MUNICIPAL CREDITWORTHINESS MODELLING BY ARTIFICIAL IMMUNE SYSTEMS

Luděk KOPÁČEK, Vladimír OLEJ
Institute of System Engineering and Informatics, Faculty of Economics and Administration,
University of Pardubice, Studentská 84, 532 10 Pardubice, Czech Republic,
e-mail: Ludek.Kopacek@upce.cz, Vladimir.Olej@upce.cz

ABSTRACT

Municipal creditworthiness represents a very important measure of financial stability. Banks, supervisors, and other institutions rely upon ratings of creditworthiness to produce representations of the risk, nevertheless rating agencies never disclose evaluated parameters and their weight. This implies effort of scientific community to find accurate models for prediction of municipal creditworthiness rating. In this paper we present a classification model for municipal rating based on algorithms from the branch of artificial immune systems; Immunos-1, Immunos-2, Immunos-99, CLONALG and CLONCLAS.

Keywords: *Municipal creditworthiness, Artificial immune system, Clonal selection, Immunos, CLONALG, CLONCLAS, Classification*

1. INTRODUCTION

Municipal creditworthiness is a creditor's measure of a municipal's past and future ability to repay debts. A commonly used measure of the financial strength of a community is its municipal bond rating. A bond rating is an assessment of the probability of timely repayment of debt produced by an independent rating agency [18]. More specifically, municipal ratings are based upon the analysis of four primary factors relating to municipal finance: economy, debt, finances, and administration/management strategies. Each of these factors is evaluated individually for its effect on the other factors in the context of the municipality's ability to repay its debt. The precise factors and related weight of these factors used in determining municipal rating are not publicly disclosed by the rating agencies.

Banks, supervisors, and other institutions rely upon these systems to produce accurate, stable representations of the risks. Financial institutions use this information in portfolio selection. Credit rating also helps a local body to reach out to a larger pool of investors. Since credit rating provides a relative assessment of risk compared to other borrowing options, it also helps in pricing of the borrowing. Stronger credit rating results in a cheaper borrowing and vice versa. Credit ratings also serve as an external validation of financial health.

Models developed to classification or explain municipal bond ratings rely on a combination of financial and socioeconomic variables [15], [22], and [26], thereby rating systems are generally combinations of statistical methods and expert systems. Statistical methods calculate risk sensitive score with the help of optimization methods, while the expert systems evaluate soft-facts using formalized expert knowledge [20]. Other models for rating classification can be based for example on decision trees and rough sets [28], grammatical evolution [1], [2], neural networks [3], or fuzzy logic [19], [23].

In our paper we present a classification model for municipal rating based on dataset created in [12]. As modelling methods we used five algorithms from the branch of artificial immune systems; Immunos-1, Immunos-2, Immunos-99, CLONALG and CLONCLAS.

2. ARTIFICIAL IMMUNE SYSTEMS

The complexity of many computational problems has led to the development of a range of innovative techniques. One area of such research is evolutionary computing. The central idea of this approach is the evolution of population of candidate solutions through the application of operators inspired by natural selection and random variation. The humane immune system is an example of a system, which maintains a population of diverse individuals. This system was taken as an inspiration for a number of artificial evolutionary systems. Immunological metaphors were extracted, simplified, and applied to create an effective classification technique.

The human immune system consists of a multilayered architecture presenting different types of defence against infectious material called pathogens. Pathogens recognizable by immune system are called antigens. The most important is the third layer-the cellular layer. This layer is composed of a variety of different cell types with different roles. These cells (especially antibodies B-cells and T-cells) are responsible for anomaly detection, which is proceeded according to affinity [8], [10], [24] (degree of similarity between a recognition cell and an antigen). The adaptive ability of the immune system is a process called affinity maturation. During an immune response the recognition cell generates many clones of itself in an attempt to gain a better match the next time the antigen is seen (the process is called clonal selection). Each clone is then mutated in proportion to the affinity between the recognition cell and the antigen (somatic hypermutation). The last step covers elimination of newly differentiated clones carrying low affinity antigenic receptors.

Mainstream thoughts about artificial immune systems is concerned on three aspects; clonal selection, negative selection, and immune network. The clonal selection theory [6], [10] assume that cells effective at recognizing pathogenic material are selected to survive and propagate.

An example of an algorithm based on clonal selection theory is CLONALG [9], CLONCLAS [25], or the group of algorithm Immunos [4]. Negative selection theory [8], [11], [17] is based on eliminating those cells whose receptors are capable of recognizing self-antigens. This process can be used for anomaly detection algorithms. Immune network theory [16], [21] proposes that the immune system maintains a network of cells that learn and maintain memory using a feedback mechanism. This theory says that even though information is learned, it can be forgotten if the information is not intensified. Algorithm aiNet which is based on this theory can be found in [8].

3. GROUP OF IMMUNOS ALGORITHM

Group of Immunos algorithm represents a population based artificial immune systems [4], [7]. One of the first attempt to use immune system principles as a basis for supervised learning [7] was called Immunos-81. This system represent predecessor of all other algorithms from this group. The immune system concepts were reduced to their most fundamental level before they were incorporated into the algorithm.

Immunos-81 was described in [7], unfortunately without detailed description so the implementation can be speculated. Carter focused his work on the description of artificial T-cell, which provided partitioning on the right problem domain. Nevertheless this can be let out, and each problem domain can be solved by separate instance of algorithm. Although Carter's work postulated applicability on multi type antigen vector, a description was given only for binary vectors of antigens.

