Experimental investigation of classification algorithms for ITS dataset

J.L. Yearwood[1], B.H. Kang[2], A.V. Kelarev[1]

[1]School of Information Technology and
Mathematical Sciences, University of Ballarat
P.O. Box 663, Ballarat, Victoria 3353, Australia
{*j.yearwood, a.kelarev*}*@ballarat.edu.au*

[2]School of Computing and Information Systems,
University of Tasmania, Private Bag 100,
Hobart, Tasmania 7001, Australia
BHKang@utas.edu.au

Abstract. This article is devoted to experimental investigation of classification algorithms for analysis of ITS dataset. We introduce and consider a novel k-committees algorithm for classification and compare it with the discrete k-means and Nearest Neighbour algorithms. The ITS dataset consists of nuclear ribosomal DNA sequences, where rather sophisticated alignment scores have to be used as a measure of distance. These scores do not form a Minkowski metric and the sequences cannot be regarded as points in a finite dimensional space. This is why it is necessary to develop novel algorithms and adjust familiar ones. We present the results of experiments comparing the efficiency of three classification methods in their ability to achieve agreement with classes published in the biological literature before. It turns out that our algorithms are efficient and can be used to obtain biologically significant classifications. A simplified version of a synthetic dataset, where the k-committees classifier outperforms k-means and Nearest Neighbour classifiers, is also presented.

1 Introduction and motivation

Classification of data is very important in machine learning, knowledge acquisition and data mining. It has numerous applications in broad areas. Many valuable results on this topic have been obtained in the literature recently (see, for example, [1–3, 5–8, 10, 12, 13, 15, 16, 20, 22–26, 29, 32, 33, 39, 38]).

Datasets of DNA, RNA and protein sequences are growing and becoming a huge and under-utilised resource in view of the rapid expansion of the whole genome sequencing and advancements of sequencing technology. Enormous amounts of nucleotide data are continuously being generated. This is why it is especially important to devise efficient classification algorithms in order to automate the analysis of nucleotide and protein sequences. To verify the efficiency of new methods for automated classification, the researchers have to use

classes and groupings that have already been considered in the biological literature. This allows them to compare results produced by new classifiers and known groupings in order to automate further classifications and create programs that lead to discoveries of biological significance.

The present paper investigates and compares the efficiency of three algorithms for supervised classification designed for an ITS dataset derived from the internal transcribed spacer regions of the nuclear ribosomal DNA in Eucalyptus as explained in [34]. We compare our classifications with known biologically significant classes published in [34]. We use general approaches to classification formulated in the language of optimization theory in [3]. This language has turned out to be particularly convenient and amenable to application in the setting of DNA sequences. Here we present the results of experimental analyses of the performance of three algorithms for supervised classification of the dataset, and compare the results with classifications published in [34]. Earlier, two different algorithms for unsupervised clustering of a set of nuclear ribosomal DNA sequences were considered in [21].

Long DNA sequences cannot be regarded as points in a finite dimensional space. Besides, rather sophisticated and biologically significant local alignment scores have to be used as a measure of similarity or distance between DNA sequences. These scores do not possess properties of the standard Euclidean norm in a finite dimensional space. Moreover, they do not satisfy axioms of the more general Minkowski metrics, which include as special cases the well known Euclidean distance, Manhattan distance, and max distance. These circumstances make it impossible to utilise previous implementations of algorithms. One has to develop novel algorithms and adjust familiar ones.

In order to achieve strong agreement between classifications produced by these machine learning algorithms and biological classifications, one must use measures of strong similarity between sequences which are biologically significant. We are using local alignment scores to develop novel classification algorithms and investigate their accuracy for a set of nuclear ribosomal DNA sequences. Our algorithms are using strong similarity measures based on local alignments, which have not been applied in this context before, and turn out to be highly efficient for DNA sequences of this kind.

