

An Empirical Comparison of NML Clustering Algorithms

Petri Kontkanen and Petri Myllymäki
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
P.O.Box 68 (Department of Computer Science)
FIN-00014 University of Helsinki, Finland
E-mail: {Firstname}.{Lastname}@hiit.fi

Abstract—Clustering can be defined as a data assignment problem where the goal is to partition the data into non-hierarchical groups of items. In our previous work, we suggested an information-theoretic criterion, based on the minimum description length (MDL) principle, for defining the goodness of a clustering of data. The basic idea behind this framework is to optimize the total code length over the data by encoding together data items belonging to the same cluster. In this setting efficient coding is possible only by exploiting underlying regularities that are common to the members of a cluster, which means that this approach produces an implicitly defined similarity metric between the data items. Formally the global code length criterion to be optimized is defined by using the intuitively appealing universal normalized maximum likelihood (NML) code which has been shown to produce optimal code lengths in the worst case sense. In this paper, we focus on the optimization aspect of the clustering problem, and study five algorithms that can be used for efficiently searching the exponentially-sized clustering space. As the suggested NML clustering criterion can be used for comparing clusterings with different number of cluster labels, the number of clusters is not known beforehand and determining it is part of the optimization process. In the empirical part of the paper we compare the performance of the suggested algorithms in the task of optimizing the NML clustering criterion using several real-world datasets.

Index Terms—minimum description length, normalized maximum likelihood, clustering, EM algorithm, K-means algorithm

I. INTRODUCTION

Although clustering is one of the central concepts in the field of unsupervised data analysis, it is also a very controversial issue, and the very meaning of the concept “clustering” may vary a great deal between different scientific disciplines (see, e.g., [1] and the references therein). However, a common goal in all cases is that the objective is to find a structural representation of data by grouping (in some sense) similar data items together. In the following we regard clustering as a partitional data assignment or data labeling problem, where the goal is to partition the data into mutually exclusive clusters so that similar data vectors are grouped together. The number of clusters is unknown, and determining the optimal number is part of the clustering problem. The data are assumed to be in a vector form so that each data item is a vector consisting of a fixed number of attribute values.

We can now identify two fundamental problems within this

framework: how to define the goodness of a clustering (data partitioning) and how to find good clusterings with respect to the chosen scoring criterion. The focus in this paper is on the latter problem.

Traditionally, the scoring problem has been approached by first fixing a distance metric, and then by defining a global goodness measure based on this distance metric. However, although this approach is intuitively quite appealing, from the theoretical point of view it introduces many problems, such as choosing a suitable distance metric and the handling of non-continuous attributes. A completely different approach to clustering is offered by the *model-based approach*, where for each cluster a data generating function (a probability distribution) is assumed, and the clustering problem is defined as the task to identify these distributions (see, e.g., [2], [3], [4]). In other words, the data are assumed to be generated by a finite mixture model [5], [6], [7]. In this framework the optimality of a clustering can be defined as a function of the fit of data with the finite mixture model, not as a function of the distances between the data vectors. See [8] for more discussion on the differences between the traditional and the model-based approaches.

In [8] we proposed a scoring criterion for clusterings, based on the idea that a good clustering is such that one can encode the cluster labels *together* with the data so that the resulting code length is minimized. The clustering criterion suggested was based on the MDL principle [9], [10], [11] which intuitively speaking aims at finding the shortest possible encoding for the data. For formalizing this intuitive goal, we adopt the modern *normalized maximum likelihood (NML)* coding approach [12], which can be shown to lead to a criterion with very desirable theoretical properties (see Section II and e.g. [11], [13], [14], [15], [16], [17]). It is important to realize that approaches based on either earlier formalizations of MDL or on more heuristic encoding schemes (see e.g. [18], [19], [20]) do not possess these theoretical properties.

This paper is a direct continuation of [8], where we introduced the NML clustering approach and derived an efficient algorithm for computing the NML criterion. In the empirical tests of [8] we concentrated on a special data consisting of measured signal strength values of radio signals originating from WLAN access points. In this paper, we will extend

this work by using several real-world datasets from the UCI repository [21]. Moreover, we will present and empirically compare five different optimization algorithms that can be used for finding good clusterings with respect to the NML scoring criterion.