These are the reasons why the result of Immunos-81 is not repeatable. This was confirmed in [4] where the Immunos-81 was reviewed and new version of the basic idea of this algorithm was designed. The first two implementations, named Immunos-1 and Immunos-2, were created by Brownlee [4] with the goal of repeating the results of the original work [7]. The third algorithm Immunos-99 was created as the extension of previous version.

3.1. Immunos-1

Although the Immunos-1 algorithm [4] is based on the ideas of Immunos-81, there are some differences. It solves single problem domain, which means that T-cells are not incorporated into the algorithm. In the Immunos-1 an uniform antigen vector structure is used. Each antigen has the same vector length and the same vector structure (order and data type). For each antigen in training phase of Immunos-1 a clone of B-cell is created. This means no data reduction. The last difference between Immunos-1 and Immunos-81 is the different way of affinity calculation. In Immunos-81 [7], the affinity values are calculated separately for each paratope (attribute) of antigen and B-cell clone. These affinities are summarized across all B-cells in antigen-group. The avidity between the unknown antigen and the given antigen group is the combination of all paratope affinities and the concentration of the given antigen-group. The algorithms will be described with this notation.

Input set with training data, in reflection of artificial immune system terminology, is named as set of antigens Ag

$$Ag = \left\{ \mathbf{ag_i} \middle| \mathbf{ag_i} = (ag_{i,1}, \ldots, ag_{i,j}, \ldots, ag_{i,l}), \mathbf{ag_i} \in S^l \right\}, \qquad (1)$$

where $\mathbf{ag_i}$ is i-th antigen of set Ag, l is number of attributes describing that antigen, $ag_{i,j}$ is j-th attribute of i-th antigen $\mathbf{ag_i}$. Analogically with the set of antigens Ag one could defined the set of antibodies Ab

$$Ab = \left\{ \mathbf{ab_i} \middle| \mathbf{ab_i} = (ab_{i,1}, \ldots, ab_{i,j}, \ldots, ab_{i,l}), \mathbf{ab_i} \in S^l \right\}, \qquad (2)$$

where $\mathbf{ab_i}$ is i-th antibody of set Ab, l is number of attributes describing that antibody, $ab_{i,j}$ is j-th attribute of antibody $\mathbf{ab_i}$. Set of classes Cl is defined as

$$Cl = \{1, 2, \ldots, n_{class}\}, \qquad (3)$$

where n_{class} is the number of class of solved problem. Function *class* is defined as projection from state space S^l into set of classes Cl

$$class : S^l \to Cl \,. \qquad (4)$$

Function *class* for each antigen $\mathbf{ag_i} \in Ag$ or antibody $\mathbf{ab_i} \in Ab$ returned given class $c_i \in Cl$. According to classes can be defined antibody sets Ab_i, which contain only antibody of one of class c_i

$$Ab_i = \left\{ \mathbf{ab_j} \middle| \mathbf{ab_j} \in Ab; class(\mathbf{ab_j}) = c_i \right\}, \qquad (5)$$

$$Ab = Ab_1 \cup Ab_2 \cup \ldots \cup Ab_i \cup \ldots \cup Ab_{n_{class}} \,. \qquad (6)$$

3.1.1. Training phase of algorithm Immunos-1

The training phase consists of the division of input antigens into groups per known class label. The B-cell population is created for each class label. No enumeration is necessary while the training phase is provided. The pseudo code of this phase is shown on fig. 1.

```
[Abm] := Function Immunos1Train (Ag)
Begin
  [Abm] = InitImmunos1(Ag);
  return [Abm];
End;
```

Fig. 1 Pseudo code of training phase of Immunos-1

Function `InitImmunos1` creates output sets of antibodies $Ab_{m,i}$, that contain antibodies of one class

$$Ag_i = \left\{ \mathbf{ag_j} \middle| \mathbf{ag_j} \in Ag; class(\mathbf{ag_j}) = c_i \right\}, \qquad (7)$$

$$Ag_i \subset Ag \,, \qquad (8)$$

$$Ag_i \cap Ag_j = \varnothing \,, \text{ for } i \neq j \,. \qquad (9)$$

For each antigen one antibody is generated. This means that for each antigen set Ag_i one set of antibodies is created

$$Ab_{m,i} = Ag_i\,, \tag{10}$$

$$Ab_m = Ab_{m,1} \cup \ldots \cup Ab_{m,i} \cup \ldots \cup Ab_{m,n_{class}}\,. \tag{11}$$

Inner representation the classifier is identical with the training set structure.

3.1.2. Classification phase of algorithm Immunos-1

During the classification phase, the class label c_i for unknown antigen $\mathbf{ag_x}$ is assigned. Process of classification of unknown antigen to one of population antibodies $Ab_{m,i}$ is based on affinity among unknown antigen and all sets of antibodies $Ab_{m,i}$. Pseudo code of this phase is on fig. 2.

```
[cx] := Function Imm1Clasify (Abm,agx)
Begin
  bestAvIndex = -1;
  bestAvidity = 0;
  For each Abm,i do
    avidity[i] = countAvidity(Abm,i,agx);
    If ( bestAvidity < avidity[i] ) then
      bestAvidity = avidity[i];
      bestAvIndex = i;
    End if;
  End for;
  return bestAvIndex;
End;
```