Although datasets of DNA sequences cannot be regarded as sets of points in a finite dimensional space, typically the datasets used for classification are fairly small because of the relatively high cost of determining each sequence. It is possible to compare the task of classifying a set of DNA sequences with a number of optimization problems in graph theory. In particular, it can be formulated as a classification problem for the set of vertices of a finite connected undirected weighted graph. Notice that an alternative model for classifying DNA code based on analogy with neural networks and FSA was introduced in [16], see also [18] and [19]. Some steps of our algorithms are analogous to the steps which occur in solutions to the minimum dominating set of vertices problem in graph theory. We refer to [10], [27] and [36] for associated graph algorithms and graph-based data mining.

2 Local alignment scores for classification algorithms

This paper investigates and compares the efficiency of three algorithms for supervised classification: discrete k-means, Nearest Neighbour, and k-committees classifiers. For preliminaries on nucleotide and protein sequences we refer to [4], [9], [11], [14] and [31].

A classification of any given set of DNA sequences is a partition of these sequences into several classes. The problem is to construct a classifier via supervised learning and then use it to determine class membership of new sequences. The initial partition is usually communicated by a supervisor to a machine learning process that constructs the classifier. We use standard terminology and notation concerning general background information on classification and data mining, which can be found, for example, in [30], [36], [37] and [39].

The novel character of our algorithms first of all comes from using local alignment scores well known in bioinformatics. Every alignment algorithm produces an alignment score which measures the similarity of the nucleotide or amino acid sequences. This score is then used to evaluate optimal local similarity between the sequences. These scores do not satisfy the axioms of Minkowski metrics, which include as special cases the standard Euclidean distance used in previous implementations, Manhattan distance, and max distance.

Long nucleotide sequences cannot be regarded as points in a finite dimensional space. Besides, it is impossible to calculate new sequences from the given ones. In particular, one cannot compute the arithmetical average, or mean, of several given sequences. Hence our methods are different from those considered before.

The alignment scores in our algorithms provide a measure of similarity that is significant biologically. To illustrate let us suppose that we have a long DNA sequence L, and an identical copy S of a segment within the sequence L. Obviously, every correct biological classification should place both L and S in one and the same class. This may however be difficult to determine using other metrics. Indeed, L and S may have seriously different values of statistical parameters. Therefore traditional statistical approaches, mapping L and S into an n-dimensional space and using standard Euclidean norm there, may not notice their similarity at all. In contrast, sequence alignment will immediately show that there is a perfect match between S and a segment in the sequence L.

Our experiments used a dataset with sequences of a region of the nuclear ribosomal DNA (nrDNA) that is often used to work out evolutionary relationships between species and genera. The dataset includes many of different species from all subgenera and sections of Eucalyptus, as well as some other genera that are closely related to Eucalyptus. For a detailed description of the dataset we refer to [34].

In this dataset we looked at the following groupings, based on phylogeny represented in [34] (in the Figures 2, 3 and 5 of [34]):

1. Stockwellia, Eucalyptopsis and Allosyncarpia (2 accessions);
2. Angophora, Corymbia;

3. Subgenera Eudesmia, Cuboidea, Idiogenes, Primitiva, Eucalyptus;
4. Subgenera Alveolatae, Cruciformes, Symphyomyrtus, Minutifructus;
5. Subgenus Acerosae (one species, two accessions — E. curtisii).

The dataset also includes one taxon, Arillastrum, which does not belong to any of the groups. This was used as an "anchor" of the phylogenetic tree in [34].

We used the BLOSUM, block substitution matrices or blocks of amino acid substitution matrices, since they "encourage local alignment algorithms to produce alignments highlighting biologically important similarities" (see [11]). Usually, higher numbered BLOSUM matrices are used for aligning sequences of closely related species, and lower number matrices are used for more distant sequences.

Alignment scores have properties that are seriously different from those of the Euclidean norms and their simple modifications discussed, for example, by Witten and Frank (2005), see Section 6.4. Hence our algorithms had to be designed differently and have been encoded with the Bioinformatics Toolbox of Matlab. We used the

$$\texttt{swalign}(\texttt{Seq1},\texttt{Seq2})$$

function of the Bioinformatics Toolbox, which returns the optimal local alignment score. Higher alignment scores correspond to lower distances between closely associated sequences.

Thus, our paper deals with novel classification algorithms for DNA sequences based on alignment scores and suitable for analysis of highly variable regions of DNA.

