Formation Recognition by Clustering-Based Method in Virtual Soccer

Aleksandr Khodos[a], Michail Panteleyev[a]

[a] Saint Petersburg Electrotechnical University "LETI", ul. Professora Popova 5, St. Petersburg, 197376, Russian Federation

Abstract

In multi-agent systems, the task of adapting to the world is extremely difficult due to interactions of agents and a constantly changing world. There is an important problem to recognize formations in multi-agent systems with team opposition, because formation can describe the interactions of team agents. Knowing an opposing team formation, an agent's behavior can be adapted to performing effective and timely actions. The paper provides a brief overview of existing approaches (home area method, classification using neural networks and machine learning and others), their main advantages and disadvantages. Also, an alternative clustering-based approach was proposed to solve the problem. The problem is solved in the virtual soccer environment RoboCup Simulation 2D. It is the most famous platform for testing models of team opposing nowadays. Features of the environment are limited perception, dynamic world changing and noisiness. Recognition algorithm was designed, implemented and tested.

Keywords

Formation recognition, Intelligent agents, Multi-agent systems, RoboCup, Virtual Soccer

1. Introduction

RoboCup is an international project to promote researches in Artificial Intelligence and Robotics areas [1]. The project includes several competitive leagues, in which the different types of real or simulated robots play soccer following a set of defined rules. In RoboCup Simulation 2D two teams (multi-agent systems), consisting of eleven autonomous players (agents), play soccer on virtual stadium, presented by central server.

In multi-agent systems, the task of adapting to the world and an opposing team is extremely difficult due to interactions of agents and a constantly changing world during a match. One of the methods to describe interactions of agents is to recognize their tactical formation.

Team formations (or team tactical scheme) are usually defined as sets of tactical lines and flanks. A set of tactical lines n1-n2-...-nk means that a formation consists of k lines, and the i-th line consists of ni players. Similarly, a set of flanks.

Knowing an opposing team formation, an agent's behavior can be adapted to performing effective and timely actions. Also, knowing allied team formation, an agent can determine his current position in the scheme (which may not fit his role).

Proceedings of the 12th Majorov International Conference on Software Engineering and Computer Systems, December 10–11, 2020, Online & Saint Petersburg, Russia

khodozzz@gmail.com (A. Khodos); mpanteleyev@gmail.com (M. Panteleyev)
0000-0002-2349-4600 (A. Khodos); 0000-0002-6077-760X (M. Panteleyev)

Features of the environment during formation recognition are limited perception (agent sees only part of the field at a certain point in time), dynamic world changing, noisiness (all the values received from the server have some errors).

The problem is solved under the project to create the team of intelligent agents. Agents are built with the concept of iterative planning. [2, 3]

2. Existing Solutions

There are many proposed solutions for formation recognition in MAS using various methods. These methods differ in speed, accuracy, the amount of information about a dataset for recognition, the ability to recognize in parts, etc. The main methods used to formation recognitions are the following.

2.1. Home Areas Method

In [4] an approach, based on the concept of a home area – area, in which the agent should generally be. It is assumed that after identifying home areas, the agent will be able to make a conclusion about the player's role in a team. Home areas are determined separately for each player, that allow to recognize in parts and in parallel. However, due to dynamic changes in the world, the player movements can create such range, which extends home area. That makes the task of determining the role of the players too difficult. Also, the team can change their tactical scheme, which will make home areas irrelevant for a while.

2.2. Classification Method

The most popular method to formation recognition in virtual soccer is classification method. Classification is the arrangement of objects (the current set of players positions) in pre-known classes (schemes). Therefore, a large number of training examples are needed for each of possible schemes, and every part of the scheme. For example, if the problem is to recognize only attacking lines (when defense lines are not in player's view). Soccer team formation classification using neural networks is described in [5, 6], other methods of machine learning are described in [7, 8]. The choice of the classification method is due to its resistance to input data noises and adaptability to dynamic changes in the world.

2.3. Other Methods

Other methods have a number of limitations; therefore, they are not widely used. These methods include methods based on statistics [9], which build a team model by characterizing the behavior of each player using data. This data can be the frequency of any action, areas in which a player generally in, etc., calculated on the statistics collected during the observed games. One of limitations is the need to observe a player for a long time in order to characterize him

3. The Proposed Approach

The proposed approach is based on searching areas of the most concentration of points in one-dimensional space, as it shown in Figure 1.

