



A variant of gravitational classification

Tomasz Górecki¹, Maciej Łuczak²

¹Faculty of Mathematics and Computer Science, Adam Mickiewicz University,
Umultowska 87, 61-614 Poznań, Poland
e-mail: tomasz.gorecki@amu.edu.pl

²Faculty of Civil Engineering, Environmental and Geodetic Sciences, Koszalin
University of Technology, Śniadeckich 2, 75-453 Koszalin, Poland
e-mail: mluczak@wilsig.tu.koszalin.pl

SUMMARY

In this article there is proposed a new two-parametrical variant of the gravitational classification method. We use the general idea of objects' behavior in a gravity field. Classification depends on a test object's motion in a gravity field of training points. To solve this motion problem, we use a simulation method. This classifier is compared to the 1NN method, because our method tends towards it for some parameter values. Experimental results on different data sets demonstrate an improvement in efficiency and that this approach outperforms the 1NN method by providing a significant reduction in the mean classification error rate.

Key words: machine learning, nearest neighbor method, dynamic classifier, gravitational classification

1. Introduction

In this paper we introduce and study a new classifier. The idea of our classifier has its origin in observations of the behavior of celestial bodies in a gravity field. In physics this is known as the n -body problem, which can be described as the problem of taking an initial set of data that gives the positions (in \mathbb{R}^3), masses and velocities of some set of n bodies, for some particular point in time, and then using that set of data to determine the motions of the n bodies, and to find their positions at other times, in accordance with the laws of classical mechanics. Objects from the training sample correspond to bodies in a gravity field. To classify a test object we place it in the field of training objects and observe its movement. If the test object hits the training object with class G_j we assign it to the class G_j .

Suppose that a training sample has been collected by sampling from a population P consisting of K subpopulations or classes G_1, \dots, G_K . The i th

observation is a pair denoted by (\mathbf{x}_i, y_i) , where \mathbf{x}_i is a p -dimensional feature vector and y_i is the label for recording class membership. The corresponding pair for an unclassified observation is denoted by (\mathbf{x}, y) . In this case \mathbf{x} is observed, but the class label y is unobserved. The object of classification is to construct a classification rule for predicting the membership of an unclassified feature vector $\mathbf{x} \in P$.

The gravitational classification method was first proposed in Wright (1977). In subsequent years many gravitational classification algorithms have been well studied by researchers (Kundu, 1999; Peng et al., 2005; Peng et al., 2009; Ravi and Gowda, 1999; Shi et al., 2003). A framework for classification based on physical fields was proposed in Ruta and Gabrys (2009) and followed in Budka and Gabrys (2011). In the present paper we study a parametrical approach to gravitational classification. In classification methods based on a gravity field the intensity of the field induced by an object is inversely proportional to the square of the distance. We propose to take a different parametrical power function to compute the intensity of the field. The use of such a parametrical kernel function, in a static manner, was proposed in Górecki and Łuczak (2010). In this work we use the kernel dynamically, the field induced by the kernel function causing movement of the test points. We consider the radius of training points as the second parameter, and investigate the influence of the parameters on the classification process, dynamically changing them for each data set. Then we try to simplify the method by finding more or less universal values of the parameters which are best on average for all data sets.

In our paper we first present the main ideas of gravity classifiers (section 2). Details of algorithms are described in section 3. The performances of the methods are compared and the error of classification is considered. Our classifiers seem to tend towards the 1NN method. Salzberg (1997) recommends comparing a new method with the most similar one. Therefore we compare our methods with the 1NN rule. The methods and data sets used are described in section 4. The results of the research are explained with a number of charts, where the differences between the classifiers are shown precisely. Section 5 contains the results (with figures) of our experiments on the described real data sets. We conclude with discussion in section 6.

2. Method

The intensity of the gravity field induced by bodies \mathbf{x}_i with masses m_i ($i = 1, \dots, n$) at point \mathbf{x} is inversely proportional to the square of the

distance. The vector of the intensity is expressed as

$$\mathbf{v}(\mathbf{x}) = G \sum_{i=1}^n \frac{m_i}{\|\mathbf{x}_i - \mathbf{x}\|^2} \frac{\mathbf{x}_i - \mathbf{x}}{\|\mathbf{x}_i - \mathbf{x}\|}, \quad (1)$$

where G is the gravitational constant. To find the trajectories of the bodies, we have to find the solution of the following second-order system of n differential equations (in vector form):

$$\mathbf{x}_i''(t) = G \sum_{\substack{1 \leq j \leq n \\ j \neq i}} m_j \frac{\mathbf{x}_j(t) - \mathbf{x}_i(t)}{\|\mathbf{x}_j(t) - \mathbf{x}_i(t)\|^3} \quad i = 1, \dots, n, \quad (2)$$

where $\mathbf{x}_i(t)$ are the positions of the bodies at time t .

