Immune K-Means: A Novel Immune Algorithm for Data Clustering and Multiple-Class Discrimination

Michał Bereta[1] and Tadeusz Burczyński[1,2]

[1] Cracow University of Technology, Institute of Computer Modeling, Artificial Intelligence Department, Cracow, Poland
[2] Department for Strength of Materials and Computational Mechanics, Silesian University of Technology, Gliwice, Poland
email: beretam@torus.uck.pk.edu.pl
email: tburczyn@pk.edu.pl

Abstract This paper presents a novel approach to data clustering and multiple-class classification problems. The proposed method is based on a metaphor derived from immune systems, the clonal selection paradigm. A novel clonal selection algorithm – Immune K-Means, is proposed. The proposed system is able to cluster real valued data efficiently and correctly, dynamically estimating the number of clusters. In classification problems discrimination among classes is based on the k-nearest neighbor method. Two different types of suppression are proposed. They enable the evolution of different populations of lymphocytes well suited to a given problem: clustering or classification. The first type of suppression enables the lymphocytes to discover the data distribution while the second type of suppression focuses the lymphocytes on the classes' boundaries. Primary results on artificial data and a real-world benchmark dataset (Fisher's Iris Database) as well as a discussion of the parameters of the algorithm are given.

1 Introduction

Clustering is an important task in machine learning and data mining. Its main goal is to find the clusters' centers that properly describe data distribution. It leads to information compression and provides the knowledge for many properties of the data. It is often used as a first step in analyzing large amount of data, data visualization, in image compression and also in classification. There are many clustering algorithms starting with the most known and simple like k-means clustering, fuzzy k-means, ending with more complex and also time demanding. Time complexity is especially important when large datasets are to be clustered. This is the point, for example, when dealing with large databases of documents in web mining tasks. K-means clustering algorithm is one of the best known clustering methods. It is simple and is often used when other clustering methods are prohibited due to their computational demands. However, k-means clustering has some drawbacks. The most important one is the need to estimate the number of clusters in advance. The original k-means algorithm is also sensitive to the initialization of clusters'

centers and the clusters are sometimes found in areas where no data exist. The solution is often to start the algorithm several times to find the best parameters. On the other hand, there is a group of evolutionary heuristic methods for data clustering (genetic, swarm, immune algorithms). The proposed method belongs to the group of methods derived from immune systems paradigm, the clonal selection paradigm. It resembles the original k-means algorithm, but it gets rid of its main drawbacks – it is able to estimate the proper number of clusters and avoids getting stuck in inappropriate areas. Compared to other immune algorithms for data clustering, its computational cost is decreased by producing a limited number of clones and proper suppression mechanisms.

When the cluster centers are properly found, one can use them for the classification of unknown data samples. A simple and well known method is the k-nearest-neighbor method (kNN), in which the group of k cluster centers, the closest to the unknown data sample, decides to which class the sample belongs by a given voting scheme. The class information in the training data can also be used during the creation of clusters (as in the LVQ method).

Traditional clustering algorithms find the data distribution in such a way that the cluster centers are placed in more or less the middle of the data subset, that constitutes the cluster. When used for classification, these cluster centers are sometimes not the best ones, especially when the number of clusters is too small. The proposed algorithm based on the immune clonal selection paradigm aims to evade this obstacle by means of a new suppression mechanism, which focuses learning on the boundaries among classes and, as a result, allows evolving additional cluster centers near the class boundaries, which, as presented in this paper, can lead to better classification results and higher compression rates.

The rest of the paper is organized as follows. In Section 2 a short description of the Artificial Immune Systems (AIS) is given. In Section 3 the novel clonal selection algorithm - Immune K-Means - is described in detail. In Section 4 the primary results of the proposed method on artificial two-dimensional data and a benchmark machine learning database (Fisher Iris database) are given and the parameters of the algorithm, its advantages and limitations are brought up for discussion. Some conclusions are drawn in Section 5.

2 Artificial Immune Systems

Artificial Immune Systems [4] try to imitate real immune systems. The main task of an immune system is to defend the organism against pathogens. Different types of cells cooperate to give a reliable system able to adapt itself efficiently to changing environment. Most AIS use only the main ideas of real immune systems, namely clonal and negative selection which deal with the evolution of B- and T-cells, respectively.