Fig. 2 Pseudo code of classification phase of Immunos-1

Function `countAvidity` counts the value of avidity [4] for antibody set $Ab_{m,i}$ and unknown antigen $\mathbf{ag_x}$. Avidity is defined as

$$avidity(\mathbf{ag_x}, Ab_{m,i}) = \frac{|Ab_{m,i}|}{\sum\limits_{\mathbf{ab_j} \in Ab_{m,i}} D(\mathbf{ag_x}, \mathbf{ab_j})}\,, \tag{12}$$

where $|Ab_{m,i}|$ is the number of antibodies in set $Ab_{m,i}$ and $D(\mathbf{ag_x},\mathbf{ab_j})$ is the metrics counted between antibody $\mathbf{ab_j}$ and antigen $\mathbf{ag_x}$. This metrics is defined as

$$D(\mathbf{ag_x}, \mathbf{ab_j}) = \sqrt{\sum_{k=1}^{l} \left(ab_{j,k} - ag_{x,k}\right)^2}\,, \tag{13}$$

where $ab_{j,k}$ is k-th attribute of antibody $\mathbf{ab_j}$ and $ag_{x,k}$ is k-th attribute of antigen $\mathbf{ag_x}$. This metric $D(\mathbf{ag_x},\mathbf{ab_j})$ is in [4] called, in the wrong meaning, as affinity. Usually affinity is defined [8], [10], [24] as similarity ratio between B-cell and the antigen so that a high value of affinity represent a significant similarity of B-cell and an antigen conversely low value means a weak similarity. In [4] affinity is used in the opposite meaning, which can cause confusion. Due to this fact, instead of the term affinity we will use the term similarity metric.

Unknown antigen $\mathbf{ag_x}$ is classified into the appropriate class of antibody population $Ab_{m,i}$ with the highest value of avidity

$$c_x = \arg\max_i (avidity(\mathbf{ag_x}, Ab_{m,i}))\,. \tag{14}$$

Avidity in equation (12), means the inverse value of the average distance between vector of unknown antigen $\mathbf{ag_x}$ and all antibody vectors $\mathbf{ab_j}$ in a given antibody population $Ab_{m,i}$.

3.2. Immunos-2

Training phase of algorithm Immunos-2 is based on creating of inner representation of classifier. Pseudo code description is given on fig. 3.

```
[E] := Function Immunos2Train (Ag)
Begin
  E = InitImmunos2(Ag);
  return E;
End;
```

Fig. 3 Pseudo code of training phase of algorithm Immunos-2

Function `InitImmunos2` creates inner representation of classifier; where Ag is the training set of antigens, and $\mathbf{e_i} \in \mathrm{E}$ are representatives of each class c_i. Training set Ag with assigned class value is divided into groups Ag_i, so that each group consists of members of the same class value c_i. Each group Ag_i has defined one representative $\mathbf{e_i}$

$$\mathbf{e_i} = \frac{\sum\limits_{\mathbf{ag_j} \in Ag_i} \mathbf{ag_j}}{|Ag_i|}\,, \tag{15}$$

where $|Ag_i|$ is the number of antigens of class c_i in the training set. As a result of the training phase we obtain set E, with representatives $\mathbf{e_i}$ for all classes

$$E = \{\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_i}, \ldots, \mathbf{e_{n_{class}}}\}. \tag{16}$$

The classification phase of algorithm Immunos-2 is formally similar to the classification phase of algorithm Immunos-1.

Inputs for classification are representatives $\mathbf{e_i}$ of classes c_i, and antigen $\mathbf{ag_x}$ with unknown assignment of class c_x. The output is class value c_x. Classification is realized by avidity calculation among all representatives $\mathbf{e_i}$ and unknown antigen $\mathbf{ag_x}$ according to

$$avidity_2 : S^l \times S^l \to \Re\,, \tag{17}$$

$$avidity_2(\mathbf{ag_x}, \mathbf{e_i}) = \frac{|Ag_i|}{D(\mathbf{ag_x}, \mathbf{e_i})}\,, \tag{18}$$

where $D(\mathbf{ag_x},\mathbf{ab_j})$ is similarity metric (13), in [4] called affinity.

Unknown antigen $\mathbf{ag_x}$ is classified into the class represented by $\mathbf{e_i}$ with the highest value of avidity

$$c_x = \arg\max_i (avidity_2(\mathbf{ag_x}, \mathbf{e_i}))\,. \tag{19}$$

3.3. Immunos-99

Immunos-99 uses different method in the training phase [4]. Groups of members with the same class value c_i are created in the same way; however these groups are in the next step transformed by modification of algorithm CLONALG. Classification phase is after that the same as in algorithm Immunos-1. Training phase description is given on fig. 4.

```
[Abm] := Function Immunos99Train (Ag,ninit,ngen,t)
Begin
  Abi = InitImmunos99(Ag,ninit);
  For gen := 1 to ngen do
       For each Abi do
        For each abj∈Abi do
          fitj = countFitness (abj,Ag);
        End for;
        [Abi',npc]= performPruning(fit,Abi,t);
        Abi'' = performCloningAndMutation(Abi');
        Abi = insertRandomAntigens(Abi'',npc);
       End for;
  End for;
  For each Abi do
       For each abj∈Abi do
        fitj = countFinalFitness (abj,Ag);
       End for;
       Abm,i = performPruning(fit,Abi,t);
  End for;
  return Abm;
End;
```

