Transactions on Machine Learning
and Data Mining
Vol. 2, No. 2 (2009) 80-99
©ISSN:1865-6781 (Journal),
ISBN: 978-3-940501-12-7,
IBaI Publishing ISSN 1864-9734

ibai Publishing
www.ibai-publishing.org

Clustering and Separating of a Set of Members in Terms of Mutual Distances and Similarities

Sergey D. Dvoenko

Tula State University, 300600, Tula, Russia,
dsd@tsu.tula.ru

Abstract. In a case of set members are presented via mutual distances or similarities well-known algorithms for clustering (K-means), grouping (Modulus), and learning (Kozinets's) are under investigation. Relationship between K-means and Modulus algorithms is shown based on idea of unbiased partitioning. The problem of learning to recognize set members (objects or features) is under investigation too. Experimental results are shown for feature recognition (Holzinger's psychological tests) and for object recognition (small classes of amino-acid sequences) problems.

Keywords: distance, similarity, dissimilarity, cluster analysis, feature grouping, pattern recognition, learning.

1 Introduction

In context of usual approach to process data, objects are vectors in some abstract mathematical space of features. Basis for intensional analysis of data, represented by a matrix of the "objects-features" type, consists in two informal assumptions of data analysis: the compactness hypothesis and the hypothesis of hidden factors.

First one refers to objects (as rows of the data matrix) and supposes that closed each other by its characteristics objects are not far in an appropriate multidimensional space and are collected in compact concentrations (classes, clusters, taxa). Objects of some compact concentration are supposed to be corresponded with some appropriate inner state of a data source.

Second one refers to features (as columns of the data matrix) and supposes that similarity in behaviour of features (as variational series) is determined by same dependence from some inner characteristic of a data source. This inner

characteristic (hidden factor) is supposed to be inaccessible for direct measurement, but its changing is indirectly reflected in behaviour of features coupled with it.

It is easy to see some sort of symmetry of these hypotheses. So, pro forma, it is possible to change object and feature ideas by simple transposition of the data matrix. But such a way usually seems to be not so natural one. As a result, traditional techniques to analyze objects (clustering, classification, pattern recognition, machine learning) and features (correlation analysis, factor analysis, grouping) are differed as a whole, but, as we can see, are actually based on similar approaches.

In modern data analysis and data mining considerably extended ways are used to represent data about a phenomenon under investigation. Specifically, it is usual way to immediately represent data as a matrix of pairwise comparisons of analyzed set members. Such results of comparisons can be nonnegative numerical values of dissimilarity or, vice versa, similarity of set members. For example, similarity of amino-acid sequences is evaluated based on mutual alignment of polymeric chains of protein molecules by means of some special algorithms [1]. It needs to note, the data matrix can't be used really to represent data in some actual problems [2].

A positively semidefinite similarity matrix can be used as a matrix of pairwise scalar products of objects in some unknown for us metric space (for example, Euclidean) with dimensionality not more than a set cardinal number. As it is known, such a scalar product matrix can be transformed in a distance matrix, and vice versa. As a result, the corresponding dissimilarity matrix can be used as the distance matrix in the same unknown space.

If set members are features, then in many cases it is sufficient to use squares (or absolute values in some cases) of weighted scalar products (correlations) of them to get the positively semidefinite similarity matrix.

As it is known, set members are represented by mutual dissimilarities or similarities in context of multidimensional scaling problem [3, 4]. Solving of this problem is aimed to build well interpreted metric space of features. The problem of factor analysis [5] is closed to it and is aimed to build well interpreted metric space of factors.

If it isn't required to restore an appropriate feature space, we use so called "featureless" approach to data analysis [6]. This approach is widely used today in a view of developing of support vector machine (SVM) and kernel function techniques [7–9] in the pattern recognition and machine learning field. As it was shown in SVM technique, a decision rule is supported in general case by mutual scalar products of objects from a learning set in countably dimensional hypothetical Hilbert space for some kernel function.

With SVM problem we fall into the quadratic programming (QP) problem, and can't reduce in general a wide variety of methods to solve different QP tasks. Specifically, it needs to use usually unfamiliar within for us so-called "solvers" [10]. But usually, it is very suitable for us to use some algorithms with known structure for some cases, at least.

Two basic approaches are known to detect compact concentrations (clusters, groups): by aggregation and by separation. As it is known, main idea of aggregation consists in finding representatives of aggregates. And main idea of separation consists in finding boundary between aggregates. As about objects, the aggregation problem is traditionally solved by cluster analysis and the separation problem – by building decision rules. As about features, the aggregation problem is traditionally solved by factor analysis, centroid and principal extremal grouping techniques (last also known as LPCA – local principal component analysis) [11–13]. But separating of features isn't so popular and traditional technique.

In this paper modifications of two widely used algorithms for aggregation (K-means) and for linear separation (learning algorithm of perceptron type) are investigated in a case of a space of initial (measured) features is not presented.

First idea consists in how correctly determine representatives of aggregates and direction vectors of linear separating hyperplanes.

Second idea consists in demonstrating of real symmetry in using simple and popular algorithms to analyze objects and features as a set members immersed in some metric space. Two experiments demonstrate new results and improved understanding of examples of real data.

2 Clustering of Objects

In cluster analysis problem objects $\omega_i \in \Omega$, $i = 1, \ldots, N$, represented by data matrix $X(N, n)$ are usually considered as vectors $\mathbf{x}_i = (x_{i1}, \ldots, x_{in})$ in n-dimensional feature space. According to cluster analysis idea objects are supposed to be concentrated in K clusters.

Well-known clustering algorithms (like K-means, Isodata [14–16], Forel family [17] for single class problem) are based on the idea of unbiased partitioning. According to it, each cluster Ω_k, $k = 1, \ldots, K$, is represented by "representative" object $\widetilde{\mathbf{x}}_k$, and the cluster center is represented by "mean" object $\overline{\mathbf{x}}_k$. It is unbiased clustering, if equalities $\widetilde{\mathbf{x}}_k = \overline{\mathbf{x}}_k$ are true for all clusters, and this is biased clustering in other cases. So, it needs to appoint mean objects as representatives, and recalculate new mean objects based on minimal distance of objects to representatives.

But in a case of featureless problem mean object $\omega(\overline{\mathbf{x}}_k)$ isn't presented in distance matrix $D(N, N)$ as the cluster center. So, closest to others in the cluster, the object $\overline{\omega}_k$ is usually used as the cluster center. But in general case, if equalities $\widetilde{\omega}_k = \overline{\omega}_k$ are true for all clusters, it appears to be biased clustering, because of center $\mathbf{x}(\overline{\omega}_k)$ isn't mean object $\overline{\mathbf{x}}_k$ in the unknown feature space.

Based on cosines theorem for a triangle, scalar product relative some $\omega_k \in \Omega$ as a point of an origin for a pair of ω_i and ω_j with distance $d_{ij} = d(\omega_i, \omega_j)$ can be calculated as $c_{ij} = (d^2_{ki} + d^2_{kj} - d^2_{ij})/2$, where $c_{ii} = d^2_{ki}$. So, main diagonal of each matrix $C_l(N, N)$, $l = 1, \ldots, N$, represents distances squared from the corresponding origin $\omega_l \in \Omega$ to other objects.

