mining, HDLSS data analysis, as well as models involving fuzzy, incomplete, uncertain, unstructured, or missing-value data. The present paper is based on, and essentially combines, elements of PR, ARM, and HDLSS models, emphasizing their interrelations and common structural principles. The association rule mining model is based on dynamic relational tables D, referred to as rule-mining data structures. Items $I_1, I_2, \ldots, I_n$ correspond to the columns of the table, while each row represents a transaction, interpreted as a selection from the item set $I = \{I_1, I_2, \ldots, I_n\}$.

The objective of ARM is to discover implication-type relations between itemsets. If $A \subset I$ and $B \subset I$, $A \cap B = \emptyset$, then an association rule

$$r : A \to B$$

is interpreted not in the strict sense of mathematical logic, but probabilistically. Specifically, both the support probability $p(A \cup B)$ and the conditional probability $p(B|A)$ must exceed predefined thresholds, known respectively as the support ($supp$) and confidence ($conf$) levels.

The classical motivating example is the supermarket basket analysis problem. This data structure differs substantially from the one commonly encountered in pattern recognition. In classical pattern recognition problems, the number of objects is often limited and each object is described by a fixed collection of features. In contrast, ARM focuses on massive collections of sparse transactions and seeks combinatorial patterns among items rather than geometric separation of classes in a feature space.

As one of the principal frameworks in contemporary Big Data Analytics, association rule mining (ARM) [1-3] requires efficient and scalable algorithmic techniques. For binary (0,1)-valued data, where the values indicate the absence or presence of an item in a transaction, a variety of highly efficient algorithms have been developed. Among them, the Apriori algorithm has become a classical benchmark and one of the most influential methods in the field. This connection has given rise to the area commonly known as Classification Rule Mining (CRM) [16-17], whose objective is to derive classification models in the form of interpretable rules extracted directly from the data. CRM may be viewed as a bridge between traditional pattern recognition and association rule mining, combining the predictive goals of classification with the descriptive and explanatory capabilities of rule discovery. The relationship between CRM and the support-set approach developed in this paper will be one of the central themes of our study.

Classification Rule Mining (CRM) aims to discover a compact collection of association rules that can be used to construct an accurate classifier. In contrast, classical association rule mining seeks to identify all rules that satisfy prescribed support and confidence thresholds.

Taken together, ARM and CRM provide a powerful framework for classification problems involving datasets with a very large number of observations. At the same time, they retain the conventional representation of objects through a relatively small set of attributes or features. This combination of scalability, interpretability, and predictive capability has made CRM an important link between data mining and pattern recognition.

1.3 High dimensional low sample size (HDLSS) data.

The next data structure considered in this work is the well-known high-dimensional low-sample-size (HDLSS) model [22]. HDLSS datasets frequently arise in modern biological and biomedical research, where the number of measured variables greatly exceeds the number of available observations. In our

study, such a structure emerged in the analysis of differential gene expression induced by typical and atypical antipsychotic drugs in the frontal cortex of mice.

To illustrate the problem, we consider the dataset *Typical and Atypical Antipsychotic Drugs Effect on Brain*, available from the Gene Expression Omnibus (GEO) repository under accession number GDS775 [4-5]. The structure of this dataset differs substantially from those encountered in the classical pattern recognition and association rule mining settings. The data consist of gene expression measurements for approximately 17,000 genes obtained from four biological samples, denoted by $\{S_1, S_2, S_3, S_4\}$. The samples are partitioned into two classes: S_1, S_2 (typical antipsychotic drugs) and S_3, S_4 (atypical antipsychotic drugs). From a purely classification perspective, this dataset is deceptively simple. Given the extremely high dimensionality of the feature space and the very small number of observations, a large number of low-complexity classifiers can perfectly separate the two classes. In particular, it is common to observe that individual genes, or small subsets of genes, provide highly accurate and often perfect discrimination between the two groups. Consequently, even simple decision rules based on thresholds, linear discriminants, or low-dimensional hyperplanes may achieve zero training error. This abundance of successful classifiers gives rise to the phenomenon that we refer to as *classifier multiplicity*. On the one hand, classifier multiplicity poses a challenge because it undermines the uniqueness and stability of selected biomarkers and complicates biological interpretation. On the other hand, it represents an opportunity, as different classifiers may correspond to distinct biological mechanisms and reveal complementary aspects of the underlying molecular processes. For this reason, the primary objective is not necessarily to identify a single optimal classifier. Instead, we propose to investigate the entire space of classifiers that are consistent with the observed data within a prescribed family of models and under constraints on the size of the feature subsets. Such an approach shifts the focus from classifier optimization to classifier enumeration (combinatorial explosion) and analysis, providing a richer framework for biological interpretation and knowledge discovery in HDLSS settings.

2 State of the art

The research presented in this paper originates from the broader field of *Big Data Analytics*, specifically from the study of high-dimensional low-sample-size (HDLSS) datasets, where a relatively small number of classified observations is described by an extremely large number of features [1–6]. Such data structures are common in modern genomics, bioinformatics, and other data-intensive scientific domains.

2.1 Logic-combinatorial pattern recognition (LCPR) model and its Multiparametric algorithmic model of score computation (MASC).