This paper is structured as follows. In Section II we discuss the basic properties of the MDL framework in general and also shortly review the optimality properties of the NML distribution. In Section III we introduce the notation and formalize clustering as a data assignment problem. We also show how the NML criterion can be computed efficiently for the clustering model class. In Section IV, we empirically compare several algorithms for finding good clusterings. Section V summarizes the main results of our work.

II. PROPERTIES OF MDL AND NML

The MDL principle has several desirable properties. Firstly, it automatically protects against overfitting in the model class selection process. Secondly, there is no need to assume that there exists some underlying “true” model, while most other statistical frameworks do. The model class is only used as a technical device for constructing an efficient code for describing the data. MDL is also closely related to the Bayesian inference but there are some fundamental differences, the most important being that MDL is not dependent on any prior distribution, it only uses the data at hand. For more discussion on the theoretical motivations behind the MDL principle see, e.g., [11], [13], [16], [17], [22], [23].

MDL model class selection is based on minimization of the stochastic complexity. In the following, we give the definition of the stochastic complexity and then proceed by discussing its theoretical properties.

Let $\mathbf{x}^n = (x_1, \dots, x_n)$ be a data sample of n outcomes, where each outcome x_j is an element of some space of observations \mathcal{X} . The n -fold Cartesian product $\mathcal{X} \times \dots \times \mathcal{X}$ is denoted by \mathcal{X}^n , so that $\mathbf{x}^n \in \mathcal{X}^n$. Consider a set $\Theta \subseteq \mathbb{R}^d$, where d is a positive integer. A class of parametric distributions indexed by the elements of Θ is called a *model class*. That is, a model class \mathcal{M} is defined as

$$\mathcal{M} = \{P(\cdot | \theta) : \theta \in \Theta\}, \quad (1)$$

and the set Θ is called a *parameter space*.

One of the most theoretically and intuitively appealing model class selection criteria is the *stochastic complexity*. Denote first the maximum likelihood estimate of data \mathbf{x}^n for a given model class \mathcal{M} by $\hat{\theta}(\mathbf{x}^n, \mathcal{M})$, i.e., $\hat{\theta}(\mathbf{x}^n, \mathcal{M}) = \arg \max_{\theta \in \Theta} \{P(\mathbf{x}^n | \theta)\}$. The *normalized maximum likelihood* (NML) distribution [12] is now defined as

$$P_{\text{NML}}(\mathbf{x}^n | \mathcal{M}) = \frac{P(\mathbf{x}^n | \hat{\theta}(\mathbf{x}^n, \mathcal{M}))}{\mathcal{C}(\mathcal{M}, n)}, \quad (2)$$

where the normalizing term $\mathcal{C}(\mathcal{M}, n)$ in the case of discrete data is given by

$$\mathcal{C}(\mathcal{M}, n) = \sum_{\mathbf{y}^n \in \mathcal{X}^n} P(\mathbf{y}^n | \hat{\theta}(\mathbf{y}^n, \mathcal{M})), \quad (3)$$

and the sum goes over the space of data samples of size n . If the data is continuous, the sum is replaced by the corresponding integral.

The stochastic complexity of the data \mathbf{x}^n given a model class \mathcal{M} is defined via the NML distribution as

$$\begin{aligned} SC(\mathbf{x}^n | \mathcal{M}) &= -\log P_{\text{NML}}(\mathbf{x}^n | \mathcal{M}) \\ &= -\log P(\mathbf{x}^n | \hat{\theta}(\mathbf{x}^n, \mathcal{M})) \\ &\quad + \log \mathcal{C}(\mathcal{M}, n), \end{aligned} \quad (4)$$

and the term $\log \mathcal{C}(\mathcal{M}, n)$ is called the (*minimax*) *regret* or *parametric complexity*. The regret can be interpreted as measuring the logarithm of the number of essentially different (distinguishable) distributions in the model class. Intuitively, if two distributions assign high likelihood to the same data samples, they do not contribute much to the overall complexity of the model class, and the distributions should not be counted as different for the purposes of statistical inference. See [24] for more discussion on this topic.