In multidimensional scaling problem it needs to restore the unknown feature space. According to multidimensional scaling idea [18] a positively semidefinite

matrix $C_l(N-1, N-1)$ with rank $n < N$ can be decomposed to $C_l = X_l X_l^T$ with data matrixes $X_l(N-1, n)$. In Torgenson's method of principal projections [4] the origin of unknown feature space is the center of gravity (centroid) of objects $\omega_i \in \Omega$, $i = 1, \ldots, N$.

It is immediately proved the cluster center $\overline{\omega}_k$ is represented by its distances to other objects $\omega_i \in \Omega$, where N_k is number of objects in Ω_k:

$$d^2(\omega_i, \overline{\omega}_k) = \frac{1}{N_k} \sum_{p=1}^{N_k} d_{ip}^2 - \sigma_k^2 \ ,$$

where the cluster dispersion σ_k^2 is

$$\sigma_k^2 = \frac{1}{N_k} \sum_{i=1}^{N_k} d^2(\omega_i, \overline{\omega}_k) = \frac{1}{2N_k^2} \sum_{i=1}^{N_k} \sum_{p=1}^{N_k} d_{ip}^2, \quad \omega_i, \omega_p \in \Omega_k \ .$$

So, K-means clustering algorithm for mutual distances is immediately developed based on idea to put the origin in the center of corresponding cluster and to get unbiased partitioning for all centers:

Step 0. Get K initial representatives $\widetilde{\omega}_k^0$, $k = 1, \ldots, K$, for example, as K most distant objects.

Step s. 1. Reallocate objects $\omega_i \in \Omega_k^s$, if $d(\omega_i, \widetilde{\omega}_k^s) \le d(\omega_i, \widetilde{\omega}_j^s)$, $j = 1, \ldots, K$.

2. Recalculate centers $\overline{\omega}_k^s$, $k = 1, \ldots, K$, based on $d(\omega_i, \overline{\omega}_k^s)$, $i = 1, \ldots, N$.

3. Stop, if $\widetilde{\omega}_k^s = \overline{\omega}_k^s$, $k = 1, \ldots, K$, else $\widetilde{\omega}_k^{s+1} = \overline{\omega}_k^s$, $k = 1, \ldots, K$, $s = s + 1$.

A positively semidefinite similarity matrix $S(N, N)$ with $s_{ij} = s(\omega_i, \omega_j) \ge 0$ can be used as a matrix of scalar products in metric space of not more than N dimensionality. Relative to $\omega_k \in \Omega$ as an origin, where $s_{ij} = (d_{ki}^2 + d_{kj}^2 - d_{ij}^2)/2$, $s_{ii} = d_{ki}^2$, distances are defined based on similarities as $d_{ij}^2 = s_{ii} + s_{jj} - 2s_{ij}$. So, K-means algorithm can be immediately used for mutual distances to get unbiased clustering.

But now the cluster center $\overline{\omega}_k$ can be represented by its similarities to other objects $\omega_i \in \Omega$, $i = 1, \ldots, N$, where N_k is number of objects in Ω_k:

$$s(\omega_i, \overline{\omega}_k) = \frac{1}{N_k} \sum_{p=1}^{N_k} s_{ip}, \quad \omega_p \in \Omega_k \ ,$$

where the cluster compactness as average similarity of its center to other objects in the cluster is

$$\delta_k = \frac{1}{N_k} \sum_{i=1}^{N_k} s(\omega_i, \overline{\omega}_k) = \frac{1}{N_k^2} \sum_{i=1}^{N_k} \sum_{p=1}^{N_k} s_{ip}, \quad \omega_i, \omega_p \in \Omega_k \ .$$

So, clustering algorithms, like K-means again, for mutual similarities relative some predefined origin can be immediately developed as above.

3 Clustering of Features

Let us show how we can use K-means for clustering of features like grouping of them based on mutual similarities.

It is easy to show the unbiased clustering minimizes cluster dispersion σ_k^2 and maximizes its compactness δ_k:

$$\sigma_k^2 = \frac{1}{2N_k^2}\sum_{i=1}^{N_k}\sum_{j=1}^{N_k} d_{ij}^2 = \frac{1}{N_k}\sum_{i=1}^{N_k} s_{ii} - \frac{1}{N_k^2}\sum_{i=1}^{N_k}\sum_{j=1}^{N_k} s_{ij} \ .$$

For normalized similarities $s'_{ij} = s_{ij}/\sqrt{s_{ii}s_{jj}}$, $s'_{ii} = 1$, therefore

$$\sigma_k^2 = 1 - \frac{1}{N_k^2}\sum_{i=1}^{N_k}\sum_{j=1}^{N_k} s'_{ij} = 1 - \delta_k \ .$$

Let us consider features as collection of columns $X_j = (x_{1j}, \ldots, x_{Nj})^T$ of the data matrix $X(N,n) = (X_1, \ldots, X_n)$. In this case similar features are correlated each other, and are represented by weighted scalar products as the correlation matrix $R(n,n)$. Grouping of features represented by squares or moduli of correlations can be built as clustering of them by similarities or by distances (after converting). So, K-means gives unbiased clustering to maximize compactness ($\omega_i, \omega_p \in \Omega_k$, and n_k is number of features in Ω_k):

$$\delta'_k = (1/n_k)\sum_{i=1}^{n_k} r^2(\omega_i, \overline{\omega}_k) = (1/n_k^2)\sum_{i=1}^{n_k}\sum_{p=1}^{n_k} r_{ip}^2 \ ,$$
$$\delta''_k = (1/n_k)\sum_{i=1}^{n_k} |r(\omega_i, \overline{\omega}_k)| = (1/n_k^2)\sum_{i=1}^{n_k}\sum_{p=1}^{n_k} |r_{ip}| \ ,$$

and to maximize criterion functions ($\omega_i \in \Omega_k$):

$$I_1 = \sum_{k=1}^{K} n_k\delta'_k = \sum_{k=1}^{K}\sum_{i=1}^{n_k} r^2(\omega_i, \overline{\omega}_k),$$
$$I_2 = \sum_{k=1}^{K} n_k\delta''_k = \sum_{k=1}^{K}\sum_{i=1}^{n_k} |r(\omega_i, \overline{\omega}_k)| \ .$$

Well-known algorithms Square and Modulus (for publications in Russian, at least) of feature extreme grouping [11–13] maximize criterion functions for "principal" π_k and "centroid" μ_k components for correlation matrix (or factors for reduced one):

$$J_1 = \sum_{k=1}^{K}\sum_{i=1}^{n_k} r^2(\omega_i, \pi_k),\ J_2 = \sum_{k=1}^{K}\sum_{i=1}^{n_k} |r(\omega_i, \mu_k)|,\ \omega_i \in \Omega_k \ .$$

These criteria evaluate partitioning quality of features for specified number of groups, where features from the same group are most correlated ones with the group component (factor).

Let the similarity matrix $S(n,n)$ consists of squares $s_{ij} = r_{ij}^2$ or moduli $s_{ij} = |r_{ij}|$ of correlations between features $\omega_i \in \Omega$ from the matrix $R(n,n)$. It is proved [19] for group Ω_k its principal component π_k, centroid component μ_k, and center $\overline{\omega}_k$ can be represented by their similarities to other features $\omega_i \in \Omega$:

$$s(\omega_i, \pi_k) = \sum_{j=1}^{n_k} \alpha_j^k s_{ij},\ \boldsymbol{\alpha}_k = (\alpha_1^k, \ldots, \alpha_{n_k}^k) \text{ is } 1^{\text{st}} \text{ eigenvector},$$
$$s(\omega_i, \mu_k) = \sum_{j=1}^{n_k} s_{ij} \ ,$$
$$s(\omega_i, \overline{\omega}_k) = (1/n_k) \sum_{j=1}^{n_k} s_{ij} \ .$$

As a result, unbiased grouping by Square (based on principal components of groups) is biased clustering, but unbiased grouping by Modulus (based on centroid components of groups) is unbiased clustering. So, *K*-means for mutual similarities to cluster features is the same as Modulus to group them. It needs to note, development of Modulus algorithm had required to prove special theorems in [11]. Here properties of Modulus are directly inherited from *K*-means algorithm.