Our interest in Logical-Combinatorial Pattern Recognition (LCPR) models [18,21-27] is motivated by the fact that the concept of support sets originated

precisely within this research tradition.

LCPR is a precedent-based pattern recognition paradigm founded on the principle of logical (or algebraic) correction of heuristic classification algorithms. The methodology addresses recognition problems in two principal stages. In the first stage, a collection of heuristic procedures is applied to classify the available objects, producing an intermediate representation that is typically encoded as a Boolean matrix. In the second stage, known as the *correction stage*, the final classification decisions are obtained by applying a Boolean function, referred to as a *logical corrector*, to the intermediate results generated by the heuristic procedures. Since its emergence in the 1970s, the development of LCPR has progressed through several major stages [22].

LCPR algorithm could be represented as a product (successive execution) of two algorithmic stages: a recognition operator (heuristics) and a decision rule stage. One of the extensive groups of LCPR heuristics is known as multiparametric metric similarity-based algorithms. For an object S to be recognized, the algorithm calculates its estimates for each class, $\Gamma_i(S),\ i = 1, 2, \ldots, l$. It is supposed, that each learning object S_{ij} (i is the class number, and j is the object number in that class) is spreading its potential over the neighboring area of descriptions. The integrative influences $\Gamma_i(S)$ over the object S is summarized as potential value of classes at the point S. And the simplest decision rule can be a best match search, when class scores $\Gamma_1(S), \Gamma_2(S), \ldots, \Gamma_l(S)$ are computed.

Consider one particular case of the MASC family. A support set of a model A is defined as a specific subsystem $\Omega_{\underline{A}}$ of the collection Ω of all subsets of the global index set $\{1, 2, \ldots, n\}$ of the considered features $x_1, x_2, \ldots, x_n$. Feature values and their vectors compose a space X with the metric ρ. As a rule, X appears as the set R of reals, or its Cartesian degree, and ρ is the Euclidean or a logical-combinatorial distance. Elements of $\Omega_{\underline{A}}$ are importantly called *support sets* of algorithms A. In score-computation algorithms, for the tractability provisions, Ω_A is determined in a number of different ways. The choice of the support-set system Ω_A plays a fundamental role in determining both the computational complexity of the algorithm and its descriptive power. In particular, the support-set approach transforms the classification problem into the study of combinatorial structures defined on the feature set, thereby establishing a direct connection between pattern recognition, combinatorial optimization, and Boolean-function theory.

FEATURES					
objects	x_1	x_2	...	x_n	classes
$\tilde{a}^1_1$	$a^1_{1,1}$	$a^1_{1,2}$	...	$a^1_{1,n}$	C_1
$\tilde{a}^1_2$	$a^1_{2,1}$	$a^1_{2,2}$	...	$a^1_{2,n}$	
...					
$\tilde{a}^1_{m_1}$	$a^1_{m_1,1}$	$a^1_{m_1,2}$	...	$a^1_{m_1,n}$	
...					...
$\tilde{a}^l_1$	$a^l_{m_{i-1}+1,1}$	$a^l_{m_{i-1}+1,2}$	...	$a^l_{m_{i-1}+1,n}$	C_m
$\tilde{a}^l_2$	$a^l_{m_{i-1}+2,1}$	$a^l_{m_{i-1}+2,2}$	...	$a^l_{m_{i-1}+2,n}$	
...					
$\tilde{a}^l_{m_l}$	$a^l_{m_l,1}$	$a^l_{m_l,2}$	...	$a^l_{m_l,n}$	

Figure 2.1 The Learning Set $T_{n,m,l}$

Forming the MASC is in several steps. First, the set Ω_A of the algorithm is introduced. As an example, we consider the case $\Omega_{A,k}$ mentioned above. Proximity measure considers all ϖ-parts ϖS_u and ϖS_t of object codes S_u and S_t together with additional parameters $\varepsilon,\ \varepsilon_1,\ \varepsilon_2, \ldots,\ \varepsilon_n$ of the model, and defines the similarity measure of parts ϖS_u and ϖS_t. $\Gamma_\varpi(S_u, S_t)$ is supposed to be some threshold type binary measure. ϖ defines the set of coordinates under consideration, and the corresponding thresholds ε_i limit the sensitive difference of the feature values, and finally, ε manages the number of coordinates where the feature values differ sensitively. Real proximity by the set of all support sets brings more parameters of the model into the force: $\gamma_i(S)$ as the significance or strength of the object code S, $\gamma_r(S)$ as the representativeness of the object code S, etc.

Having $\gamma_i(S)$ and $\gamma_r(S)$, $\Gamma_\varpi\,(S_u, S_t)$ is transformed into the form $\Gamma_\varpi\,(S_u, S_t) = \gamma_i(S)\gamma_r(S)B_\varpi(S_u, S_t)$, and $\Gamma\,(S_u, S_t) = \sum_{\varpi\in\Omega_A}\Gamma_\varpi\,(S_u, S_t)$.

Finally, the summary estimates for the elements S for the classes K_i is defined as $\Gamma_i(S) = \sum_{S_t\in K_i}\Gamma\,(S, S_t)$. Concluding, based on the learning set $T_{n,m,l}$, and for a particular feasible object description S, we defined the similarity measure of S to the class K_i using a large set of parameters such as $\Omega_{\underline{A}},\ \varepsilon,\ \varepsilon_1,\ \varepsilon_2, \ldots,\ \varepsilon_n, \gamma_i(\bullet), \gamma_r(\bullet)$. This was the first part of the algorithm – the scoring or estimation part. This part is followed by the decision-making part, with 2 new parameters, δ_1 and δ_2.