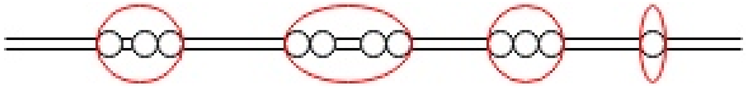


Figure 1: Areas of the most concentration of points in one-dimensional space

Thus, the task of formation recognition in virtual soccer reduced to two: tactical lines and flanks recognition. Transition to two tasks illustrated in Figure 2.

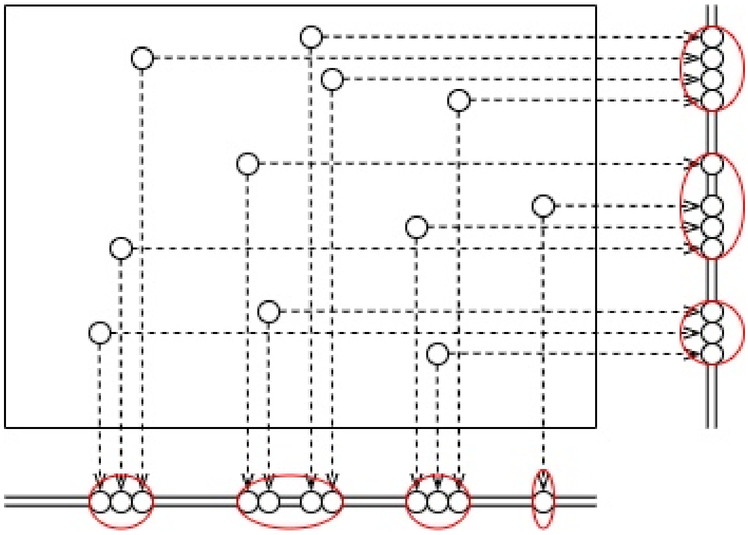


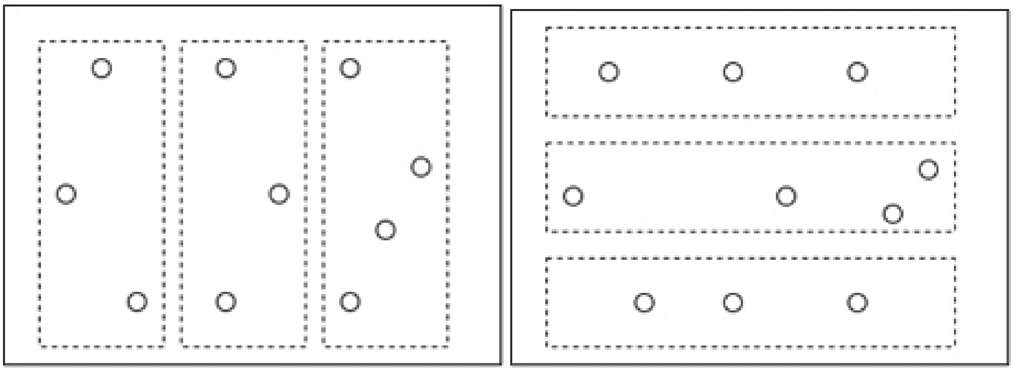
Figure 2: Transition to tactical lines and flanks recognitions

In a virtual soccer environment players position points appear as a result of visual perception. To solve the problem using this approach is not necessary to have information about the position of all the players. Thus it is possible to recognize a part of formation.

Search of areas of the most concentration of points can be solved by clustering – grouping a set of objects in such way that objects of one group (cluster) have more common characteristics than objects from different groups [10]. Unlike classification, which arranges each object to one of pre-known classes, clustering divides objects into new groups.

3.1. The Choice of Clustering Method

The most common clustering methods, such as k-means, k-medians, etc, have a limitation – the number of clusters have to be known in advance. For the most of formation recognition problems (including soccer), we cannot know the number of clusters in advance, thus it is impossible to use these methods.



(a) (b)

Figure 3: Results of lines (a) and flanks (b) recognition

In turn, the FOREL clustering method [11] does not require calculating the number of clusters. The principle of this method is to find the area with a fixed cluster radius, which covers as many points as possible. It is achieved by shifting the area towards the centroid of the local concentration area. After the area is stabilized, all objects inside it will be added to the new cluster and removed from the set. The process is repeated until the entire set is clustered.

