Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Pattern Recognition Letters 24 (2003) 65–73

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Classification of binary vectors by using ΔSC distance to minimize stochastic complexity

Pasi Fränti *, Mantao Xu, Ismo Kärkkäinen

Department of Computer Science, University of Joensuu, P.O. Box 111, Fin-80101 Joensuu, Finland

Received 17 October 2001; received in revised form 28 February 2002

Abstract

Stochastic complexity (SC) has been employed as a cost function for solving binary clustering problem using Shannon code length (CL distance) as the distance function. The CL distance, however, is defined for a given static clustering only, and it does not take into account of the changes in the class distribution during the clustering process. We propose a new ΔSC distance function, which is derived directly from the difference of the cost function value before and after the classification. The effect of the new distance function is demonstrated by implementing it with two clustering algorithms.

Keywords: Clustering; Stochastic complexity; Algorithms; Distance functions

1. Introduction

Binary vector classification has been widely used in DNA computing and human chromosome study, and in solving taxonomy problems from biomedical area. Statistical models are usually applied to describe and solve prediction and taxonomy problems. For example, Rissanen (1987, 1996) has introduced a model known as stochastic complexity (SC), which is an extensible explanation for Shannon information theory (Kontkanen et al., 1999).

To be a cost function of classification, SC needs to be approximated by a simple model (Rissanen, 1987). Gyllenberg et al. (1994, 1997, 2000) has given a simple and practical approximation of SC for binary vector classifications. Thereafter, SC has been employed as a generic evaluation function in solving binary clustering problems as follows. The clustering problem is first formulated as an optimization problem. Approximation solutions are then found for every reasonable number of groups. SC is applied for measuring the goodness of the various clustering results.

Individual clustering can be generated using any algorithms such as the generalized Lloyd algorithm (GLA) (Linde et al., 1980), and the randomized local search (RLS) Fränti and Kivijärvi, 2000. A better idea is to integrate the SC cost function directly in the clustering algorithm as done in Gyllenberg et al. (1997) and Fränti et al. (2000). The *vector-to-cluster* distance for the classification

[*] Corresponding author. Tel.: +358-13-251-7931; fax: +358-13-251-7955.

E-mail address: franti@cs.joensuu.fi (P. Fränti).

0167-8655/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0167-8655(02)00189-7

of the vectors must then be re-defined correspondingly. *Euclidean distance* (L_2-norm) provides the optimal classification of the data vectors for the minimization of the MSE, but not for the SC. The optimal classification for SC is given by the Shannon code length (CL) function (Gyllenberg et al., 1997). It represents the entropy of the binary vector when coded by the probability model of the particular cluster.

Surprisingly, the CL distance introduces a new problem that never arises with the L_2-distance. This is illustrated in Fig. 1, in which we classify the black point according the two existing clusters. The probability distribution of the leftmost cluster indicates the point belongs to this class with a low probability. Nevertheless, the probability distribution of the rightmost cluster have zero variance in the horizontal dimension ($\sigma_x = 0$) resulting in zero probability and infinite entropy. As a consequence, the point will be classified to the leftmost cluster.

This infinite entropy problem happens often in the classification of multi-dimensional binary data vectors. It has therefore been necessary to make modifications to the existing clustering algorithms when SC has been applied as a cost function. Previously, the problem has been solved in Gyllenberg et al. (1997) and Fränti et al. (2000) by applying the clustering algorithms first using the sub-optimal but less problematic L_2-distance. The CL distance is then applied in the last stage of the algorithm when the global clustering structure has already settled down and only fine-tuning of the solution takes place. The drawback of this approach is that similar patch should be made for every clustering algorithm that is to be applied with the SC.

In this paper, we propose a more general solution to the infinity problem by proposing a new ΔSC distance function. The distance function is derived directly from the difference of the cost function value before and after the classification. It therefore implicitly takes into account the change in the class distribution caused by the re-classification of the data vector, and in this way, avoids the infinity problem. The ΔSC is general in the sense that it applies to any clustering algorithm and no more patches are therefore needed. The effect of the new distance function is demonstrated by implementing it with two clustering algorithms.