We just mention 2 optimization problems that appear here.

a) Effective computation of the score functions.

Here, to show the result, we bring one example of the score computation formulas from [25]:

$$\Gamma_i(S) = \sum_{j=m_{i-1}+1}^{m_i}\gamma(S_j)\left(\sum_{t=0}^{\varepsilon}C^{k-t}_{\rho(S,S_j)}C^{t}_{n-\rho(S,S_j)}\right).$$

This is a more or less simple formulae that computes the combinatorial value of $\Gamma_i(S)$. Support set model $\Omega_{A,k}$ is internal element of this formula, and $\rho(S, S_j)$ denotes the threshold Hamming distance of elements S and S_j.

There exist a large set of similar formulas to optimize the stage of score calculation. The core supposition in these formulas is the regular structure of the set Ω_A. For the general case $\Omega_A \subseteq E^n$ the simple formula computations might be unknown. Moreover, it is known that this task is equivalent to the Boolean function minimization, which is known as a hard problem. The idea is, that the subset of E^n is to be represented through the simple constructions of E^n such as chains, layers, spheres or subcubes (intervals). Fore simple subsets it can be derived simple formulas alike the above one, but in general, this is a serious problem for MASC machine learning models. And our goal is a bit different – it is to weight the feature subsets in a proper classification system.

b) Weighting parameters and the support sets.

Consider the validation scenario of a classification problem. Usually, in this, along with the table $T_{n,m,l}$ a similar testing table $V_{n,k,l}$ is used, or the cross-validation mechanism over the $T_{n,m,l}$ is applied. Consider the target function that characterizes the validation success and then let that Ω_A, ε, ε_1, $\varepsilon_2, \ldots,$ ε_n, $\gamma_i(\bullet), \gamma_r(\bullet), \delta_1, \delta_2$ is the point of optimality of the target function for the given input data. This is the standard step of optimization for classification. But we are mainly interested in part of $\Omega_{\triangleq}$ among the whole set of parameters. In first step all these sets have the same weight and these weights are not a visible part among the optimization parameters. As a second stage for optimization, consider weights for the vertices $\Omega_A \subseteq E^n$ and compute the optimal values of weights to this support sets when other parameters are fixed after the first optimization stage. The highest weights correspond to an important part of the vertices of E^n, for classification by the validation scenario. And this way effectively mines the important collections of features for successful classifications. These two optimization steps provide correct weight determination of support sets and other interpretable parameters in the system, but optimization deals with extremely large data dimensions and at this time there is not known effective computational approaches to these problems. And of course, the direct application to the HDLSS classification problem is not possible.

2.2 Support Set Probably Approximately Correct model.

Probably Approximately Correct (PAC) [6,14,19] is the theoretical model for statistical learning by precedents. In this context - support sets may provide an extension of PAC algorithmic model – from the features area to the feature-subsets area. Several fundamental questions arise when designing and analyzing models and algorithms that learn from precedents. Let us start by introducing the PAC learning framework that helps in defining framework of learnable concepts in terms of characterization of sample precedents needed to achieve an approximate solution; the time and space complexity of the learning algorithm, which also depends on the cost of the computational representation of the concepts/classifications.

Recall the input set $\mathcal{X}$, – the input set of all possible instances. In particular, it is supposed to be a finite subset of the real vector domain R^n, which means that instances are coded by real values of n attributes $a_1, a_2, \ldots, a_n$. The set of all possible labels or target values is denoted by $\mathcal{Y} = \{1, 2, \ldots, l\}$, the so-called l class classification case.

A concept c: $\mathcal{X} \to \mathcal{Y}$ is a mapping from $\mathcal{X}$ to $\mathcal{Y}$. Since $\mathcal{Y} = \{1, 2, \ldots, l\}$, we can identify c with the hypergraph K over the base set $\mathcal{X}$. Over the edge K_i of K, concept c takes the value i. Thus, in the following, we equivalently refer to a concept of learning as a mapping from $\mathcal{X}$ to $\{1, 2, \ldots, l\}$, or introducing a collection of disjoint hyperedges of $\mathcal{X}$.

In statistical interpretation we assume that in each problem, examples are independently and identically distributed (i.i.d.) according to some fixed but unknown to us distribution D. The learning problem is then formulated as follows.

The learner considers a fixed set of possible concepts H, called a hypothesis set, which may not coincide with C. Learner receives a sample $S = (x_1, \ldots, x_m)$ drawn i.i.d. according to D as well as the labels $(c(x_1), \ldots, c(x_m))$ of these samples, which are based on a specific target concept $c \epsilon C$ to learn. The task is to use the labeled sample S to select a hypothesis $h_S \epsilon H$ that has a small generalization error with respect to the concept c. Generalization means extension of correct classification to the elements of the set $K \backslash S$. The generalization error of a hypothesis $h \epsilon H$, also referred to as the true error or just error of h is denoted by $R(h)$ and defined as follows.

Definition 1. Generalization error.

