UDC 519.254 MSC 68H30

Volatility in Classification

A. A. Rubchinsky

National Research University "Higher School of Economics"
International Laboratory of Decision Choice and Analysis
5a, Vorontsovo Pole str., Moscow, Russia
Department of Applied Mathematics and Informatics
Dubna International University for Nature, Society and Man
19, Universitetskaya str., Dubna, Moscow region, Russia, 141980

The goal of the presented work consists in the construction of the new three-levels scheme of authomatical classification. This scheme is based on the newly introduced notion of ***volatility*** of separate clusters as well as of whole classification. The property is exactly defined and efficiently calculated. It describes the stability, exactness, validity of subsets of the given initial set – in essence, their possibility (or impossibility) to be selected as clusters. The suggested algorithm finds the clusters with arbitrary levels of volatility, including the conventional case of zero volatility. The clusters in USA, Russia and Sweden stock market (for crisis period of 2008-2010) and deputies clusters based on voting results in the 3rd State Duma between September 2001 and January 2002 (the period including the creation of the party "United Russia" 01.12.2001) were constructed by the suggested algorithm. Analyzing clusters constructed basing on the voting results for every of the considered months, it has turned out that the clustering volatility was equal to zero in September and October, drastically increased in November and slightly decreased in December and January. But several indices (i.e. concordance of parties' positions) did not show sensible jumps near this political "bifurcation point". The other considered various model examples demonstrated the results well-coordinated with geometrical intuition.

Key words and phrases: cluster analysis, automatic classification, volatility, cut in graph, stock market, State Duma.

1. Introduction

The well-known clustering problem consists in selection from a given set of objects several non-intersecting subsets (usually called clusters, aggregates, blocks, classes, etc.). It is required that every cluster consists of objects that are in some sense closely connected, similar in appearance, while objects belonging to different clusters are as unlike as possible, significantly distinct. In classification problems it is required additionally that the selected clusters form a division of the initial set, but the abandoning of this requirement seems more realistic in the considered situations. Informal character of the clustering problem, its various modifications, statements and applications, numerous approaches and methods of its solution are comprehensively described in several monographs and reviews (see, for instance [1–7]).

The goal of the presented work consists in elaboration of a clustering algorithm, which takes into account the ***volatility*** of subsets of the given initial set. That important property describes their stability, exactness, validity — in essence, their possibility (or impossibility) to be selected as clusters. Volatility is determined formally for separate candidates as well as for the whole clustering problem.

Let us consider some examples without giving exact definition of volatility but rather to give some hints to its future definition. Clusters with different volatility are shown in Fig. 1. In Fig. 1a and 1b three considered clusters have volatility 0, despite the fact that selection of clusters in Fig. 1b is more difficult than selection of the same clusters in Fig. 1a. The clusters shown in Fig. 1c have different volatilities. Intuitively cluster 1 has the same volatility 0, cluster 2 has some small volatility, and volatility of cluster 3 exceeds volatility of cluster 2. Finally the cluster 3 in Fig. 1d practically disappears (its volatility is close to the maximal number 1), meanwhile clusters 1 in all the pictures has the same volatility, as well as cluster 2 in Fig. 1c and 1d.

Received 30$^{\text{th}}$ December, 2013.