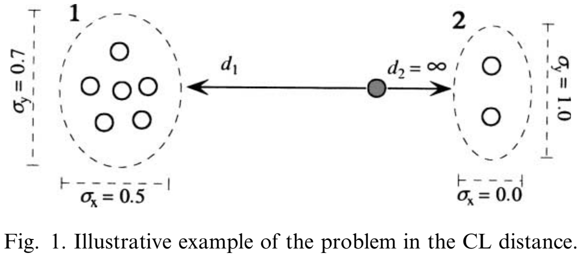


Fig. 1. Illustrative example of the problem in the CL distance.

The rest of the paper is organized as follows. In Section 2, we define the clustering problem of binary vectors, and give the simplified formalization of the SC. The SC function is then applied within two clustering algorithms as the cost function, and the CL distance is employed in the RLS and GLA algorithms as a practical vector-to-cluster distance. In Section 3, we introduce the new ΔSC function derived from the SC difference of the old and new classification when a data vector is moved from one class to another. In Section 4, we make performance comparisons of the different variants including the RLS and GLA algorithms, and the ΔSC, CL and L_2 distance functions.

2. Clustering by minimizing SC

We use the following notations:

N:	number of data vectors,
M:	number of groups,
D:	dimension of vectors,
X:	set of N data objects $X = \{x_1, x_2, \ldots, x_N\}$,
P:	partition indices of $x_i : P = \{p_i \mid i = 1, \ldots, N\}$,
C:	set of cluster centroids $C = \{c_j \mid j = 1, \ldots, M\}$.

The goal of the clustering is to partition a given set of N data vectors into a number of groups so that a given cost function is minimized. In the clustering process, we must solve both the number of clusters (M) and their location (c_i). The clustering result is described by the partition (P) of the data set by giving for each vector (x_i) the cluster index (p_i) of the group, which it belongs to. We consider a set of d-dimensional binary data vectors.

2.1. Stochastic complexity

SC can be applied to the clustering by finding the minimum description of the data via using a clustering model. SC measures the information content of the data, and it is defined as the shortest possible code length for the data obtainable by using a set of class distributions. SC includes both the model parameters and the coding of the data in the measurement.

Suppose that we have classified the data vectors into M groups described by the partition of the data. The model of the class j can then be described by the probability distribution within the class in each dimension:

$$c_{ij} = n_{ij}/n_j \tag{1}$$

where n_j is the number of binary vectors in the class j, and n_{ij} is the number of vectors having the ith coordinate value 1. The probability vector c_j of the class j is also the centroid (average vector) of the cluster.

The simplified approximation of the SC function in Gyllenberg et al. (1997) can be described using the class distribution models as:

$$\mathrm{SC} = \sum_{j=1}^{M} n_j \sum_{i=1}^{d} h\left(\frac{n_{ij}}{n_j}\right) + \sum_{j=1}^{M} -n_j \log\left(\frac{n_j}{N}\right) + \frac{d}{2}\sum_{j-1}^{M} \log \max(1, n_j) \tag{2}$$

where h measures the entropy of a binary distribution

$$h(p) = -p\log(p) - (1-p)\log(1-p) \tag{3}$$

Since every vector is classified to some group, it is known that $\sum n_j = N$. Moreover, $N \log N$ in the middle term is a constant and, therefore, the equation can be simplified as:

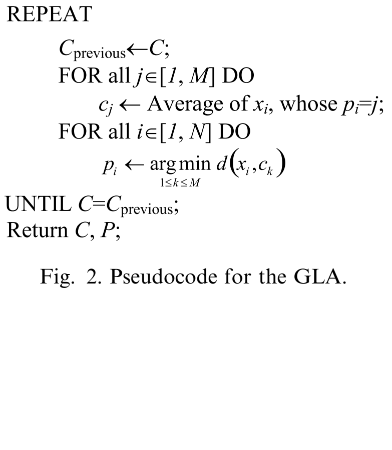
$$\mathrm{SC} \approx \sum_{j=1}^{M} n_j \sum_{i=1}^{d} h\left(\frac{n_{ij}}{n_j}\right) + \sum_{j=1}^{M} -n_j \log n_j + \frac{d}{2} \times \sum_{j=1}^{M} \log \max(1, n_j) \tag{4}$$