Given a hypothesis $h \epsilon H$, a target concept $c \epsilon C$, and an underlying distribution D, the generalization error of h vs. c is defined as $R(h) = \Pr_{x \sim D}[h(x) \neq c(x)]$.

The generalization error of a hypothesis is not directly accessible to the learner since both the distribution D and the target concept c are unknown. However, the learner can measure the empirical error of a hypothesis on the labeled sample S.

Definition 2. Empirical error.

Given a hypothesis $h \epsilon H$, a target concept $c \epsilon C$, and a sample $S = (x_1, \ldots, x_m)$, the empirical error of h vs. c is defined by $\dot{R}(h) = \frac{1}{m} \sum_{i=1}^{m} 1_{h(x_i) \neq c(x_i)}$.

Thus, the empirical error of $h \epsilon H$ is its average error over the sample S, while the generalization error is its expected error based on the distribution D. There exist a number of guarantees relating these two quantities with high probability, under some general assumptions. What is evident, is that $E\left[\dot{R}(h)\right] = R(h)$.

The following introduces the entire PAC learning framework. We denote by $size(c)$ the maximal cost of the computational representation of $c \epsilon C$.

$\mathcal{X}$ is the input set of possible instances codded by vectors of R^n; D is probability distribution over the $\mathcal{X}$; $\mathcal{Y} = \{1, 2, \ldots, l\}$ is the set of class labels; C is the set of classifications to learn.

H is hypotheses set to be chosen by the algorithm, $A : S \to h_S$

A concept class* C *is said to be PAC-learnable if there exists an algorithm A and a polynomial function $poly\ (\circ, \circ, \circ, \circ)$ such that for any $\varepsilon > 0$ and $\delta > 0$, for all distributions D on $\mathcal{X}$ and for any target concept $c \epsilon C$, the following holds for any sample size $m \geq poly(1/\varepsilon, 1/\delta, n, size(c))$:

$\Pr_{S \sim D^m}[R(h_S) \leq \varepsilon] \geq 1 - \delta$.

If further A runs in $poly(1/\varepsilon, 1/\delta, n, size(c))$, then C is said to be efficiently PAC-learnable. When such an algorithm A exists, it is called a PAC-learning algorithm for C.

A concept class C is thus PAC-learnable if the hypothesis returned by the algorithm after observing a number of points, polynomial in $1/\varepsilon$ and $1/\delta$, is approximately correct (error at most ε) with high probability (at least $1 - \delta$),

which justifies the PAC terminology. $\delta > 0$ is used to define the confidence $1-\delta$ and $\varepsilon > 0$ – the accuracy $1-\varepsilon$. Note that if the running time of the algorithm is polynomial in $1/\varepsilon$ and $1/\delta$, then the sample size m must also be polynomial if the full sample is received by the algorithm. Several key points of the PAC definition are worth emphasizing. First, the PAC framework is a distribution-free model: no particular assumption is made about the distribution $\mathbf{D}$ from which examples are drawn. Second, the training sample and the test examples used to define the error are drawn according to the same distribution $\mathbf{D}$. This is a necessary assumption for generalization to be possible, in most cases. Finally, the PAC framework deals with the question of learnability for a concept class $\mathbf{C}$ and not a particular concept. Note that the concept class $\mathbf{C}$ is known to the algorithm, but of course target concept $\mathbf{c}\epsilon\ \mathbf{C}$ is unknown. In many cases, in particular when the computational representation of the concepts is not explicitly discussed or is straightforward, we may omit the polynomial dependency on $\mathbf{n}$ and $\mathbf{size}(\mathbf{c})$ in the PAC definition and focus only on the sample complexity.

PAC-learning interpretation of HDLSS support set classification model:

- Parameters: $\mathbf{n}$ is extraordinarily large, and $\mathbf{m}$ is extremely small in HDLSS. In this regard, the processing of small groups of features with subsequent integration of the results becomes relevant, instead of analysis of the complete features domain.
- The concept $\mathbf{c} \in \mathbf{C}$ refers to both – the object as a whole, and to tits ω-parts. The set of concepts C obeys the general characterization of classes in pattern recognition — to the compactness hypotheses, described by potential function theory, which is further formalized in terms of expanders and discrete isoperimetry [7,10-12].
- Elements of hypotheses set $\mathbf{H}$ of support set HDLSS are the simplest decision stamps [19], or majority type algorithms [25], and their attribute sets may be restricted by support sets $\omega \in \mathbf{\Omega}_{\underline{A}} \subseteq \mathbf{\Omega}$. In general, $\mathbf{\Omega}$ is supposed to be the power set of $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$. The presence of $\mathbf{\Omega}_{\underline{A}}$ implemented in PAC definitions is by the use of the notion $\mathbf{h}_{\varpi \mathbf{S}}$ instead of $\mathbf{h}$.
- In PAC, it is given $\mathbf{S}$, and $\mathbf{c} \in \mathbf{C}$ implicitly, indicates the existence of the corresponding $\mathbf{h} \in \mathbf{H}$. The support set PAC requires constructing all hypotheses $\mathbf{h}$ satisfying the given error threshold.

