STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVII**, Number 2, 2012

A STUDY ON USING ASSOCIATION RULES FOR PREDICTING PROMOTER SEQUENCES

MARIA IULIANA BOCICOR

Abstract. The problem of promoter identification in DNA sequences is of major importance within bioinformatics. As the conditions for a region of DNA to function as a promoter are not known, machine learning based classification models are still developed to approach this problem. Relational association rules are a particular type of association rules and describe numerical orderings between attributes that commonly occur over a data set. This paper aims to investigate a classification model based on relational association rules mining for the problem of promoter sequences prediction. Some further extensions to this model are introduced and we provide a comparison of the original algorithm with its newly introduced versions.

1. Introduction

Association rule mining means searching attribute-value conditions that occur frequently together in a data set [4, 9]. Ordinal association rules [5] are a particular type of association rules, which specify ordinal relationships between record attributes that hold for a certain percentage of the records in a data set. However, in real world data sets, attributes with different domains and relationships between them, other than ordinal, exist and therefore ordinal association rules are not powerful enough to describe data regularities. Consequently, Serban et al. introduced in [8] *relational association rules* in order to be able to capture various kinds of relationships between record attributes.

The problem of predicting whether a DNA sequence contains or not a promoter region is an important problem within bioinformatics, mainly because determining the promoter region in the DNA is a significant step in the process

Received by the editors: May 5, 2012.

2010 *Mathematics Subject Classification.* 68P15, 68T05.

1998 *CR Categories and Descriptors.* I.2.6[**Computing Methodologies**]: Artificial Intelligence – *Learning*; H.2.8[**Information systems**]: Database Applications – *Data Mining*.

Key words and phrases. Bioinformatics, Promoter Sequences Prediction, Machine Learning, Association Rule Mining.

of detecting genes. This classification problem was already approached both in the biological and computer science literature and several machine learning methods have proven to be very suitable and efficient.

In this paper we aim to investigate some extensions to a relational association rules based classification model for the problem of promoter sequences prediction, model that we previously introduced in [2]. The modified model will be evaluated on the data set that was used in [2] and the obtained results will be analysed and interpreted.

The rest of the paper is organized as follows. The problem of promoter sequences classification, as well as an existing classification model based on relational association rules are introduced in Section 2. Section 3 presents two extensions to this model. Experimental evaluations, analysis and comparisons of the three algorithms are given Section 4. Conclusions and further work are outlined in Section 5.

2. Background

In this section we will briefly present the problem of promoter sequences prediction, then review some fundamental aspects related to the relational association rules based classifier that we previously introduced for solving this problem [2].

2.1. **Promoter Sequences Prediction.**

There are two processes that are involved in the synthesis of proteins from the DNA molecules. During the first process, called *transcription*, a single stranded RNA molecule, called messenger RNA is synthesized from one of the strands of DNA corresponding to a gene (a gene is a segment of the DNA that codes for a type of protein). This process begins with the binding of an enzyme called RNA polymerase to a certain location, that determines which of the two strands of DNA will be transcript and in which direction. This exact site is recognized by the RNA polymerase due to the existence of certain regions of DNA placed near the beginning of a gene, regions called *promoters*. The *promoter sequences prediction problem* refers to determining if a given DNA sequence contains or not a promoter region.

Because determining the promoter regions in the DNA is an important step in the process of detecting genes, the problem of promoter identification is of major importance within bioinformatics. As the conditions for a DNA sequence to function as a promoter are not known, machine learning methods are suitable to approach this problem because they can learn useful descriptions of concepts when given only instances - DNA sequences that are assumed to contain underlying but unknown patterns of base pairs [10].

2.2. **Promoter sequences Classifier using Relational Association Rules - *PCRAR*.** *Relational association rules* were introduced in [8] as an extension to association rules, in order to be able to discover various kinds of relations or correlations that exist between data in large data sets. Classical association rules discard any quantitative information that may exist between record attributes in data sets, but many times this type of information can give valuable insights into the problem at hand. Therefore, the extension of classical association rules towards ordinal and more general, relational association rules allows the uncovering of much stronger rules that consequently achieve superior data mining, or classification.

In [2], we introduced *PCRAR* - a supervised learning technique for the prediction of promoter sequences, based on relational association rules mining. We have started from the intuition that in the problem of deciding if a DNA sequence contains or not promoter regions, relationships between the nucleotides that form the DNA sequence [7] may be relevant.