2.1 General Concept of Artificial Immune Systems

B-cells with different receptors' shapes try to bind to antigens (training and testing data). The best fitted B-cells become stimulated and start to proliferate and produce clones, which are then mutated at very high rates (somatic hyper-mutation). After this process is repeated, it is likely that there will emerge a better B-cell (better solution). The whole process is called the clonal selection. T-cells undergo different type of evolution.

They are created in thymus and learn to recognize none of the self-cells presented to them. If a T-cell recognizes any of the self-cells it is destroyed in the thymus. The mature T-lymphocytes do not react to self-cells and thus can protect the organism from auto-destruction. This evolution is known as a negative selection and it has been used by several researches for such problems as computer security, novelty or anomaly detection. These tasks can be considered as two-class discrimination problems. The clonal selection paradigm has been mainly used for data compression, data and web mining, clustering and optimization.

In this paper attention is drawn to the clonal selection mechanism. It offers a number of interesting features from the standpoint of computation. The most important one is the ability to remember, generalize and classify previously encountered substances. The adaptive immunological response of the population of B-cells depends on several mechanisms such as recognition, stimulation, proliferation, hyper-mutation and suppression. The recognition of the antigen by a B-cell depends on the level of binding between them. The level of binding can be simulated by means of a given metric (or more general by some measure of similarity), for example Euclidean metric, which was also used in this research. The hipermutation of the offspring of the stimulated cells can be easily done by random changes in feature vectors describing B-cells. Suppression plays an important role in the whole process of immunological evolution. It is not fully understood how the real immune system regulates the population of lymphocytes over the time, however, as it is pointed out further in the paper, in artificial immune systems the proper suppression mechanism is crucial in evolving populations that have desirable properties. Carefully choosing the suppression mechanism one can achieve populations of lymphocytes of different properties.

2.2 Immune Algorithms for Clustering and Classification Problems

In the recent years AIS have been used for different tasks. Some researchers applied AIS to clustering and visualization tasks. There are many works where negative selection was applied to two-class discrimination problems, especially in anomaly detection, computer security etc. [3, 2, 6] where there is an evident problem of self – non-self discrimination. Different versions of clonal selection algorithm have usually been used for unsupervised learning. Little research have been conducted on the abilities of AIS as a supervised learning systems for multiple-class classification, i.e. class information has seldom been used for evolution of lymphocytes. The work of Watkins [5] is one of the exceptions. He adopted the algorithm of a resource limited artificial immune system developed by Timmis [7] and developed a supervised learning algorithm AIRS (Artificial Immune Recognition System) by adding into the learning procedure class information as an additional feedback for evolving cells. The method proposed in this paper differs significantly from the one of Watkins.

3 Immune K-Means – a Novel Clonal Selection Algorithm

A new clonal selection algorithm is proposed. It is a simple but robust clustering algorithm. Two ideas that allow for the control of the number of clusters are the way in which the clones are generated and the suppression mechanism which removes useless B-lymphocytes (clusters' centers). In the proposed method, the clone generating procedure

is a more efficient one when compared to the other methods used in immune algorithms because in Immune K-Means each lymphocyte produces only one clone (or one for each class when the class labels are considered during training). By means of different suppression mechanisms it is possible to evolve lymphocytes with different properties suitable for different tasks. Two suppression mechanisms are proposed in this section, the first one for data clustering and the second one for classification.

3.1 Immune K-Means for Data Clustering

The current implementation of the proposed method assumes that both antigens and B-cells are real valued vectors. The algorithm goes as follows:

1. Generate an initial population of B-cells as a set of random real valued vectors.
2. For each antigen a_i (a sample in the training set) find its nearest B-cell.
3. For each B-cell create its clone as a mean vector of all antigens for which a given B-cell is the nearest neighbor. If a clone is the same as the parent, mutate the clone by adding to each dimension a random value from range [*-mut, mut*]. It allows the clones to escape from the places where there are no clusters. Add the clones to the population of B-cells.
4. For each antigen find its new nearest B-cell.
5. Count the stimulation level of each B-cell. The stimulation level for the j-th B-cell is counted according to the equation (1):

$$stimulation\_level(B_j) = \sum_{a_i : a_i \in NN(B_j)} \exp^{-beta * euc\_dist(B_j, a_i)} \tag{1}$$

   where $NN(B_j)$ is a subset of antigens for which B_j is the nearest neighbor, *beta* is a positive constant and *euc_dist* stands for the Euclidean distance.
6. Sort B-cells in a descending order according to their stimulation level.
7. Perform the suppression.
8. Repeat step 3 – 7 until the termination condition is satisfied (in the current implementation it is a given number of iterations).

The first proposed suppression goes as follows:

Suppression I. Starting from the less stimulated B_j, for each $a_i \in NN(B_j)$ find B_{new} as a_i's new NN (nearest neighbor) among B-cells (except for the B_j) and calculate the distance to B_{new}. If a condition (2)

$$(new\_dist - old\_dist) \leq alpha \tag{2}$$

is satisfied for each $a_i \in NN(B_j)$, remove B_j and find a new NN for each $a_i \in NN(B_j)$. The values *old_dist* and *new_dist* are the Euclidean distances between the given antigen a_i and B-cells B_j and B_{new}, respectively.

The training samples are normalized to the range [0, 1]. The parameters of the algorithm are easily tuned. As tests showed, the parameter *beta* does not significantly influence the evolution and can be set to 1 and thus removed from tuning. The parameter *mut* should decrease over the iteration to 0, with a small starting value (like 0.05). The most important parameter is *alpha*. It indicates the maximum allowed change in distance

to the NN of each antigen while trying to remove a given B-cell. The initial value of this parameter should be small (like 0.001), and then, which can be confusing at first, it should grow over the iterations. The final value depends on the training set and it can be considered as a density measure of the B-cells' population. The bigger the final value, the less B-cells there will be in the final population.

Figure 1 depicts a result of applying the Immune K-Means algorithm to two-dimensional data. The algorithm is able to learn the structure of data easily.

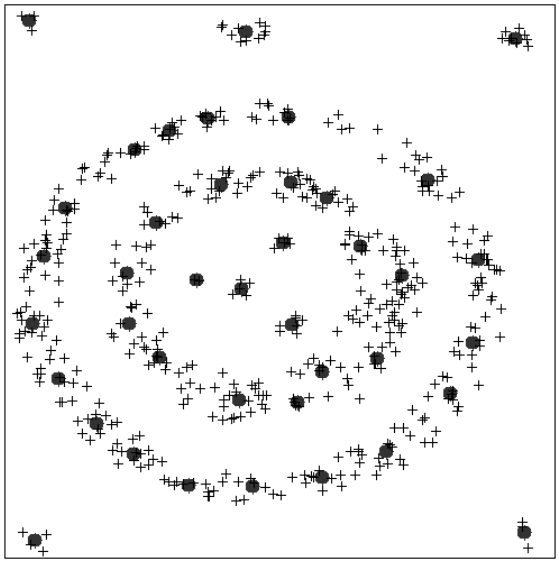

Figure 1. Immune K-Means algorithm is easily able to evolve a population of B-cells (black dots) that properly represent the structure of the data (crosses). The starting value of the parameter *alpha* was 0.001, the final value was 0.07. The iterations number was set to 25. There was only one B-cell in the initial population

3.2 Immune K-Means for Classification

Immune K-Means algorithm in the form described in previous section is an unsupervised learning algorithm as it does not use the information of the class labels of the training samples (antigens). However, it can be used for multiple-class discrimination. After the unsupervised learning, each lymphocyte is labelled as representing the class, to which the most of the antigens (training data) belong and for which this lymphocyte was the nearest neighbor. It is seen easily, that the class information of antigens is not utilized during the learning process. That fact does not cause any problem, when the system is trained for data compression and classification is not a main issue. But on the other hand, if the lymphocytes are to be good classifiers, using class information during learning seems to be a good idea. For that reason another type of suppression is proposed. While performing this type of suppression during learning it is necessary to count the class labels for each lymphocyte everytime when the nearest neighbors are found for the antigens. Class labels are found for each cell by finding the class that has

the biggest number of representatives among the antigens for which the given B-cell is the nearest neighbor. The difference is also while creating an offspring of each B-cell: each cell creates one clone for each class as the mean of antigens (from that class) for which the given B-cell is the nearest neighbor. The suppression goes as follows:

Suppression II. For each lymphocyte B_j , for each $a_i \in NN(B_j)$ find its new NN, B_{new}, among B-cells (except for the B_j) and check whether the condition (3)

$$C(B_j) = C(a_i) \quad AND \quad C(B_{new}) \neq C(a_i) \tag{3}$$

is satisfied. If the condition (3) is satisfied for at least one $a_i \in B_j$, B_j cannot be removed from the population, otherwise, permanently remove B_j and find a NN for each $a_i \in B_j$. $C(B_j)$ and $C(a_i)$ are the class labels of the B-cell and the antigen, respectively. This mechanism does not allow the removal of B-cells in the situation when at least one training sample was classified correctly by a given B-cell and it would be misclassified by another B-cell while trying to remove the first one. A given B-cell B_j is removed permanently only when its removal does not cause the growth of the total number of misclassified training samples.

The suppression II utilizes the class information during the learning and thus makes the Immune K-Means algorithm a supervised learning algorithm. Both variations of the proposed method share most of the steps. The most important difference is in the suppression step. Additionally one should easily observe that the parameter *alpha* does not play any role while using suppression II and thus it can be eliminated from tuning. Also, as tests revealed, sorting of B-cells according to their bounding degree seems to have no influence on the behavior of the algorithm and its final results when suppression II is used. Sorting is a necessary step while using suppression I, as it leads to an attempt to remove the B-cells from the edges of data clusters and thus enables the evolution of cells representing the inner distribution of data. On the contrary, suppression II is expected to focus the learning on the class discrimination and place the B-cells near the class boundaries. It is somewhat similar in meaning to the concept of SVM method, where only some of the training data necessary for learning the class boundaries (so called support vectors) are considered, while the rest are discarded. Of course, it is possible to use both types of suppression in the same time, which would result in developing B-cells of two types: those describing the inner distribution of data and those somewhat describing the class boundaries.

The proposed algorithm resembles in some parts the well known k-means clustering algorithm, which is fast and easy to implement. However, in contrast to k-means method, clonal selection algorithms are able to evolve a proper number of clusters. Combining these two approaches results in a clustering method that has all the positives and is free of the limitations. The biggest advantage of the new algorithm is that each B-cell creates only one clone in each iteration (or one clone for each class). The most important concept in this novel algorithm are proper suppression mechanisms which are able to decide when to remove useless (depending on some criterion) B-cells. The concept of the suppression based on the usefulness of the given cell rather than on the similarity among cells was adopted. This approach to performing suppression emerged during the development of the Two-Level AIS [1]. In that system, starting from the worst subpopulation of lymphocytes, the whole subpopulation was temporarily removed and the system was

evaluated whether there is any loss in the number of recognized antigens by the rest of subpopulations. If there was no loss, the subpopulation was removed permanently. The main concept of removing only the cells that are useless for the whole system can be easily adopted here.

Examples of the results of applying the proposed method to artificial two-dimensional data and a real world benchmark dataset are presented in the next section.

4 Results of Simulations

In this section some preliminary results for multiple-class classification with Immune K - Means are given. The influence and meaning of the parameters of the algorithm are also discussed.

4.1 Two-dimensional Data

Figure 2 shows the results of applying Immune K-Means algorithm for three class classification problem. Three groups of samples constituting three classes can be seen in the picture. Three approaches were used: suppression I, suppression II and suppression I together with suppression II.