2.3 Support set classification empirical error

Let's move on to considering support set modifications and extensions of the PAC model. Empirical error function in PAC is a combinatorial, not statistical concept, in a sense that the distribution $\mathbf{D}$ is not a supposition in there. If desired, some distribution function may be weakly approximated on base of the learning set. The concept of empirical error function is based entirely on learning set, on parameters $\mathbf{n}$, $\mathbf{m}$, $\mathbf{l}$, and sets $\omega \in \mathbf{\Omega}$, without a generalisation visionary. Given the learning set $\mathbf{S}$ of the problem, with class labels of its elements, we have the split of $\mathbf{S}$ into the classes, $\mathbf{S} = \mathbf{S_1} + \mathbf{S_2} + \ldots + \mathbf{S_l}$, and sizes $\mathbf{m} = \mathbf{m_1} + \mathbf{m_2} + \ldots + \mathbf{m_l}$, corresponding to the split. Let $\mathbf{x} \in \mathbf{S}$. $\mathbf{c}(\mathbf{x})$ is the true class label of $\mathbf{x}$. For $\omega\mathbf{x}$, its class label is not defined directly, but presumably, it belongs

to the same class $\mathbf{c}(\mathbf{x})$ in hereditary logical assumptions. Explaining this, let us consider ω parts of learning elements $\mathbf{x}, \mathbf{x}^{'} \in \mathbf{S}$ under the conditions $\mathbf{1}_{\omega\mathbf{x}\doteq\omega\mathbf{x}'}$ and $\mathbf{1}_{\mathbf{c}(\omega\mathbf{x})\neq\mathbf{c}(\omega\mathbf{x}')}$: an element $\ddot{\mathbf{r}}\,(x, h, \omega)$ of the error $\ddot{\mathbf{R}}\,(h, \omega)$ (further normalized). Condition $\mathbf{1}_{\omega\mathbf{x}\doteq\omega\mathbf{x}'}$ represents the threshold equality of elements $\mathbf{x}, \mathbf{x}^{'}$ by MASC parameters $\varepsilon,\ \varepsilon_{\mathbf{1}},\ \varepsilon_{\mathbf{2}}, \ldots,\ \varepsilon_{\mathbf{n}}$, having its simplest case as the ordinary equality of $\mathbf{x}, \mathbf{x}^{'}$. Conditions $\mathbf{1}_{\omega\mathbf{x}\doteq\omega\mathbf{x}'}$ and $\mathbf{1}_{\mathbf{c}(\omega\mathbf{x})\neq\mathbf{c}(\omega\mathbf{x}')}$ may appear in an ω-truncated fragment of original learning set information $\mathbf{1}_{\mathbf{x}\neq\mathbf{x}'}$ with $\mathbf{1}_{\mathbf{c}(\mathbf{x})\neq\mathbf{c}(\mathbf{x}')}$. When ω is applied on pairs of elements of different classes, $\mathbf{x}, \mathbf{x}^{'}$, making them similar, then it becomes impossible to differentiate the classes of these ω parts functionally. The classification function obeys the following two rules:

Classification Rule 1. Pairs of objects that are equal or threshold-equal in terms of their ω-parts must be assigned to the same class.

Classification Rule 2. Pairs of objects that are equal or threshold-equal in terms of their ω-parts, but from different classes, inevitably introduce an error that must be minimized.

Let us construct a graph $\mathbf{G}_{\omega}$, with vertex set $\mathbf{S}$, and edges corresponding to all $\mathbf{x}, \mathbf{x}^{'}$, with $\omega\mathbf{x} \doteq \omega\mathbf{x}^{'}$. Note that the relation codded by edge connection in $\mathbf{G}_{\omega}$ is not transitive, so this relation is not an equivalence. Then which is the structural characterisation of relation $\omega\mathbf{x} \doteq \omega\mathbf{x}^{'}$? Let $\mathbf{x_p}, \mathbf{x_q} \in \mathbf{S}$ are vertices of graph $\mathbf{G}_{\omega}$, connected by a chain of edges. According to *Rule 1.*, a valid classification function can't allocate ω-parts of $\mathbf{x_p}$ and $\mathbf{x_q}$ to different classes. To prove this, consider, recursively, pairs of neighbor vertices (edge vertices) on the chain, starting from $\omega\mathbf{x_p}$, and/or from $\omega\mathbf{x_q}$ correspondingly (from two endpoints of the chain). If the chain length is 1, then the claim is evident. Here we can say even more, - if the vertices are equal, then the class labels defined by the function must coincide and be equal to the labels of these vertices, and if the class labels are not equal initially, then the function must drop one of these labels getting an error in classification. The situation is similar in an increase of the chain length, which we are going to consider now. For a longer chain let us continue with the recursion steps from two endpoints, until the two current vertices coincide in a neighborhood edge. Since the vertices $\mathbf{x_p}$ and $\mathbf{x_q}$ distribute their own class labels over the neighborhood, at each recursion step, from the two sides we come to an edge with two neighboring vertices with these class labels, and these labels must coincide since the obtained neighboring vertices are threshold equal. The possible label inequality is against the *Rule 1.* and we get:

Proposition 1.

An arbitrary valid classification function in a support set classification model, by the *Rule 1.*, is a mapping from the set of all connected components of graph $\mathbf{G}$ of the threshold equality of ω-parts of learning objects, to the set of class labels.