The NML distribution (2) has several important theoretical optimality properties. The first one is that NML provides a unique solution to the minimax problem

$$\min_{\hat{P}} \max_{\mathbf{x}^n} \log \frac{P(\mathbf{x}^n | \hat{\theta}(\mathbf{x}^n, \mathcal{M}))}{\hat{P}(\mathbf{x}^n | \mathcal{M})}, \quad (5)$$

as posed in [12]. The minimizing \hat{P} is the NML distribution, and the minimax regret

$$\log P(\mathbf{x}^n | \hat{\theta}(\mathbf{x}^n, \mathcal{M})) - \log \hat{P}(\mathbf{x}^n | \mathcal{M}) \quad (6)$$

is given by the parametric complexity $\log \mathcal{C}(\mathcal{M}, n)$. This means that the NML distribution is the *minimax optimal universal model*. The term universal model in this context means that the NML distribution represents (or mimics) the behaviour of all the distributions in the model class \mathcal{M} . Note that the NML distribution itself typically does not belong to the model class.

A related property of NML involving expected regret was proven in [17]. This property states that NML also minimizes

$$\min_{\hat{P}} \max_g E_g \log \frac{P(\mathbf{x}^n | \hat{\theta}(\mathbf{x}^n, \mathcal{M}))}{\hat{P}(\mathbf{x}^n | \mathcal{M})}, \quad (7)$$

where the expectation is taken over \mathbf{x}^n and g is the worst-case data generating distribution. The minimax expected regret is also given by $\log \mathcal{C}(\mathcal{M}, n)$.

III. NML CLUSTERING

Let us assume that our problem domain consists of m discrete variables X_1, \dots, X_m and that the variable X_i has K_i values. The data $\mathbf{x}^n = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ consists of observations $\mathbf{x}_j = (x_{j0}, x_{j1}, \dots, x_{jm}) \in \mathcal{X}$, where

$$\mathcal{X} = \{1, 2, \dots, K_1\} \times \dots \times \{1, 2, \dots, K_m\}. \quad (8)$$

We assume that the possibly originally continuous variables have been discretized. One reason for focusing on discrete data is that in this case we can model the domain variables by multinomial distributions without having to make restricting

assumptions about unimodality, normality etc., which is the situation we face in the continuous case.

A *clustering* of the data set \mathbf{x}^n is here defined as a partitioning of the data into mutually exclusive subsets, the union of which forms the data set. The number of subsets is a priori unknown. The *clustering problem* is the task to determine the number of subsets, and to decide to which cluster each data vector belongs.

Formally, we can notate a clustering by using a *clustering vector* $\mathbf{z}^n = (z_1, \dots, z_n)$, where z_j denotes the index of the cluster to which the data vector \mathbf{x}_j is assigned to. Denote the *clustering variable* by Z so that \mathbf{z}^n is a sample from the distribution of Z . The number of clusters, say K_0 , is implicitly defined in the clustering vector, as it can be determined by counting the number of different values appearing in \mathbf{z}^n . It is reasonable to assume that K_0 is bounded by the size of our data set, so we can define the *clustering space* \mathcal{Z} as the set containing all the clusterings \mathbf{z}^n with the number of clusters being less or equal to n . Hence the clustering problem is now to find from all the $\mathbf{z}^n \in \mathcal{Z}$ the optimal clustering \mathbf{z}^n .

For solving the clustering problem we obviously need a global optimization criterion that can be used for comparing clusterings with different number of clusters. To formalize this, we first need to explicate the type of probabilistic models we consider.

As in [8], we use the finite mixture model family here. The corresponding model class with K_0 components is denoted by $\mathcal{M}(K_0)$ and

$$\mathcal{M}(K_0) = \{P_{\text{FM}}(\cdot | \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta_{K_0}\}. \quad (9)$$

The basic finite mixture assumption is that given the value of the clustering variable Z , the primary variables (X_1, \dots, X_m) are independent. Consequently, we have

$$\begin{aligned} P_{\text{FM}}(Z = z, X_1 = x_1, \dots, X_m = x_m | \boldsymbol{\theta}) \\ = P(Z = z | \boldsymbol{\theta}) \cdot \prod_{i=1}^m P(X_i = x_i | Z = z, \boldsymbol{\theta}). \end{aligned} \quad (10)$$