The method really does not require calculation the numbers of clusters. Instead, it is necessary to calculate cluster radius R in advance.

3.2. Determination of Cluster Radius

Cluster radius is a value that determines whether an object in the cluster, according to the following principle: if the distance from the cluster centroid to the point is less than R, then this point belongs to the cluster. If R is too small or too large, then recognition will not make sense.

3.2.1. Dependence on Distance Between Furthest Players

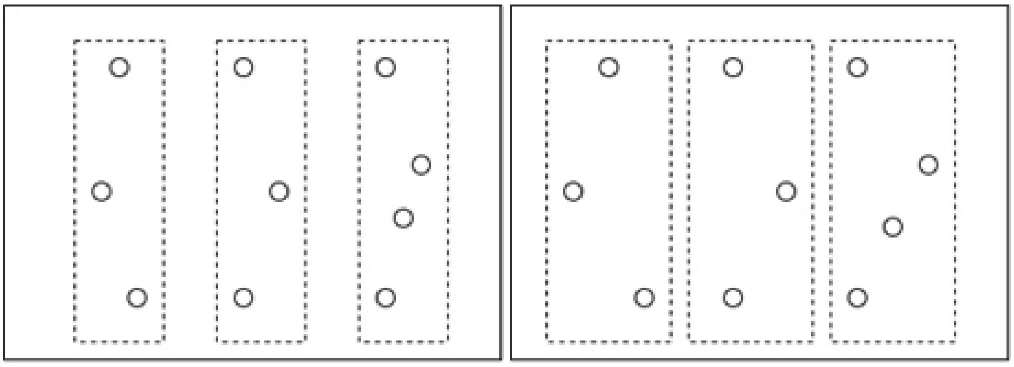
To obtain this dependence, consider an example of tactical lines and flanks recognition in Figure 3.

Obviously, x coordinate of the players' positions is key, y coordinate does not matter to recognize tactical lines. Also, y coordinate is key, x coordinate does not matter to recognize flanks.

A function of the main component c depending on the task, which returns the desired point coordinate x or y of a player position, as follows:

$$c(x,y) = \begin{cases} x\text{, if the task is to recognize tatcital lines} \\ y\text{, otherwise} \end{cases} \tag{1}$$

An example of tactical lines recognition from two different sets of the positions of the players showed in Figure 4. Here are two identical schemes “4-3-3” – a ratio between players and lines are the same, but tactical lines in Figure 4b are wider.



(a) (b)

Figure 4: Formation 4-3-3 variations

The width of tactical lines depends on the distance between the furthest players along x coordinates. Similar, the width of flanks depends on the distance between furthest players along y coordinates. The distance between the furthest players r can be calculated as follows:

$$r = \max_{i,j \in \text{players}} \{| \ c(xi, yi) - c(xj, yj) \ |\} \tag{2}$$

3.2.2. Dependence on Number of Players

Recognition parts of the formations was considered (for example, only players of the team, who are in opponent's half of the field). Expected lines number of partial recognizing is defined as K. Knowing that the number of expected lines for ten players is three, K for n players can be obtained as follows:

$$K = \frac{3n}{10} \tag{3}$$

3.2.3. Final Formula of Cluster Radius

If the players are evenly spaced, then clusters will cover all space between the furthest players. Most of tactical schemes consist of three lines (defense, midfield, attack) and three flanks (left flank, center, right flank). Expected lines number is defined as k. Therefore, to cover the entire space with three clusters (it is not guaranteed that there will be exactly three clusters, the algorithm will correctly determine another number of pronounced clusters) cluster radius R should be:

$$R = \frac{r}{2k} = \frac{\max_{i,j \in \text{ players}} \{|c(xi, yi) - c(xj, yj)|\}}{6} \tag{4}$$

To partial recognition, cluster radius R should be:

$$R = \frac{r}{2K} = \frac{5 \cdot \max_{i,j \in \text{ players}} \{|c(xi, yi) - c(xj, yj)|\}}{3n} \tag{5}$$

3.3. Formation Recognition Algorithm

As stated above, if distance from the cluster centroid to the point is less than R, then this point belongs to the cluster. That idea is described in algorithm below:

Algorithm 1: Cluster forming

```
formCluster (O, P, R)
    inputs : set of object O; point P; cluster radius R
    output: cluster C
    foreach o ∈ O do
        if distance(objoect, P) < R then
            C ← C ∪ o
    return C
```

The pseudocode of FOREL algorithm is demonstrated below:

Algorithm 2: FOREL clustering

```
FOREL (O, R)
    inputs : set of object O; cluster radius R
    output: set of clusters Cs
    while O ≠Ø do
        P ←randomObjectFrom(O)
        C ←formCluster(O, P, R)
        T ←centroid(C)
        while T ≠P do
            P ←T
            C ←formCluster(O, P, R)
            T ←centroid(C)
        Cs ← Cs ∪ {C}
        O ←O / C
    return Cs
```

Cluster radius is computed by following algorithm:

Algorithm 3: Calculating cluster radius

```
clusterRadius (O)
    input  : set of objects O
    output : cluster C
    r = 0
    foreach p1 ∈ O do
        foreach p2 ∈ O do
            if distance(p1, p2) > R then
                r ← distance(p1, p2)
    return 5 * R / (3 * |O|)
```

New positions of each player computed by following algorithm:

Algorithm 4: Update position by main component

```
updatePosition (O, M)
    inputs : set of object O; main component M
    output : updated objects O
    foreach p ∈ O do
        if M = 'x' then
            p.x ← 0
        else
            p.y ← 0
    return O
```

Centroid of points in one-dimensional space is arithmetical mean of this points. Thus, in view of the above, formation recognition algorithm is demonstrated below:

Algorithm 5: Formation recognition

```
formationRecognition (O, R)
    inputs : set of object O; cluster radius R
    output: set of clusters Cs
    foreach p ∈O do
        updatePosition(p)
    R = clusterRadius(O) while O ≠∅ do
        P ←randomObjectFrom(O)
        C ←formCluster(O, P, R)
        T ←arithmeticalMean(C)
        while T ≠P do
            P ←T
            C ←formCluster(O, P, R)
            T ←arithmeticalMean(C)
        Cs ← Cs ∪ {C}
        O ←O / C
    return Cs
```

4. The Results of Experiments

The main criteria for c of the obtained algorithm are ability to recognize any formation or a part of formation, speed and noise resistance.

4.1. Recognitions of Different Team Formations

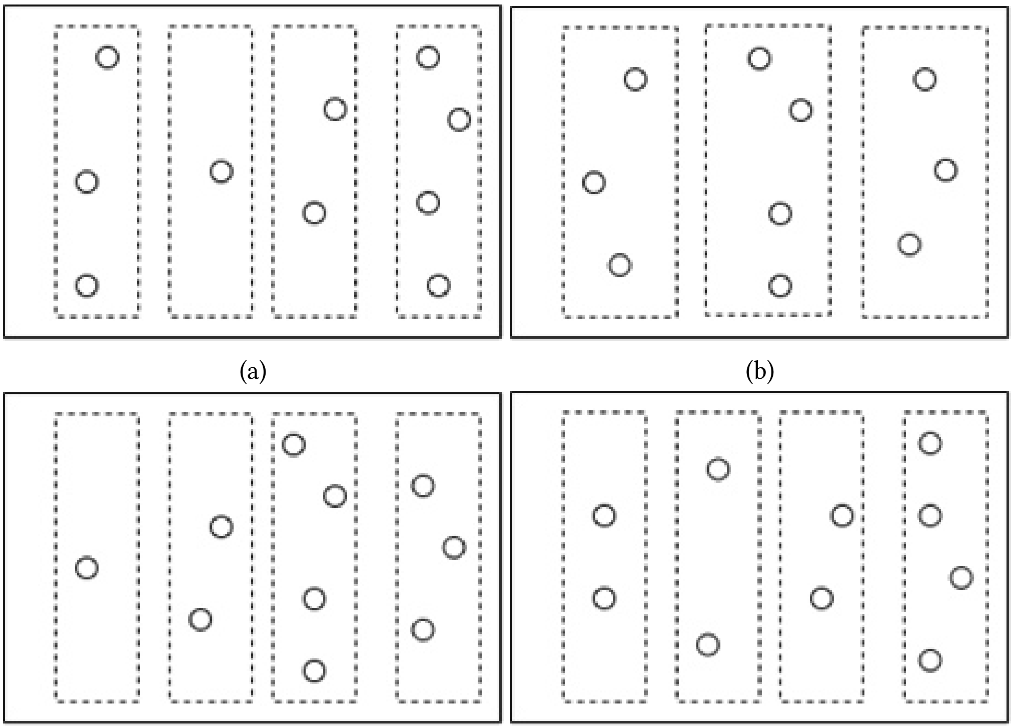
Recognitions of different team formations were carried out, some schemes are presented in Figure 5, results of their recognition are in Table 1. It was concluded that the algorithm works correctly on all the defined formations.