In this regard we will treat the problem in terms of empirical error $\ddot{\mathbf{R}}\,(\omega, \mathbf{h})$, which is the classification error of the table $\varpi\mathbf{T}_{\mathbf{n},\mathbf{m},\mathbf{l}}$ itself.

3 Support Set Classifier Model

Restriction of support set classification hypothesis $\mathbf{h}$ to work on ϖ part off data may only incorporate additional errors. Any features group, such as ϖ, provides certain quality of correct classification depending on the algorithm of analysis. It is clear that expansion of the set ϖ of features, the classifier $\mathbf{h} \in \mathbf{H}$ is based on, $\ddot{\mathbf{R}}(\mathbf{h}, \omega)$ measure will decrease. At least we suppose that $\mathbf{H}$ obeys this property. That is, if α and β are vertices (feature collections) in Figure 2.1, $\alpha \subseteq \beta$, then $\ddot{\mathbf{R}}(\mathbf{h}, \alpha) \geq \ddot{\mathbf{R}}(\mathbf{h}, \beta)$. This property of monotonous decrease of $\ddot{\mathbf{R}}(\mathbf{h}, \omega)$ as the set ϖ expands, is an essential element of our model. Generalizing $\mathbf{R}(\mathbf{h})$ to $\ddot{\mathbf{R}}(\mathbf{h}, \omega)$ we may consider any appropriate measure of false classification that behaves monotonically such as the measure $\ddot{\mathbf{R}}(\mathbf{h}, \omega)$.

Expanding $\mathbf{R}(\mathbf{h})$ up to the $\ddot{\mathbf{R}}(\mathbf{h}, \omega)$ in PAC model, we apply a specific approach in the different style of use of hypotheses set $\mathbf{H}$. Instead of optimizing the error by choosing a hypothesis $\mathbf{h}$ from $\mathbf{H}$, we consider a single-element or a very narrow subset of $\mathbf{H}$, hereby transferring the optimization over the selection of the proper subset ϖ. However, the classifiers considered in this work are distinguished by an extreme simplicity, which is aimed at processing large data amounts, trying to reduce the computational difficulties that arise here.

PAC extension with support sets raises new questions such as: search for not one solution but characterization of the whole spectrum of solutions with $\ddot{\mathbf{R}}(\mathbf{h}, \omega) \leq \varepsilon$; and secondly, to characterize the set of solutions, where ω is minimal, based on a particular $\mathbf{h}$ and $\ddot{\mathbf{R}}(\mathbf{h}, \omega) \leq \varepsilon$.

We gave a framework that extends PAC model from individual attribute-based algorithms to the attribute-collection-based algorithms, which is an essential enlargement of PAC. At the same time, the statistical theoretical extension that appears here will not be considered just because of such theory may have only limited and theoretical interest. We prefer practical work on designing heuristics such as the mentioned majority selection and experimental error definition as $\ddot{\mathbf{R}}(\mathbf{h}, \omega)$. $\ddot{\mathbf{R}}(\mathbf{h}, \omega)$ does not correspond exactly to majority algorithm but the details of definitions are not sensitive and we prefer to work with simplified constructions.

Support set concept, in previous paragraphs is introduced in two terms. In MASC support sets are the technical element of optimization. Optimization may weight them and the higher score support sets may form an important part of classification structures. Higher the number of attributes, larger is the initial collection of support sets. In HDLSS model the direct use of MASC optimization draws to intractability of algorithms. It was also discussed the PAC interpretation of support sets, which expands the PAC model from attributes to the sets of attributes, and a number of new combinatorial problems appear here. The most valuable issues, to notice them again, is the whole solution set characterization problem, and the new notion introduced for empirical error, that accepts only limited number of discrete values.

In the continuation of the presentation we will come to one more model, the association rule mining (ARM) model of data mining, within the framework of which we will get the right interpretation of the mentioned problems.

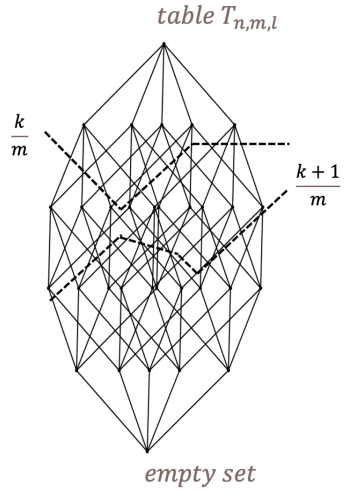

Figure 3.1 Structure of support sets and the support value levels.

Introduce the parameter **SUPP** (support), serving as a threshold for a permissible empirical error. We borrow this concept from the area of association rule mining (ARM) domain. Since there are $\mathbf{m}$ objects in our model, the effective values of **SUPP** accept the form $\frac{\mathbf{k}}{\mathbf{m}}$, $\mathbf{k} = \mathbf{0}, \mathbf{1}, \ldots, \mathbf{m}$. Integrating the monotonity of $\dot{\mathbf{R}}(\mathbf{h})$ and the discreteness of the values of **SUPP**, we obtain the left side picture for $\dot{\mathbf{R}}(\mathbf{h}) =$**SUPP**. At each level $\frac{\mathbf{k}}{\mathbf{m}}$ we get a monotone Boolean function and in integration we arrive to a structure of $\mathbf{k}$ nested monotone Boolean functions.