However, the simplified Eq. (4) above could make SC negative. The first part of the SC function measures the intra-class information as the code length when every data vector is coded according to the class probability model. The code length is calculated by multiplying the number of vectors in each cluster (n_j) by the average entropy (h) of the cluster. The second part measures the inter-class information as the code length of the partition. It can be calculated by the number of vectors in each cluster (n_j) multiplied by the average entropy of the corresponding cluster index. The third part measures the information of the model as the code length of the class distribution when described by a series numbers between $[1..n_j]$.

2.2. Clustering algorithm

The SC can be applied for the clustering problem as follows. We first find approximation solutions for every reasonable number of groups using any clustering algorithm. The solutions are then evaluated, and the one that minimizes the SC is the final result of the clustering. This search strategy can use any clustering algorithm to find the individual solutions. In the following, we recall two clustering algorithms: the GLA by Linde et al. (1980), and the RLS by Fränti and Kivijärvi (2000).

The pseudocode of the GLA is shown in Fig. 2. The algorithm takes any initial solution (here the partition P) as an input, and iteratively fine-tunes the solution by repeating two operations in turn. The first operation calculates the centroids of the clusters, and the second operation re-partition the data vectors according to the new set of centroids.



```
GLA($X$, $P$) return $C$, $P$
REPEAT
    $C_{\mathrm{previous}} \leftarrow C$;
    FOR all $j \in [1, M]$ DO
        $c_j \leftarrow$ Average of $x_i$, whose $p_i = j$;
    FOR all $i \in [1, N]$ DO
        $p_i \leftarrow \arg\min_{1 \le k \le M} d(x_i, c_k)$
UNTIL $C = C_{\mathrm{previous}}$;
Return $C$, $P$;
```

Fig. 2. Pseudocode for the GLA.

```
RLS($X$, $M$) return $C$, $P$
$C \leftarrow$Set of randomly chosen data vectors;
FOR all $i \in [1, N]$ DO
        $p_i \leftarrow \arg\min_{1 \le k \le M} d(x_i, c_k)$
FOR $a \leftarrow 1$ To NumberOfIterations DO
        $C_{\text{new}} \leftarrow C$;
        $c_j \leftarrow$ Randomly chose data vector from $X$;
        $C_{\text{new}}$, $P_{\text{new}} \leftarrow$ LocalRepartition($X$, $P$, $C_{\text{new}}$, $j$);
        GLA($X$, $P_{\text{new}}$);
        IF SC(X, $C_{\text{new}}$, $P_{\text{new}}$) < SC(X, $C$, $P$) THEN
              $C \leftarrow C_{\text{new}}$; $P \leftarrow P_{\text{new}}$;
Return $C$, $P$;
```

Fig. 3. Pseudocode for the RLS.

The algorithm is iterated until no more improvement appears in the solution. The method is simple to implement, and has been widely used for the clustering problem as such, or integrated with more complicated methods.

The pseudocode of the RLS is shown in Fig. 3. The method takes any initial solution, which is then improved by a sequence of operations. At each iteration phase, the algorithm creates a new candidate solution by making a small change to the current clustering structure. First, a randomly chosen cluster centroid (c_j) is replaced by a randomly chosen data vector (x_i). This moves the cluster location to another part of the vector space. The partition is then adjusted by a local repartition operation, which consists of the two steps as shown in Fig. 4. In the first step, the old cluster is removed by re-partitioning its data vectors to other clusters. In the second step, the newly created cluster is populated by attracting data vectors from the neighboring clusters. The modified clustering is fine-tuned by the application of the GLA. The new candidate solution is then evaluated and accepted only if it improves the previous solution. Otherwise, the candidate solution is discarded and the previous solution remains as the starting point for the next iteration.

```
LocalRepartition($X$, $P$, $C$, $j$) return $P$, $C$;
FOR all ($p_i$=$j$) DO
        $p_i \leftarrow \arg\min_{1 \le k \le M} d(x_i, c_k)$
FOR all $i \in [1, N]$ DO
        $p_i \leftarrow \arg\min_{k=j \vee k=p_i} d(x_i, c_k)$
```

Fig. 4. Pseudocode for the local repartition operation.