Fig. 4 Pseudo code of training phase of Immunos-99

The first step of this phase is initialization (function `InitImmunos99` on fig. 4). Set of antigens Ag, is divided into groups Ag_i in this phase. Group Ag_i consists of antigens belonging into class c_i. In the next step, are from set Ag randomly selected antigens, and for them is created exactly one antibody, which is included into set Ab. Size of set Ab is related to size of set Ag by parameter n_{init}

$$|Ab| = \lfloor |Ag| \times n_{init} \rfloor, \tag{20}$$

$$Ab \subseteq Ag . \tag{21}$$

The set of antigens Ab is then divided into groups Ab_i according to class value. In each generation, for each antibody $\mathbf{ab_j} \in Ab_i$ there is calculated a coefficient indicating how well antibody $\mathbf{ab_j}$ identify antigens, which belong to the same class. This coefficient is labeled as fit_j. The value of fit_j for each antibody $\mathbf{ab_j} \in Ab_i$ is defined as

$$fit_j = fitness(\mathbf{ab_j}, Ag) = \frac{correct}{incorrect}, \tag{22}$$

$$correct = \sum_{\substack{\mathbf{ag_x} \in Ag \\ class(\mathbf{ag_x}) = class(\mathbf{ab_j})}} score(\mathbf{ab_j}, \mathbf{ag_x}), \tag{23}$$

$$incorrect = \sum_{\substack{\mathbf{ag_x} \in Ag \\ class(\mathbf{ag_x}) \neq class(\mathbf{ab_j})}} score(\mathbf{ab_j}, \mathbf{ag_x}). \tag{24}$$

Value of $score(\mathbf{ab_j},\mathbf{ag_x})$ is based on the calculation of metric $D(\mathbf{ab}_j,\mathbf{ag_x})$ between antibody $\mathbf{ab}_j$ and antigen $\mathbf{ag_x}$ (13). This value of $score(\mathbf{ab_j},\mathbf{ag_x})$ is given by

$$score(\mathbf{ab_j}, \mathbf{ag_x}) = |Ab_i| - index_j , \tag{25}$$

where $index_j$ is an ordinal number of antibody $\mathbf{ab_j} \in Ab_i$ in list ordered ascending by value $D(\mathbf{ab_j},\mathbf{ag_x})$ and $|Ab_i|$ is number of antibody in set Ab_i. For antibody $\mathbf{ab_j}$ with the lowest value of $D(\mathbf{ab_j},\mathbf{ag_x})$ is highest value of $score(\mathbf{ab_j},\mathbf{ag_x}) = |Ab_i|$ and for antibody $\mathbf{ab_k}$ with the highest value of $D(\mathbf{ab_j},\mathbf{ag_x})$ is value of $score(\mathbf{ab_j},\mathbf{ag_x}) = 1$. Calculation of value fit_j is not affected by absolute extent of value of $D(\mathbf{ab_j},\mathbf{ag_x})$, only by their relation. For all antibodies we obtain vector

$$\mathbf{fit} = \left(fit_1, fit_2, \ldots, fit_j, \ldots, fit_{|Ab_i|}\right). \tag{26}$$

Function `performPruning(fit,Abi,t)` prune away low quality antibodies from set Ab_i. Low quality is defined as value of fit_j lower than the given threshold t. Set Ab_i' is, after elimination, formulated as

$$Ab_i' = \left\{\mathbf{ab_j} \middle| \mathbf{ab_j} \in Ab_i; fit_j > t\right\}. \tag{27}$$

If the value t = -1, then as a threshold the average value of components of vector **fit** of set Ab_i is used. Description of set Ab_i' is as follows

$$Ab_i' = \left\{\mathbf{ab_j} \middle| \mathbf{ab_j} \in Ab_i; fit_j > \min\left(fit_{avg}, 1\right)\right\}, \tag{28}$$

where fit_{avg} is the average value of components of vector **fit**. This method will be called dynamical thresholding or dynamical elimination.

Function `performCloningAndMutation(Abi')` adds such antibodies that were created by clonal selection and mutation into set of antibodies

$$Ab_i = Ab_i' \cup Ab_{clone} , \tag{29}$$

$$Ab_{clone} = \bigcup_{j=1}^{|Ab_i|} Ab_{clone}(\mathbf{ab_j}), \tag{30}$$

where $Ab_{clone}(\mathbf{ab_j})$ is set of identical clones created from antibody $\mathbf{ab_j}$. For each antibody $\mathbf{ab_j} \in Ab_i$ is created $n_{clone}(\mathbf{ab_j})$ of identical clones

$$n_{clone}(\mathbf{ab_j}) = \left\lfloor \frac{r(\mathbf{ab_j})}{\sum_{\mathbf{ab_k} \in Ab_i} r(\mathbf{ab_k})} |Ag_i| + 0{,}5 \right\rfloor, \tag{31}$$

$$r(\mathbf{ab_j}) = \frac{index_j}{|Ab_i|} , \tag{32}$$

where $index_j$ is ordinal number of antibody $\mathbf{ab_j} \in Ab_i$ in list ordered ascending by value fit_j and $|Ab_i|$ is number of antibody in set Ab_i. For antibody $\mathbf{ab_j}$ with the lowest value of fit_j is value of $index_j = 1$, for antibody $\mathbf{ab_j}$ with the highest value of fit_j is value $index_j = |Ab_i|$. The number of clones is not affected by absolute extent of value fit_j, only their relation.

All antibodies $\mathbf{ab_k}$ from set $Ab_{clone}(\mathbf{ab_j})$ are mutated. Mutation is inversely proportional to the value of $r(\mathbf{ab_j})$. Clones created from antibodies with high value of fit_j, have low mutation rate, in contrast to clones created from antibodies with low value of fit_j, which are more affected by mutation. Mutation is done for each attribute of antibody $\mathbf{ab_k} \in Ab_{clone}(\mathbf{ab_j})$ separately.

Function `insertRandomAntigens(`Ab_i''`,`n_{pc}`)` adds into the set Ab_i randomly selected antigens from Ag_i. The number of antigens is given by n_{pc}, which is defined as number of eliminated antibodies by function `performPruning(`**`fit`**`,`Ab_i`,`t`)`.