4 Quasi-hierarchical Clustering

It is well-known difficult theoretical problem to define suitable number of clusters. As concerns applied problem, first answer is: it would rather specify suitable number K to get clustering (grouping), for example, by *K*-means, Square or Modulus. Second answer is: it would rather find suitable clustering (grouping) to get number K, for example, by Isodata or by hierarchical clustering [14–16, 20].

It needs to note, the hierarchical clustering is the special problem itself in data analysis. But, if this problem doesn't arise, then "yet another" algorithm can be used to define number of clusters (groups). It is suitable to name it as "quasi-hierarchical" one [21]. It can be represented like the following.

Starting from $K = 1$, let us get a least compact subset Ω_k. Let us use a pair of most different set members in it as representatives and build partitioning of Ω_k on two subsets Ω_k and Ω_{K+1} by some clustering algorithm. For $K+1$ subsets let us use two new $\widetilde{\omega}_k$ and $\widetilde{\omega}_{K+1}$ representatives, and $K-1$ previously defined $\widetilde{\omega}_1, \ldots, \widetilde{\omega}_{k-1}, \widetilde{\omega}_{k+1}, \ldots, \widetilde{\omega}_K$ ones. Let us build by the same way partitioning of all N set members on $K+1$ clusters. Stop, when $K = N$.

It is obvious, almost whatever known clustering algorithm with specified number K (*K*-means, at least) can be used here. This idea reduces the problem of finding cluster (group) initial representatives to finding only a pair of most different objects (features) from a least compact cluster (group).

A sequence of partitions, started from $\Omega_1 = \Omega$ and finished with single-element sets $\Omega_1, \ldots, \Omega_N$, is a result. Some partitions in this sequence are in subsequences of hierarchical ones. Two partitions on K and $K+1$ sets are hierarchy, if breaking the least compact set on two subsets Ω_k and Ω_{K+1} immediately

produces unbiased partition on $\Omega_1, \ldots, \Omega_{K+1}$. In this case the partition on K sets is so called "stable" partition for number K.

It is natural to use stable partitions as suitable ones to get result of clustering (grouping) and to specify number K. But biased partitions as the basis for quasi-hierarchical algorithm disintegrate (split to parts) hierarchical subsequences of partitions, reducing the set of suitable numbers K.

This algorithm gives a set of dendrograms like hierarchical ones, but it points to breaks in the hierarchy, and gives better partitions at these moments. If the single hierarchy sequence of partitions appears to be the result of quasi-hierarchical clustering, then this algorithm is proved to be equivalent [21] to the algorithm of cutting of minimal spanning tree (for distances) or the algorithm of cutting of maximal correlation path (for correlations).

5 Separating of Objects

As it is supposed above, a set of objects $\omega_i \in \Omega$ is placed in n-dimensional space. Each object $\omega_i = \omega(\mathbf{x}_i)$ is represented as the vector $\mathbf{x}_i = (x_{i1}, \ldots, x_{in})$, and all of objects are represented by the data matrix $X(N, n)$. Each feature forms corresponding coordinate axis and is represented as observations $X_j = (x_{1j}, \ldots, x_{Nj})^T$, $j = 1, \ldots, n$. In a case of linear separability of a set Ω of objects, for example, on Ω_1 and Ω_2 classes, it needs to build a decision rule like $g(\mathbf{a}, \mathbf{x}) = \sum_{l=1}^{n} a_l x_l + a_0 = (\mathbf{a} \circ \mathbf{x}) + a_0$, where $(\mathbf{a} \circ \mathbf{x})$ is a scalar product of vectors $\mathbf{x}$ and $\mathbf{a}$, $\mathbf{a} = (a_1, \ldots, a_n)$ is a direction vector of a separating hyperplane, a_0 is its offset from the origin, $g(\mathbf{a}, \mathbf{x}) \geq 0$ for $\mathbf{x} \in \Omega_1$, and $g(\mathbf{a}, \mathbf{x}) < 0$ for $\mathbf{x} \in \Omega_2$.

Scalar products $c_{ij} = (\omega_i \circ \omega_j)$ for all pairs of objects $\omega_i, \omega_j \in \Omega$ relative anyone of coordinate points treated as the origin (let it be ω_0) can be represented by distances $c_{ij} = (d_{0i}^2 + d_{0j}^2 - d_{ij}^2)/2$. Here $d_{0p} = d(\omega_0, \omega_p)$ is a distance from an object ω_p to the origin, $d_{pq} = d(\omega_p, \omega_q)$ is a distance between objects ω_p and ω_q. It is evident $c_{ii} = d_{0i}^2$. So, the object ω_0 is represented by its distances to other objects from the learning set Ω.

If classes Ω_1 and Ω_2 are separable ones, then convex covers of sets of different classes aren't intersect each other, at least. If convex covers of separable sets aren't contact each other, then some separating hyperplanes can be built in the gap between them. Usually, the most distant from convex covers of separated sets hyperplane is used in a case for lack of other a priori information. This optimal separating hyperplane provides minimal number of recognition errors.

Let $\mathbf{y} \in \Omega_1$ and $\mathbf{z} \in \Omega_2$ be closest points of convex covers of sets Ω_1 and Ω_2. The optimal hyperplane to separate them is determined by the direction vector $\mathbf{a} = \mathbf{y} - \mathbf{z}$ and by the offset $a_0 = -((\mathbf{y} - \mathbf{z}) \circ (\mathbf{y} + \mathbf{z}))/2$. Let's formulate scalar products $c_{ai} = (\mathbf{a} \circ \mathbf{x}_i)$, $i = 1, \ldots, N$, based on distances

$$c_{ai} = \sum_{l=1}^{n} a_l x_{il} = \sum_{l=1}^{n} (y_l - z_l) x_{il} = c_{yi} - c_{zi} = (d_{0y}^2 - d_{0z}^2)/2 - (d_{yi}^2 - d_{zi}^2)/2 \ .$$

Let's formulate the offset a_0 based on distances too

$$a_0 = -(1/2)\sum_{l=1}^{n}(y_l + z_l)(y_l - z_l) = -(c_{yy} - c_{zz})/2 = -(d_{0y}^2 - d_{0z}^2)/2 \ .$$

Let's take the object $\omega_a = \omega(\mathbf{a})$ and formulate the decision rule $g(\mathbf{a}, \mathbf{x})$:

$$g(\omega(\mathbf{a}), \omega(\mathbf{x})) = g(\omega_a, \omega) = (\omega_a \circ \omega) + a_0 = c_{a\omega} + a_0 \ .$$

As a result, it needs to calculate $(\omega_a \circ \omega) + a_0 = -(d_{y\omega}^2 - d_{z\omega}^2)/2$ to recognize new object ω, where $d_{y\omega} = d(\omega_y, \omega)$ is a distance between objects $\omega_y = \omega(\mathbf{y})$ and ω, and $d_{z\omega} = d(\omega_z, \omega)$ is a distance between objects $\omega_z = \omega(\mathbf{z})$ and ω.