The GLA and the RLS are both applicable for the clustering task and also rather simple to implement. The RLS is less sensitive to the initialization because it is capable of making global changes in the clustering structure (by random swapping of the clusters), and therefore, correct the incorrect settlement of the initial clustering. If the GLA is to be used, it should be repeated several times in order to reduce the dependency on the initialization.

2.3. Shannon code-length distance

The clustering algorithms employ a distance function d, which measures the vector-to-cluster distance, and is used for the classification of the vectors during the clustering process. Usually the distance function is defined as the Euclidean distance (L_2-norm) between the data vector x_i and the particular cluster centroid c_j:

$$d_{\mathrm{E}}(x_i, c_j) = \sqrt{\sum_{k=1}^{d} \|x_{ik} - c_{jk}\|^2} \tag{5}$$

This gives optimal classification for the minimization of the MSE but not for the SC. The optimal classification for the SC is given by the Shannon code-length function $\mathrm{CL}(x_i, c_j)$ in Gyllenberg et al. (1997).

$$d_{\mathrm{CL}}(x_i, c_j) = -\sum_{i=1}^{d} \left((1 - x_i)\log(1 - c_{ij}) + x_i \log c_{ij}\right) - \log\frac{n_j}{N} \tag{6}$$

It measures the code length when the data vector (the summation term in the equation) and its class index (second term in the equation) are coded using the given model. In principle, the CL distance is well defined, but in practice, it has a fundamental problem in its definition, which will be explained by the following example.

Consider a single cluster c_1 consisting of the following three data vectors: $x_1 = (0,0,0)$, $x_2 = (0,1,0)$ and $x_3 = (1,0,0)$. The corresponding class probability distribution of the cluster is $c_1 = (0.33, 0.33, 0.00)$. The distances of the vectors can now be calculated using the CL distance:

$$d_{\mathrm{CL}}(x_1, c_1) = 0.58 + 0.58 + 0.00 = 1.17$$
$$d_{\mathrm{CL}}(x_2, c_1) = 0.58 + 1.58 + 0.00 = 2.17$$
$$d_{\mathrm{CL}}(x_3, c_1) = 1.58 + 0.58 + 0.00 = 2.17$$

The second term of Eq. (6) is omitted in this example for simplicity. Let us then consider a fourth vector $x_4 = (0, 0, 1)$, which is equally close to the cluster according to the Euclidean distance, but the CL distance gives the following result:

$$d_{\mathrm{CL}}(x_4, c_1) = 0.58 + 0.58 + \infty = \text{undefined}$$

The problem can appear when there is a uniform bit distribution in any dimension, and the data vector has different value in the same position. The homogenous bit distribution indicates that there is no uncertainty and the entropy of the contradicting value would therefore approach infinite. This is a serious flaw especially in the local re-partition procedure of the RLS. It creates new clusters starting from a singular cluster, which evidently has uniform bit distribution. As a consequence, no other data vectors (except equal ones) can ever be classified to this cluster.

The problem of the CL distance is that even though it measures the uncertainty of the classification, it does not take into account the uncertainty of the model itself. In other words, the bit distribution of the class model is indeed homogeneous, but the model is only an approximation and subject to change during the clustering process. Zero-probability is therefore not a feasible approximation of the classification.