3 The discrete *k*-means classifier based on alignment scores

The k-means algorithm and the nearest neighbour algorithms are classification techniques used most often and are implemented in the WEKA environment. Complete explanations of these methods are given, for example, by Witten and Frank [37], see Chapter 4.

It is explained in [37] that when the traditional k-means algorithm is used for classification, it simply computes the centroids of classes in the training set, and then uses the centroids to classify new elements as they become available. Traditional k-means algorithm uses standard Euclidean distances and computes the arithmetical average, or mean, of the points in each class. It is explained in Section 4.8 of [37] on p. 137 that the task of finding the centroid of a class C in the k-means algorithm is equivalent to finding a solution x to the following optimization problem:

$$\text{minimize} \sum_{y \in C} ||x - y||^2 \quad \text{subject to } x \in \mathbb{R}^n \tag{1}$$

In other words, the centroid of a class C is the point of the n-dimensional space $\mathbb{R}^n$ with the minimum sum of squares of distances to all known points c in the

class C. Every new point in the n-dimensional Euclidean space is then assigned to the class represented by its nearest centroid. The running time complexity of the k-means algorithm is $O(k)$.

Our first algorithm is a natural modification of the k-means classifier. Instead of trying to represent DNA sequences as some points in a Euclidean space, the new algorithm uses alignment scores to establish similarity between sequences. The easiest way to view these sequences is then to regard each sequence in the dataset as a vertex of a weighted undirected graph. In this case it is impossible to compute the arithmetical mean of a set of sequences in a class as it is done in the traditional k-means algorithm. For the alignment scores there does not exist a simple arithmetical calculation computing a DNA sequence that is the "midpoint" or "mean" of the given DNA sequences in order to use it as a new centroid.

This motivates substantial adjustments to condition (1). In this paper we used a modification of the k-medoids algorithm, explained for example in [17]. First, we had to choose centroids in the same given set of the DNA sequences, the way it is done in the k-medoids algorithm. Second, the squares of the alignment scores do not make geometrical sense, and we have modified them using the alternative formulation from Section 2.2 of [3]. In order to emphasize this property of our classification algorithm, here we call it a *discrete k-means classifier*.

Our algorithm operates on the set of given sequences only and does not create any new sequences as means of the given ones. As a centroid of the class C our algorithm uses a solution x to the following optimization problem:

$$\text{minimize} \quad \left(\max_{y \in C} ||x - y|| \right) \quad \text{subject to} \;\; x \in C. \tag{2}$$

A solution to this problem is a sequence with minimum largest distance to all other sequences in the class. We can think of this approach as a way of approximating the class by a sphere centred at the centroid. Then the optimization problem minimizes the radius of the sphere. After the centroids have been found, each new sequence is then assigned to the class of its nearest centroid. The running time of this algorithm is $O(k)$.

Every alignment score between each pair of the given sequences is found once during a pre-processing stage of the algorithm, and then these scores are looked up in a table during the search for centroids. The average success rates of this method for classifying new sequences in comparison with the classes obtained and published in [34] are represented in Figure 1.

4 Nearest neighbour classifier based on alignment scores

The second algorithm we implemented is an analogue of the nearest neighbour classification algorithm, see [37], Chapter 4. The standard nearest neighbour classifier implemented in WEKA could not be applied directly to the dataset of nuclear ribosomal DNA, because it handles data represented as points in an n-dimensional Euclidean space. Thus we had to encode a new version of

the nearest neighbour algorithm based on optimal local alignments of the given sequences.

The situation in this case is much simpler compared to the case of the k-means algorithm, and all modifications for the case of the nearest neighbour classifier are straightforward. We have found the average success rates of this method comparing classes produced by our algorithm for various alignment scores with the five classes obtained in [34]. The results on the accuracy of this algorithm are presented in Figure 1. As we see, the nearest neighbour algorithm turns out to be substantially more accurate.

The nearest neighbour algorithm compares each new sequence with all previous sequences, and assigns it to the class of the nearest known sequence using the alignment scores. The running time of this algorithm is $O(n)$, where n is the number of all sequences in the dataset. Since $n > k$, we see that the process of applying the nearest neighbour algorithm to classify new sequences is slower.