Furthermore, we assume that the distribution of $P(Z | \boldsymbol{\theta})$ is multinomial with parameters $(\pi_1, \dots, \pi_{K_0})$, and each $P(X_i | Z = k, \boldsymbol{\theta})$ is multinomial with parameters $(\sigma_{ik1}, \dots, \sigma_{ikK_i})$. The whole parameter space is then

$$\begin{aligned} \Theta_{K_0} = \{ & (\pi_1, \dots, \pi_{K_0}), \\ & (\sigma_{111}, \dots, \sigma_{11K_1}), \dots, (\sigma_{mK_01}, \dots, \sigma_{mK_0K_m}) : \\ & \pi_k \geq 0, \sigma_{ikl} \geq 0, \pi_1 + \dots + \pi_{K_0} = 1, \\ & \sigma_{ik1} + \dots + \sigma_{ikK_i} = 1, \\ & i = 1, \dots, m, k = 1, \dots, K_0 \}, \end{aligned} \quad (11)$$

and the parameters are defined by $\pi_k = P(Z = k)$, $\sigma_{ikl} = P(X_i = l | Z = k)$.

Our optimality criterion for clustering is based on information-theoretical arguments, in particular on the Minimum Description Length (MDL) principle. Intuitively, the MDL principle aims at finding the shortest possible encoding for the data, in other words the goal is to find the most

compressed representation of the data. Compression is possible by exploiting underlying regularities found in the data — the more regularities found, the higher the compression rate. Consequently, the MDL optimal encoding has found all the available regularities in the data; if there would be an “unused” regularity, this could be used for compressing the data even further.

What does this mean in the clustering framework? We suggest the following criterion for clustering: *the data vectors should be partitioned so that the vectors belonging to the same cluster can be compressed well together*. This means that those data vectors that obey the same set of underlying regularities are grouped together. In other words, the MDL clustering approach defines an implicit multilateral distance metric between the data vectors.

In [8], we suggested the following formalization of general optimality criterion for finding the optimal clustering $\hat{\mathbf{z}}^n$:

$$\hat{\mathbf{z}}^n = \arg \max_{\mathbf{z}^n} P(\mathbf{x}^n, \mathbf{z}^n | \mathcal{M}(K_0)). \quad (12)$$

From the coding point of view, definition (12) means the following: If one uses separate codes for encoding the data in different clusters, then in order to be able to decode the data, one needs to send with each vector the index of the corresponding code to be used. This means that we need to encode not only the data \mathbf{x}^n , but also the clustering \mathbf{z}^n , which is exactly what is done in (12).

There are, naturally, several ways to define the joint probability $P(\mathbf{x}^n, \mathbf{z}^n | \mathcal{M}(K_0))$. In [8], we compared the MDL and Bayesian approaches and the conclusion was that the MDL approach has several advantages over the Bayesian one. Firstly, the MDL principle does not assume that the chosen model class is correct. It even says that there is no such thing as a true model or model class, as acknowledged by many practitioners. The model class is only used as a technical device for constructing an efficient code. Secondly, there is no need to define a prior distribution for the parameters. The choice of the prior has a major effect on the quality of the results, as shown in [8]. Since there is no automatic way to choose the optimal prior, the Bayesian approach has a disadvantage here. Finally, the empirical results of [8] clearly favored the MDL approach, especially in the more complex cases. For these reasons, in the following we will only concentrate on the MDL approach.

As mentioned in Section II, MDL model selection is based on the minimization of the stochastic complexity, which is the minus logarithm of the NML distribution. Assuming i.i.d., the NML distribution for the finite mixture model can be written as (see [8])

$$\begin{aligned} P_{\text{NML}}(\mathbf{x}^n, \mathbf{z}^n | \mathcal{M}(K_0)) \\ = \frac{\prod_{k=1}^{K_0} \left(\frac{h_k}{n}\right)^{h_k} \prod_{i=1}^m \prod_{l=1}^{K_i} \left(\frac{f_{ikl}}{h_k}\right)^{f_{ikl}}}{\mathcal{C}_{\text{FM}}(\mathcal{M}(K_0), n)}, \end{aligned} \quad (13)$$