The infinite values could be avoided by preventing the centroids to take values 0 and 1. This can be achieved, for example, given binary data vector x, by taking the centroid values to be mean vector of x and c_j. If some coordinate of c_j equals to 0 or 1 value, the number of vectors in jth cluster, n_j can be taken as a parameter of CL distance function. Obviously, c_j can be replaced with a new vector, which is the centroid of jth cluster after vector x is put into cluster j.

$$c_{ij} = \frac{n_j c_{ij} + x_i}{n_j + 1} \quad c_{ij} \neq x_i \tag{7}$$

If $c_j = x$, CL distance is adopted as $\log_2(N)$. Another condition on CL distance value is considered as follows:

$$x_i \log c_{ij} + (1 - x_i)\log(1 - c_{ij}) = 0$$
$$c_{ij} = x_i, \;\; c_j \neq x \tag{8}$$

This patch, however, does not remove the problem itself as it merely assigns a low probability instead of a zero value. Additional modifications have therefore been necessary for the clustering algorithms so that the CL distance could have been used properly in the GLA and in the RLS. For example, in the algorithms presented in Gyllenberg et al. (1997) and Fränti et al. (2000) the CL distance is applied only in the last step of the clustering process when the global clustering structure has already settled down, and only fine-tuning of the solution takes place. The problem of this approach is that it is not trivial to determine the stage of the clustering process, when it would be safe enough to start to use the CL distance.

To sum up, the problem with the CL distance is fundamental in its nature. It is therefore better to fix it than to find patch for every clustering algorithm that is to be applied.

3. ΔSC distance function

We introduce a new vector-to-cluster distance function denoted as ΔSC distance. It is based on a design paradigm, in which the distance function is derived directly from the difference of the cost function value before and after the classification of a data vector. The main advantage of this design philosophy is that it implicitly takes into account of the changes in the clustering model caused by the classification. It is also general in the sense that it does not depend on the chosen clustering algorithm and should therefore be applicable with any distance-based clustering method.

The ΔSC distance function is always defined relative to a given model. We can therefore assume that we have a model, for which we can calculate the SC value. If we then consider the distance calculation as a movement of the data vector from one group to another, we can define the distance function as the difference in the SC of the clustering before and after the movement of the data vector. Given two classes j_1, j_2 and a binary vector x, which we consider to move from the class j_1 to class j_2, the SC function in (2) value after the movement is:

$$\begin{aligned}
\mathrm{SC} \approx{} & \sum_{j\neq j_1,j_2}^{M} n_j \sum_{i=1}^{d} h\left(\frac{n_{ij}}{n_j}\right) + \sum_{j\neq j_1,j_2} (n_j \log 2(n_j)) \\
& + \frac{d}{2}\sum_{j\neq j_1,j_2}^{M} \log\max(1,n_j) - (n_{j_1}-1) \\
& \times \log(n_{j_1}-1) - (n_{j_2}+1)\log(n_{j_2}+1) \\
& + \sum_{i=1}^{d}\left((n_{j_1}-1)h\left(\frac{n_{ij_1}-x_i}{n_{j_1-1}}\right)\right. \\
& \left. + (n_{j_2}+1)h\left(\frac{n_{ij_2}+x_i}{n_{j_2}+1}\right)\right) \\
& + \frac{d}{2}(\log\max(1,n_{j_1}-1) \\
& + \log\max(1,n_{j_2}+1)) + N\log N \qquad (9)
\end{aligned}$$

We can then calculate the difference between the SC function values of the old clustering (before the movement) and the new one (after the movement) as:

$$\begin{aligned}
\text{SC-diff}(x,j_1,j_2) ={} & \sum_{i=1}^{d}\left((n_{j_1}-1)h\left(\frac{n_{ij_1}-x_i}{n_{j_1}-1}\right)\right. \\
& - n_{j_1}h\left(\frac{n_{ij_1}}{n_{j_1}}\right) \\
& + (n_{j_2}+1)h\left(\frac{n_{ij_2}+x_i}{n_{j_2}+1}\right) \\
& \left. - n_{j_2}h\left(\frac{n_{ij_2}}{n_{j_2}}\right)\right) + (n_{j_1}-d/2) \\
& \times \log n_{j_1} + (n_{j_2}-d/2) \\
& \times \log n_{j_2} - (n_{j_2}+1-d/2) \\
& \times \log(n_{j_2}+1) - (n_{j_1}-1-d/2) \\
& \times \log(n_{j_1}-1) \quad n_{j_1} > 1
\end{aligned}$$