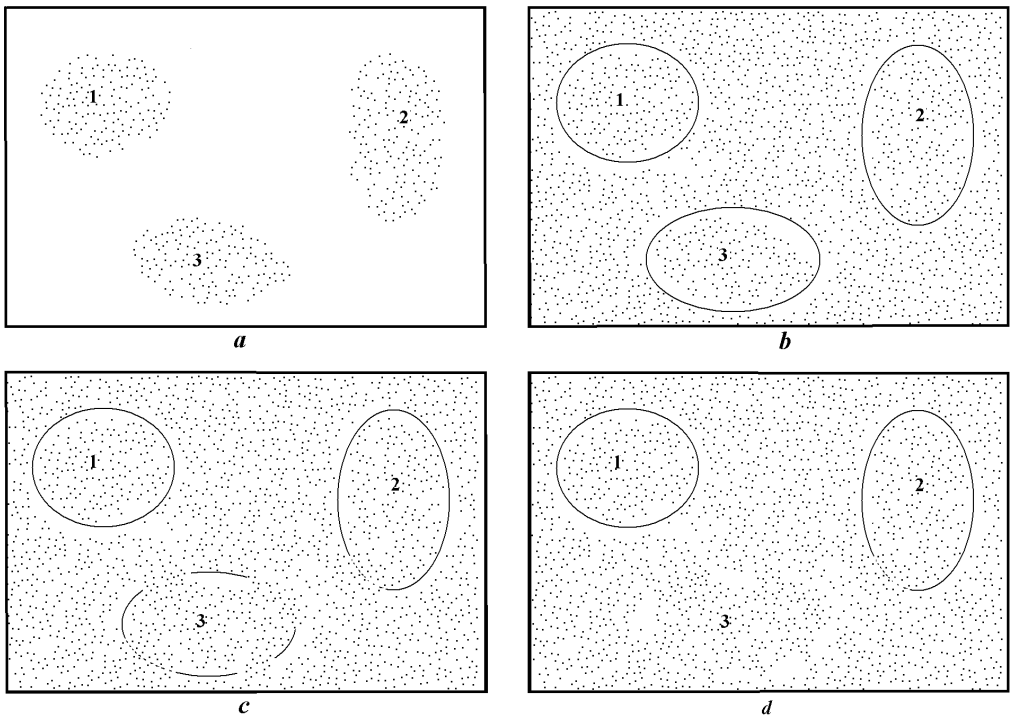

Figure 1. Clusters with different volatility

Usually the notions of volatility, stability, and so on are connected with the process of changes of a considered system in dependence on time or other external parameters. In the suggested approach to clustering, however, this it is not the case. Volatility is determined for a given clustering problem. The essence is that the suggested clustering algorithm (like some other ones) consists of repeating randomized steps. At every step a family of subsets (candidates for clusters) is constructed. Clear-cut clusters with zero volatility are absolutely the same at every algorithm run. Less clear clusters can be slightly different or/and occur not at every algorithm run. This reasoning enables to formulate a simple formal criterion, whose maximization defines volatility of a considered cluster. The volatility of the whole clustering problem is determined as weighted sum of volatilities of the found clusters.

It seems that high level of volatility corresponds to difficulty of a clustering problem, and realization of this connection led to the new clustering algorithm. The suggested algorithm finds all the clusters whose volatility does not exceed an arbitrary feasible level of volatility (including the conventional case of zero volatility). Moreover, the feasible level of volatility is one of very few external parameters of the suggested algorithm. It is one that essentially depends upon human decision.

2. The Structure of the Clustering Algorithm

The algorithm is determined as a three-level procedure. The external level correspond to the repeatedly constructions of a family of different divisions of the initial set of objects into $2, 3, \ldots, k$ clusters. The selection of reasonable clusters, basing on all the found families, is exposed — as a separate algorithm — in Section 3.

Every of the above-mentioned families of divisions is determined as a result of one run of the suggested Divisive-Agglomerative Classification Algorithm (DACA). The algorithm is described in Section 4. It presents the intermediate level of the suggested clustering algorithm.

DACA itself is based on the dichotomy algorithm that is the internal level of the suggested general procedure. This algorithm is a new version of so called frequency approach, suggested by Girvan and Newman in 2002. As well as some other algorithms, including spectral and kernel ones, it also gives some approximation of the balanced cut problem (*NP*-complete combinatorial problem of division of a graph into two subgraphs). In difference to them, the suggested algorithm allows us to understand the shortages of divisions, obtained even by exact solution of balanced cut problem. But the using of the suggested algorithm in the framework of DACA allows overcoming the above shortages despite the fact that some dichotomies can be wrong. The two levels of the algorithm (internal and intermediate ones) were comprehensively considered in [8], available at `http://www.hse.ru/org/hse/wp/wp7`. Therefore they are exposed here briefly enough.

3. New Frequency Dichotomy Algorithm

In the Girvan-Newton frequency algorithm [2, 8] a path, connecting a next pair of vertices, is traced independently of all the already traced paths. Taking into account all the already traced paths yet can obtain cuts between two sets of vertices whose all the edges have the same maximal frequency. Then concurrent removal of all the edges with the maximal frequency defines the desired dichotomy of the graph. The algorithm is as follows.