Function `countFinalFitness(`$\mathbf{ab_j}$`,`Ag`)` counts value fitness in the same way as function `countFitness(`$\mathbf{ab_j}$`,`Ag`)`, only the calculation of $score_{fin}(\mathbf{ab_j},\mathbf{ag_x})$ is done differently

$$score_{fin}(\mathbf{ab_j},\mathbf{ag_x}) = \begin{cases} 1 & if\ \mathbf{ab_j} = \arg\min\limits_{\mathbf{ab_k} \in Ab_i} (D(\mathbf{ab_k},\mathbf{ag_x})) \\ 0 & else \end{cases} . \quad (33)$$

As the result of training phase of Immunos-99 we obtain set of memory cells Ab_m, which are defined as

$$Ab_m = \bigcup_{i=1}^{n_{class}} Ab_{m,i} , \quad (34)$$

where $Ab_{m,i}$ is the set of memory cells corresponding to class c_i.

Classification phase is the same for algorithm Immunos-99 as for Immunos-1.

4. CLONALG AND CLONCLAS

A description of training phase of algorithm CLONALG [9] by pseudo code is introduced on fig. 5. The inputs are as follows: set of antibodies Ag, number of generations n_{gen}, population size of antibodies n_{Ab}, population size of memory cells n_m, number of antibodies for selection n_s, clonal factor β, number of randomly generated antibodies for diversity preservation n_d.

First step of training phase is initialization (function `initClonalg`), which prepare fundamental structures for training. Set of memory cells and remainder cells are randomly created. Size of these sets is given by

$$|Ab_m| = n_{Ab} , \quad (35)$$

$$n_r = |Ab_r| = n_{Ab} - n_m , \quad (36)$$

Initialization is done so that sets Ab_m and Ab_r have the same distribution of antibodies among classes as it is among classes of Ag set. Set of antibodies Ab is then defined as

$$Ab = Ab_m \cup Ab_r . \quad (37)$$

```
[Abm] := Function ClonalgTrain
(Ag,ngen,nAb,nm,ns,β,nd)
Begin
  [Abm,Abr] = initClonalg(Ag,nAb,nm);
  Ab = Abm ∪ Abr;
  For i := 1 to ngen do
    For each agi∈Ag do
      af = affinity(agi,Ab);
      Abs = select(Ab,af,ns);
      C = clone(Abs,β,af);
      C′ = mutateClones(C,af);
      af′ = affinity(agi,C′);
      abc = select(C′,af′,1);
      Abm = insert(Abm,abc);
      Abr = replace(Abr,nd,af);
    End for;
  End for;
  return Abm;
End;
```

Fig. 5 Pseudo code of training phase of CLONALG

In the second step is calculated affinity among selected antigen $\mathbf{ag_i}$ and population of antibodies Ab. In [9] the affinity calculation used metric according to similarity metric (13). Resulting vector $\mathbf{af}$, has components af_j defined as

$$\mathbf{af} = (af_1, af_2, \ldots, af_j, \ldots, af_{n_{Ab}}), \quad (38)$$

$$af_j = D(\mathbf{ag_i},\mathbf{ab_j}). \quad (39)$$

It must be stated that low similarity metric value af_j according to (39) means high similarity rate of antigen $\mathbf{ag_i}$ and antibody $\mathbf{ab_j}$, and vice versa.

Selection of best antibodies $\mathbf{ab_j}$ is based on the lowest value of af_j according to antigen $\mathbf{ag_i}$. For each selected antibody $\mathbf{ab_j} \in Ab_s$. Number of clones nc_j is given by

$$nc_j = \left\lfloor \frac{\beta \cdot n_{Ab}}{i} \right\rfloor, \quad (40)$$

where i is index in list of antibodies from set Ab_s ordered descending by value of affinity af_j. The best antigen has generated the most clones. Clones created from antibody $\mathbf{ab_j}$ are included in class C_j.

As the result of `clone` function we obtain the set of all clones C, prepared for mutation. Clones generated from antibodies with low value of similarity metric af_j have low mutation rate, in contrary to clones generated from antibodies with high value of this metric. In consequence, high-quality antibodies are affected by process of mutation only slightly contrary to low-quality antibodies. Mutation rate is based on α, calculated as

$$\alpha = \exp^{-\frac{af_{max}}{af_j}} , \quad (41)$$

where af_j is the value of similarity metric (13) corresponding to antibody $\mathbf{ab_j}$, and af_{max} is the maximal value of this metric for the whole set of antibodies Ab.

For all newly generated antibodies $\mathbf{ab_k}' \in C'$ is calculated value of similarity metric af_j'. Resulting vector $\mathbf{af'}$ is described as

$$\mathbf{af}' = \left(af_1', af_2', \ldots, af_j', \ldots, af_{|C'|}' \right) \quad (42)$$

where value af_j' is given by (39). Consequently, antibody $\mathbf{ab_c}$ with the best value of affinity is chosen from set C'. If the value of affinity of antibody $\mathbf{ab_c} \in C'$ is better then affinity of the best memory cell $\mathbf{ab_{best}} \in Ab_m$, than antibody $\mathbf{ab_c}$ replace this memory cell (function `replace`).

In the final step of each cycle, within `replace` function, n_d antibodies are changed from set Ab_r (with worst value of affinity) for newly generated antibodies $\mathbf{ab_y} \in S^l$. This process ensures the diversity in the evolution of of antibodies Ab. The result of the training phase of CLONALG algorithm is the set of memory cells Ab_m. In the classification phase the unknown antigen $\mathbf{ag_x}$ is classified into class c_x, which correspond to memory cell $\mathbf{ab_{best}} \in Ab_m$ with the best value of affinity.