$$\begin{aligned}
\text{SC-diff}(x,j_1,j_2) ={} & \sum_{i=1}^{d}\left((n_{j_2}+1)h\left(\frac{n_{ij_2}-x_i}{n_{j_2}-1}\right)\right. \\
& \left. - n_{j_2}h\left(\frac{n_{ij_2}}{n_{j_2}}\right)\right) \\
& + (n_{j_2}-d/2)\log n_{j_2} \\
& + (d/2-n_{j_2}-1)\log(n_{j_2}+1) \\
& n_{j_1} = 1 \qquad (10)
\end{aligned}$$

The SC-diff takes zero value if $j_1 = j_2$. Negative values are obtained when the movement of the vector improves the solution, and positive values otherwise. The SC-diff could now be applied as such in the cases when we re-classify a vector in an existing solution.

In the SC-diff function we assume that the given vector is already classified into some class. In general, however, this is not the case but we must be able to define a more general distance function that depends only on the vector x_i and on the candidate cluster c_j. For example, in the repartition procedure of the RLS algorithm, we classify vectors whose previous class has been removed. More general ΔSC function can be derived from (10) as follows.

The classification can be considered as a two-step procedure, in which we first remove the vector x_i from the class j_1 and then add it to the class j_2 For a given vector x_i the cost of the removal is constant. This means that the parameters N, n_{j_1}, n_{ij_1} are fixed in the classification, and as a consequence, we can consider only the cost of adding the vector in class j_2 and ignore the removal part in the formula. Thus, the ΔSC ($j_1 \neq j_2$) can be defined merely as the cost of the addition

$$\begin{aligned}
\Delta\mathrm{SC}(x,C_{j_2}) ={} & \sum_{i=1}^{d}\left((n_{j_2}+1)h\left(\frac{n_{ij_2}+x_i}{n_{j_2}+1}\right)\right. \\
& \left. - n_{j_2}h\left(\frac{n_{ij_2}}{n_{j_2}}\right)\right) \\
& + (n_{j_2}-d/2)\log n_{j_2} \\
& + (d/2-n_{j_2}-1)\log(n_{j_2}+1) \\
& + \log N \qquad (11)
\end{aligned}$$

This gives the same result as the SC-diff with the difference of a constant. The only exception is when we measure the distance of x_i to the cluster, in which it is already included ($j_1 = j_2$). In this case, we should use $(n_{j_1} - 1)$ as the class size instead of n_{j_1} because the class size does not increase due to the classification. Thus, if the previous classification is known, we should apply the following equation for this special case:

$$\begin{aligned}\Delta\mathrm{SC}(x, C_{j_2}) = \sum_{i=1}^{d} &\left(n_{j_1} h\left(\frac{n_{ij_1}}{n_{j_1}}\right) - (n_{j_1} - 1) h\left(\frac{n_{ij_1} - x_i}{n_{j_1} - 1}\right)\right) \\ &+ (d/2 - n_{j_1}) \log n_{j_1} \\ &+ (n_{j_1} - 1 - d/2) \log(n_{j_1} - 1) \\ &+ \log N \\ &\quad j_1 = j_2, \; n_{j_1} > 1\end{aligned}$$

$$\Delta\mathrm{SC}(x, C_{j_2}) = \log N \quad j_1 = j_2, \; n_{j_1} = 1 \tag{12}$$

Hence, ΔSC distance as in Eq. (11) is applicable as vector-to-cluster distance in all cases, although it underestimates the distance in the case when the vector is already included in the class. The special case of (12) should therefore be used when applicable to give more exact value.

4. Test results

We use three binary data sets to test the new method: DNA-1, DNA-2, Normal. The features of the first two sets (DNA-1 and DNA-2) were extracted from analysis of DNA samples of fishes (presence or absence of given DNA fragment) in biological research experiments. There are 215 52-dimensional binary vectors in (DNA-1) and 260 60-dimensional vectors in DNA-2. The third set (Normal) was artificially created by generating 265 binary vectors into 10-dimensional vector space with 12 clusters.