The main idea of this classifier is the following. In a supervised learning scenario for predicting promoter sequences, two sets containing positive and negative instances are given. These sets will be used for training the classifier, which actually refers to discovering binary relational association rules between nucleotides in the given DNA sequences. We detect in the training data sets all the interesting binary relational association rules (rules between two attributes), with respect to the user-provided support and confidence thresholds. After the training is completed, when a new instance (DNA sequence) has to be classified (positive - if it contains a promoter region, or negative, otherwise), we reason as follows. Considering the binary rules discovered during training using the sets of positive and negative instances, the probability to assign the new instance to the positive class will be computed. If this probability is greater than or equal to 0.5, then the query instance will be classified as a positive instance, otherwise it will be classified as a negative instance. For more details about the relations that were used, how the data was preprocessed and the way in which the probabilities of assigning a new sequence to the positive or the negative class were computed, we refer the reader to [2].

3. Extensions of the *PCRAR*

This section aims to present two extensions we propose for the relational association rules based classifier introduced in [2] and used for promoter sequences prediction. The first method was developed in order to investigate how the confidence of the relational association rules discovered in the training data influences the accuracy of the classification task. The second refers to the length of the generated rules. As the classifier mentioned in Subsection

2.2 generated and used only binary rules (rules of length 2), we now aim to investigate the use of rules of any length.

3.1. **Confidence Based Probability Computation.** After the training phase of the classifier introduced in [2] ($PCRAR$) is completed, during the testing phase, two probabilities are computed for each new DNA sequence: P_+ - the probability that the sequence belongs to the positive class, i.e., it contains a promoter region and P_- - the probability that it belongs to the negative class, i.e., it does not contain a promoter. The way these probabilities are computed depends solely on the total number of generated association rules (positive and negative) and on the number of rules that the new sequence verifies or not, not taking into consideration the confidences of the rules.

We propose in this subsection a new way of computing the conditional probabilities for a new DNA sequence, which is based on the confidences of the generated relational association rules. It can be proven that the sum of the probabilities of the two possible outcomes (an instance to be classified as *positive* or *negative*) is 1.

We first introduce some notations that will be used in the following:

- S - a new DNA sequence, that must be classified as containing or not a promoter region.
- RAR_+/ RAR_- - the set of relational association rules having a minimum *support* and *confidence*, determined using the training data set containing the positive/negative instances.
- $RAR_+(S)/RAR_-(S)$ - a subset of RAR_+ ($RAR_+(S) \subseteq RAR_+$), respectively RAR_- ($RAR_-(S) \subseteq RAR_-$), containing the positive/negative rules that are verified in the sequence S. The set of relational association rules generated for the positive/negative instances, that are not verified in the sequence S will be denoted by $NRAR_+(S)$, respectively by $NRAR_-(S)$, where $NRAR_+(S) \subseteq RAR_+$ and $NRAR_-(S) \subseteq RAR_-$. It is obvious that $NRAR_+(S) \bigcup RAR_+(S) = RAR_+$ and that $NRAR_-(S) \bigcup RAR_-(S) = RAR_-$.
- $conf(\mathcal{R})$ - the confidence of an arbitrary relational association rule $\mathcal{R}$.

The steps we propose for computing the conditional probabilities are:

- Determine s_+ the total sum of the confidences of all the rules from the set RAR_+ and s_- the total sum of the confidences of all the rules from the set RAR_-:

$$s_+ = \sum_{\mathcal{R} \in RAR_+} conf(\mathcal{R}) \qquad s_- = \sum_{\mathcal{R} \in RAR_-} conf(\mathcal{R})$$

- Determine $s_+(S)$ the total sum of the confidences of the rules from $RAR_+(S)$ and $sn_+(S)$ the total sum of the confidences of the rules from $NRAR_+(S)$:

$$s_+(S) = \sum_{\mathcal{R} \in RAR_+(S)} conf(\mathcal{R}) \qquad sn_+(S) = \sum_{\mathcal{R} \in NRAR_+(S)} conf(\mathcal{R})$$