Minimax frequency algorithm of graph dichotomy. The input of the algorithm is an undirected connected graph G. There are two integer algorithm parameters:

- maximal initial value f of edge frequency (typical value is $10 - 20$);
- number of repetition T for statistics justification (typical value is $1000 - 3000$ independently of the initial graph size).

1. Preliminary stage. Frequencies in all the edges are initialized by integer numbers uniformly distributed on the segment $[0, f - 1]$.
2. Cumulative stage. The operations of steps 2.1 – 2.3 are repeated T times:
   2.1. Random choice of a pair of vertices of graph G.
   2.2. Construction of a minimal path (connecting the two chosen vertices, whose longest edge is the shortest one among all such paths) by Dijkstra algorithm. The length of an edge is its current frequency.
   2.3. Frequencies modification. 1s are added to frequencies of all edges belonging to the path found at the previous step 2.2.
3. Final stage.
   3.1. The maximal (after T repetitions) value of frequency f_{max} in edges is saved.
   3.2. The operations of steps 2.1 – 2.3 are executed once.
   3.3. The new maximal value of frequency f_{mod} in edges is determined.
   3.4. If $f_{\mathrm{mod}} = f_{\mathrm{max}}$, go to step 3.2; otherwise, go to the next step 3.5.
   3.5. Deduct one from frequencies in all edges forming the last found path.
   3.6. Remove all the edges, in which frequency is equal to f_{max}.
   3.7. Find two connectivity components of the modified graph. The two constructed sets of vertices form the solution of the considered dichotomy problem.

Let us to consider the expression

$$R(A, B) = d(A, B) \times \left(\frac{1}{|A|} + \frac{1}{|B|} \right), \tag{1}$$

determined for any cut (A, B) of the graph $G(V, E)$, where $B = V \backslash A$, $d(A, B)$ is equal to the number of edges, connecting sets A and B. This function is well known as ratio cut criterion, determined for the set of all the cuts of the graph $G(V, E)$. It is known [1, 4] that spectral and kernel methods find cuts, approximately minimizing criterion (1).

Taking into account hundreds of computational experiments with various data (partially presented in [8], it is possible to make the following informal conclusions.

1. Well known ratio cut criterion (and, hence, approximating it spectral and kernel methods) can give intuitively wrong answers in relatively simple cases.
2. All the stochastically stable dichotomies found by the suggested minimax algorithm are intuitively correct and minimize criterion (1).
3. All the stochastically unstable dichotomies found by the suggested minimax algorithm are intuitively incorrect.

However, the notion of stability itself is not exactly defined. Between clear stable and clear unstable cases there is a "gray zone" of weak instability. Like other situations of such a type, occuring in most fields of pure and applied mathematics, these situations arise when a system is in transition process from one stable state to another one. Therefore they are — in some sense — inevitable. Examples in Section 6 demonstrate that such phenomena can really occur in clustering problems.

4. Divisive-Agglomerative Classification Algorithm

In this Section we describe the intermediate algorithm (named DACA), whose flow-chart is shown in Fig. 2. Its main idea consists in consecutive execution of divisive and agglomerative operations. It is done in order to keep strong properties of the suggested method of dichotomy and to be got rid of its shortages.

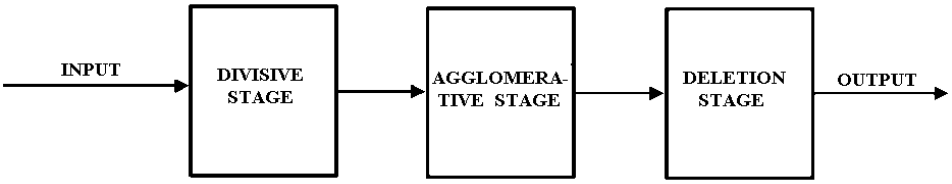

Figure 2. Flow-chart of DACA

The only algorithm parameter of DACA is the maximal number K of parts in the divisive stage. Its input is an arbitrary undirected graph. Let us consider the blocks of flow-chart in Fig. 2 separately.