We study first the DNA-1 and DNA-2 sets when clustered using the RLS and GLA methods. The RLS was performed 80 iterations. The results in Figs. 5–8 show that the ΔSC distance and L_2-distance

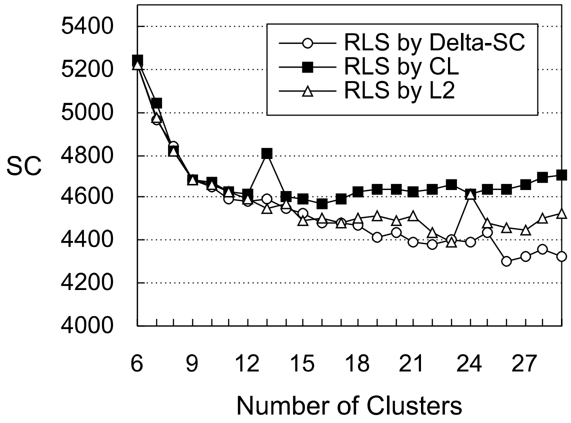


Fig. 5. Clustering results by the RLS algorithm for DNA-1.

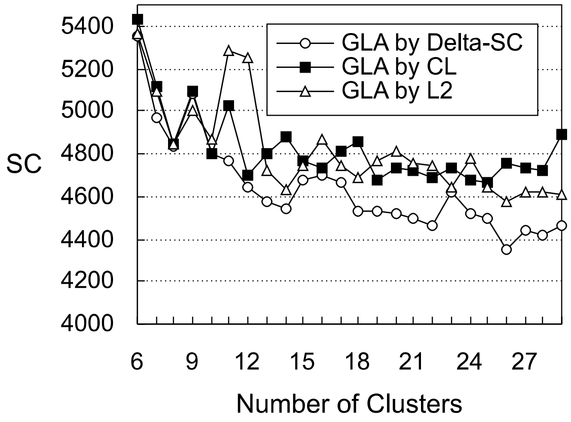


Fig. 6. Clustering results by the GLA algorithm for DNA-1.

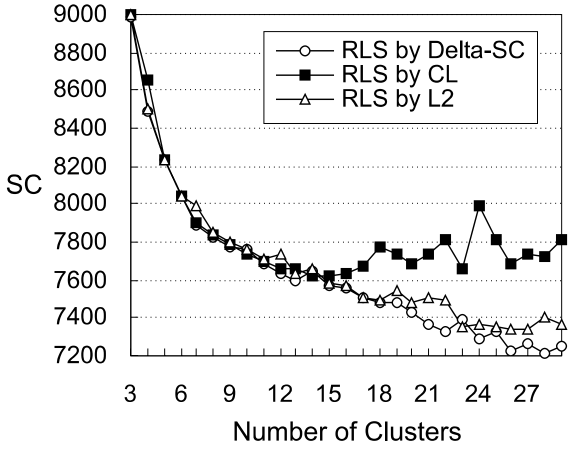


Fig. 7. Clustering results by the RLS algorithm for DNA-2.

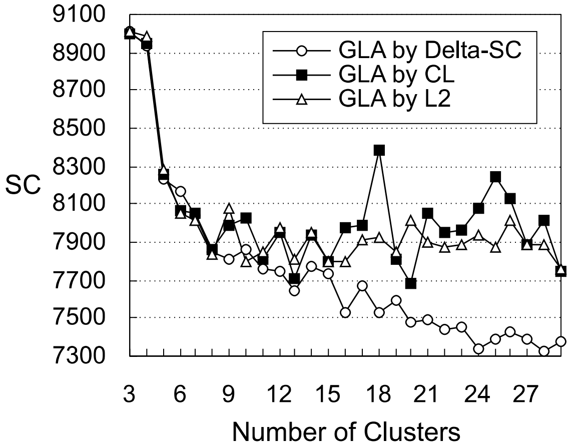


Fig. 8. Clustering results by the GLA algorithm for DNA-2.

come up with much better results than CL distance in RLS algorithm. It seems that there is no big difference between ΔSC distance and L_2-distance when they are employed in RLS. The difference, however, can be significant when the correct number is to be determined in the stepwise search.