So, the object ω is represented by its distances to objects ω_y and ω_z only, and belongs to a smaller distance class (by distance to the "closest point" of its convex cover). It doesn't need distances to the origin, so it doesn't need coordinates.

On the other part, it needs to calculate $(\omega_a \circ \omega) + a_0 = c_{y\omega} - c_{z\omega} - (c_{yy} - c_{zz})/2$ to recognize new object ω, where $c_{y\omega} = (\omega_y \circ \omega)$ is a scalar product of objects ω_y and ω, and $c_{z\omega} = (\omega_z \circ \omega)$ is a scalar product of ω_z and ω. It is true, that $c_{yy} = c_{zz} = 1$ for normalized scalar products, so $(\omega_a \circ \omega) + a_0 = c_{y\omega} - c_{z\omega}$.

So, the object ω is represented by its normalized scalar products with objects ω_y and ω_z only, and belongs to a nearest class (by similarity with the "closest point" of its convex cover).

But in this case it needs to define the origin. It is convenient to define the origin as the "gravity" center $\overline{\omega}$, and to represent it by distances $d(\overline{\omega}, \omega_i)$ to other objects $\omega_i \in \Omega$ from the learning set (method of principal projections).

So, the scalar product of objects ω_i and ω_j relative to center $\overline{\omega}$ is represented as $c_{ij} = (d^2(\overline{\omega}, \omega_i) + d^2(\overline{\omega}, \omega_j) - d^2(\omega_i, \omega_j))/2$, and the object ω being recognized is represented by its scalar products $c_{y\omega}$ and $c_{z\omega}$ with objects ω_y and ω_z by the same way.

Let the data matrix $X(N, n)$ be given and the distance matrix $D(N, N)$ with elements $d_{ij} = d(\omega_i, \omega_j)$ be calculated. Let's take two vectors $\mathbf{x}_p$ and $\mathbf{x}_q$, and find its convex linear combination $\mu\mathbf{x}_p + (1 - \mu)\mathbf{x}_q$, where $0 \le \mu \le 1$. Let's take two objects $\omega_p = \omega(\mathbf{x}_p)$ and $\omega_q = \omega(\mathbf{x}_q)$. Each of them is represented by distances to other objects $d_{pi} = d(\omega_p, \omega_i)$ and $d_{qi} = d(\omega_q, \omega_i)$, $i = 1, \ldots, N$. Let's note the linear combination of vectors $\mathbf{x}_p$ and $\mathbf{x}_q$ as an object $\omega_\mu = \omega(\mu\mathbf{x}_p + (1 - \mu)\mathbf{x}_q)$ and represent it by distances to other objects $d_{\mu i} = d(\omega_\mu, \omega_i)$, $i = 1, \ldots, N$.

Let the matrix $C(N, N)$ of scalar products $(\omega_i \circ \omega_j)$ of objects $\omega_i, \omega_j \in \Omega$ be calculated relative the initial origin ω_0, where $c_{ij} = (\mathbf{x}_i \circ \mathbf{x}_j) = \sum_{l=1}^{n} x_{il}x_{jl}$. Let's center the data matrix relative the vector $\mu\mathbf{x}_p + (1 - \mu)\mathbf{x}_q$, so specify the object ω_μ as the origin, and calculate a matrix of scalar products $C_\mu(N, N)$ with elements represented by elements of the matrix $C(N, N)$:

$$\begin{aligned} c_{ij}^{\mu} = \sum_{l=1}^{n}(x_{il} - \mu x_{pl} - (1 - \mu)x_{ql})(x_{jl} - \mu x_{pl} - (1 - \mu)x_{ql}) = \\ c_{ij} - \mu c_{ip} - (1 - \mu)c_{iq} - \mu c_{jp} - (1 - \mu)c_{jq} + \\ \mu^2 c_{pp} + 2\mu(1 - \mu)c_{pq} + (1 - \mu)^2 c_{qq} \ . \end{aligned}$$

If $i = j$, then

$$c_{ii}^{\mu} = c_{ii} - 2\mu c_{ip} - 2(1-\mu)c_{iq} + \mu^2 c_{pp} + 2\mu(1-\mu)c_{pq} + (1-\mu)^2 c_{qq} \ .$$

So, diagonal elements $c_{ii}^{\mu} = d^2(\omega_\mu, \omega_i)$ define distances from the object ω_μ to other objects $\omega_i \in \Omega$ in the learning set

$$d^2(\omega_\mu, \omega_i) = \mu d^2(\omega_i, \omega_p) + (1-\mu)d^2(\omega_i, \omega_q) - \mu(1-\mu)d^2(\omega_p, \omega_q) \ .$$

And the object ω_μ is represented by its scalar products $c_{\mu i} = (\omega_\mu \circ \omega_i)$ with other objects relative the origin ω_0:

$$c_{\mu i} = \sum_{l=1}^{n} x_{il}(\mu x_{pl} + (1-\mu)x_{ql}) = \mu c_{ip} + (1-\mu)c_{iq} \ .$$

6 Learning Algorithm

Let's modify the well-known (for publications in Russian, at least) learning algorithm of perceptron type, developed by B.N. Kozinets [22] for feature space.

In our case of the distance matrix $D(N, N)$ is given only, this algorithm, like the original one, must find a formulation ω_a of the optimal hyperplane to separate two classes Ω_1 and Ω_2, or must state, that convex covers of sets being separated intersect each other.

This algorithm finds for limited number of steps such two objects $\omega^+ \in \Omega_1$ and $\omega^- \in \Omega_2$, that the distance between them $d(\omega^+, \omega^-)$ exceeds the distance $d(\omega_y, \omega_z)$ between closest points $\omega_y \in \Omega_1$ and $\omega_z \in \Omega_2$ of convex covers of sets Ω_1 and Ω_2 not more, than for value $\varepsilon d(\omega^+, \omega^-)$, where $0 < \varepsilon < 1$ is sufficiently small previously specified value. To recognize new object ω it needs to calculate the value $(\omega_a \circ \omega) + a_0 = -(d^2(\omega^+, \omega) - d^2(\omega^-, \omega))/2$ or calculate another value $(\omega_a \circ \omega) + a_0 = (\omega^+ \circ \omega) - (\omega^- \circ \omega) - ((\omega^+ \circ \omega^+) - (\omega^- \circ \omega^-))/2$.

If $d(\omega^+, \omega^-) < \eta$, where $0 < \eta < 1$ is sufficiently small previously specified value, then convex covers of sets being separated are supposed to be inseparable. Here is the algorithm in both forms for scalar products and for distances.

Step 0. Define two objects $\omega_0^+ \in \Omega_1$ and $\omega_0^- \in \Omega_2$, for example, as most distant ones from classes Ω_1 and Ω_2. For next steps all objects $\omega_k \in \Omega_1 \cup \Omega_2$ from the learning set are cyclically scanned for some ordering.