1. DIVISIVE STAGE. The graph is consecutively divided into two subgraphs by the minimax algorithm of dichotomy. For the division at every step the graph with the maximal number of vertices is selected. The number of divisions is equal to $K-1$. The output of the stage is the family of inserted classifications $D=(D_2,D_3,\ldots,D_K)$ into $2,3,K$ classes.
2. AGGLOMERATIVE STAGE. Every classification D_j into j classes determines the subfamily of classification into j classes (D_j itself), into $j-1$ classes (obtained by the union of subgraphs, connected be the maximal number of edges), and so on, in correspondence with the convenient agglomeration scheme (joining subsets, connected by the maximal number of edges). Denote the constructed classifications as $C_j^j, C_{j-1}^j, \ldots, C_2^j$ and present them as follows (these classifications are diagonally placed):

$$C_2^2, C_2^3, \ldots, C_2^{k-1}, C_2^k$$
$$C_3^3, C_3^4, \ldots, C_2^k$$
$$\ldots\ldots\ldots\ldots$$
$$C_{k-1}^{k-1}, C_{k-1}^k$$
$$C_k^k$$

3. DELETION STAGE. In every row after deletion of all the doubling classifications at least one classification remains. Write all of them out:

$$F_2^1, \ldots, F_2^{n_2} \text{ (into 2 classes)}, F_3^1, \ldots, F_3^{n_3} \text{ (into 3 classes)}, \ldots, F_k^1 = C_k^k \text{ (into classes)},$$

This list is the output of the suggested DACA.

The essential remaining problem is the follows one: how to select (automatically) the correct classification or the most reasonable separate clusters from the list, found at the above described inter-mediate stage. This problem is considered in the next Section 5.

5. Clusters Selection Algorithm

Before starting the algorithm description, let us describe its input in more detail. Assume r is the number of independent runs of DACA. Because every run uses random numbers (for instance, for consecutive choice of pair of vertices in the internal minimax algorithm, described in Section 3), DACA produces at every run a family of classifications. Generally speaking, these families can be different, though they coincide in many simple cases. Moreover, the quantitative measure of their coincidence (that will be defined in this section) can be considered as a formal measure of complexity of a given clustering problem.

Let us introduce some necessary definitions and notations. Assume U_i is the set of all the clusters included in all the classifications found by DACA at its i-th run. All the elements of $U_i (i = 1, \ldots, r)$ are candidates for clusters. For simplicity, they are named "clusters".

Assume F be an arbitrary family of clusters, belonging to different sets U_i. It will be convenient to present F as follows:

$$\mathrm{F} = \langle F_{i_1}, \ldots, F_{i_d} \rangle \text{ where } F_{i_k} \in U_{i_k} (k = 1, \ldots, d), \text{ and } s < t \text{ implies } i_s < i_t. \qquad (2)$$

Denote

$$A(F) = \cap F_j, B(F) = \cup F_j, \alpha(F) = |A(F)|/|B(F)|, \qquad (3)$$

where intersection and union are taken over all sets F from family F. It is clear that α(F) cannot exceed 1. Family F, such that $\alpha(\mathrm{F}) > 0.5$, are named ***α-stable***. The proximity of α(F) to 1 means the stability of a set in all the runs there it appears. This notion is illustrated in Fig. 3. The three families P, Q and R are separately shown in Fig. 4 in more detail.

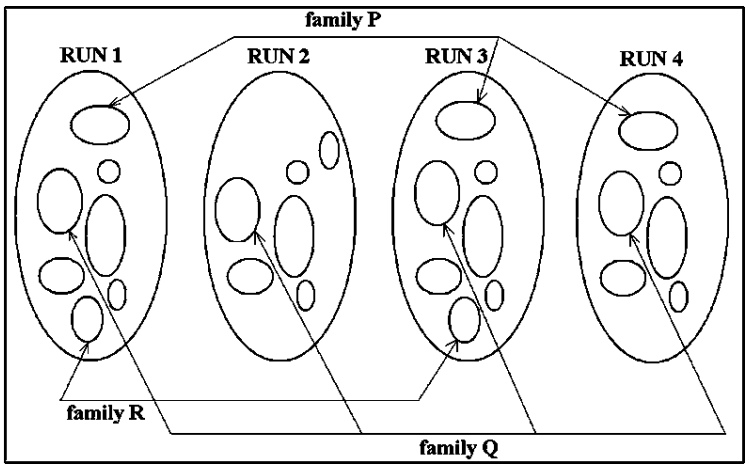

Figure 3. Results of 4 runs

Fig. 3 demonstrates that clusters from family P appear 3 times of 4, clusters from family Q appear 4 times of 4, and clusters from family R appear 2 times of 4. These examples lead to the notion of another kind of stability. Denote $c(\mathrm{F}) = |\mathrm{F}| = d$ (see (2)). Assume $\beta(\mathrm{F}) = c(\mathrm{F})/r$. This parameter shows, how many times of r runs components of F are included in found families. Finally, assume $\gamma(\mathrm{F}) = \alpha(F) \times \beta(\mathrm{F})$.