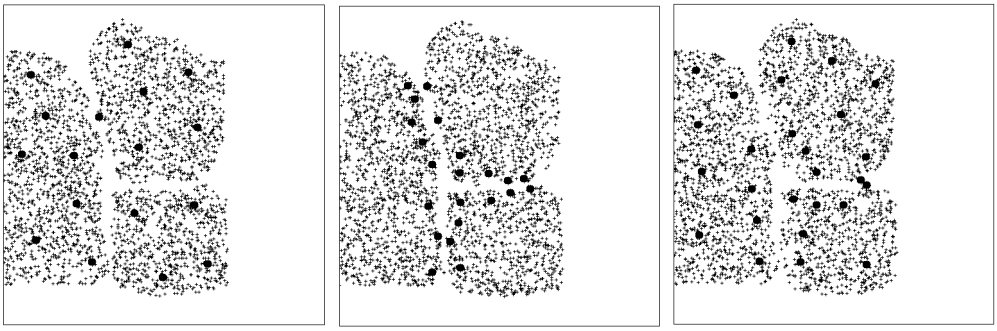

Figure 2. Immune K-Means algorithm with different suppression mechanisms. The left picture shows the results for suppression I, the picture in the middle for suppression II, the rightmost picture for suppression I & II. Note the additional B-cells (black dots) near the class boundaries on the rightmost picture as a result of suppression II. The starting value of the parameter *alpha* was 0.004, the final value was 0.13. The iterations number was set to 20. There were 5 B-cells in the initial population

As it can be seen, suppression I can find the data distribution properly when no class information is provided. When B-cells evolved in that way are used for classification (after labelling them with class info as it was described earlier), the results achieved were 96% of correctly classified data. It is obvious that the false classified samples were those lying near the class boundaries. However, the distribution of data was correctly discovered. On the other hand, when the suppression II was applied, the B-cells were created near the boundaries and the classification achieved was 100%. Information on data distribution is lost, which could be a disadvantage. If both needs have to be satisfied (data distribution and class boundaries information) both types of the suppression can

be applied, i.e. a given B-cell is removed from the population if, and only if, both suppression conditions allow it to be removed. The classification was 100% for the third case, while the inner data distribution was also revealed. The result is shown in the rightmost picture of Figure 2.

4.2 Results for Fisher's Iris Database

In this section results for the well known Fisher's Iris Database are provided. Iris database is a database of 150 samples of three species of iris flowers, each one described by four real values. Each class is represented by 50 instances. The proposed algorithm was evaluated by means of five-fold cross-validation method. The whole dataset was divided randomly into five parts and each part was used in turn as a testing set while the rest was used as the training set. The procedure was repeated three times and an average was counted. In each run the minimum value of *alpha* was 0.0005, the number of iterations was set to 30, and the starting size of the population was 20. Parameter k is the parameter of the kNN classification method. Table 1 shows the best results achieved for each parameter settings for the training data and Table 2 for the testing data.

Table 1. Results for Iris database. Percentage of correctly classified train data for different parameter settings.

k	*alpha_max*	0.01	0.05	0.1	0.15	0.2	0.22
1	Suppression I	100.0	100.0	98.8	96.7	94.8	90.8
	Suppression I & II	100.0	100.0	100.0	100.0	100.0	100.0
3	Suppression I	96.7	96.8	97.3	94.3	89.9	89.8
	Suppression I & II	96.4	97.0	97.2	96.7	95.1	95.2
4	Suppression I	96.7	96.2	95.6	91.4	87.4	84.6
	Suppression I & II	96.6	96.5	95.4	94.2	92.7	92.3
5	Suppression I	96.9	96.8	96.0	91.1	82.2	79.9
	Suppression I & II	97.1	96.8	96.8	95.5	85.4	86.3
6	Suppression I	96.9	96.8	96.3	90.4	78.3	68.2
	Suppression I & II	96.9	96.7	95.8	94.5	88.2	82.6

Table 2. Results for Iris database. Percentage of correctly classified test data for different parameter settings.

k	*alpha_max*	0.01	0.05	0.1	0.15	0.2	0.22
1	Suppression I	95.6	95.6	95.1	94.7	93.8	89.8
	Suppression I & II	95.8	95.6	95.1	94.9	94.7	94.9
3	Suppression I	96.0	96.0	95.6	93.8	88.7	89.6
	Suppression I & II	96.0	95.6	96.0	95.1	94.7	95.3
4	Suppression I	96.0	96.2	95.1	92.2	87.1	83.1
	Suppression I & II	96.0	95.1	95.1	94.4	94.2	93.1
5	Suppression I	96.9	96.4	94.4	90.0	85.6	76.7
	Suppression I & II	96.4	96.9	95.3	94.4	87.3	83.8
6	Suppression I	96.0	96.0	94.9	92.0	78.2	67.3
	Suppression I & II	96.4	96.0	95.3	94.9	86.2	84.0