Step k. Let, for example $\omega_k \in \Omega_1$. Take the object ω_{k-1}^+ as the origin. Define new object ω_k^+, and leave the object $\omega_k^- = \omega_{k-1}^-$ without modification

1. $\rho = \dfrac{(\omega_k \circ \omega_{k-1}^-)}{(\omega_{k-1}^- \circ \omega_{k-1}^-)} = \dfrac{d^2(\omega_{k-1}^+, \omega_k) + d^2(\omega_{k-1}^+, \omega_{k-1}^-) - d^2(\omega_k, \omega_{k-1}^-)}{2d^2(\omega_{k-1}^+, \omega_{k-1}^-)}$;
2. $\mu = \dfrac{(\omega_k \circ \omega_{k-1}^-)}{(\omega_k \circ \omega_k)} = \dfrac{d^2(\omega_{k-1}^+, \omega_k) + d^2(\omega_{k-1}^+, \omega_{k-1}^-) - d^2(\omega_k, \omega_{k-1}^-)}{2d^2(\omega_{k-1}^+, \omega_k)}$;
3. if $\rho \le \varepsilon/2$, then $\omega_k^+ = \omega_{k-1}^+$,
   if $\rho > \varepsilon/2$ and $\mu \ge 1$, then $\omega_k^+ = \omega_k$,
   if $\rho > \varepsilon/2$ and $\mu < 1$, then $\omega_k^+ = \omega_\mu$;
4. Stop, if ω_k^+ and ω_k^- remain unchanged.

For the item 3 of this algorithm the object ω_μ is represented by its distances

$$d^2(\omega_\mu, \omega_i) = \mu d^2(\omega_i, \omega_k) + (1-\mu)d^2(\omega_i, \omega_{k-1}^+) - \mu(1-\mu)d^2(\omega_k, \omega_{k-1}^+)$$

to objects $\omega_i \in \Omega$ from the learning set and by its scalar products

$$(\omega_\mu \circ \omega_i) = \mu(\omega_i \circ \omega_k) + (1-\mu)(\omega_i \circ \omega_{k-1}^+)$$

with objects $\omega_i \in \Omega$ from the learning set relative the origin ω_0.

Kozinets's algorithm has one important property. In a case of separability relative η it is warranted none of training set $\Omega_1 \cup \Omega_2$ members is projected into interval of the length $(1-\varepsilon)\, d(\omega^+, \omega^-)$ in the middle between points ω^+ and ω^- on the "axis" of the direction object ω_a, where η and ε are sufficiently small previously specified values. Such an interval is bounded by two points on the "axis" of ω_a, where point a_0' is near Ω_1, and point a_0'' is near Ω_2:

$$a_0' = -\left((\omega^+ \circ \omega^+) - (\omega^+ \circ \omega^-)\right) + \varepsilon d(\omega^+, \omega^-)/2\,,$$

$$a_0'' = -\left((\omega^+ \circ \omega^-) - (\omega^- \circ \omega^-)\right) - \varepsilon d(\omega^+, \omega^-)/2\,.$$

We usually use the cross-validation technique to evaluate statistical stability of the learning result.

After learning in our case of separability relative η false negative or false positive errors are encountered, when object ω under validation appears to be single one in the "gap" between convex covers of sets to be separated, but from the "wrong" side of the separating hyperplane.

Nevertheless, if object ω can be separated, it is easy to remove the error by changing the hyperplane offset as $a_0' + \eta$ toward Ω_1 or as $a_0'' - \eta$ toward Ω_2, where $0 < \eta < 1$ is sufficiently small previously specified value to ensure separability.

So, this algorithm is appeared to be simple and it is suitable to use it for preliminary data understanding, at least. Parameters ε and η are very suitable to use them in different optimization tasks for this algorithm. Proving of convergence of this algorithm is simple too, because of distance between closest points $\omega_y \in \Omega_1$ and $\omega_z \in \Omega_2$ of convex covers of sets is limited, and distance between points $\omega^+ \in \Omega_1$ and $\omega^- \in \Omega_2$ converges to it.

It needs to say, this learning algorithm is investigated as one of perception type in [23] and as simple version of SVM one in [24].

7 Separating of Features

Investigation of features $(X_1, \ldots, X_n) = X(N, n)$ is known as the standard problem. Feature interrelations are usually represented by the correlation matrix $R(n, n)$. The problem to analyze feature interrelations is usually stated as the problem of grouping and is targeted to find "representatives" of feature groups.

If members of the learning set $\Omega = \Omega_1 \cup \Omega_2$ are features $\omega_i \in \Omega$, represented by their correlation coefficients, then squares or moduli of correlation coefficients specify the matrix $S(n, n)$ of pairwise normalized similarities of features.

Let the matrix $S(n,n)$ of pairwise similarities $s_{ij} = s(\omega_i, \omega_j) \geq 0$ of features $\omega_i \in \Omega$ be given. If this is the positively semidefinite matrix, then it can be used as the matrix of scalar products of members $\omega_i \in \Omega$ of the learning set in some unknown metric space with dimensionality not higher than n. Such a similarity matrix has n nonnegative eigenvalues with possibly zero ones.

In contrast to traditional approach, let's analyze feature interrelations as the problem of separating. Let us show how we can use Kozinets's algorithm to separate features.

If new set member ω is supposed to be immersed in the same metric space, then it needs to calculate $(\omega_a \circ \omega) + a_0 = s_{y\omega} - s_{z\omega} - (s_{yy} - s_{zz})/2$ to recognize it, where $s_{y\omega} = s(\omega_y, \omega)$ is similarity of members ω_y and ω, and $s_{z\omega} = s(\omega_z, \omega)$ is similarity of members ω_z and ω. If similarities are normalized like $s_{ij}/\sqrt{s_{ii}s_{jj}}$, then $s_{yy} = s_{zz} = 1$. So, the formulation of the decision rule is $(\omega_a \circ \omega) + a_0 = s_{y\omega} - s_{z\omega}$.

So, it needs to calculate $(\omega_a \circ \omega) + a_0 = s(\omega^+, \omega) - s(\omega^-, \omega)$ to recognize new feature ω, when elements $\omega^+ \in \Omega_1$ and $\omega^- \in \Omega_2$ have been found on the learning stage by Kozinets's algorithm. On step k of this algorithm for $\omega_k \in \Omega_1$ values $\rho = \mu = s(\omega_k, \omega^-_{k-1})$ are calculated, and decisions are made: $\omega^+_k = \omega^+_{k-1}$, if $\rho \leq \varepsilon/2$; $\omega^+_k = \omega_k$, if $\rho \geq 1$; $\omega^+_k = \omega_\mu$, if $\varepsilon/2 < \rho < 1$. In the last case new element ω_μ for current step is represented by its similarities $s(\omega_\mu, \omega_i) = \mu s(\omega_i, \omega_k) + (1-\mu)s(\omega_i, \omega^+_{k-1})$ with elements $\omega_i \in \Omega$ of the learning set.

It needs to note, if we want to separate features instead of to group them, we can evaluate quality of the separation by the standard cross-validation technique.

8 Two Experiments

First experiment shows using of clustering and learning algorithms developed here to find group of features and gives improved understanding of well-known real data. Second experiment shows using of learning algorithm to separate objects and gives improved result relative previous one for real data too.

For first experiment let's consider well-known data collected by K. Holzinger [5] about correlations of 24 psychological tests to investigate mentality of individuals. All tests are classified by him in 5 groups: four tests of Spatial Relations (1–4), five tests of Verbal Ability (5–9), four tests of Perceptual Speed (10–13), six tests of Memory (14–19), and five tests of Deduction (20–24). These data are interesting, because of investigation background.

First, data were collected in 1934 as a result of questioning of 145 school children in a suburb of Chicago to demonstrate Bi-factor technique of factor analysis. These data were considered to be complex ones, because of tests from 5^{th} group of Deduction had tendency to be distributed among other groups.