The number $V(\mathrm{F}) = 1 - \gamma(\mathrm{F})$ is named the ***volatility of the family*** F. Assume that all the families F (see (2)) are ordered in correspondence with their volatilities increasingly: $\mathrm{F}_1, \mathrm{F}_2, \ldots$, so that $s < t$ implies $\mathrm{F}_s < \mathrm{F}_t$. Assume $C_i = A(\mathrm{F}_i), i = 1, 2, \ldots$ (see (3)). These sets are shown in the middle column in Fig. 4. Finally, assume the number V^* — the maximal feasible volatility – is given.

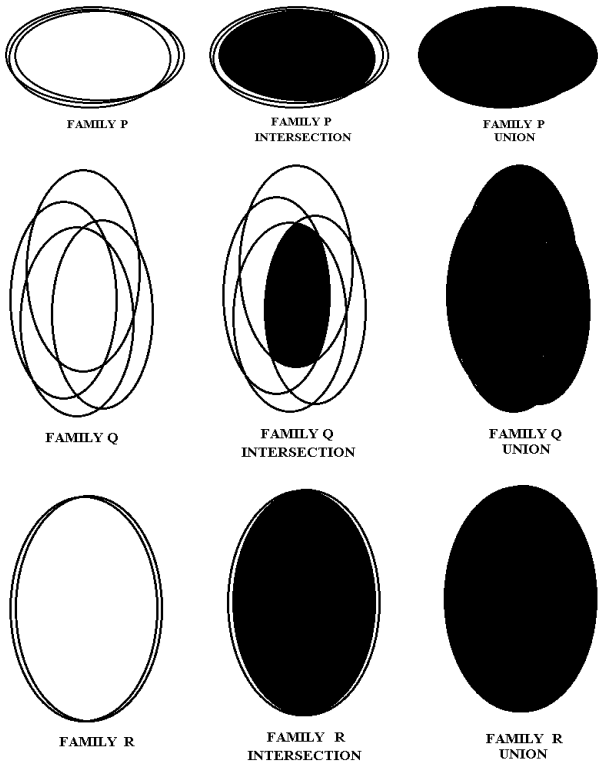

Figure 4. Family intersection and union

The following steps of the **Algorithm of Clusters Selection** define the suggested solution of a given clustering problem.

Algorithm of Clusters Selection

1. Find all the ***α-stable*** families $\mathrm{F}_1, \mathrm{F}_2, \ldots, \mathrm{F}_m$ (see (3)).
2. Select among them all the families F such that $V(\mathrm{F}) \leqslant V^*$ (they are named the ***feasible*** ones).
3. Order feasible families F_i in correspondence with $V(\mathrm{F}_i)$ increasingly.
4. Define sets $C_i = A(\mathrm{F}_i)$ $(i = 1, 2, \ldots, k)$.
5. Assume $D_1 = C_1$, current $i_c = 1$.
6. If sets $D_1, \ldots, D_t$ are found, consider consecutively $i > i_c$ till one of the following two events occur:
   - C_i does not intersect with $D_1, \ldots, D_t$;
   - $i = k + 1$.

   In the 1-st case assume $D_{t+1} = C_i, i_c = i, t = t + 1$ and return to step 6.
7. Consider all the clusters $D_1, \ldots, D_t$ and eliminate every cluster containing other clusters from the list.
8. Stop.