where h_k is the number of times Z has value k in \mathbf{z}^n , f_{ikl} is the number of times X_i has value l when Z has value k ,

and $\mathcal{C}_{\text{FM}}(\mathcal{M}(K_0), n)$ is given by (see [8])

$$\mathcal{C}_{\text{FM}}(\mathcal{M}(K_0), n) = \sum_{h_1 + \dots + h_{K_0} = n} \frac{n!}{h_1! \dots h_{K_0}!} \prod_{k=1}^{K_0} \left(\frac{h_k}{n}\right)^{h_k} \cdot \prod_{i=1}^m \mathcal{C}_{\text{MN}}(K_i, h_k), \quad (14)$$

and $\mathcal{C}_{\text{MN}}(K, n)$ is given by

$$\mathcal{C}_{\text{MN}}(K, n) = \sum_{h_1 + \dots + h_K = n} \frac{n!}{h_1! \dots h_K!} \prod_{k=1}^K \left(\frac{h_k}{n}\right)^{h_k}. \quad (15)$$

The stochastic complexity for the finite mixture model can now be written as

$$\begin{aligned} SC(\mathbf{x}^n | \mathcal{M}(K_0)) \\ = - \sum_{k=1}^{K_0} h_k \cdot \log \frac{h_k}{n} \sum_{i=1}^m \sum_{l=1}^{K_i} f_{ikl} \cdot \log \frac{f_{ikl}}{h_k} \\ + \mathcal{C}_{\text{FM}}(\mathcal{M}(K_0), n), \end{aligned} \quad (16)$$

While the terms $\mathcal{C}_{\text{MN}}(K, n)$ can be computed in linear time in n (see [25]), the sum (14) is clearly exponential and thus computationally infeasible. In [8], however, we presented an efficient recursive formula for computing this sum,

$$\begin{aligned} \mathcal{C}_{\text{FM}}(\mathcal{M}(K_0), n) = \sum_{r_1 + r_2 = n} \frac{n!}{r_1! r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2} \\ \cdot \mathcal{C}_{\text{FM}}(\mathcal{M}(K_0^*), r_1) \cdot \mathcal{C}_{\text{FM}}(\mathcal{M}(K_0 - K_0^*), r_2), \end{aligned} \quad (17)$$

where $1 \leq K_0^* \leq K_0 - 1$. A straightforward quadratic-time algorithm based on this formula presented in [8] allows the use of NML for practical clustering problems.

The clustering space \mathcal{Z} , however, is obviously exponential in size, which means that in practice we need to resort to combinatorial search algorithms in our attempt to solve the clustering problem. The search algorithm used in the empirical tests in [8] was a simple stochastic greedy algorithm. In the next section, we will compare five different algorithms for finding good clusterings using several real-world datasets from the UCI repository.

IV. EMPIRICAL RESULTS

In this section, we will present two sets of results. The first set concentrates on finding the number of clusters and the actual clustering minimizing the stochastic complexity (16). In the second set of experiments, we will test how long it takes for each of the five algorithms to find the minimum SC value.

The first search algorithm candidate is a simple stochastic greedy (SG) algorithm, which was suggested in our previous paper [8]. The details of SG are described in Algorithm 1.

Since our definition of clustering is based on the finite mixture model, the standard mixture learning algorithm, EM (Expectation-Maximization) is a natural choice as a clustering search algorithm. The EM algorithm is iterative and consists of two alternating steps. In the E-step, the current parameters

Algorithm 1 The stochastic greedy algorithm.

```

Choose a random initial clustering
repeat
  Choose a random data vector
  Move the chosen data vector to the cluster locally optimizing
  the SC score
until converged

```

of the mixture model are used to fractionally assign each data vector to the clusters. In the M-step, the parameters are updated based on the fractional assignments. See [26], [27] for more details on the EM algorithm. To obtain an actual clustering from the fractional assignments, in this work the most probable cluster for each data vector is chosen after the EM algorithm has converged.

Our third candidate algorithm is the K-means algorithm (KM), sometimes called the CEM algorithm [28], is a simple modification to the EM algorithm. The difference is that in the E-step, each data vector is fully assigned to the most probable cluster, i.e., no fractional assignments are used.