- Determine $s_-(S)$ the total sum of the confidences of the rules from $RAR_-(S)$ and $sn_-(S)$ the total sum of the confidences of the rules from $NRAR_-(S)$:

$$s_-(S) = \sum_{\mathcal{R} \in RAR_-(S)} conf(\mathcal{R}) \qquad sn_-(S) = \sum_{\mathcal{R} \in NRAR_-(S)} conf(\mathcal{R})$$

- Calculate the probability $P_+(S)$ to classify the instance S as a *positive* one as:

$$P_+(S) = \frac{1}{2}\left(\frac{s_+(S)}{s_+} + \frac{sn_-(S)}{s_-}\right) \tag{1}$$

The probability $P_-(S)$ to classify the instance S as a *negative* one could be computed in the same way, but it can be easily proven that the sum of the probabilities of the two possible outcomes (an instance to be classified as *positive* or *negative*) is 1. Therefore, if $P_+(S) \geq 0.5$ then the instance S will be classified as a *positive* instance, otherwise it will classified as a *negative* instance.

3.2. ***K*-length Rules Generation.** The *PCRAR* is based on an algorithm for the discovery of interesting ordinal association rules, called *DOAR* and introduced in [1]. This algorithm identifies ordinal association rules using an iterative process that consists in length-level generation of candidate rules, followed by the verification of the candidates for minimum support and confidence compliance.

In [2] we considered that a certain rule with a length greater than two is verified if all its binary subrules are verified and for computing the probability to classify a new instance as positive or negative we took into account only the number of verified/unverified rules. Therefore, for the *PCRAR* classifier, it was sufficient to generate only the binary interesting rules. This also led to a very fast training for this classifier. Here we propose, as another version of the algorithm, the generation of rules of any length k, the maximum length being, obviously, the number of attributes of an instance (for any instance S, let us denote its number of attributes $|S|$). A k-length rule is verified by an

instance if all its $k-1$ binary subrules are verified. As soon as the training phase is completed, meaning that all k-length rules ($k \in \{2, 3, \ldots, |S|\}$) have been generated, when a new DNA sequence must be classified, we compute the positive and negative probabilities for this sequence, as described in the previous subsection.

4. Experiments

In this section we provide experimental evaluations of the algorithms described in Section 3.

4.1. **Case study.** The data set we used to test the performances of the *PCRAR*-derived [2] algorithms is the one that was used in [2]. It is entitled "E. coli promoter gene sequences (DNA) with associated imperfect domain theory" and it was taken from the UCI Repository [3]. The data set is composed of 106 DNA sequences. Half of these represent positive instances, i.e. they contain promoter regions, while the other 53 are negative instances. As mentioned before, the relation definition and data pre-processing phases of the algorithm remain unchanged, only the training and testing phases being modified, as described in the Section 3.

The algorithms are compared by examining the classification accuracies and validation times and analysis for each one are made with different values for the confidence threshold: $\{0.4, 0.42, 0.45, 0.47, 0.48, 0.5, 0.52, 0.55, 0.6\}$. The minimum support threshold is fixed at 0.9. As described in [2], in order to decrease the number of considered attributes, we eliminate those attributes that have a very small correlation with the target output - whose correlation value is below a small positive threshold ϵ. To identify the optimal value of the threshold ϵ, a grid search method is applied, for each algorithm. The chosen values for ϵ are: $\{10^{-3}, 5 \cdot 10^{-3}, 10^{-2}, 5 \cdot 10^{-2}\}$. For each value of ϵ a cross-validation using a "leave-one-out" methodology is performed during the training phase of every algorithm, the best value of the threshold being indicated by the best accuracy (smaller error) obtained. We mention that the experiments were carried out on a PC with an Intel Core i5 Processor, at 2.53 GHz with 4 GB of RAM and that the validation time includes the computation time of the grid search procedure.

4.2. **Comparative results.** In the following the extensions of the *PCRAR* classifier [2] introduced in Section 3 will be referred to using the following abbreviations:

- *BRPC* - Binary Rules, with Probability computation based on the Confidence of the rules (Subsection 3.1)

- *KRPC* - K-length Rules, with Probability computation based on the Confidence of the rules (Subsection 3.2)

First observation we make is concerned to the number of generated relational association rules (for the entire positive and negative data sets), for each of the three algorithms. As both *PCRAR* [2] and *BRPC* determine binary rules, that only depend on the input data set, it is obvious that these two algorithms will always generate the same number of rules, for a certain value of the confidence threshold and of ϵ. However, the number of rules generated by *KRPC* will be significantly greater, as this algorithm determines rules of any length, starting from the set of generated binary rules. The maximum possible length for a k-length rule is $k = 57$ (the number of attributes in an instance), but the maximum length of rules that were actually generated was $k = 8$, for the confidence thresholds 0.4 and 0.42. Clearly, in all three cases, the number of rules increases as the confidence threshold decreases.