The result with the GLA is quite different from that of the RLS, mainly because the variance of the results is much greater. The CL distance still performs worse than the L_2-distance, but the ΔSC distance is now clearly better than the L_2-distance almost with respect to every number of clusters. It is expected that the correct result would be reached more reliably using the ΔSC distance. The drawback of the ΔSC distance is takes much more time to compute than the L_2-distance.

Table 1 summarizes the clustering and classification results for the Normal data set. It is the only data set for which the real classification is known, and thus, classification rate could be calculated. The results show that employing RLS by ΔSC distance gives the best performances both in terms of best clustering result (smallest SC values), and the highest classification rate. The RLS algorithm found the correct number of clusters also with the L_2-distance and CL distance but the corresponding classification rates were smaller. The GLA, however, was able to find the correct result (12 clusters) only by using the ΔSC distance.

5. Conclusions

We proposed a new vector-to-cluster distance in the classification problems of binary vectors by minimizing SC. The distance function was applied both in the GLA and RLS algorithms.

Experiments show that RLS by ΔSC distance gives the best clustering performance in minimizing SC among all the variants considered, and the highest classification rate. In most cases, the modified CL distance performs even worse than the L_2-distance. It is somehow difficult for the stepwise GLA to deliver satisfactory results in solving the correct number of clusters, even by using the ΔSC distance. The L_2-distance is moderately effective to classify simple data. Among the three distances, the ΔSC distance is the most precise to minimize stochastic complexity.

Our approach by using ΔSC distance is general in its nature as the same design paradigm can be applied with any other cost function too.

Table 1
The classification results of the RLS and GLA algorithms with the L_2-distance, CL-distance and ΔSC-distance

	RLS algorithm			GLA algorithm			Real classification
	L_2	CL	ΔSC	L_2	CL	ΔSC	
SC	1865.2	1867.2	1857.6	1924.1	1920.6	1862.5	1856.5
Number of clusters	12	12	12	6[a]	6[a]	12	12
Classification rate	96.98%	92.83%	98.87%	74.34%	87.55%	91.32%	100%

[a] The classification rates are calculated from the clustering of 12 clusters even though smaller SC-value was found with 6 clusters.

References

Fränti, P., Kivijärvi, J., 2000. Randomized local search algorithm for the clustering problem. Pattern Analysis and Applications 3 (4), 358–369.

Fränti, P., Gyllenberg, H.G., Gyllenberg, M., Kivijärvi, J., Koski, T., Lund, T., Nevalainen, O., 2000. Minimization stochastic complexity using GLA and local search with applications to classifications of bacteria. Biosystems 57 (1), 37–48.

Gyllenberg, M., Koski, T., 2000. Probabilistic Models for Bacterial Taxonomy, TUCS Technical Report, No. 325, Turku Center for Computer Science, Finland, February 2000.

Gyllenberg, M., Koski, M., Verlaan, M., 1994. Clustering and quantization of binary vectors with stochastic complexity. Proc. IEEE Internat. Symposium on Information Theory, Trondheim, Germany, 1994.

Gyllenberg, M., Koski, T., Verlaan, M., 1997. Classification of binary vectors by stochastic complexity. Journal of Multivariate Analysis 63, 47–72.

Kontkanen, P., Myllymäki, P., Silander, T., Tirri, H., 1999. On the accuracy of stochastic complexity approximations. In: Gammerman, A. (Ed.), Causal Models and Intelligent Data Management, Chapter 9. Springer-Verlag.

Linde, Y., Buzo, A., Gray, R., 1980. An algorithm for vector quantizer design. IEEE Trans. Communications 28 (1), 84–95.

Rissanen, J., 1987. Stochastic Complexity and the MDL Principle. Econometric Reviews 6, 85–102.

Rissanen, J., 1987. Stochastic complexity. Journal of Statistics Society 4 (10), 223–239.

Rissanen, J., 1996. Fisher information and stochastic complexity. IEEE Trans. on Information Theory 42 (1), 40–47.