Each of the described algorithms needs to be initialized prior to the iterative updating procedure. In our tests, we started each algorithm simply by choosing a random clustering. To test the importance of the initialization, we added two hybrid methods to our set of candidate search algorithms. The first hybrid algorithm (KMSG) starts by running the K-means algorithm until convergence and then switches to the stochastic greedy search. The second algorithm (EMSG) is the same except that the EM algorithm is used as an initializer.

It should be noted that we also tested several purely greedy algorithms, such as bottom-up and top-down clustering. However, we noticed very early that these algorithms are very slow and converge to highly suboptimal local optima and consequently were dropped from our tests.

Having fixed the set of candidate search algorithms, the next task is to define a strategy for finding the optimal number of clusters and the actual clustering. Since all the five algorithms converge to a local optimum of the stochastic complexity, the natural strategy is to restart the algorithms several times from different starting points.

Although the NML scoring criterion can be used for comparing clusterings with different number of clusters, the framework does not offer an explicit way to directly infer the optimal number of clusters (K). Consequently, the second part of our search strategy is to vary the parameter K . The complete search strategy is described in Algorithm 2.

In the first batch of results we tested which of the five algorithms finds the best clusterings in terms of the stochastic complexity. Description of the datasets and the results can be found in Figure 1. For all the five algorithms, the minimum SC value found and the corresponding number of clusters is recorded. For each dataset, the minimum stochastic complexity over the algorithms is in boldface.

The first thing to notice about the results is that all the five algorithms seem to end up choosing a similar number of clusters. This means that all the algorithms are useful in the

dataset	size	#attrs	SG		KM		EM		KMSG		EMSG	
			K	SC	K	SC	K	SC	K	SC	K	SC
Australian	690	15	2	5834.5	2	5884.6	2	5844.5	2	5833.8	2	5834.5
Balance	625	5	2	3795.0	3	3811.5	2	3800.5	3	3809.3	2	3795.0
Dermatology	366	35	6	8556.0	5	9083.7	5	8792.0	6	8556.0	6	8556.0
Diabetes	768	9	4	5137.7	3	5245.9	3	5182.5	3	5158.0	5	5144.3
Ecoli	336	8	4	2088.8	3	2116.4	3	2090.9	3	2089.0	3	2089.0
Hepatitis	155	20	3	2266.9	3	2294.0	3	2287.8	3	2266.9	3	2266.9
Ionosphere	351	35	15	10011.3	13	10970.6	12	10339.8	17	10013.0	15	10012.7
Iris	150	5	4	632.6	3	634.6	4	633.5	3	633.9	4	632.6
Liver	345	7	2	1689.6	3	1727.4	2	1702.0	3	1702.9	2	1689.6
Lymphography	148	19	5	2057.3	5	2094.8	5	2074.5	5	2057.3	5	2057.3
Vehicle	846	19	13	10722.2	11	11227.0	13	10781.6	13	10712.8	13	10710.0
Tic-Tac-Toe	958	10	18	8921.5	17	9291.0	17	8888.4	19	8939.4	17	8888.4
Wine	178	14	3	2402.2	3	2440.8	3	2403.7	3	2402.2	3	2402.2
Yeast	1484	9	5	9338.3	6	9543.1	4	9385.3	5	9383.0	4	9327.6

Fig. 1. The minimum SC scores and the number of clusters chosen by the candidate algorithms for the UCI datasets.

Algorithm 2 The search strategy used in our tests.

```

repeat
  for all  $D$  in datasets do
    for  $K = 1$  to 20 do
      Choose a random initial  $K$ -clustering for dataset  $D$ 
      for all  $A$  in {SG, KM, EM, KMSG, EMSG} do
        Run the algorithm  $A$  until converged
      end for
    end for
  end for
until 50 restarts have been made

```

task of choosing the optimal number of clusters with respect to the stochastic complexity. However, when we look at the actual SC values, there are significant differences between the algorithms. Since SC can be interpreted as a quality of a clustering, these differences are important. The SG algorithm and the hybrid EMSG are clearly the best ones. One interesting observation is that EMSG beats SG clearly in some of the more complex cases, i.e., when the size of data and the optimal number of clusters is bigger, while EMSG is never significantly worse than SG. The KMSG algorithm is also reasonable good, but it is practically always worse than EMSG.

