Why so many clustering algorithms — A Position Paper

Vladimir Estivill-Castro
School of Electrical Engineering and Computer Science,
University of Newcastle,
Callaghan, NSW, 2308
Australia
vlad@cs.newcastle.edu.au

ABSTRACT

We argue that there are many clustering algorithms, because the notion of "cluster" cannot be precisely defined. Clustering is in the eye of the beholder, and as such, researchers have proposed many induction principles and models whose corresponding optimization problem can only be approximately solved by an even larger number of algorithms. Therefore, comparing clustering algorithms, must take into account a careful understanding of the inductive principles involved.

Keywords

Clustering, Inductive Principle, Clustering criterion

1. INTRODUCTION

Clustering is a central task for which many algorithms have been proposed. Regularly, new variants of older methods, or new approaches emerge while industry demands the establishment of benchmarks for contrasting these algorithms. Why is it that there is such a large family of clustering algorithms and such benchmarks are so difficult to develop?

We argue here that there is much confusion at what is at stake and what is what should be compared. We will first indicate that the diversity in clustering algorithms is in fact due to the diversity of *induction principles* and *models*. Thus, we will explain what are inductive principles and what are models. Then, for a single family of models and an inductive principle, there are many algorithms to approach the resulting optimization problem. However, it is within the context of a fixed inductive principle that benchmarks can be established for comparing clustering algorithms. Thus, there are many clustering algorithms because there are many algorithms for each inductive principle and there are many inductive principles. Why are there so many inductive principles? Because clustering is in part in the eye of the beholder. Inductive principles are just mathematical formalizations of what researchers believe is a definition of *cluster*. What constitutes a cluster, or a good clustering, has biases because of the background of researchers and because of the application. This domain knowledge is what constitutes the belief that there are subgroups among a bulk of data about objects, and such beliefs mold the structures used to represent those groups. Thus, differences on inductive principles are philosophical but fundamental. Nevertheless, we propose some mechanisms that may be used to compare inductive principles and then to compare clustering algorithms.

The paper is organized as follows. First, we provide the necessary definitions, notation and examples to clarify the notions of *inductive principle*, *models* and *algorithms*. Section 3 presents some problems derived from the lack of explicit discussion of models and induction principles when presenting an algorithm. It intends to illustrate that diversity is necessary and useful; but has lead to some confusion about the properties of algorithms. The effect of this lack of clarity and detail is a difficulty to compare algorithms. Section 4 summarizes our recommendations. The Appendix. presents an illustration of the points in Section 3.

2. DEFINITIONS

Cluster analysis has been identified as a core task in data mining [30]. The definitions in the literature of what constitutes clustering reflect the different philosophical points on the matter. The top-down view regards clustering as the segmentation of a heterogeneous population into a number of more homogeneous subgroups [3]. A bottom-up view defines clustering as finding groups in a data set "*by some natural criterion of similarity*" [12]. There are others who believe that the fundamental question is if two items are or not in the same cluster. Specifying the natural criterion of similarity of points is a step towards indicating how to account for similarity between groups. Some intuitive notion of what constitutes cohesive groups result in an induction principle. In fact, most authors find difficult to describe clustering without some suggestion to grouping criteria. For example, "*the objects are clustered or grouped based on the principle of maximizing the inter-class similarity and minimizing the intra-class similarity*" [30, Page 25].

Clustering applications have grown since all sorts of scientific disciplines are based on built or discovered classifications that structure knowledge. In KDD, this is much more apparent, since we are supplied with constantly growing data sets for which we are to discover those classifications. Clustering generates concepts provides generalization, data summarization and is an inductive process. Given a data set, any clustering (produced by an algorithm or a human) is a hypothesis to suggest (or explain) groupings in the data. This hypothesis is selected amongst a large set of possibilities and is represented (structured) in some way. It becomes a model for the data, and can potentially constitute a mechanism to classify unseen instances of the data.

What discriminates one hypothesis over another given the same data set? This criterion is what we refer to as the mathematical formulation of the inductive principle. It has also received the name *clustering criterion* [20; 28; 27]. Thus, models are the structures we are willing to use to represent clusters, while the induction principle selects a "best fit" model given a data set.