As reported in [2], the best result for *PCRAR*: **104** correctly classified instances, out of 106 instances (which means an error of 0.018867) was obtained for a confidence threshold of 0.4 and for $\epsilon = 10^{-2}$. The best result obtained by *BRPC* is the same: **104** out of 106, for the same minimum confidence, but it was obtained for $\epsilon = 10^{-3}$. *KRPC*, on the other hand, has proven a worse performance, as the best obtained result is **97** correctly classified instances, out of 106 (an error of 0.084905), for a confidence of 0.6 and with $\epsilon = 10^{-3}$. Figure 1 illustrates a comparison of the accuracies obtained by these three algorithms, for the minimum confidence 0.4 and for $\epsilon = 10^{-2}$ and $\epsilon = 10^{-3}$.

From Figure 1 we can observe that *BRPC* outperforms *PCRAR* [2], for $\epsilon = 10^{-3}$, which means that it is also important to consider the confidences of the rules in the classification process. Regarding the accuracies obtained by *KRPC*, we can observe that they are worse than those obtained by the other two algorithms. The higher classification error that appears in the case of *KRPC* may be due to the fact that the number of generated rules is significantly higher than in the other cases and consequently even if an instance verifies a certain number of stronger rules (with a high confidence), there may still remain a very large number of unverified rules (whose confidences, even if smaller, when summed up, could far exceed the sum of the confidences of the verified rules). Another thing we need to mention about the *KRPC* is that the accuracies it obtains increase with the confidence threshold. This is normal, because as the confidence threshold increases, the number of generated rules decreases and therefore the problem that we mentioned above is less likely to appear.

As the training time is also a relevant feature for comparing different classifiers, we will now refer to the running times of the algorithms. It is important

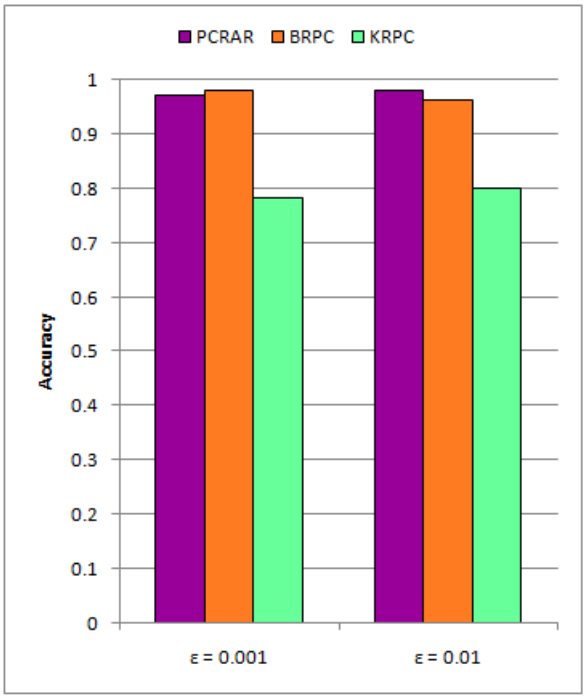

FIGURE 1. Comparative results: confidence threshold = 0.4

to know that these are actually validation times, i.e. overall times in which each version of the *PCRAR* classifier [2] performs the validation (this including the training time). We mention that both the algorithms that only consider binary rules have very low computational times, while the one that generates rules of any length clearly runs much slower. *PCRAR* [2] and *BRPC* have similar running times for all tested confidence thresholds: except for the minimum confidence 0.45, where they differ with approximately 3 seconds, for all the other values of the confidence thresholds, these two algorithms have at most 1.5 seconds difference. Figure 2 comparatively illustrates the running times for *PCRAR* [2] and *BRPC*. Although we cannot say that *PCRAR*'s [2] confidence-time function is monotonic (more specifically, decreasing), we may observe a decreasing tendency. On the other hand, the confidence-time function for *BRPC* is strictly decreasing. Concerning *KRPC*, its running times are significantly higher and, as expected, they decrease as the confidence threshold increases. The minimum validation time for this algorithm, for the confidence threshold of 0.6 is more than three times higher than that of the other two