5 The k-committees classifier based on alignment scores

This section is devoted to a novel k-committees algorithm. The idea behind this algorithm is natural. However, we have not found it in the literature. In particular, this algorithm is not discussed in the monograph by Witten and Frank [37]. Thus, we are developing this algorithm independently as a new one. Our description of the algorithm originates from the general formulation of classification methods in the language of optimization theory given in [3].

Instead of using a single centroid to represent each class, we select a few representatives in each class. These representatives form a *committee* of the class. Let us denote the number of the representatives chosen in each class by r. When the training stage is complete, during the classification stage every new sequence is then assigned to the class of its nearest committee member. If every class has the same number r of committee members and it is desirable to indicate this number explicitly, then we call our method the (k, r)-*committees* algorithm, or the k-*committees of* r *representatives* algorithm.

Average success rates of classification algorithms						
	discrete k-means classifier	k-committees classifier				Nearest Neighbour classifier
		$r = 2$	$r = 3$	$r = 4$	$r = 5$	
Blossum30	64.75	70.69	74.66	78.00	78.98	85.26
Blossum40	72.87	76.04	77.85	79.05	80.70	85.11
Blossum50	76.80	79.07	79.70	80.26	81.12	85.37
Blossum60	71.87	75.50	77.07	79.52	80.39	86.94
Blossum70	70.41	74.82	77.53	79.06	79.11	86.22
Blossum80	76.86	77.99	78.75	80.96	82.74	86.97
Blossum90	76.64	77.98	78.64	79.78	81.77	85.94
Blossum100	72.78	76.10	77.71	79.31	81.02	86.90

Fig. 1. Tenfold cross validation

It is explained in Section 2.2 of [3] that the k-means algorithms achieves high accuracy in situations where every class can be represented by one centroid. If approximations like this are not accurate, then higher success rates can be achieved by using several representatives.

The set of representatives selected in a class will be called a *committee* of the class. As a committee of r representatives of the class C in our algorithm uses the points $x_1, \ldots, x_r$ defined by

$$\text{minimize} \quad \max_{y \in C} \left(\min_{i=1,\ldots,r} ||x_i - y|| \right) \qquad \text{subject to} \quad x_1, \ldots, x_r \in C, \qquad (3)$$

i.e., the set X of r points from the finite set C such that the largest distance from any point y in C to the set X achieves a minimum.

Intuitively speaking, this means that the k-committees algorithm approximates every class by a union of 'spheres', i.e., sets of points with given members of the committee as their centroids. When the committees have been prepared, the algorithm assigns every new sequence to the class of its nearest committee member. The running time of this algorithm is $O(kr)$.

We used the standard tenfold cross validation, explained in [37], Section 5.3, and investigated the efficiency of the classes produced by our algorithms for