Our idea for a new classification method is as follows. Objects from the training sample correspond to bodies in a gravity field. We assume that the masses of the objects are the same ($m_j = 1$) and each of them is a sphere in R^p with a radius r (the same for all training objects) and with center at \mathbf{x}_i . The training objects are motionless, and they have no influence on each other. We also assume that the intensity of the field induced by the objects does not need to be inversely proportional to the square of distances; it can be any other power function. Thus the vector of intensity is

$$\hat{\mathbf{v}}(\mathbf{x}) = G \sum_{i=1}^n \frac{\mathbf{x}_i - \mathbf{x}}{\|\mathbf{x}_i - \mathbf{x}\|^\alpha}, \quad (3)$$

where $\alpha > 1$ is a parameter.

Then, to classify a test object \mathbf{x} , we put it in the field (it appears at position \mathbf{x}). The object starts moving and at some time hits one of the training objects (i.e. at time t , its distance from some training point \mathbf{x}_k is lower than r). We classify the test object by placing it in the class of the object \mathbf{x}_k .

To find the trajectory of a test object we have to find the solution of the following second-order differential equation (in vector form):

$$\mathbf{x}''(t) = G \sum_{j=1}^n \frac{\mathbf{x}_j - \mathbf{x}(t)}{\|\mathbf{x}_j - \mathbf{x}(t)\|^\alpha}, \quad (4)$$

where $\mathbf{x}_j \in R^p$ are training points, and $\mathbf{x}(t)$ is the position of test object \mathbf{x} at time t .

To solve the problem we use simulation methods. We also introduce some simplifications of the main method. We examine three different models:

1. The first model is the most sophisticated, but it corresponds strictly to the main analytical problem. The time of a test object's motion is divided into many short periods. A test object is placed in the field of training objects with an initial velocity of 0. In each period the object moves with uniformly accelerated linear motion. As the periods tend to 0, the model becomes similar to the main analytical problem.
2. The second model is a simplification of the first one. We assume that the test object moves with constant velocity (some fixed initial velocity). In each period only the direction of motion changes (it corresponds to the direction of field intensity).
3. The third model is the simplest. The test objects move rectilinearly. The direction of a test object is constant for the whole time of movement and equals the direction of field intensity at the initial point of the object.

Figure 1 shows the classification areas of the above three methods for the same training data set. For the first model there are some more or less chaotic (random) disturbances of the classification regions. In real motion in a gravity field, a test object often covers a long distance before hitting one of the training objects. The destination object can be quite far from the starting position of the test point. This is similar to the behavior of our first model. It is possible that, in finite time, a test object fails to collide with any training object. It may be thrown out of the classification area or go into a more or less stable orbit.

The second model gives smoother classification borders. It very rarely happens that a test object is not classified. Practically all test points reach one of the training objects.

In the third model, it very often happens that the trajectory line does not intersect any sphere of the training objects. There are large unclassified areas.

All of the models have the same properties. As the radius r increases, the models become similar to the nearest neighbor method. However, for large values of radius r , a test object in its start position can be closer than r to many training points. Then we have to resolve this conflict, and this

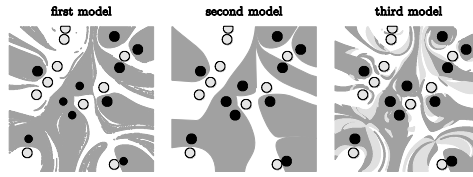


Figure 1. Comparison of classification areas. Black (class ‘1’) and white (class ‘2’) dots represent the observations from the training set, while the color of the area specifies areas of classification (light gray – unclassified region, gray – class ‘1’, white – class ‘2’)