Algorithm CLONCLAS is based on algorithm CLONALG and has similar features. Description of training phase of CLONCLAS algorithm by pseudo code is introduced on fig. 6. Inputs are the same as for CLONALG algorithm.

```
[Ab_m] := Function ClonclasTrain (Ag,n_gen,n_Ab,n_m,
n_s,β,n_d)
Begin
  [Ab_m, Ab_r] = initClonalg(Ag,n_Ab,n_m);
  Ab = Ab_m ∪ Ab_r;
  For each ag_i∈Ag do
    For i := 1 to n_gen do
      af = affinity(ag_i,Ab);
      Ab_s = select(Ab,af,n_s);
      C = clone(Ab_s,β,af);
      C' = mutateClones(C,af);
      af' = affinity(ag_i,C');
      ab_c = select(C',af',1);
      Ab_m = insert(Ab_m,ab_c);
      Ab_r = copyPopulation(C');
      Ab_r = replace(Ab_r,n_d,af);
     End for;
  End for;
return Ab_m;
End;
```

Fig. 6 Pseudo code of training phase of CLONCLAS

In comparison with algorithm CLONALG, order of cycles is changed. In CLONALG algorithm, the whole set of antigens is iterated in each generation. In contrast to CLONCLAS algorithm where set of antigens is iterated only once (outer cycle), but each antigen $\mathbf{ag_i} \in Ag$ interact with set of antibodies Ab in given number of generations n_{gen}. The rest of both algorithms are the same.

Algorithm CLONCLAS contains additional function `copyPopulation`, which realize partial or total replacement of set Ab_r by antibodies from set C'. Whether the replacement will be partial or total depends on relative size of both sets:

- If the relation between sets is given by equation $|C'| > |Ab_r| = n_r$, the set Ab_r is totally replaced by n_r best antibodies from set C' according to value of similarity metric for current antigen $\mathbf{ag_i}$.
- If the relation between sets is given by equation $|C'| = |Ab_r|$, the set Ab_r is totally replaced by whole set C'.
- If the relation between sets is given by equation $|C'| = nc < |Ab_r|$, nc worst antibodies from the set Ab_r are replaced by antibodies from the set C' (according to value of similarity metric for current antigen $\mathbf{ag_i}$).

Classification phase is the same as in CLONALG algorithm.

5. EXPERIMENTAL SETUP

The dataset used for experiments in this paper was taken from [12]. This dataset covers data about 452 municipalities from the Czech Republic (micro-region Pardubice), table 1. Parameters obtained for each municipality fall into three categories; economic, debt, financial. This dataset was used in [13], [14] too.

Table 1 Municipal creditworthiness parameters design [12]

Type of parameters	Parameters
Economic	$x_1 = PO_r$, PO_r is population in the r-th year. $x_2 = PO_r/PO_{r-s}$ is population in the year r-s, and s is the selected time. $x_3 = U$, U is the unemployment rate in the municipality. $x_4 = \sum_{i=1}^{e}(EP_i/TEP)^2$, EP_i is the employed population of the municipality in the i-th economic sector, i=1,2,...,e, TEP is the total number of employed population, e is the number of the economic sector.
Debt	$x_5 = DS/PR$, $x_5 \in <0,1>$, DS is debt service, PR are periodical revenues. $x_6 = TD/PO$, TD is total debt. $x_7 = STD/TD$, $x_7 \in <0,1>$, STD is short term debt.
Financial	$x_8 = PR/CE$, $x_8 \in R^+$, CE are current expenditures. $x_9 = OR/TR$, $x_9 \in <0,1>$, OR are own revenues, TR are total revenues. $x_{10} = KE/TE$, $x_{10} \in <0,1>$, KE are capital expenditures, TE are total expenditures. $x_{11} = CR/TR$, $x_{11} \in <0,1>$, CR are capital revenues. $x_{12} = LA/PO$, [Czech Crowns], LA is the size of the municipal liquid assets.

All data were classified into seven categories according to creditworthiness.

Class1 – cover municipalities with high ability to meet its financial obligation, low debt and excellent budget implementation.

Class2 – this rating means that the municipality has very good ability to meet its financial obligation.

Class3 – municipalities with good ability to meet their financial obligation are included here.

Class4 – classification into this category consider municipalities with stable economy, good budget implementation but with medium debt.

Class5 – if municipality meets its financial obligation only under favourable economic conditions, it is ranked as class5.

Class6 – cover highly indebted municipalities that meet their financial obligations only with difficulty.

Class7 – inability of municipality to meet its financial obligations is characteristic for this class.

Final assignment of municipalities into classes is shown on the histogram (see fig. 7). It is obvious that most municipalities are covered by class 3 and 4.