The constructed sets $D_1, \ldots, D_s$ form the output of the external stage 3. Sets $D_1, \ldots, D_s$ are the found clusters. The ***volatility $V(D)$ of cluster D*** is defined as volatility of family F such that $D = A(\mathrm{F})$. The volatility of the whole clustering problem is defined as the weighted sum of all the found clusters:

$$V = \sum_{i=1}^{s} V(D_i)|D_i| / \sum_{i=1}^{s} |D_i|. \tag{4}$$

In order to resume this Section, let us describe the operation of Step 1 – **Construction of α-stable Families**. The algorithm is rather simple. We construct the list of all families F, such that $\alpha(\mathrm{F}) > 0.5$. Assume we have already the current list of such different families $\mathrm{F}_1, \ldots, \mathrm{F}_s$. Assume $\mathrm{F} = \langle \mathrm{F}_{i_1}, \ldots, \mathrm{F}_{i_d} \rangle$ is one of constructed families, presented in form (2). Consider arbitrary set F_i from any set U_i, where $i > i_d$. Check the new family $\mathrm{F}' = \langle \mathrm{F}_{i_1}, \ldots, \mathrm{F}_{i_d}, F \rangle$ for the condition $\alpha(\mathrm{F}') > 0.5$ (it is a simple operation). If this condition holds, F' is added to the list.

The same operations is executed

1. for all the elements of U_i;
2. for all i $(i_d < i \leqslant r)$;
3. for all the families of the current list.

The algorithm stops then no new family cannot be added to the current list. Initially all the separate sets from every $U_i (i = 1, \ldots, r)$ form the current list.

It is worthwhile to remark that the algorithm is fast enough, because for almost all pairs of two sets from different U_i their intersection is empty and therefore all the chains $F_{i_1}, \ldots, F_{i_d}$ are very quickly terminated.

6. Comparisons and Examples

Many difficult model examples as well as comparisons with other known clustering methods were considered in the cited work [8]. However, in all the given there examples volatility is very close to 0. In this publication we focus our attention on some real clustering problems, in which volatility essentially differs from 0. Moreover, its level is one of the most significant and meaningful characteristics of the considered situations. This circumstance underlines the expedience of introduction of this parameter.

1. Stock market analysis. The results are presented in terms of the found groups of stock for USA, Russia and Sweden stock market. In the considered case the initial data consist of pair-wise correlations for 2008–2010 years: 500 USA stocks; 266 Sweden stocks; 151 Russian stocks. It is required to find clusters in these data (or be sure in their absence), basing on the given correlation matrices.

USA market. Volatilities of the found clusters are presented in the following table:

NoNo	1	2	3	4	5	6	7	8
Volatility	0.000	0.125	0.167	0.167	0.286	0.356	0.549	0.583

It is possible to add that all the clusters are formed by companies engaged in the same or close fields. The cluster with the minimal volatility 0 includes only the companies engaged in the same field (gold mining). The increasing of volatility is accompanied (as a trend) by the widening of field of activity of included firms. The obtained clustering results at least do not contradict to common sense.

Russia market. Only 2 clusters are revealed in Russian stock market. Both groups of companies are engaged in electrical power production. In both cases volatility is equal to 0.15. One group consists of 18 companies, the other consists of 5 companies. The correlation between stock, included in the same cluster, is significantly less than in USA market. This circumstance demonstrates the significant difference between these two markets.

Sweden market. Under the same algorithm no clusters in Sweden data are revealed.

2. Deputies Clusters in Duma. In this case the activity of Russian Duma (parliament) was analyzed for period of 5 months, since 01.09.2001 till 01.02.2002. This period seems important, because the significant political event – occurrence of new party "Unified Russia" – happened 01.12.2001. In more detail the situation is described in book [9]. Five families of classifications (corresponding to the considered five months period) are found. For every separate month all the votings (200 – 500) are considered. To i-th deputy ($i = 1, 2, \ldots, 479$) a vector $v_i = (v_1^i, v_2^i, \ldots, v_n^i)$ is related, where n is the number of votings in the months,

$$v_j^i = \begin{cases} 1, & \text{if } i\text{-th deputy voted for } j\text{-th proposition;} \\ -1, & \text{if } i\text{-th deputy voted against } j\text{-th proposition;} \\ 0, & \text{otherwise.} \end{cases}$$

The dissimilarity d_{st} between s-th and t-th deputies is defined as usual Euclidian distance between vectors v_s and v_t. The dissimilarity matrix $D = (d_{st})$ is the initial one for clustering algorithm, described in Section 2. The volatilities of all the clusters are presented in the following table:

Table 1

Volatility in duma clusters

September	0; 0; 0; 0; 0
October	0; 0; 0; 0; 0; 0; 0; 0
November	0; 0; 0.022; 0.200; 0.260; 0.315
December	0; 0; 0; 0.010; 0.012; 0.074; 0.125
January	0; 0; 0; 0.020; 0.035; 0.060; 0.144

The main conclusion is as follows. The volatility was equal 0 in September and October 2001; it significantly increased just before the key event – the creation of "United Russia", and slightly decreased after this event. In the cited book [9] several known indices of Duma did not show signifycant features at this period.

References

1. A Survey of Kernel and Spectral Methods for Clustering / M. Filippone, F. Camastra, F. Masulli, S. Rovetta // Pattern Recognition. — 2008. — Vol. 41, No 1. — Pp. 176–190.
2. *Girvan M., Newman M. E. J.* Community Structure in Social and Biological Networks // Proc. Natl. Acad. Sci. USA. — 2002. — Vol. 99, No 12. — Pp. 7821–7826.
3. *Gordon A. D.* Classification. — London: Chapman & Hall/CRC, 1999.
4. *Luxburg U.* A Tutorial on Spectral Clustering // Statistics and Computing. — 2007. — Vol. 17, No 4. — Pp. 395–416.
5. *Mirkin B.* Mathematical Classification and Clustering. — Dordrecht: Kluwer Academic Publishers, 1996.
6. *Mirkin B.* Clustering for Data Mining: A Data Recovery Approach. — London: Chapman & Hall/CRC, 2005.
7. *Mirkin B.* Core Concepts in Data Analysis: Summarization, Correlation, Visualization. — London: Springer, 2010.
8. *Rubchinsky A.* Divisive-Agglomerative Classification Algorithm Based on the Minimax Modification of Frequency Approach. — Moscow: NRU HSE, 2010.
9. *Алескеров Ф. Т. и др.* Влияние и структурная устойчивость в Российском парламенте (1905–1917 и 1993–2005 гг.). — М.: ФИЗМАТЛИТ, 2007. [Aleskerov F. T. et al. Influence and Structure Stability in Russian Parliament. — Moscow: FIZMATLIT. — 2007. — 312 p.]

УДК 519.254 MSC 68H30

Волатильность в классификации

А. А. Рубчинский

Национальный исследовательский университет «Высшая школа экономики»
Международная научно-учебная лаборатория анализа и выбора решений
ул. Воронцово поле, д. 5а, Москва, Россия
Кафедра прикладной математики и информатики
Международный университет природы, общества и человека «Дубна»
ул. Университетская, д. 19, Дубна, Московская область, Россия, 141980

Целью данной работы является разработка новой трёхуровневой схемы автоматической классификации, основанной на введённом понятии — ***волатильности***, как отдельных кластеров, так и классификации в целом. Волатильность представляет собой точно определяемую и эффективно вычисляемую величину, которая определяет стабильность, точность, надёжность некоторых подмножеств исходного множества вариантов — короче говоря, возможность (или невозможность) их выбора в качестве кластеров. Предложенный алгоритм находит кластеры с заданным максимальным уровнем волатильности, включая и традиционные кластеры, обладающие волатильностью, близкой к нулевой. Кластеры на фондовых рынках США, России и Швеции (за период кризиса 2008–2010 годов) и депутатские кластеры, определяемые голосованиями в 3-й Думе с 01.09.2001 по 31.01.2002 – периода, включающего в себя образование партии «Единая Россия» 01.12.2001, — были построены предложенным алгоритмом. При анализе кластеров, построенных по результатам голосований для каждого месяца в отдельности, оказалось, что волатильность кластеризации в сентябре и октябре равна 0, резко возрастает в ноябре и слегка убывает в декабре и ноябре. Другие методы (типа индексов согласованности между фракциями и др.) не показывают «политической бифуркации» в рассматриваемом периоде. Рассмотрены также разнообразные модельные примеры, для которых результаты классификации хорошо согласуются с геометрической интуицией.

Ключевые слова: кластерный анализ, автоматическая классификация, разрез в графе, волатильность, фондовый рынок, Государственная Дума.