Second, results of extremal grouping of features in 1970 appeared to be different for different starting groups for developed algorithms, and couldn't find Holzinger's groups [12]. As before, differences between results were usually depended on tests from the 5^{th} group again. So, if such tests fell into "wrong" group, then they usually resulted in ejected "proper" tests from this group.

Today in this paper quasi-hierarchical clusterings of tests are obtained based on K-means for following cases: moduli of correlations (MC), squares of correlations (SC), distances based on moduli of correlations (DMC), and distances based on squares of correlations (DSC). Clustering quality is evaluated by criterion function I_1 for SC and DSC cases and by criterion function I_2 for MC and DMC cases. For DMC and DSC cases criterion functions I_1 and I_2 are evaluated based on the expression $\sigma_k^2 = 1 - \delta_k$. In addition, the hierarchical clusterings of tests (H) are obtained for all these cases (MCH, SCH, DMCH, and DSCH).

Table 1. Quality of Clusterings of Psychological Tests

Number of Groups	MCH	MC	DMC & DMCH	MC & DMC	SCH & DSCH	SC & DSC	
1	7.94	7.94	7.94	-	3.43	3.43	-
2	9.64	**9.64**	9.17	-	4.52	**4.52**	-
3	10.91	11.17	10.57	-	5.85	6.04	-
4	11.94	12.00	11.92	-	7.16	7.42	-
5	12.76	13.11	12.66	13.34	8.10	8.34	8.99
6	13.65	13.81	13.36	14.16	8.99	9.25	9.96
7	14.35	14.66	14.25	14.92	10.02	10.40	10.92
8	15.18	15.32	15.23	15.62	11.09	11.47	11.83
9	15.89	16.03	15.94	16.32	12.03	12.39	12.76
10	16.53	16.84	16.61	16.96	12.96	13.29	13.65
11	17.35	17.49	17.33	17.65	13.88	14.26	14.42
12	17.88	18.14	18.05	18.17	14.78	15.19	15.18
13	18.53	18.78	18.64	18.83	15.72	16.03	16.08
14	19.17	19.40	19.29	19.47	16.58	16.79	16.96
15	19.79	20.00	19.89	19.97	17.41	17.67	17.83
16	20.39	20.50	20.39	20.56	18.17	18.51	18.66
17	20.89	21.09	20.98	21.11	19.05	19.34	19.35
18	21.48	21.64	21.55	21.64	19.89	20.03	20.15
19	22.05	22.07	22.11	22.07	20.72	20.85	20.93
20	22.49	22.57	22.54	22.57	21.52	21.65	21.67
21	22.98	23.03	23.03	23.03	22.26	22.39	22.39
22	23.45	23.45	23.45	23.45	22.86	23.04	23.04
23	23.72	23.72	23.72	23.72	23.52	23.52	23.52
24	24.00	24.00	24.00	24.00	24.00	24.00	24.00

Results are reported in Table 1. In this table criterion values for unstable clusterings are showed by bold font. It is clear SC and DSC clusterings are the same (complete SC & DSC and SCH & DSCH columns).

If the quasi-hierarchical clustering resulted in sequence of stable clusterings only, then the hierarchy of clusterings is obtained. In this case quality of clusterings can't be improved (DMC & DMCH column). In other case, if the quasi-hierarchical clustering resulted in the sequence with unstable clusterings, then

Table 1 shows, that quality of clusterings in the hierarchy doesn't exceed one in the corresponding complete quasi-hierarchical clustering (MCH column against MC column and SCH & DSCH column against complete SC & DSC column). Breaking of the hierarchy is the moment of unstable clustering, so, later clusterings appear to be with the better quality.

Table 2. Group Patterns of Psychological Tests

	Group of Tests	HZ	MC	DMC & DMCH SC & DSC	MCH	SCH & DSCH
1.	Spatial Relations	1–4	1, 3 2 4, 22	1, 3 2 4	1, 3 2 4, 22	1, 3 2 4, 21
2.	Verbal Ability	5–9	5–9	5–9	5–9	5 6–9
3.	Perceptual Speed	10–13	10–13	10–12 13, 21	10–12, 24 13	10–12 13
4.	Memory	14–19	14, 16 15, 17 18, 19	14, 16 15 17, 18 19, 22	14, 16 15, 17 18, 19	14, 16 15 17, 18 19
5.	Deduction	20–24	20, 23 21, 24	20, 23 24	20, 23 21	20, 23 22 24
	Number of Clusters	5	10	12	11	14
	Distance to HZ Groups	0	52	64	66	70

But Table 1 shows Holzinger's groups are stable ones for all cases to represent mutual similarity and dissimilarity of tests in this paper (incomplete MC & DMC and SC & DSC columns started from 5 groups). More over, corresponding sequences of clusterings appear to be hierarchies.

At the same time, as it is expected in general, Holzinger's groups haven't been restored for none of cases used above. So, the clustering most closed to Holzinger's one is defined for each of cases based on Hamming distance between them. These clusterings are reported in Table 2.

Their distances to Holzinger's original groups (HZ) are shown in last row. Complexity of these data is uncovered by following: original test groups usually need to be broken onto more small clusters to separate themselves from others.

Columns of Table 2 show all cases of clusterings. Members of each small cluster are placed on standalone row. It is most difficult to separate tests from 5^{th} group of Deduction again. In different attempts tests 21, 22, and 24 are extracted

as isolated ones as a result of breaking of other groups to small clusters. This result corresponds with the idea of a probably special role of these tests.

As a result, hierarchical clusterings for these data (MCH, SCH, DMCH, and DSCH cases) are worst of all relative to original HZ case. The MC clustering for 10 groups of tests agrees with HZ one best of all among others (DMC, SC, and DSC cases). This result corresponds to the idea about multiextremal character of criterion functions J_1 and J_2 reported in [12], and to the same character of criterion functions I_1 and I_2 here.

However, stability of Holzinger's groups and stability of all next fragmentations of them indicates that original groups of psychological tests are compact subsets of features. It is easy to see the Bi-factor analysis problem of these tests can be considered as the problem of learning to recognize of test groups, so, as one to get classification of them.

So, linear decision rules are built to separate each test group from others by Kozinets's algorithm for MC case. These decision rules completely separate each test group against others.

Table 3. Cross-validation Results

Test Groups	n_1	n_2	n_{11}	n_{12}	n_{21}	n_{22}	p_{11}	p_{12}	p_{21}	p_{22}
1	4	20	2	2	0	20	0.50	0.50	0.00	1.00
2	5	19	5	0	0	19	1.00	0.00	0.00	1.00
3	4	20	4	0	0	20	1.00	0.00	0.00	1.00
4	6	18	5	1	0	18	0.83	0.17	0.00	1.00
5	5	19	3	2	0	19	0.60	0.40	0.00	1.00
Total	24	96	19	5	0	96	0.79	0.21	0.00	1.00

This result of learning is evaluated by the standard cross-validation technique. Results of cross-validation in a case of "one against others" are reported in Table 3.

Here n_1 is number of tests from the group given (1^{st} "right" class), n_2 is number of tests from other groups (2^{nd} "wrong" class), n_{11} is number of correctly recognized tests from the group given, n_{22} is number of correctly recognized tests from other groups, n_{12} is number of incorrectly recognized tests from the group given, n_{21} is number of tests from other groups incorrectly recognized as tests of the group given. So, following probabilities are evaluated for sensitivity (p_{11}), false negative errors (p_{12}), false positive errors (p_{21}), and specificity (p_{22}).