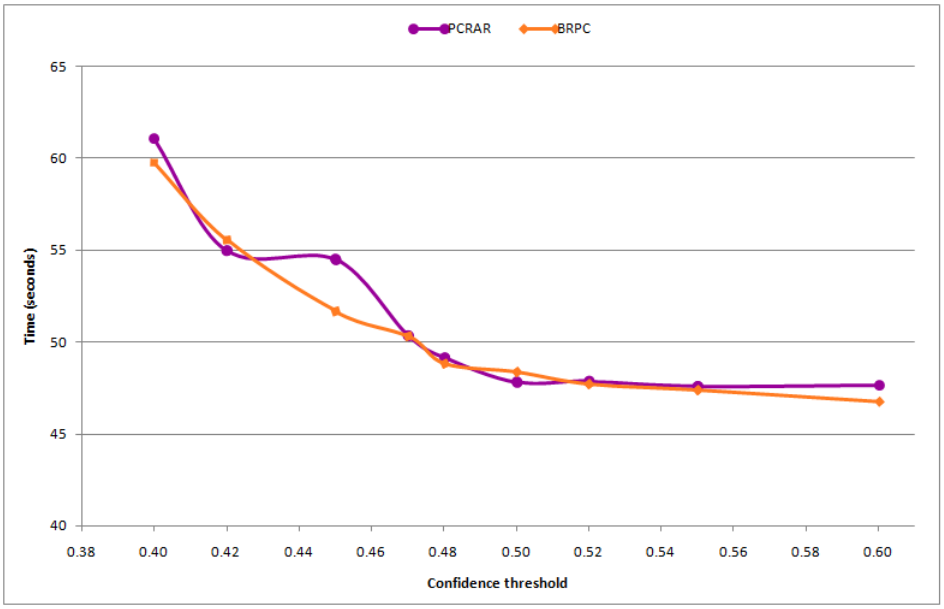

FIGURE 2. Comparative running times: *PCRAR* [2] and *BRPC*

algorithms. The confidence-time function that is generated for *KRPC* is exponential, as shown in Figure 3. In this figure, the Y axis represents hours, not seconds, as in Figure 2.

4.3. **Discussion.** We have experimented with three relational association rules based algorithms in order to obtain results for the promoter prediction problem. The original model - *PCRAR* [2] generates binary relational association rules in the training phase and then in the testing phase uses the number of rules that are verified by the new instance in order to classify it. The first extension that we introduced also generates binary relational association rules in the training phase, but then uses the confidences of the rules in order to classify a new instance. Finally, the second extension generates rules of any length, and uses the same method of classification as the first extension.

The obtained results demonstrate that the algorithms that generate and use only binary relational association rules perform better than the one generating rules of any length, both in terms of classification accuracy and validation times. This leads us to the conlusion that, for the considered problem, binary rules are sufficiently relevant in order to obtain a good classification of a DNA sequence as a *promoter* or a *non-promoter*.