The traditional KM and EM algorithms are the worst of the candidate algorithms. Especially KM is in some cases extremely poor, which is alarming since KM is one of the most frequently used clustering algorithms. Furthermore, the EM algorithm beats KM every time, which suggests that it is easier to find good quality clusterings by exploiting the “soft clustering” space than by working in the “hard clustering” space alone. This observation was also made in [29].

In the second set of experiments we recorded how much CPU time (in seconds) each algorithm required for finding their respective optimal clustering. The results can be found in Figure 2. For these experiments, we used otherwise the same search strategy as before except that the number of restarts was only 10 and the results were averaged over 5 runs of Algorithm 2.

The most important thing to notice from these results is

dataset	SG	KM	EM	KMSG	EMSG
Australian	1.0	0.3	0.2	0.4	0.3
Balance	1.3	3.4	0.2	0.5	0.2
Dermatology	11.0	14.4	19.7	8.2	7.4
Diabetes	2.0	14.3	5.4	3.7	2.7
Ecoli	0.6	3.2	3.3	0.9	0.2
Hepatitis	0.8	2.2	1.5	0.6	0.5
Ionosphere	121.2	11.8	7.2	132.7	103.5
Iris	0.3	0.4	0.3	0.3	0.2
Liver	0.4	2.5	0.2	0.3	0.2
Lymphography	1.2	2.2	2.1	0.6	0.8
Vehicle	52.6	17.7	48.1	59.9	59.6
Tic-Tac-Toe	1428.5	14.0	30.0	1217.4	240.3
Wine	0.3	1.8	4.4	0.3	0.3
Yeast	19.5	25.4	0.9	15.6	3.5

Fig. 2. The CPU times (in seconds) spend by the candidate algorithms in finding the optimum clusterings.

that the hybrid EMSG algorithm, which we above found to produce comparable or better results than SG, is almost always significantly faster than the SG algorithm proving the intuitive argument that choosing a good initial clustering is important. This makes the EMSG algorithm a clear overall winner in our experiments. The KMSG algorithm is also faster than SG, but slower than EMSG. It is also noteworthy that KM and EM are often much slower than the other algorithms even though they produce inferior results. This makes the applicability of KM and EM even more questionable in the setting used here.

V. CONCLUSION

In this paper, we have extended our previously suggested framework for data clustering based on the idea that a good clustering is such that it allows efficient compression when the data are encoded together with the cluster labels. As a first extension we introduced five optimization algorithms for minimizing the stochastic complexity. Secondly, using these algorithms, we conducted an extensive set of experiments with several real-world datasets. In the first part of the tests we recorded the number of clusters chosen and the quality of the

actual clusterings found by the algorithms. The idea of the second batch of tests was to see how much CPU time each algorithm requires for finding the best solution.

In the empirical results we found out that all the five algorithms were useful if the goal is to find the NML optimal number of clusters. However, the quality of the individual clusterings found by the more traditional KM and EM algorithms was questionable. These algorithms were also found to be slow. The most interesting observation was that the novel hybrid EMSG algorithm produced the best results and was also significantly faster than the SG algorithm used in our previous work.

In these tests, our search strategy was a very simple one. It is a natural topic of our future research to test more elaborate strategies, such as trying to find the optimal number of clusters in a more efficient way than what we did here. Another interesting extension is the development of stochastic greedy type of SC optimization algorithms that would be capable of exploiting the soft clustering search space in a similar manner EM does.