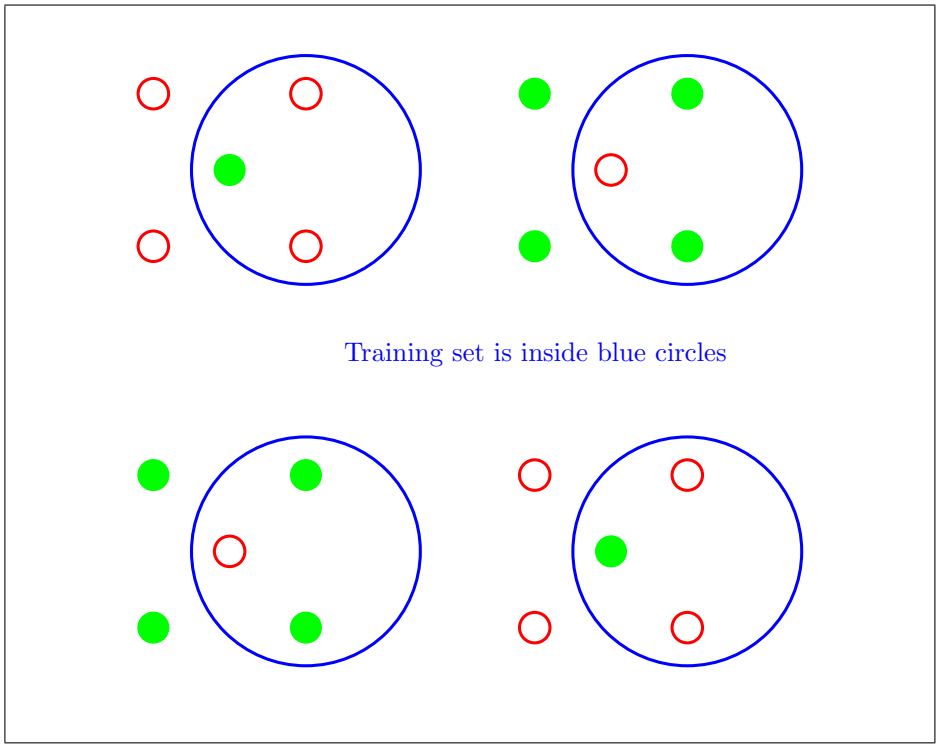

Fig. 2. Example where k-committees classifier outperforms k-means and NN.

classifying new sequences that have not been included in the training set. With regard to the ITS dataset, the k-committees algorithm plays an intermediate role. To illustrate experimental results on performance of the k-committees classifiers we have included Figure 1 using tenfold cross validation.

Notice that in some cases the k-committees algorithm can have higher success rates than both the Nearest Neighbour and the discrete k-means algorithms. We have generated a special synthetic "noisy" dataset, where the k-means algorithm turns out more accurate than the Nearest Neighbour, and the k-committees method happens to be the most accurate one. The idea behind this example is illustrated in Figure 2 prepared in Python with PyX package. There are two classes in the dataset of this small illustrating diagram. They are represented by small red circles and small green disks, respectively. The success rates of the algorithms we have looked at are equal to 0% for Nearest Neighbour, 50% for k-means, and 75% for the k-committees algorithm with $r = 2$.

Thus, our algorithms can be used for classifying new sequences not considered before as they become available. The nearest neighbour classification algorithm is more accurate but requires aligning each new sequence with every known ones and is substantially slower than the k-means algorithm.

6 Conclusions

The nearest neighbour, k-means, and k-committees classification algorithms based on alignment scores are suitable for practical analysis of datasets of this type, and have sufficiently high success rates.

Our algorithms use highly biologically significant local alignment scores as an indication of distance between sequences and achieve relatively high level of accuracy when compared with known biological classifications. The experimental results we have obtained demonstrate that the success rates of our algorithms are significantly higher than those of analogous methods using traditional Euclidean norms and presented, for example, on the WEKA web site (see [35]).

For ITS dataset the nearest neighbour classification algorithm with alignment scores has turned out to be more accurate but much slower than the k-means algorithm with these scores. For the ITS dataset the k-committees algorithm turns out to be intermediate in accuracy. Hence it can be used in situations where it is possible to spend CPU time on pre-processing data and prepare the committees of representatives in advance so that future classification questions are answered more quickly than by the Nearest Neighbour classifier, and more accurately than by the k-means classifier. On the other hand, we have shown that there exist 'noisy' datasets where the k-committees classifier outperforms both the discrete k-means method and the Nearest Neighbour classifier.

Our k-committees method concentrates on careful selection of a very small number of members for each committee. This is why it should be regarded as a modification of the k-means algorithm. In contrast, various known modifications of the Nearest Neighbour algorithm focus on "pruning", or "editing", or "condensing" the whole large class by removing "noise" or "outliers", and

then deal with whole large sets of prototypes. Although it might be possible to imagine that the k-means algorithm is just a Nearest Neighbour algorithm where all classes have been "condensed" to just a single point, it is standard in the literature to regard the k-means algorithm as being different from the Nearest Neighbour algorithm and its close relatives, because of completely different approaches and complexities which occur in the pruning of a large set as compared to a careful selection of just one centroid. The same holds true of selecting a very few points as representatives. Thus, it is more accurate to regard our k-committees algorithm and other modifications of the k-means algorithm as being different from other versions derived from the Nearest Neighbour algorithm. In order to explain the difference for students in an undergraduate unit a lecturer could compare the k-means and k-committees algorithms with elections of the presidents and state governments, and the Nearest Neighbour algorithm and its refinements with polls, statistical surveys and sampling processes.