Table 3 shows that average rate of correct recognition of single group of tests against others for cross-validation isn't so high (79%). It is clear, tests of 1^{st} (Spatial Relations) and 5^{th} (Deduction) groups are recognized with a quite low quality (50% and 60%). As about group of Deduction this result isn't so unexpected.

But Table 3 shows cross-validation recognition errors are of single type, specifically, in a case of "right" test is recognized as "wrong" one only. It is easy to see, alone group contains small number of tests against taken together other groups. So, a "shape" of each group of tests in hypothetical space appears to be unstable and can determine low result of the cross-validation quality of recognition (high error rate).

So, only false negative error is encountered, because of set member ω under validation appears to be single one in the "gap" between convex covers of sets to be separated, and from the "wrong" side of the separating hyperplane.

As a result, it is easy to remove this error by changing the hyperplane offset toward Ω_2 as $a_0'' - \eta$ to ensure separability of the point $\omega^- \in \Omega_2$ and the member ω under validation.

Finally, all Holzinger's groups of tests are correctly recognized by cross-validation technique after such a type adjusting of decision rules for 1^{st}, 4^{th}, and 5^{th} groups.

This result proves the idea that Holzinger's groups are compact sets of features.

For second experiment let's consider a collection of proteins selected by Dr. S.-H. Kim from Lawrence Berkeley National Lab. This collection contains 420 remote protein domains of 51 fold classes from the SCOP database [25]. This set of proteins provides as a little similarity of amino-acid sequences within each family, as possible. The least threshold of 27% of the similarity degree was accepted for pairwise alignments by the Fasta program [1].

As a result, there were small fold classes relative the whole size of the protein collection, where smallest ones contain 3 domains only, and largest ones contain 38, 31 and 20 domains.

One of fundamental ideas of molecular biology consists in that the primary structure of a protein, i.e. a sequence of amino-acid residues, holds essential information to establish unambiguously its spatial structure.

As a rule, a fold pattern remains the same within large groups of evolutionarily allied proteins. As a result, the set of essentially different spatial structures is much less than the set of known proteins. Therefore, the problem to estimate the spatial structure of a given protein falls into the field of the pattern recognition problem.

Because of initial matrix $S(N,N)$ had five small negative eigenvalues, all similarities $s(\omega_i, \omega_j)$ in it were normalized to $s_{ij}/\sqrt{s_{ii}s_{jj}}$ and squared to get positively semidefinite similarity matrix with eigenvalues in the range from $\lambda_1 = 15.752$ to $\lambda_{418} = 0.336$, where $\lambda_{419} = \lambda_{420} = 0$.

Therefore, it was supposed objects $\omega_i,\ i = 1,\ldots,N$, were represented by scalar products $(\omega_i \circ \omega_j)$ with other ones $\omega_j,\ j = 1,\ldots,N$, in unknown metric space with dimensionality not more, than 420 of size.

To use well-known clustering and recognizing algorithms without modifications it was convenient in [2] to represent immediately each object ω_i by such similarities $s_{ij},\ j = 1,\ldots,N$, as a vector in 420-dimensional space. Such a space was denoted in [2] as so-called projection (not usual feature) space.

Preliminary analysis of this collection of proteins reported general poor separability as a whole. As a result, it was poor separability of fold classes in cross-validation "one against others": only 14 ones from 51 were recognized with 80,6% of reliability (75 from 93 objects). Table 4 shows names and sizes of such well-separated fold classes.

Table 4. Well-separated 14 Classes

Class	Name	Size	Sensitivity	Method	Improved
1	Globin	12	92% (11)	PP	100% (12)
2	Cytochrome C	7	100% (7)	1NN	100% (7)
6	EF Hand	13	85% (11)	3NN	92% (12)
7	Cyclin	4	75% (3)	1NN	100% (4)
8	Cytochrome P450	5	80% (4)	PP	100% (5)
11	Cupredoxins	9	78% (7)	3NN	100% (9)
14	Crystallins/protein S/yeast killer toxin	5	60% (3)	PP	100% (5)
21	Acid proteases	5	60% (3)	PP, 3NN	100% (5)
23	Lipocalins	6	50% (3)	PP	83% (5)
25	Barrel-sandwich hybrid	6	67% (4)	PP, 3NN	100% (6)
39	Periplasmic binding protein I	7	86% (6)	3NN	100% (7)
47	N-terminal nucleophile aminohydrolases	4	100% (4)	1NN	100% (4)
49	C-type lectin	6	83% (5)	1NN	100% (6)
50	Protein kinases (PK), catalytic core	4	100% (4)	3NN	100% (4)

So, total reliability appeared to be not more than 22% only (93 from 420 objects). Table 4 also shows sensitivity levels and used methods. Some results were achieved by building a posteriori probability decision rule (PP), other ones – by nearest neighbor (NN) algorithm [2].

Another idea used here consists in not to represent objects immediately as vectors in high-dimensional projection space with unknown characteristics, but represent objects only by mutual similarities in unknown metric space.

But in this case, it needs to use modified versions of clustering and recognition algorithms, as described above.

It is easy to see, alone fold class contains small number of objects against taken together other classes. A "shape" of some alone class in hypothetical space appears to be unstable one and can determine low result of the cross-validation quality of recognition (high error rate). Experiments prove that cross-validation recognition errors appear to be of single type, specifically, "right" object is recognized as "wrong" one only. Therefore, the offset a_0 needs to be changed as $a_0'' - \eta$ to get better result.

The last column in Table 4 shows the total result appears to be much better. Table 4 shows result for well-separated 14 classes in cross-validation again by

Kozinets's algorithm with 2,6% of error rate (only 2 from 93 objects). The total reliability for all fold classes improves up to 73% (307 from 420 objects).

Table 5. Poor-separated 9 Classes

Class	Name	Size	Sensitivity	Improved
10	Common fold of difteria toxin/ trnscription factors/cytochrome	5	20.0% (1)	20.0% (1)
12	C2 domain	3	33.3% (1)	33.3% (1)
33	Thioredoxin fold	5	20.0% (1)	20.0% (1)
36	S-adenosyl-L-methionine-dependent methyl-transferases	5	20.0% (1)	20.0% (1)
41	Lysozyme	4	25.0% (1)	25.0% (1)
13	Viral coat and capsid proteins	15	6.7% (1)	46.7% (7)
17	OB-fold	17	5.9% (1)	11.8% (2)
24	Double-stranded beta-helix	6	16.7% (1)	33.3% (2)
44	Cystatin	7	14.3% (1)	42.9% (3)

From other side, this result appears to be reduced from possible the best one by the set of 9 classes with low level of sensitivity, less than 50% (Table 5).

Experiments show, that it is impossible to improve sensitivity for classes 10, 12, 33, 36, 41 by adjusting of decision rule as above. Sensitivity for classes 13, 17, 24, 44 has been improved in very small degree.

And at last, improved sensitivity for other 28 classes is reported in Table 6.

9 Conclusions

In a case of difficulties to restore unknown feature space, a set of objects under investigation can be represented by results of mutual comparisons as dissimilarity or similarity matrix. This is the standard approach in many cases today, such as for expert opinions, protein sequences, signatures, etc.

From other hand, a set of features under investigation is usually represented by the matrix of pairwise correlation coefficients, while a space of features is usually restored later, as the problem, for example, of hidden factors evaluating.