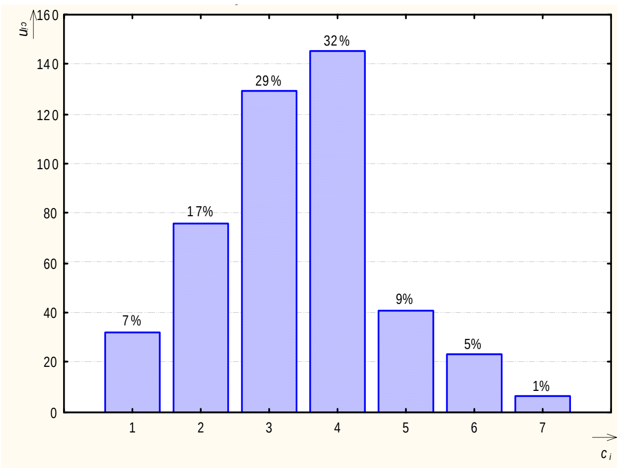

Fig. 7 Histogram of assignment of municipalities into classes

6. RESULTS

The dataset described in previous chapter was divided into two parts; training data and test data. A set of training data represents a group of antigens which represents input to learning phase for all algorithms. Elements of training set represent unknown antigen $\mathbf{ag_x}$ which represents input to the classification phase. Sets of antibodies in algorithm description represent inner representation of training set. Both datasets included the same distribution of patterns (municipalities) across all classes. In the training phase we used training data and on the basis of test data classification we were able to evaluate accuracy of each algorithm.

Each experiment was performed repeatedly with calculation of average accuracy Acc_{avg}, maximal accuracy Acc_{max} and standard deviation σ. In table 2 results for algorithms Immunos-1 and Immunos-2 that do not need any setup of parameters are introduced.

Table 2 Given results for algorithm Immunos-1 and Immunos-2

Algorithm	Acc_{avg}[%]	Acc_{max}[%]	σ
Immunos-1	74.31	74.31	0
Immunos-2	61.47	61.47	0

Although these algorithms do not need parameter setup, so their usage is easy, the accuracy is quite low. Hence the modelling of municipal creditworthiness by these algorithms cannot be recommended. Table 3 contains the best classification results for Immunos-99 algorithm. Given results were obtained using dynamical thresholding (value of parameter t = -1).

Table 3 Given results for algorithm Immunos-99

n_{gen}	n_{init}[%]	Acc_{avg}[%]	Acc_{max}[%]
40	90	**89.76**	**90.27**

Results of the experiments with algorithm CLONALG are given in table 4. Results obtained by algorithm CLONCLAS are introduced in table 5.

Table 4 Given results for algorithm CLONALG

n_{gen}	n_{Ab}	n_m	n_s	β	n_d	Acc_{avg}[%]	Acc_{max}[%]	σ
10	425	340	42	0.2	60	**88.41**	92.04	2.52
10	425	300	42	0.1	60	87.27	**92.92**	2.54

Table 5 Given results for algorithm CLONCLAS

n_{gen}	n_{Ab}	n_m	n_s	β	n_d	Acc_{avg} [%]	Acc_{max} [%]	σ
10	465	372	139	0.1	70	**89.20**	91.15	1.36
10	445	310	133	0.3	66	87.43	**93.81**	3.03

Within the scope of experiments, it is obvious that average accuracy of classification is better for Immunos-99, nevertheless algorithms CLONALG and CLONCLAS are much successful in maximal accuracy.

7. CONCLUSIONS

In this paper it was described the concept of municipal creditworthiness modelling by artificial immune systems. With regards to the character of particular algorithms, algorithms based on clonal selection and affinity maturation were chosen for this task. To be specific, algorithms CLONALG, CLONCLAS, and group of Immunos algorithms were used.

The difference between algorithms Immunos-1 and Immunos-2 is in representation of obtained knowledge (fig. 1, fig. 2). Algorithm Immunos-1 differs from algorithm Immunos-99 in the way of learning phase (fig. 1, fig. 4). The difference between algorithm group CLONALG and CLONCLAS and algorithm group Immunos is in usage of obtained knowledge (fig. 5, fig. 6).

The article contains detailed description of these algorithms. Results obtained in the course of municipal creditworthiness modelling indicate that basic algorithms Immunos-1 and Immunos-2 cannot be recommended for this task. These algorithms are simple and do not need any parameter setup, however obtained results are insufficient in comparison with the other algorithms. Algorithms Immunos-99, CLONALG, and CLONCLAS need additional parameter setup but their results of classification are much better. In comparison with all algorithms, algorithm Immunos-99 achieved the best average results of classification. Best maximal values were obtained by algorithm CLONCLAS. For the utilization in practice it is better to use algorithms with stabile results of classification than maximal values under the specific conditions. For this reason we can fully recommend algorithm Immunos-99 for municipal creditworthiness modelling. For the experiments Weka [27] and WEKA Classification Algorithms tool [5] were used.

ACKNOWLEDGEMENT

This work was supported by the scientific research of Czech Science Foundation, under Grant No: 402/08/0849 with title Model of Sustainable Regional Development Management.

REFERENCES

[1] BRABAZON, A.-O'NEILL, M. Credit Classification Using Grammatical Evolution. *Informatica*, no.30, 2006, pp.325-335.

[2] BRABAZON, A.-O'NEILL, M. Bond-Issuer Credit Rating with Grammatical Evolution. Applications of Evolutionary Computing. Lecture Notes in Computer Science, vol.3005, 2004, pp.270-279.

[3] BRENNAN, D.-BRABAZON, A. Corporate Bond Rating Using Neural Networks. *In Proc. of the International Conference on Artificial Intelligenc*e, IC-AI '04, Las Vegas, Nevada, USA, CSREA Press, vol.1, 2004, pp.161-167.

[4] BROWLEE, J. Immunos-81, The Misunderstood Artificial Immune System. Technical Report, no.3-01, Centre for Intelligent Systems and Complex Processes (CISCP), Faculty of Information and Communication Technologies, Swinburne University of Technology, Australia, 2005.

[5] BROWLEE, J. WEKA Classification Algorithms [computer program]. Ver.1.6. Sourceforge.NET, 2006 [citation 2008-09-09]. Available from <http://sourceforge.net/projects/wekaclassalgos>.