The best achieved result for test data reported in Table 1 is 96.9%. As the suppression

II, when used alone, does not depend on the value of *alpha*, it is not mentioned in Table 1 nor Table 2. The best result achieved when the suppression II was used was 95.8% of correctly classified test data. The results are comparable with the results achieved by other standard classifiers. Watkins reports 96.7% correctly classified test data for the algorithm AIRS [8], when testing was performed by means of the same five-fold cross-validation approach.

Figure 3 and Figure 4 show how the performance of the Immune K-Means algorithm decreases as the final value of the parameter *alpha* increases. The increase of the final value of *alpha* leads to the decrease in the final population size. As it was mentioned earlier, *alpha* is responsible for the density of the lymphocytes' distribution. The less lymphocytes there are in the population, the worse the classification performance is, especially when k increases in the kNN classification method. As it can be expected, when both types of suppression are used, the decrease in the performance is less significant, as the additional B-cells exist nearby the class boundaries. It is obvious that using bigger values of k in kNN method leads to worse results when there are few cluster centers. As a result, even while using both types of suppression, performance decreases significantly for big values of k. Figure 5 shows how the percentage of recognized test data depends on the value of k when only the second type of suppression is used. In that case, the parameter *alpha* does not have any influence on the evolving B-cells. As Figure 2 suggested, in the case of suppression II, the use of big values of k should lead to worse results, as the B-cells are located close to the boundaries. This is also the point when the algorithm is tested on Iris database, which can be seen in Figure 5. The best choice for k while using the suppression II is $k = 1$.

It can be observed that the results for suppression II are a little worse than when both suppressions (or only suppression I) with the proper value of *alpha* are used. However, there is one more thing to discuss at this point. Figure 6 shows how the population size depends on the final value of *alpha*. For small values of *alpha*, the population size is big – 110 B-cells in the final population results in a small compression rate, as the training dataset size was 120 (in each run of the five-fold cross-validation the size of the training dataset is 4*30=120). The population size decreases when the final *alpha* increases, however, it leads to the decrease in the performance, too. For bigger values of *alpha*, the result of applying suppression II together with suppression I can be seen, as the population size is bigger because the additional B-cells are created. When only the suppression II is used, the final population size does not depend on the value of *alpha* and its average value was 9 – 10 B-cells which gives significantly better compression of information. The same population size, when only the suppression I is used, is achieved for value of *alpha* of more or less 0.22. However, it can be seen that the performance of such a population is definitely worse than when the suppression II is used. It shows that suppression II is able to evolve B-cells that are more dedicated to classification especially when bigger compression rates are required. On the other hand, the B-cells evolved by suppression II seems to be over-trained, which can be seen from Table 1 and Table 2 – the percentage of recognized training antigens is always 100% while using $k = 1$. This shows that suppression II can lead to B-cells with less generalization abilities than suppression I. However, the difference in population size (for similar classification performance) while using these different suppression types can be significant. The computations are also much faster when only the suppression II is used.