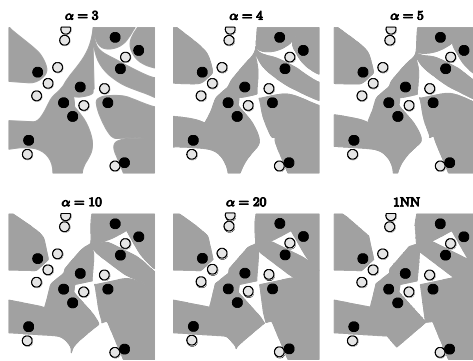


Figure 2. Classification areas for different values of the parameter α . Black (class ‘1’) and white (class ‘2’) dots represent the observations from the training set, while the color of shading specifies areas of classification (light gray – unclassified region, gray – class ‘1’, white – class ‘2’)

has a significant influence on the classification error rate. Because of this, we should prefer to take smaller values of radius r .

As the value of the parameter α increases, the influence of distant training points decreases. Thus also in this case the model tends towards the 1NN method. Figure 2 shows classification areas for different values of the parameter α .

Our research established that the second model is the best for classification problems. The classification error rate of the other models is similar and is much higher than in the second model. Thus, for the remainder of this paper, we consider only the second method.

3. Algorithm

In our method (the second model) a test object has constant velocity norm. For each test object, the total time of its motion is divided into many short periods. In each period the test object covers a distance of $1/10$ of the radius r (constant for the training objects). In this way, the gravity constant G does not influence the velocity norm and can be fixed at $G = 1$. In the total time the test object covers the distance of twice the maximum distance between all training points (if it does not hit any training object). If it does not reach any training object in the total time, then its class is chosen randomly. This has a negligible influence on the classification results, because such points are very few (Figure 1, second model). Also if the start position of a test object is closer than r to more than one training object, we classify it randomly. The classification of one test point can be presented in pseudocode:

Algorithm 1 Classification of one test point \mathbf{x} .

Require:

(\mathbf{x}_i, y_i) — positions and classes of training points

r — constant radius of training points

\mathbf{x} — position of the test point

t — constant time period

$\mathbf{v}(\mathbf{x})$ — velocity vector with constant norm and varied

Ensure: y — class of the test point \mathbf{x}

1: **while** $\forall_i \|\mathbf{x} - \mathbf{x}_i\| \geq r$ **do**

2: $\mathbf{x} := \mathbf{x} + t \mathbf{v}(\mathbf{x})$

3: **end while**

4: $y := y_i$, where $\|\mathbf{x} - \mathbf{x}_i\| < r$

Thus there are two parameters in our model: the radius $r > 0$ and the power function exponent $\alpha > 1$. We introduce a new parameter β , where $0 \leq \beta \leq 1$. To compute the radius r from β we calculate all nonzero distances between training points. Let us denote the number of these as m . Then we take the $\lfloor \beta \cdot m \rfloor$ lowest of them (or the lowest one if $\lfloor \beta \cdot m \rfloor = 0$) and calculate the arithmetic mean. Half of the mean is the radius r . In this procedure the lowest possible radius equals half the minimal nonzero distance between training points. The radius is usually a little greater than the shortest possible one.

We finally obtain a classification method depending on two parameters: $\alpha > 1$ and $0 \leq \beta \leq 1$. The greater is the value of α , the less influence on

classification comes from distant training points. The greater is the value of β , the greater is the radius r .

4. Computational experiments

4.1. Data sets

We conducted experiments on some real data sets. The learning capability of our algorithm was measured by the accuracy of a classifier, learned from training samples, in classifying unseen features. We compared our algorithm with the widely used nearest neighbor method. A total of 12 data sets were selected. These data sets have been extensively adopted for benchmark study. A summary of some of their properties is given in Table 1.

Table 1. A summary of the data sets used in the experiments.

Data set	Dimension of feature space	Number of classes	Number of instances in classes	Number of all instances
balance	4	3	49,288,288	625
beetles	2	3	21, 31, 22	74
diabetes	8	2	500,268	768
hayes	5	3	51, 51, 30	132
heart_h	10	5	31, 23, 24, 15, 168	261
hepatitis	16	2	26, 111	137
iris	4	3	50,50,50	150
lungcancer	55	3	9, 13, 10	32
parkinsons	22	2	147, 48	195
thyroid	5	3	150,35,30	215
vertebral	6	3	60, 150, 100	310
wine	13	3	59,71,48	178

The data set *beetles* comes from Seber (1984). Other data sets originate from Frank and Asuncion (2010). These data sets are representative for a wide range of biometry classification problems.