It is nice that our experimental results are in good agreement with theoretical expectations and could be viewed as a sign confirming the strength and validity of data mining theory.

7 Acknowledgements

Preliminary steps of our research have been initiated during joint work on IRGS grant K14313 of the University of Tasmania. The first author was supported by Queen Elizabeth II Fellowship and Discovery grant DP0211866 from Australian Research Council. The second author has been supported by several grants from Asian Office of Aerospace Research and Development. The third author was supported by Discovery grant DP0449469 from Australian Research Council and a RIBG grant of the University of Ballarat.

References

1. Bagirov, A.M. and Yearwood, J.L.: A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems, European J. Operational Research 170 (2006), 578-596.
2. Bagirov, A.M., Rubinov, A.M. and Yearwood, J.: A global optimization approach to classification, Optim. Eng. 3 (2002), 129–155.
3. Bagirov, A.M., Rubinov, A.M., Soukhoroukova, N.V. and Yearwood, J.: Unsupervised and supervised data classification via nonsmooth and global optimization, Top 11 (2003), 1–93.
4. Baldi, P. and Brunak, S.: "Bioinformatics: The Machine Learning Approach", Cambridge, Mass, MIT Press, 2001.
5. Cho, W.C. and Richards, D.: BayesTH-MCRDR algorithm for automatic classification of web documents, Proc. 17th Australian Joint Conf. on Artificial Intelligence, AI'2004, December 6-10, 2004, Cairns, Australia.
6. Cho, W.C. and Richards, D.: Automatic construction of a concept hierarchy to assist Web document classification, Proc. 2nd Internat. Conf. on Information Management and Business, IMB2006, 13-16 February, 2006, Sydney, Australia.

7. Compton, P., Hoffmann, A., Motoda, H. and Yamaguchi, T.: PKAW2000 - Proc. of the 6th Pacific Rim Knowledge Acquisition Workshop, 2000.
8. Dazeley, R., Kelarev, A.V., Yearwood, J.L. and Mammadov, M.A.: Optimization of multiple classifiers in data mining based on string rewriting systems, in preparation.
9. Durbin, R., Eddy, S.R., Krogh, A. and Mitchison, G.: "Biological Sequence Analysis", Cambridge University Press, 1999.
10. Geamsakul, W., Yoshida, T., Ohara, K., Motoda, H., Yokoi H. and Takabayashi, K.: Constructing a decision tree for graph-structured data and its applications, Fundamenta Informaticae, 66 (2005)(1-2), 131-160.
11. Gusfield, D.: "Algorithms on Strings, Trees, and Sequences", Computer Science and Computational Biology, Cambridge University Press, Cambridge, 1997.
12. Hofmann, A., Kang, B.H., Richards, D. and Tsumoto, S.: "Advances in Knowledge Acquisition and Management", Proceedings of AKAM2006, Guilin, China, 2006.
13. Hoffmann, A., Motoda, H. and Scheffer, T.: "Discovery Science", Lecture Notes in Artificial Intelligence 3735, Springer, 2005.
14. Jones, N.C. and Pevzner, P.A.: "An Introduction to Bioinformatics Algorithms", Cambridge, Mass, MIT Press, 2004. `http://www.bioalgorithms.info/`
15. Kang, B.H.: PKAW2004 - "Pacific Knowledge Acquisition Workshop", Auckland, New Zealand, 2004.
16. Kang, B.H., Kelarev, A.V., Sale, A.H.J. and Williams, R.N.: A new model for classifying DNA code inspired by neural networks and FSA, Pacific Knowledge Acquisition Workshop, PKAW2006, Lect. Notes Computer Science 4303 (2006), 187–198.
17. Kaufman, L. and Rousseeuw, P.J.: "Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley & Sons, New York, 1990.
18. Kelarev, A.V.: "Ring Constructions and Applications", World Scientific, River Edge, 2002.
19. Kelarev, A.V.: "Graph Algebras and Automata", Marcel Dekker, New York, 2003.
20. Kelarev, A.V., Kang, B.H., Sale, A.H.J. and Williams, R.N.: Labeled directed graphs and FSA as classifiers of strings, 17th Australasian Workshop on Combinatorial Algorithms, AWOCA 2006, 12–16 July 2006, Uluru (Ayres Rock), Northern Territory, Australia, 93–109.
21. Kelarev, A., Kang, B. and Steane, D.: Clustering algorithms for ITS sequence data with alignment metrics, Advances in Artificial Intelligence, 19th Australian Joint Conference on Artificial Intelligence, AI06, Lect. Notes Artificial Intelligence 4304 (2006), 1027–1031.
22. Kelarev, A.V., Ryan, J., Yearwood, J.L.: Cayley graphs as classifiers for data mining: the influence of asymmetries, Discrete Mathematics, accepted for publication.
23. Kelarev, A.V., Ryan, J., Yearwood, J.L.: A combinatorial algorithm for the optimization of multiple classifiers in data mining based on graphs, J. Comb. Math. & Comb. Computing, accepted.
24. Kelarev, A.V., Yearwood, J.L. and Mammadov, M.A.: A formula for multiple classifiers in data mining based on Brandt semigroups, Semigroup Forum, to appear soon.
25. Kelarev, A.V., Yearwood, J.L. and Vamplew, P.W.: A polynomial ring construction for classification of data, Bulletin Austral. Math. Soc., accepted.
26. Kelarev, A.V., Yearwood, J.L. and Watters, P.: Rees matrix constructions for clustering of data, submitted.
27. Lau, H.T.: "A Java Library of Graph Algorithms and Optimization", CRC Press, 2008.