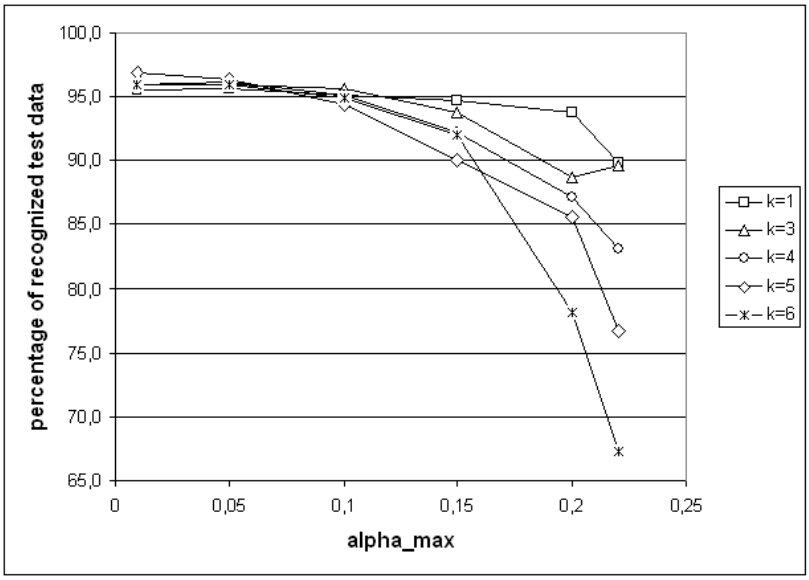

Figure 3. Percentage of recognized test data for different values of k (suppression I). Results for Iris database.

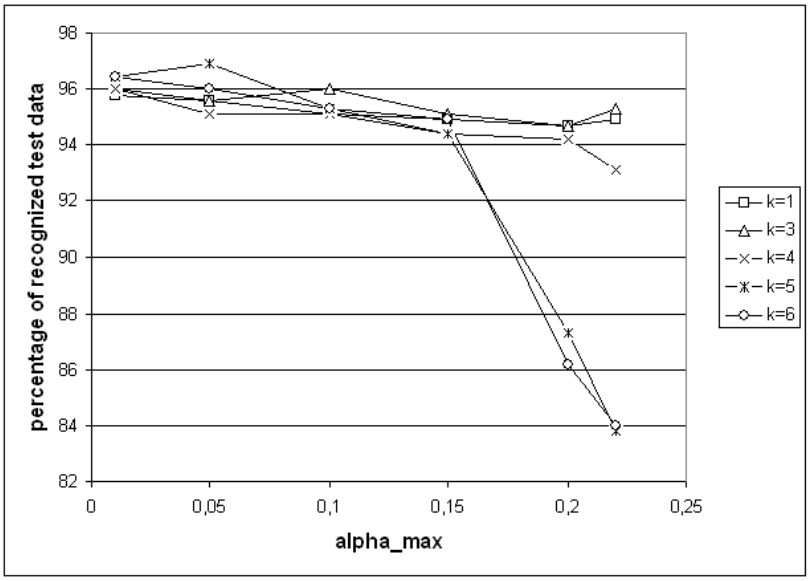

Figure 4. Percentage of recognized test data for different values of k (suppression I & II). Results for Iris database.

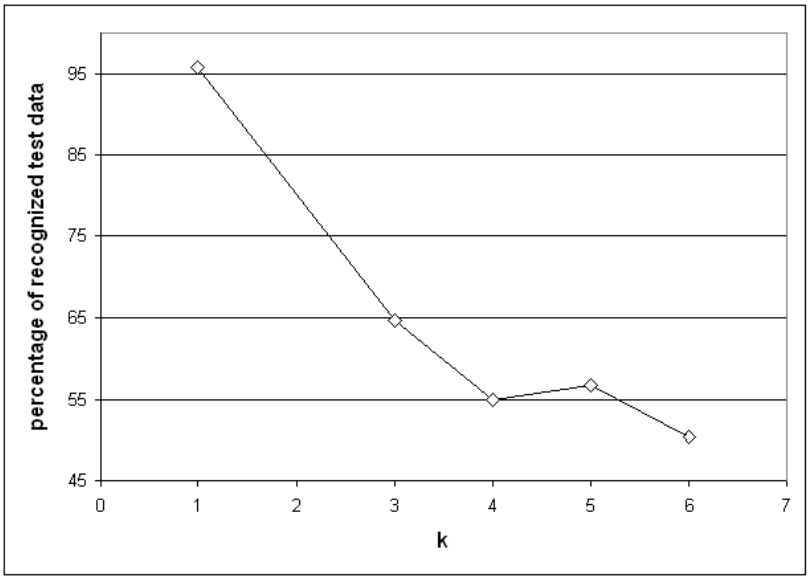

Figure 5. Percentage of recognized test data (suppression II). Results for Iris database.