4.2. Experimental setup

We employed a bootstrap method in estimating the classification accuracy. We used 100 repetitions for each data set. The 100 values of classification accuracy are averaged to get the final estimate. Clearly this number of repetitions is somewhat arbitrary and may not be optimal. However, since the bootstrap samples were fixed for all methods and data sets, the comparison should be fair. We carried out experiments for β from 0 in steps of 0.005

up to 0.05 and for $\alpha \in \{3, 4, \dots, 15, 20, 50\}$, but we show in the tables only values for $\alpha \in \{3, 4, \dots, 10, 20, 50\}$, due to table size limitations.

5. Results

We experimented on a range of values for β . For a fixed data set, the same value of parameter β can give a lower error rate for different values of parameter α . However, for each data set the best value of β is different. For some data sets the error rate curve is constant, while for others it increases or decreases. For example, the results for fixed $\alpha = 5$ are shown in Table 2.

Table 2. Sensitivity of bootstrap classification error rates with respect to parameter β ($\alpha = 5$)

Data set	0	0.005	0.01	0.015	0.02	0.025	0.03	0.035	0.04	0.045	0.05
balance	24.32	24.32	24.32	24.37	24.32	24.34	24.39	24.38	24.41	24.46	24.54
beetles	6.61	6.61	6.61	6.61	6.61	6.61	6.61	6.61	6.61	6.61	6.61
diabetes	33.10	33.03	33.17	33.26	33.51	33.56	33.42	33.50	33.49	33.69	33.70
hayes	59.28	59.24	59.16	59.08	59.12	59.08	59.06	59.10	59.13	59.21	59.37
heart_h	54.97	54.96	54.99	55.07	54.99	54.94	54.90	54.78	54.69	54.55	54.62
hepatitis	27.63	27.61	27.56	27.48	27.32	27.34	27.54	27.51	27.48	27.25	27.09
iris	4.19	4.19	4.19	4.19	4.19	4.19	4.19	4.19	4.19	4.21	4.21
lungcancer	64.07	64.07	64.07	64.17	64.17	64.32	64.32	64.32	64.24	64.24	64.14
parkinsons	15.42	15.42	15.46	15.44	15.47	15.38	15.43	15.43	15.59	15.75	15.90
thyroid	7.18	7.17	7.17	7.18	7.18	7.18	7.18	7.17	7.17	7.17	7.17
vertebral	20.39	20.39	20.39	20.47	20.48	20.40	20.30	20.26	20.09	19.94	19.93
wine	27.16	27.19	27.22	27.19	27.22	27.40	27.58	27.91	28.15	27.89	27.71

It may be of interest to have one total value of the parameter β for all data sets. To find such β we normalize the error rates for each data set and take means for each β . Thus we have a function of average normalized errors for all data sets depending on the parameter β (Figure 3). It is seen that the most universal value is $\beta = 0.01$. All further investigations are performed only for this value of β .

We recognize that this selection of parameters is not a universally best choice for all practical problems. In our experiments it is selected mainly to demonstrate the potential of gravity classifiers. For practitioners, however, the parameters must be fine-tuned to the problem domain. Nonetheless it seems that the value $\beta = 0.01$ is a fairly good first choice for a wide range of data sets.

Next we have to choose the parameter α . The estimated mean values of classification error rate are listed in Table 3. A lower error rate means a higher classification accuracy. The winner in each data set is given in bold. We can see that for almost all data sets there exists α such that our

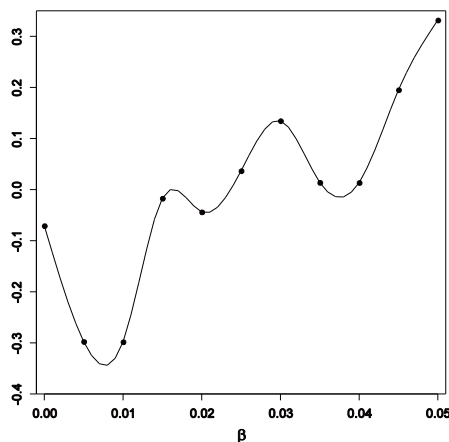


Figure 3. Normalized mean bootstrap error rates for different values of β and $\alpha = 5$

method is better than the nearest neighbor method. The only exceptions are the data sets *balance* and *wine*. However we can see that the error decreases as the value of α increases. Because the model tends towards the 1NN method, we can hope that the error also tends to the 1NN error rate and attains it for some (maybe large) value of the parameter α .