[6] BURNET, F. M. The Clonal Selection Theory of Immunity. Vanderbilt University Press, Nashville, TN, 1959.

[7] CARTER, J. H. The Immune System as a Model for Classification and Pattern Recognition. *Journal of the American Medical Informatics Association*, vol.7, no.1, 2000, pp.28-41.

[8] CASTRO, L. N.-TIMMIS, J. I. Artificial Immune Systems: A New Computational Intelligence Approach, London: Springer-Verlag, 2002.

[9] CASTRO, L. N.-ZUBEN, F. J. Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation*, Special Issue on Artificial Immune Systems. vol.6, no.3, 2002, pp.239-251.

[10] CASTRO, L. N.-ZUBEN, F. J. The Clonal Selection Algorithm with Engineering Applications. *In Proc. of the Genetic and Evolutionary Computation Conference*, Workshop on Artificial Immune Systems and Their Applications, USA, Las Vegas: Morgan Kaufman, 2000, pp.36-40.

[11] DASGUPTA, D. et. all. Artificial Immune Systems and Their Applications. Springer-Verlag, 1999.

[12] HÁJEK, P. Municipal Creditworthiness Modelling by Computational Intelligence Methods, Ph.D. Thesis, University of Pardubice, Pardubice, 2006.

[13] HÁJEK, P.-OLEJ, V. Hierarchical Structure of Fuzzy Inference Systems Design for Municipal Creditworthiness Modelling. *WSEAS Transaction on Systems and Control*, vol.2, no.2, 2007, pp.162-169.

[14] HÁJEK, P.-OLEJ, V. Municipal Creditworthiness Modelling by Kohonen's Self-Organizing Feature Maps and LVQ Neural Networks. *In Proc. of the 9th International Conference on Artificial Intelligence and Soft Computing*, ICAISC 08, Lecture Notes in Artificial Intelligence, Zakopane, Poland, Springer Berlin Heidelberg New York, 2008, pp.52-61.

[15] INGRAM, R. A Descriptive Analysis of Municipal Bond Price Data for Use in Accounting Research. *Journal of Accounting Research*, Autumn 1985, pp.595-618.

[16] JERNE, N. K. Towards a Network Theory of the Immune System. *Annals of Immunology*, vol.125, no.1, 1974, pp.373-389.

[17] KIM, J.-BENTLEY, P. The Artificial Immune Model for Network Intrusion Detection. *In Proc. of the 7-th European Conference on Intelligent Techniques and Soft Computing*, EUFIT'99, Aachen, Germany, 1999.

[18] LAMB, R.-RAPPAPORT, S. Municipal Bonds, New York: McGraw-Hill, 1987.

[19] LIU, X.-LIU, W. Credit Rating Analysis with AFS Fuzzy Logic. Advances in Natural Computation, Lecture Notes in Computer Science, Springer-Verlag, vol.3612, 2005, pp.1198-1204.

[20] MAGNI, C. A. Rating and Ranking Firms with Fuzzy Expert Systems: The Case of Camuzzi. MPRA Paper University Library of Munich, Germany, vol. 5889, 2007.

[21] PERELSON, A. S. Immune Network Theory. *Immunol. Rev.*, vol.110, 1989, pp.5-36.

[22] RAMAN, K. K. Assessing Credit Risk on Municipal Short-term Debt. *Advances in Accounting*, 1986, pp.171-180.

[23] ROMANIUK, S. G. Fuzzy Rule Extraction for Determining Creditworthiness of Credit Applicants. Technical report, No. TR20/92, National University of Singapore, 1992.

[24] WATKINS, A.-BOGGESS, L. A New Classifier Based On Resource Limited Artificial Immune Systems. *In Proc. of Congress on Evolutionary Computation*, Part of the 2002 IEEE World Congress on Computational Intelligence held in Honolulu, HI, USA, IEEE Computer Society Press, 2002, pp.1546-1551.

[25] WHITE, J.-GARRETT, S. M. Improved Pattern Recognition with Artificial Clonal Selection. *In Proc. Artificial Immune Systems: Second International Conference*, Edinburgh, UK, ICARIS-2003, Berlin: Springer Verlag, 2003, pp.181-193.

[26] WILSON, E.-HOWARD, T. Information for Municipal Bond Investment Decisions: Synthesis of

Prior Research, An Extension and Policy Implications. Research in Governmental and Non-Profit Accounting, 1985, pp.213-263.

[27] WITTEN, I. H.-FRANK, E. Data Mining: Practical Machine Learning Tools and Techniques. Second Edition. Morgan Kaufmann, 2005.

[28] ZHOU, X.Y. - ZHANG, D.F. - JIANG, Y. A New Credit Scoring Method Based on Rough Sets and Decision Tree. *Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, Springer-Verlag*, 2008, vol. 5012, pp.1081-1089, ISSN 0302-9743.

Received September 11, 2009, accepted January 26, 2010

BIOGRAPHIES

Luděk Kopáček (MSc. Ph.D.) was born on 11.3.1980. In 2004 he graduated (master degree) with distinction at the Faculty of Electrical Engineering at Czech Technical University in Prague. Since 2004 he is as Ph.D. student at doctoral study program Informatics in Public Administration at the Faculty of Economics and Administration at University of Pardubice. His thesis is focused on artificial immune system and creditworthiness modelling.

Vladimír Olej (prof. MSc. Ph.D.) was born in Poprad, Slovakia. Since 2002 he is working as a professor at the University of Pardubice. He has published a number of papers concerning fuzzy logic, neural networks, and genetic algorithms.