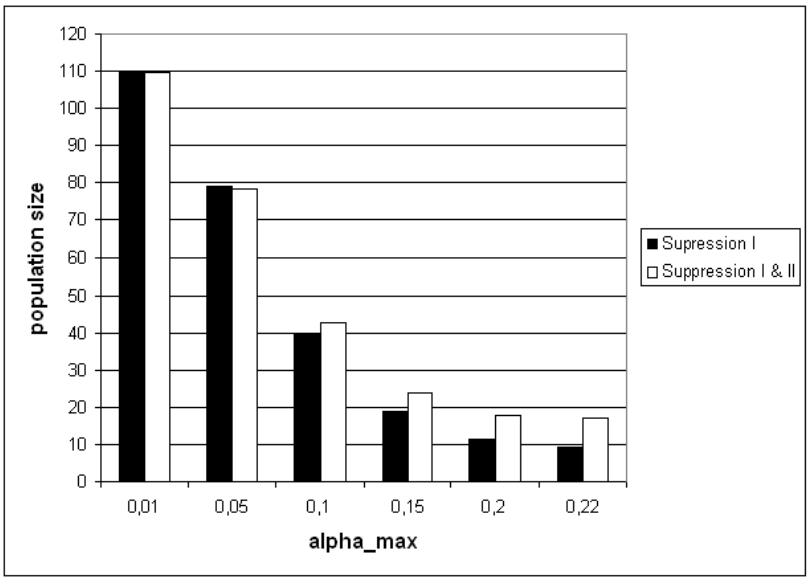

Figure 6. Dependence of the population size on the final value of *alpha*.

5 Conclusion

The proposed Immune K-Means algorithm is a new, promising approach to data clustering and classification. It is easy but robust. It has a small number of parameters, that have clear interpretations, they are easily tuned and their impact on the performance of the algorithm is easily predictable. The proposed method is less computationally demanding than other immunological algorithms (like AIRS [8]) as there are less clones produced by the B-cells in each iterations. Selecting different types of suppression can lead to significantly different results, to populations that have different properties desirable in different applications. In this paper two suppression mechanisms were presented, one for efficient data clustering and data distribution learning, the second one well suited for discovering the class boundaries. It was shown how the different types of suppression used in the same time lead to better classification results. The possible improvement of the generalization abilities of evolved B-cells will be studied in the future.

Acknowledgement

The research is financed from The Foundation For Polish Science.

Bibliography

[1] M. Bereta and T. Burczyński. Hybrid immune algorithm for feature selection and classification of ECG signals. In T. Burczyński, W. Cholewa, and W. Moczulski, editors, *Recent Developments in Artificial Intelligence Methods*, AI-METH Series, pages 25–28, Gliwice, 2005.

[2] D. Dasgupta and S. Forrest. Novelty detection in time series data using ideas from immunology. In *ISCA 5th International Conference on Intelligent Systems*, pages 19–21, Reno, Nevada, June 1996.

[3] D. Dasgupta and S. Forrest. *An Anomaly Detection Algorithm Inspired by the Immune System*, chapter 14 in the book entitled Artificial Immune Systems and Their Applications, pages 262–277. Springer-Verlag, Inc., January 1999.

[4] L.N. de Castro and J. Timmis. *Artificial Immune Systems: A New Computational Approach*. Springer-Verlag, London. UK., September 2002.

[5] D. Goodman, L. Boggess, and A. Watkins. Artificial immune system classification of multiple-class problems. In *Artificial Neural Networks in Engineering (ANNIE-2002)*, 2002.

[6] P.D'haeseleer, S. Forrest, and P. Helman. An immunological approach to change detection: algorithms, analysis, and implications. *In Proceedings of the 1996 IEEE Symposium on Computer Security and Privacy (1996).*

[7] J. Timmis and M. J. Neal. A Resource Limited Artificial Immune System for Data Analysis. *Research and Development in Intelligent Systems XVII*, pages 19–32, December 2000. Proceedings of ES2000, Cambridge, UK.

[8] Andrew Watkins, Jon Timmis, and Lois Boggess. Artificial immune recognition system (AIRS): An immune-inspired supervised machine learning algorithm. *Genetic Programming and Evolvable Machines*, 5(3):291–317, September 2004.