Table 3. Bootstrap error rates with respect to the parameter α ($\beta = 0.01$)

Data set	1NN	3	4	5	6	7	8	9	10	15	20	50
balance	21.62	47.05	26.83	24.32	22.98	22.41	22.13	22.08	22.04	22.11	22.10	22.13
beetles	6.82	7.51	6.54	6.61	6.61	6.64	6.53	6.50	6.57	6.68	6.68	6.75
diabetes	32.85	36.71	35.05	33.17	32.87	32.88	32.86	32.86	32.84	32.89	32.99	33.01
hayes	59.05	60.07	59.67	59.16	59.20	58.96	58.87	59.09	58.90	58.97	58.99	58.89
heart_h	56.10	55.71	53.67	54.99	55.78	55.84	55.99	56.06	56.02	56.15	56.24	56.14
hepatitis	27.11	27.25	27.91	27.56	26.84	26.49	27.02	26.92	26.96	26.85	26.93	27.09
iris	4.29	5.10	4.26	4.19	4.30	4.34	4.25	4.22	4.25	4.34	4.38	4.33
lungcancer	56.33	68.07	66.62	64.07	60.72	58.47	58.43	57.16	56.54	55.61	56.40	55.40
parkinsons	16.38	16.75	15.26	15.46	15.65	16.13	16.28	16.31	16.36	16.35	16.34	16.35
thyroid	6.60	14.71	9.17	7.17	6.78	6.60	6.38	6.40	6.38	6.44	6.47	6.54
vertebral	18.48	34.67	23.52	20.39	19.30	18.91	18.97	18.75	18.68	18.49	18.44	18.48
wine	26.29	28.07	27.41	27.22	27.07	27.09	26.87	26.88	26.76	26.71	26.60	26.44

We may use the mean ratio of error rates across data sets as a measure of relative performance (Bauer and Kohavi, 1999). However, Webb (2000) reported that this measure could increase the difficulty of error comparison in different data sets. Therefore he adopted the geometric mean ratio rather than arithmetic mean ratio:

$$\text{err}_g = \sqrt[n]{\prod_{i=1}^n \frac{a_i}{b_i}}, \quad (5)$$

where n is the total number of data sets and a_i, b_i are classification error rates. The geometric mean ratio of a_i/b_i enjoys the desirable property that if it is greater than one then the geometric mean ratio of b_i/a_i will be less than one. We therefore calculate the geometric mean ratios of error rates for our method with respect to the 1NN method, the results are shown in Table 4. From the table we see that, on average (for the best value of parameter α), our algorithm performs better than the 1NN method (Figure 4). This is in accordance with the above-mentioned observations.

As Salzberg, 1997 (1997) recommends, for the statistical comparison of methods we used the exact binomial test. Let n be the number of data sets, s be the number of times that our method is better than the 1NN rule, and f be the number of times that our method fails. Suppose that $s > f$, so our method is better than 1NN. We calculate the probability that our method “wins” over 1NN at least as many times as observed in the experiment. This will be our p -value. It can be computed as

$$n!(0.5)^n \sum_{j=s}^n \frac{1}{j!(n-j)!}. \quad (6)$$

In our case we have $n = 12, s = 10, f = 2$. Hence the p -value is equal to 0.01929. Thus it is highly unlikely that the rules have the same accuracy. We can reject the null hypothesis with high confidence.

In Figure 5 a few curves of normalized error rates are shown. For each data set the best value of α is rather different, but we can choose one universal value of α for all data sets. We can see that for $\alpha = 9, \dots, 15$ our method is better on average than the 1NN method (Table 4).