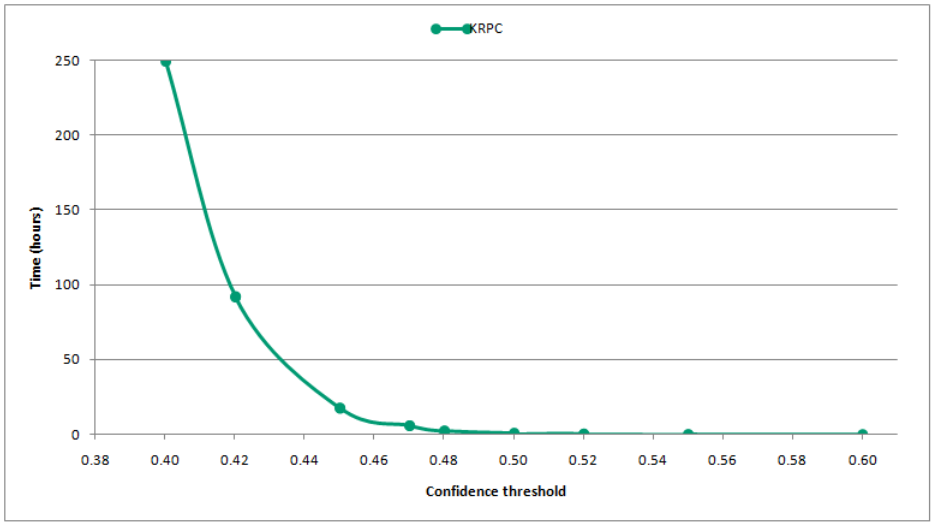

FIGURE 3. Running time for *KRPC*

5. Conclusions and Further Work

In the present study we introduced two extensions to a relational association rules based classification model for the problem of promoter sequences prediction and we experimentally evaluated and compared the three algorithms.

The algorithms were tested on a data set containing 106 *E. coli* DNA sequences [3], among which 53 contained promoter regions (positive instances) and 53 did not (negative instances). The tests showed that the two algorithms that generate only binary rules obtain very good performances, *better* than those reported by all other classifiers already applied in the literature for promoter sequences recognition and they need very low training times (less than two minutes). However, the third algorithm proved to obtain worse accuracies, when compared to the other two relational association rules classifiers, but still, when compared to the other classifiers already applied in the literature (see [2]), it is the *third best*. The drawback of this last algorithm is that its training times are very high.

Further work will be made in order to improve the accuracy of the relational association rules based classifiers by using supervised learning to identify the most appropriate values for the confidence threshold and for the attribute elimination threshold ϵ, i.e. the values that minimize the classification error as well as the execution time. We will also focus on hybridizing this classification

model, by combining it with other machine learning based predictive models [6].

ACKNOWLEDGEMENT

I thank my thesis advisor, Professor Gabriela Czibula for the ideas and suggestions for this paper. This work was possible with the financial support of the Sectoral Operational Programme for Human Resources Development 2007-2013, co-financed by the European Social Fund, under the project number POSDRU/107/1.5/S/76841 with the title Modern Doctoral Studies: Internationalization and Interdisciplinarity.

References

[1] Câmpan, A., Serban, G., Truta, T. M., Marcus, A., *An Algorithm for the Discovery of Arbitrary Length Ordinal Association Rules*, In DMIN'06, The 2006 International Conference on Data Mining, Las Vegas, USA, 2006, pp. 107–113.

[2] Czibula, G., Bocicor, M. I., Czibula, I. G., *Promoter Sequences Prediction Using Relational Association Rule Mining*, Evolutionary Bioinformatics **8**, 2012, pp. 181-196.

[3] Frank, A., Asuncion, A., *UCI Machine Learning Repository*, University of California, Irvine, School of Information and Computer Sciences, 2010, URL:http://archive.ics.uci.edu/ml.

[4] Han, J., *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers Inc., 2005, San Francisco, CA, USA.

[5] Marcus, A., Maletic, J. I., Lin, K. I., *Ordinal association rules for error identification in data sets*, Proceedings of the tenth international conference on Information and knowledge management, CIKM '01, ACM, 2001, pp. 589–591.

[6] Mitchell, T. M., *Machine Learning*, New York: McGraw-Hill, 1997.

[7] Saenger, W., *Principles of Nucleic Acid Structure*, Springer-Verlag, 1984.

[8] Serban, G., Câmpan, A., Czibula, I. G., *A Programming Interface for Finding Relational Association Rules*, International Journal of Computers, Communications and Control **I/2006**, Proceedings of the International Conference on Computers, Communications and Control, ICCCC 2006, Oradea, 2006, pp. 934-944.

[9] Tan, P. N., Steinbach, M., Kumar, V., *Introduction to Data Mining, (First Edition)*, Addison-Wesley Longman Publishing Co. Inc., 2005, Boston, MA, USA.

[10] Tung, N.T., Yang, E., Androulakis, I.P, *Machine Learning Approaches in Promoter Sequence Analysis*, In Machine Learning Research Progress, Nova Science Publishers, Inc, 2008.

Babeş-Bolyai University, Department of Computer Science, 1, M. Kogălniceanu street, 400084 Cluj-Napoca, Romania

E-mail address: iuliana@cs.ubbcluj.ro