REFERENCES

- [1] A. Jain, M. Murty, and P. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
- [2] P. Smyth, "Probabilistic model-based clustering of multivariate and sequential data," in *Proceedings of the Seventh International Conference on Artificial Intelligence and Statistics*, D. Heckerman and J. Whittaker, Eds., Morgan Kaufmann Publishers, 1999, pp. 299–304.
- [3] C. Fraley and A. E. Raftery, "How many clusters? Which clustering method? Answers via model-based cluster analysis," *The Computer Journal*, vol. 41, no. 8, pp. 578–588, 1998.
- [4] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman, "Autoclass: A Bayesian classification system," in *Proceedings of the Fifth International Conference on Machine Learning*, Ann Arbor, June 1988, pp. 54–64.
- [5] B. Everitt and D. Hand, *Finite Mixture Distributions*. London: Chapman and Hall, 1981.
- [6] D. Titterton, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions*. New York: John Wiley & Sons, 1985.
- [7] G. McLachlan, Ed., *Mixture Models: Inference and Applications to Clustering*. New York: Marcel Dekker, 1988.
- [8] P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, and H. Tirri, "An MDL framework for data clustering," in *Advances in Minimum Description Length: Theory and Applications*, P. Grünwald, I. Myung, and M. Pitt, Eds., The MIT Press, 2006.
- [9] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 445–471, 1978.
- [10] —, "Stochastic complexity," *Journal of the Royal Statistical Society*, vol. 49, no. 3, pp. 223–239 and 252–265, 1987.
- [11] —, "Fisher information and stochastic complexity," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 40–47, January 1996.
- [12] Y. M. Shtarkov, "Universal sequential coding of single messages," *Problems of Information Transmission*, vol. 23, pp. 3–17, 1987.
- [13] A. Barron, J. Rissanen, and B. Yu, "The minimum description principle in coding and modeling," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2743–2760, October 1998.
- [14] P. Grünwald, "The minimum description length principle and reasoning under uncertainty," Ph.D. dissertation, CWI, ILLC Dissertation Series 1998-03, 1998.
- [15] J. Rissanen, "Hypothesis selection and testing by the MDL principle," *Computer Journal*, vol. 42, no. 4, pp. 260–269, 1999.
- [16] Q. Xie and A. Barron, "Asymptotic minimax regret for data compression, gambling, and prediction," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 431–445, March 2000.
- [17] J. Rissanen, "Strong optimality of the normalized ML models as universal codes and information in data," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1712–1717, July 2001.
- [18] B. Dom, "An information-theoretic external cluster-validity measure," IBM Research, Tech. Rep. RJ 10219, 2001.
- [19] M. Plumbley, "Clustering of sparse binary data using a minimum description length approach," Department of Electrical Engineering, Queen Mary, University of London, Tech. Rep., 2002, unpublished manuscript.
- [20] M.-C. Ludl and G. Widmer, "Clustering criterion based on minimum length encoding," in *Proceedings of the 13th European Conference on Machine Learning*, ser. Lecture Notes in Computer Science, T. Elomaa, H. Mannila, and H. Toivonen, Eds., vol. 2430. Springer, 2002, pp. 258–269.
- [21] S. Hettich, C. Blake, and C. Merz, "UCI repository of machine learning databases, University of California, Irvine, Dept. of Information and Computer Sciences," 1998, <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
- [22] P. Grünwald, "Minimum description length tutorial," in *Advances in Minimum Description Length: Theory and Applications*, P. Grünwald, I. Myung, and M. Pitt, Eds., The MIT Press, 2006, pp. 23–79.
- [23] J. Rissanen, "Lectures on statistical modeling theory," August 2005, available online at www.mdl-research.org.
- [24] V. Balasubramanian, "MDL, Bayesian inference, and the geometry of the space of probability distributions," in *Advances in Minimum Description Length: Theory and Applications*, P. Grünwald, I. Myung, and M. Pitt, Eds., The MIT Press, 2006, pp. 81–98.
- [25] P. Kontkanen and P. Myllymäki, "A linear-time algorithm for computing the multinomial stochastic complexity," *Information Processing Letters*, vol. 103, no. 6, pp. 227–233, 2007.
- [26] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.
- [27] P. Kontkanen, P. Myllymäki, and H. Tirri, "Constructing Bayesian finite mixture models by the EM algorithm," ESPRIT Working Group on Neural and Computational Learning (NeuroCOLT), Tech. Rep. NC-TR-97-003, 1996.
- [28] M. Meila and D. Heckerman, "An experimental comparison of several clustering and initialization methods," in *UAI'98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, G. F. Cooper and S. Moral, Eds., 1998, pp. 386–395.
- [29] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," in *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM'02)*, November 2002, pp. 600–607.