Table 4. Geometric mean ratios of bootstrap error rates for our method with respect to the 1NN method

α	3	4	5	6	7	8	9	10	15	20	50	Best
Error	1.2714	1.0811	1.0303	1.0125	1.0056	1.0006	0.9972	0.9967	0.9985	1.0008	1.0002	0.9801

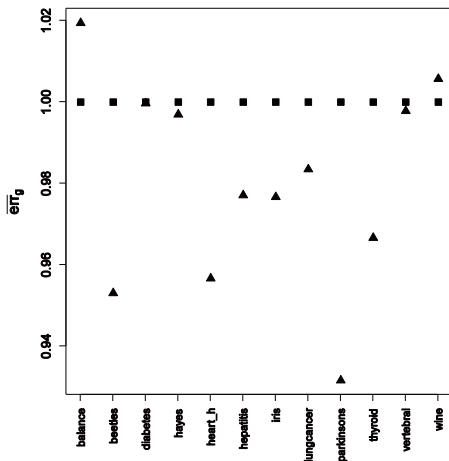


Figure 4. Ratios of the classification error rates (■ – 1NN, ▲ – our best method)

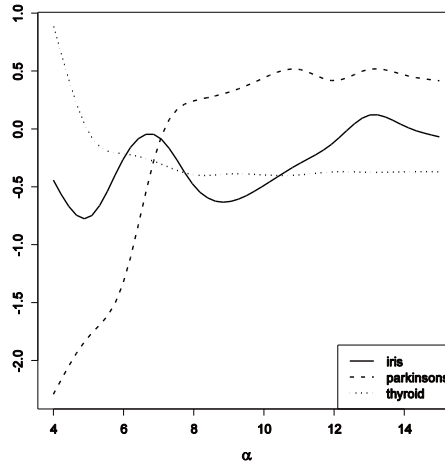


Figure 5. Normalized bootstrap error rates for different data sets

6. Conclusions

In this paper we have introduced and studied a new classification method having its origin in the behavior of bodies in a gravity field. Our research shows that there are a number of data sets for which the method is better than the nearest neighbor method. To achieve a lower error rate we have to choose an appropriate value of the parameter α for each data set. However for some fixed α the method also has a lower error rate on average than the 1NN method.

Due to the high nonlinearity and complicated dynamics involved in gravity classifiers, the method does not easily lead to a rigorous theoretical analysis. However the experiments that we have conducted justify the potential and usefulness of our method.

Of course, the classification performance of the new algorithm needs to be further evaluated, considering additional real and artificial data. Reducing the number of training samples by editing (to remove noise) and condensing (to obtain a smaller subset) can improve the performance of the method. This may be a direction of future research.

REFERENCES

- Bauer E., Kohavi R. (1999): An experimental comparison of voting classification algorithms: bagging, boosting, and variants. *Machine Learning* 36(1-2): 105–139.
- Budka M., Gabrys B. (2011): Electrostatic field framework for supervised and semi-supervised learning from incomplete data. *Natural Computing* 10(2): 921–945.
- Chen C.-Y., Hwanga S.-C., Oyanga Y.-J. (2005): A statistics-based approach to control the quality of subclusters in incremental gravitational clustering. *Pattern Recognition* 38(12): 2256–2269.
- Frank A., Asuncion A. (2010): UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml> Irvine, CA: University of California, School of Information and Computer Science.
- Górecki T., Łuczak M. (2010): Some methods of replacing the nearest neighbor method. *Communications in Statistics-Simulation and Computation* 39(2): 262–276.
- Kundu S. (1999): Gravitational clustering—a new approach based on the spatial distribution of the points. *Pattern Recognition* 32(7): 1149–1160.
- Peng L.Z., Chen Y.H., Yang B., Chen Z.X. (2005): A novel classification method based on data gravitation. In *Proceedings of International Conference on Neural Networks and Brain (ICNN&B '05)*, Beijing, China 3: 667–672.
- Peng L., Yang B., Chen Y., Abraham A. (2009): Data gravitation based classification. *Information Sciences* 179(6): 809–819.
- Ravi T.V., Gowda C.K. (1999): Clustering of symbolic objects using gravitational approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 29(6): 888–894.
- Ruta D., Gabrys B. (2009): A framework for machine learning based on dynamic physical fields. *Natural Computing* 8(2): 219–237.
- Salzberg S.L. (1997): On comparing classifiers: pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery* 1(3): 317–328.
- Seber G.A.F. (1984): *Multivariate Observations*. New York. Wiley.
- Shi Y., Song Y., Zhang A. (2003): A shrinking-based approach for multi-dimensional data analysis. In *Proceedings of the 29th international conference on Very large data bases (VLDB '03)*, Berlin, Germany. 29: 440–451.
- Webb G.I. (2000): Multiboosting: a technique for combining boosting and wagging. *Machine Learning* 40(2): 159–196.
- Wright W.E. (1977): Gravitational clustering, *Pattern Recognition* 9(3): 151–166.