The picture in Figure 3.1 presents the general structure of the power set of the finite binary learning set $\mathbf{T_{n,m,l}}$ (the set of its rows). Below we will bring the detailed structure of this set but now it is enough to imagine that the larger subsets/samples of $\mathbf{T_{n,m,l}}$ are placed higher in this picture. For example, the line marked as $(\mathbf{k}+\mathbf{1})/\mathbf{m}$ restricts, at the tope, all those sets ϖ that provide an error $(\mathbf{k}+\mathbf{1})/\mathbf{m}$ or less. This line defines the Boolean function $\mathbf{f_{k+1}}$ with zero values placed on this line and below it. The unit (true) values of the function are above the line and correspond to sets ϖ with an error that is strictly higher than $(\mathbf{k}+\mathbf{1})/\mathbf{d}$. Vertices located in the region formed by lines $\mathbf{k}/\mathbf{m}$ and $(\mathbf{k}+\mathbf{1})/\mathbf{m}$ are minimal sets with an error strictly less than $(\mathbf{k}+\mathbf{1})/\mathbf{m}$, i.e. this set contains all support sets for the value $\mathbf{SUPP} = (\mathbf{k}+\mathbf{1})/\mathbf{m}$. Support sets correspond to the lower units included in this area.

Finding the lower units of $\mathbf{f_{k+1}}$ is equivalent to deciphering this function, which is a complex procedure. Alternatively, and equivalently, one may try to build the set of all upper zeros of $\mathbf{f_{k+1}}$, which also uniquely represents the function $\mathbf{f_{k+1}}$.

In case of the error $\ddot{\mathbf{R}}(\mathbf{h}, \omega)$ it is to consider the set of pairs of learning samples, that we demonstrate in an example.

Consider a simple table $\mathbf{T_{5,5,2}}$ for two classes $\mathbf{C_1}$ and $\mathbf{C_2}$, with $\mathbf{m_1} = |\mathbf{C_1}| = \mathbf{2}$ and $m_2 = |\mathbf{C_2}| = \mathbf{3}$, and let all 5 attributes are binary:

x_1	x_2	x_3	x_4	x_5	
1	1	0	0	1	C_1
0	0	0	1	1	
1	1	0	0	0	C_2
0	1	1	0	1	
1	0	1	1	1	

Figure 3.2 A 2 class and 5 binary attribute classification learning set.

x_1	x_2	x_3	x_4	x_5	row pair
0	0	0	0	1	1,3
1	0	1	0	0	1,4
0	1	1	1	0	1,5
1	1	0	1	1	2,3
0	1	1	1	0	2,4
1	0	1	0	0	2,5

Figure 3.3 XOR of row pairs of rows from different classes.

We have given a simple example of a 2-class classification problem with a training set $\mathbf{T}_{\mathbf{5},\mathbf{5},\mathbf{2}}$, which represents the ω-part of some HDLSS table, $|\omega| = 5$. Here, the feature domains are also simplified, reducing them to the binary case. Comparison of feature values, which should have been performed using thresholds, is here reduced to logical addition modulo 2. After comparing pairs of rows from different classes, a single binary table is obtained, and for the general case of feature value domains, we would also obtain a similar binary table. We will denote this table by P.

Proposition 2.

Frequent subsets in table P are one or the union of threshold classification errors $\ddot{\mathbf{r}}(x, \mathbf{h}, \omega)$ of one or several members $x \in \mathbf{T}_{\mathbf{n},\mathbf{m},\mathbf{l}}$.

4 Conclusion

In this paper, we proposed new algorithms for support set generation in high-dimensional classification problems. The proposed methods are of two types. The first is based on monotone Boolean reconstruction, while the second, which is more practical from a computational perspective, is based on association rule mining. From a technical standpoint, the support set problem differs from classical ARM in that it considers the region of the n-dimensional unit cube that is complementary to the region covered by all frequent itemsets. The construction of support sets can therefore exploit techniques developed for frequent itemset generation. Alternatively, the same objective can be achieved through monotone Boolean recognition methods based on chain decomposition and oracle-guided procedures. The main contribution of this work is the convergence of ideas originating from machine learning, monotone Boolean recognition, and association

rule mining. This integration provides a solution to the algorithmic problem of generating all support sets of a classification table, thereby identifying the minimal subsets of features that ensure the required classification accuracy on the available training data. Although the proposed methodology is applicable to a broad range of classification problems characterized by a large number of features, the motivating application considered in this study is whole-genome classification based on gene expression data. In this context, the obtained results should be regarded as preliminary. Owing to the combination of extremely high dimensionality and limited sample size, there remains a possibility that some of the discovered support sets correspond to random combinations of unrelated features rather than biologically meaningful patterns. To assess the biological relevance of the generated support sets, we performed a functional analysis of the genes involved using the KEGG Pathways, Reactome, and Gene Ontology databases. This analysis provides an additional level of validation for the support sets identified by the proposed algorithms.