28. Lee, K., Kay, J. and Kang, B.H.: KAN and RinSCut: lazy linear classifier and rank-in-score threshold in similarity-based text categorization, Proc. ICML-2002 Workshop on Text Learning, University of New South Wales, Sydney, Australia , 2002, 36–43.
29. Lee, K.H. and Kang, B.H.: A new framework for uncertainty sampling: exploiting uncertain and positive-certain examples in similarity-based text classification, Proc. Internat. Conf. on Information Technology: Coding and Computing, ITCC2004, Las Vegas, Nevada, 2004, 12pp.
30. Luger, G.F.: "Artificial Intelligence. Structures and Strategies for Complex Problem Solving", Addison-Wesley, 2005.
31. Mount, D.: "Bioinformatics: Sequence and Genome Analysis". Cold Spring Harbor Laboratory, 2001. `http://www.bioinformaticsonline.org/`
32. Park, G.S., Park, S., Kim, Y. and Kang, B.H.: Intelligent web document classification using incrementally changing training data set, J. Security Engineering 2 (2005), 186–191.
33. Sattar, A. and Kang, B.H.: "Advances in Artificial Intelligence", Proceedings of AI2006, Hobart, Tasmania, 2006.
34. Steane, D.A., Nicolle, D., Mckinnon, G.E., Vaillancourt, R.E. and Potts, B.M.: High-level relationships among the eucalypts are resolved by ITS-sequence data, Australian Systematic Botany 15 (2002), 49–62.
35. WEKA, Waikato Environment for Knowledge Analysis, `http://www.cs.waikato.ac.nz/ml/weka`, viewed 20.06.2008.
36. Washio, T. and Motoda, H.: State of the art of graph-based data mining, SIGKDD Explorations, Editorial: Multi-Relational Data Mining: The Current Frontiers, Editors: Saso Dzeroski and Luc De Raedt, SIGKDD Exploration, 5 (2003)(1), 59–68.
37. Witten, I.H. and Frank, E.: "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations", Morgan Kaufmann, 2005.
38. Yada, K., Motoda, H., Washio, T. and Miyawaki, A.: Cousumer behavior analysis by graph mining technique, New Mathematics and Natural Computation, 2 (2005)(1), 59-68.
39. Yearwood, J.L. and Mammadov, M.: "Classification Technologies: Optimization Approaches to Short Text Categorization", Idea Group Inc., 2007.