Table 1
Results of formations recognition

Formation	Resut of recognition
a	4-2-1-3
b	3-4-3
c	3-4-2-1
d	4-2-2-2

4.2. Recognitions of Parts of Formation

For the experiment, the formation 4-2-1-3 was divided into all possible parts of the scheme, which are shown in Figure 6, results of their recognition are in Table 2.



(a) (b) (c) (d)

Figure 5: Different team formations

Table 2
Results of parts recognition of the formation 4-2-1-3

Part of formation	Resut of recognition
a	4-2
b	4-2-1
c	1-3
d	2-1-3

The results of this experiments show that the algorithm correctly recognizes the parts of formations.

Thus, in some situations, it is possible to save some time during recognition by not collecting actual information about positions of all the players. An example of such situation: the agent is moving quickly from defense to attack, he should understand positions of those defenders, that can potentially cut off the attack. Therefore, it is possible to recognize only a part of the opponents' scheme in front of the player.

Another conclusion is that recognition can be carried out in parts, and then can be combine the results. In this case, there will be no time loss. It is also possible to recognize the formation by two or more players, who can combine results after recognition.

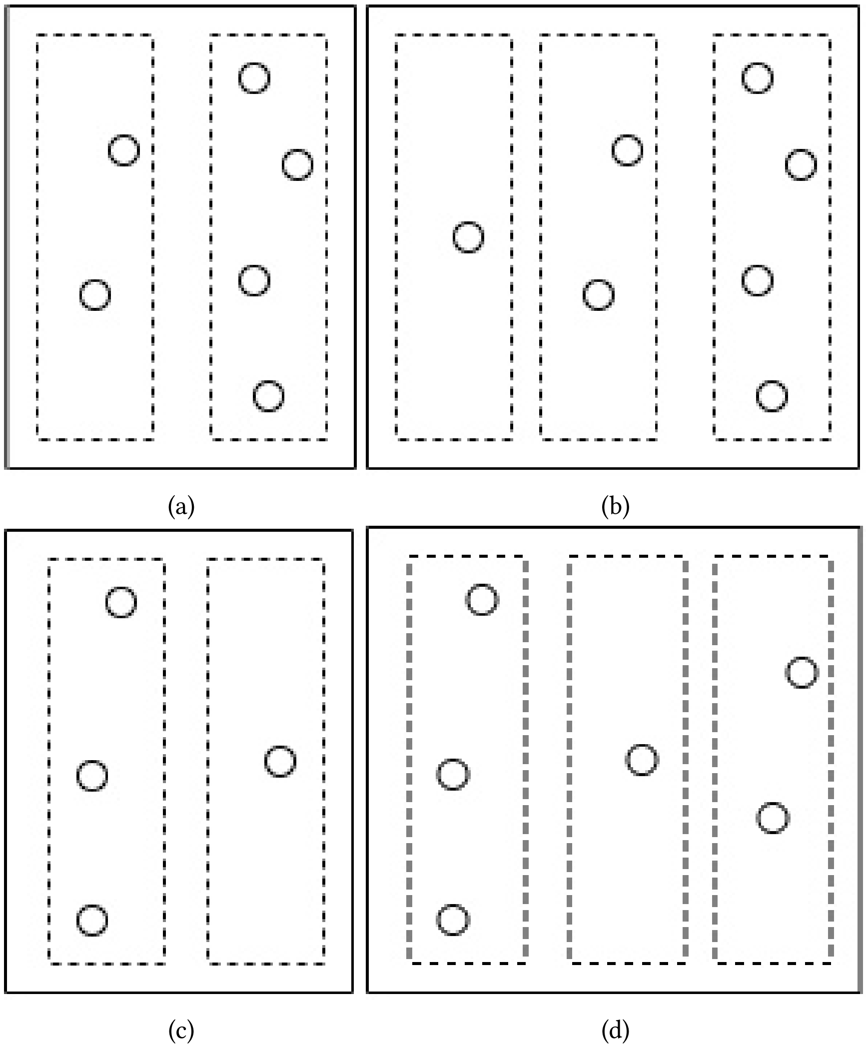


Figure 6: Parts of formation 4-2-1-3

4.3. Algorithm Noise Robustness

It is known from the paper [12], that standard deviation of a player position after calculating Cartesian coordinates, which are used in algorithm, takes value in range from 0.04 to 0.54, depending on the method. Formation recognition accuracy was measured depending on the standard deviation of players positions.

Thus, accuracy is 1 when standard deviation is less than 0.8. Thus, it is concluded that algorithm is robustness to any noise, which added to value by central server in environment RoboCup Simulation 2D.

Table 3
Formation recognition accuracy at different level of standard deviation

Standard deviation (m)	Accuracy
0.0	1
0.1	1
0.2	1
0.3	1
0.4	1
0.5	1
0.6	1
0.7	1
0.8	0.998
0.9	0.989
1.0	0.942
1.1	0.880
1.2	0.771
1.3	0.647
1.4	0.514
1.5	0.375

5. Conclusion

The paper investigates the problem of team formation recognition. Clustering-based method was proposed to solve this problem. Recognition algorithm was designed, implemented and tested.

Unlike other popular approaches described in literature, the proposed approach allows to recognize parts of the scheme. That can be useful in designing intelligence agents. Also, it was proven that the algorithm is noise robustness.