References

1. Agrawal R. et al. (1996) Fast discovery of association rules. In Fayyad U.M. et al. (eds.), Advances in knowledge discovery and data mining 12, no. 1. American Association for Artificial Intelligence, Menlo Park, CA, pp. 307-328.
2. Agrawal R., Imielinski T., Swami A. (1993) Database mining: A performance perspective. In IEEE Transaction on Knowledge and Data Engineering, 5(6), 914-925.
3. Agrawal R., Srikant R. (1994) Fast algorithms for mining association rules. In 20th VLDB Conference.
4. Arakelyan A., Aslanyan L., Boyajyan A., On Knowledge-based Gene Expression data analysis", Selected Revised Papers of 9th CSIT conference, IEEE Xplore, pp. 1-6, 2013.
5. Arakelyan A., Aslanyan L., Boyajyan, A. (2013) High-throughput Gene Expression Analysis Concepts and Applications. Sequence and Genome Analysis II – Bacteria, Viruses and Metabolic Pathways. iConcept Press Ltd, USA, 71-95.
6. Aslanyan L., Recognition Method Based on Classification by Disjunctive Normal Forms, Translated from Kibernetika, No. 5, pp. 103-110, September-October, 1975.
7. Aslanyan L. (1979) The discrete isoperimetry problem and related extremal problems of discrete spaces. Probl. Kibern. 36, 85–127.
8. L. H. Aslanyan, H. E. Danoyan, "On the Optimality of the Hash-Coding Type Nearest Neighbour Search Algorithm", Selected Revised Papers of 9th CSIT conference, IEEE Xplore, pp. 1-6, 2013.
9. Aslanyan L., Danoyan H., Complexity of Hash-Coding Type Search Algorithms with Perfect Codes, Journal of Next Generation Information Technology (JNIT), Volume 5, Number 4, November 30, 2014, ISSN : 2092-8637(Print), 2233-9388(Online), pp. 26-35.

10. Aslanyan L., Sahakyan H. (2017) The Splitting technique in monotone recognition. Discrete Applied Mathematics, 216, 502–512.
11. Aslanyan L., Sahakyan H. (2017) Cube-split technique in quantitative association rule mining. Information Theories and Applications, 24(3), 3-20.
12. Aslanyan, L. and Sahakyan, H., 2026. Hybrid block balanced searches for nearest neighbors. *Pattern Recognition and Image Analysis*, *36*(1), pp.163-174.
13. Dmitriev A., Zhuravlev Yu., Krendelev F. (1966) On mathematical principles of classification. Discrete Analysis, 7, IM SO AS USSR, 3-11 (In Russian).
14. Ge L., Juba B., Vorobeychik Y., 2024. Learning linear utility functions from pairwise comparison queries. *arXiv preprint arXiv:2405.02612*.
15. Kulhandjian M., Aslanyan L., Sahakyan H., Kulhandjian H., D'Amours C., Multidisciplinary Discussion on 5G from the Viewpoint of Algebraic Combinatorics, "CSIT 2019 Revised Selected Papers, IEEE conference proceedings", 2019, pp. 69-76.
16. Liu, Bing, Wynne Hsu, and Yiming Ma. "Integrating classification and association rule mining." In *Proceedings of the fourth international conference on knowledge discovery and data mining*, pp. 80-86. 1998.
17. Liu, Bing, Yiming Ma, and Ching-Kian Wong. "Classification using association rules: weaknesses and enhancements." In *Data mining for scientific and engineering applications*, pp. 591-605. Boston, MA: Springer US, 2001.
18. Martinez-Trinidad Francisco J., Guzman-Arenas A. (2001) The logical combinatorial approach to pattern recognition, an overview through selected works. Pattern Recognition 34, 741-751.
19. Mohri M., Rostamizadeh A., Talwalkar A. (2012) Foundations of Machine Learning. The MIT Press, 414p.
20. Sahakyan H., Aslanyan L., Discrete Tomography with Distinct Rows: Relaxation, CSIT 2017, Revised Selected Papers, IEEE Xplore, 2018.
21. Sahakyan H., Aslanyan L., Ryazanov V., On the Hypercube Subset Partitioning Varieties, "CSIT 2019 Revised Selected Papers, IEEE conference proceedings", 2019, pp. 83-88. doi: 10.1109/CSITechnol.2019.8895211
22. Von Borries G. F., Wang, H. (2009) Partition clustering of high dimensional low sampling size data base on p-values. *Journal Computational Statistics & Data Analysis,* 53(12), 3987-3998.
23. Yablonskii S.V., Chegis L.A. (1955) On tests for electric circuits. Uspekhi Mat. Nauk, 10(4), 182-184.
24. Zhuravlev Yu.I. (1978) On an Algebraic Approach to Solving Recognition or Classification Problems. Problemi Kibernetiki Moscow, Nauka, 33, 5-68.
25. Zhuravlev Yu. I. (1988) Elected Scientific Works. Moscow, (In Russian).
26. Zhuravlev Yu. I., Ryazanov V. V., Aslanyan L. H., and Sahakyan H. A., On a Classification Method for a Large Number of Classes, Pattern Recognition and Image Analysis, 2019, ISSN 1054-6618, Vol. 29, No. 3, pp. 366–376.
27. Zhuravlev Yu. I., Ryazanov V. V., Ryazanov V. V., Aslanyan L. H., and Sahakyan H. A., Comparison of Different Dichotomous Classification Algorithms, Pattern Recognition and Image Analysis, 2020, Vol. 30, No. 3, pp. 303–314.