Under such conditions, it is convenient to consider a collection of objects or features as a set of members immersed in metric space to establish explicitly unified approach to analyze them.

In this paper modifications of two algorithms are proposed for cluster analysis (K-means) and for machine learning (Kozinets's linear decision rule). Two experiments are shown to demonstrate new results and improved understanding of examples of real data.

Table 6. Separating of 28 Classes

Class	Name	Size	Sensitivity	Improved
3	Four-helical bundle	8	12.5% (1)	75.0% (6)
4	Ferritin	8	12.5% (1)	87.5% (7)
5	Four-gelical cytokines	11	9.1% (1)	100% (11)
9	Immunoglobin beta-sandwitch	31	22.6% (7)	71.0% (22)
15	Galastose-binding domain	4	25.0% (1)	75.0% (3)
16	ConA lectins/glucanases	8	12.5% (1)	50.0% (4)
18	Beta-Trefoil	5	20.0% (1)	60.0% (3)
19	Reductase/isomerase/elongation factor common domain	4	25.0% (1)	75.0% (3)
20	Trypsin serine proteases	6	16.7% (1)	66.7% (4)
22	PH domain	7	14.3% (1)	57.1% (4)
26	TIM barrel	38	2.6% (1)	100% (38)
27	Flavodoxin	9	11.1% (1)	55.6% (5)
28	Adenine nucleotide alpha hydroclase	4	25.0% (1)	50.0% (2)
29	Rossmann-fold domains	14	21.4% (3)	78.6% (11)
30	Thiamin-binding	3	33.3% (1)	66.7% (2)
31	P-loop containing NTP hydrolases	9	11.1% (1)	55.6% (5)
32	Thioredoxin fold	9	11.1% (1)	88.9% (8)
34	Ribonuclease H motif	9	11.1% (1)	55.6% (5)
35	Phosphoribosyltransferases (PRTases)	3	33.3% (1)	66.7% (2)
37	Alpha/beta-Hydrolases	12	8.3% (1)	100% (12)
38	Phosphorylase/hydrolase	5	20.0% (1)	60.0% (3)
40	Periplastic binding protein II	7	14.3% (1)	57.1% (4)
42	Cysteine proteinases	4	25.0% (1)	75.0% (3)
43	Beta-Grasp	8	12.5% (1)	75.0% (6)
45	Ferredoxin	20	5.0% (1)	55.0% (11)
46	Zincin	7	14.3% (1)	100% (7)
48	ADP-ribosylation	4	25.0% (1)	100% (4)
51	Beta-Lactamase/D-ala carboxypeptidase	3	33.3% (1)	66.7% (2)

Acknowledgments

This work was partially supported by the Russian Foundation for Basic Research grants 06-01-00412, 08-01-99003, 08-01-12023, by the INTAS grant 04-77-7347. I would like to thank my colleagues in the INTAS Project M. Schlesinger and V. Hlavac for the kindly provided book [23] published in Russian; B. Flach, V. Hlavac, J. Kittler, and V. Mottl for discussion during the Project Coordination Meeting in Prague. I would like to thank N. Zagoruiko for discussion of this work. And I would like to thank I. Muchnik for helpful discussions of methods of extremal grouping during my visit to Rutgers University some years ago.

References

1. Pearson, W.R.: Rapid and Sensitive Sequence Comparison with FASTP and FASTA. Methods in Enzymology. 183, 63–98 (1990)
2. Mottl, V.V., Dvoenko, S.D., Seredin, O.S., Kulikowski, C.A., Muchnik, I.B.: Featureless Pattern Recognition in an Imaginary Hilbert Space and Its Application to Protein Fold Classification. In: Perner, P. (ed.) MLDM 2001. LNCS, vol. 2123, pp. 322–336. Springer Berlin / Heidelberg (2001)
3. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling. 2nd ed. Chapman & Hall/CRC (2001)
4. Torgenson, W.S.: Theory and Methods of Scaling. John Wiley & Sons, N.Y. (1958)
5. Harman, H.H.: Modern Factor Analysis. Univ. Chicago Press, Chicago (1976)
6. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore (2005)
7. Vapnik, V.N.: Statistical Learning Theory. Adaptive and Learning Systems. John Wiley & Sons, N.Y. (1998)
8. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer Verlag, N.Y. (1995)
9. Aizerman, M.A., Braverman, E.M., Rozonoer, L.I.: The Method of Potential Functions in Theory of Machines Learning (in Russian). Nauka, Moscow (1970)
10. The MOSEK Optimization Tools Manual. Version 6.0 (Revision 53). MOSEK ApS, Denmark (2009)
11. Braverman, E.M.: Methods for the Extremal Grouping of Parameters and the Problem of Determining Essential Factors. Automation and Remote Control. 1, 108–116 (1970)
12. Lumel'skii, V.Ya.: Parameter Grouping on the Basis of the Square Coupling Matrix. Automation and Remote Control. 1, 117–127 (1970)
13. Braverman, E.M., Muchnik, I.B.: Structured Methods of Empirical Data Processing (in Russian). Nauka, Moscow (1983)
14. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer Science + Business Media LLC (2006)
15. Webb, A.R.: Statistical Pattern Recognition. John Wiley & Sons (2002)
16. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, N.Y. (2001)
17. Zagoruiko, N.G.: Applied Methods of Data and Knowledge Analysis (in Russian). Sobolev Institute of Mathematics, Novosibirsk (1999)
18. Young, G., Householder, A.S.: Discussion of a Set of Points in Terms of Their Mutual Distances. Psychometrika. 3, 19–22 (1938)
19. Dvoenko, S.D.: Clustering of a Set, Described by Paired Distances and Similarities Between Its Elements (in Russian). Journal of Applied and Industrial Mathematics. 12(1), 61–73. (2009)
20. Ward, J.: Hierarchical Grouping to Optimize an Objective Function. J. of ASA. 58, 236–244 (1963)
21. Dvoenko, S.D.: Restoration of Spaces in Data by the Method of Nonhierarchical Decompositions. Automation and Remote Control. 62(3), 467–473. (2001)
22. Kozinets, B.N.: The Recurrent Algorithm to Separate Convex Covers of Two Sets (in Russian). In: Vapnik, V.N. (ed.) Algorithms of Learning for Pattern Recognition, pp. 43–50. Soviet Radio, Moscow (1973)
23. Schlesinger, M.I., Hlavac, V.: Ten Lectures on Statistical and Structural Pattern Recognition. Springer Science + Business Media LLC (2002)

24. Frank, V., Hlavac, V.: An Iterative Algorithm Learning the Maximal Margin Classifier. Pattern Recognition. 36(9), 1985–1996 (2003)
25. Dubchak, I., Muchnik, I., Mayor, C., Dralyuk, I., Kim, S.-H.: Recognition of a Protein Fold in the Context of the SCOP Classification. Proteins: Structure, Function, and Genetics, 35, 401–407 (1999)

Vitae

Dr. Sergey Dvoenko received his Ph.D. degree in Computer Science by Institute of Control Sciences (Moscow) of Russian Academy of Sciences in 1992, and received his Dr.Sci. degree in Computer Science by Computing Center (Moscow) of Russian Academy of Sciences in 2002. He is a professor in Tula State University, Russia, since 2003. He is a member of Russian Federation Association for Pattern Recognition and Image Analysis (RAPRIA). His research interests include cluster analysis and data mining, machine learning and pattern recognition, image analysis, hidden Markov models and Markov random fields in applied problems.