The algorithm is actively used in agent designing to participation in RoboCup Simualtion 2D competitions. The problem of formation recognition by two or more players is still open.

References

[1] RoboCup Federation official website, https://www.robocup.org/, last accessed 2020/11/15

[2] Panteleev M.G., The concept of designing intelligent real-time agent on the basis of advanced iterative planning models [In Russian]. Proc. of the 13 National Conference on Artificial Intelligence with International Participation KII-2012. - Belgorod: BGTU. – vol. 3. - pp. 25-33. (2012)

[3] Panteleev M.G., A formal model of proactive iterative action planning for real-time intelligent agents [In Russian]. Proc. of the 14 National Conference on Artificial Intelligence with International Participation KII-2014. - Kazan: RIC School. – vol. 1 - pp. 323-334. (2014)

[4] 4) Riley, P., Veloso, M., Kaminka G.: An empirical study of coaching. In Asama, H., Arai, T., Fukuda, T., Hasegawa, T. (eds.), Distributed Autonomous Robotic Systems 5, pp. 215–224. Springer-Verlag, (2002)

[5] Nakashima, T., Uenishi, T., Narimoto, Y.: Off-line learning of soccer formations from game logs. In: World Automation Congress (WAC), pp. 1-6, (2010)
[6] Ayanegui-Santiago, H.: Recognizing Team Formations in Multi-agent Systems: Applications in Robotic Soccer. Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, pp. 163–173, (2009)
[7] Faria, B.M., Reis, L.P., Lau, N., Castillo, G.: Machine Learning Algorithms applied to the Classification of Robotic Soccer Formations and Opponent Team. Proceedings of the 2010 IEEE Conference on Cybernetics and Intelligent Systems (CIS) and Robotics, Automation and Mechatronics (RAM), Singapore, pp. 344-349 (2010)
[8] Almeida, R., Reis, L.: Analysis and Forecast of Team Formation in the Simulated Robotic Soccer Domain. In L. Seabra Lopes et al. (Eds.): EPIA 2009, LNAI 5816, pp. 239–250, 2009. Springer-Verlag Berlin Heidelberg (2009)
[9] Kuhlmann, G., Stone, P., Lallinger, J.: The champion UT Austin Villa 2003 simulator online coach team. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020. Springer, Heidelberg (2004)
[10] Zaki M.J., Meira W., Data mining and analysis: fundamental concepts and algorithms. Cambridge University Press, New York (2014)
[11] Zagoruiko, N.G.: Applied Methods of Data and Knowledge Analysis. Institute of Mathematics SD RAS, Novosibirsk (1999)
[12] de Boer R., Kok J., Groen F.: UvA Trilearn 2002 team description – Faculty of Science, University of Amsterdam (2002)