For simplicity we assume that the data $\mathcal{D}$ is given as n d-dimensional vectors $\{\vec{x}_1, \ldots, \vec{x}_n\} \subset \Re^d$ (although, some other forms of input are possible, like n object identifiers and a similarity function s that measures the separation $s(i,j)$ between each pair i,j of objects). Lets look at three examples of induction principles. The literature of parametric statistics (sometimes referred as statistical inference) will attempt to fit a probabilistic model to the data. Typically, mixture models of normal distributions. In this case, the researchers is inclined to believe that the data $\mathcal{D}$ is the result of a fixed number k of classes, each of which is distributed according to a multivariate normal distribution $N_{\vec{\mu_i},\Sigma_i}$ and that they are combined by a vector of non-negative proportions $\vec{\pi}^T = (\pi_1, \ldots, \pi_k)$ (with $\sum_{i=1}^{k} \pi_i = 1$). Then, the probability distribution is given by $Prob(\vec{x}) = \sum_{i=1}^{k} \pi_i N_{\vec{\mu_i},\Sigma_i}(\vec{x})$. Most importantly, the researcher works under the assumption that each data item was independently drawn form that mixture, and believes that future (new) data will result from such a mixture and what is actually missing are the parameters $\vec{\pi}$, $\vec{\mu_i}$ and Σ_i. Once these parameters are estimated (discovered), we can know the proportions in the clusters, their location and scatter, and for each individual item, the probability that it belongs to the i-th cluster. We can use the model to do predictions, etc.

But how to chose the set $(\vec{\pi}, (\vec{\mu_i}, \Sigma_i)_{i=1,\ldots,k})$ that best explains the data. What is the inductive principle? A traditional approach is Maximum Likelihood(ML). This says "*choose the model that maximizes the probability of the data being generated by such model*" [34]. For many models, the observed data has probability different of zero, but we asses the model as best in proportion to its likelihood. This allows an explicit mathematical expression for the inductive principle. We have an optimization problem!

$$\begin{aligned} \text{Maximize } & \mathrm{ML}(\vec{\pi}, (\vec{\mu_i}, \Sigma_i)_{i=1,\ldots,k}) \\ &= Prob(\mathcal{D} | (\vec{\pi}, (\vec{\mu_i}, \Sigma_i)_{i=1,\ldots,k}). \end{aligned} \quad (1)$$

How we solve this multi-variate optimization problem? What is the common algorithm used in this case? By equating the gradient of $\log \mathrm{ML}$ to zero, sufficient (but not necessary) conditions for the optima are obtained that lead to an iterative algorithm (heuristic) [47; 50]. This iterative algorithm converges to local optima an is the well-known EXPECTATION MAXIMISATION method [9].

But what if we are convinced that each data item belongs to one and only one class, instead of having a degree of membership to each class assessed by a probability? What if we change our model? What can we use then as our inductive principle? Because we are to partition a set into more homogeneous clusters, we need to asses homogeneity. The statistical theory of multivariate analysis of variance suggests a starting point for evaluating homogeneity. This starting point is the structure of total scatter matrix T. A traditional measure of the size of this matrix is the *trace*. The trace of a square and symmetric matrix is just the sum of its diagonal elements, and thus minimizing $trace[T]$ is exactly the least sum of squares loss function (known in the statistics literature as L_2 [46]):

$$\text{minimize } L_2(C) = \sum_{i=1}^{n} \mathrm{EUCLID}^2(\vec{x}_i, \mathrm{REP}[\vec{x}_i, C]), \quad (2)$$

where $\mathrm{EUCLID}(\vec{x}, \vec{y}) = [(\vec{x} - \vec{y})(\vec{x} - \vec{y})^T]^{1/2} = [\sum_{j=1}^{D} |x_j - y_j|^2]^{1/2}$ is the Euclidean metric; $C = \{\vec{c}_1, \ldots, \vec{c}_k\}$ is a set of k *centers* , or representative points of $\Re^d$; and for $i = 1, \ldots, n$, the point $\mathrm{REP}[\vec{x}_i, C]$ is the closest point in C to $\vec{x}_i$. Note that Equation (2) is the search for a set C of k representatives. This is illustrative of representative-based clustering: the partition into clusters is defined by assigning each $\vec{x}_i$ to its representative $\mathrm{REP}[\vec{x}_i, C]$.

The k-MEANS method produces an approximate solution to Equation (2) by iteratively refining the partition encoded by the representatives.

The inductive principle formalized in Equation (2) says "*pick the model (set of k representatives) that minimizes the total squared error*". The problem with means is that they are very sensitive to outliers. The value $\mathrm{EUCLID}^2(\vec{x}_i, \mathrm{REP}[\vec{x}_i, C])$ is the squared error of representing $\vec{x}_i$ by its representative and will be dramatically large when $\vec{x}_i$ is an outlier. A more robust estimator of location is the median [46]. This corresponds to the induction principle that says "*pick the model (set of k representatives) that minimizes the total absolute error*". The mathematical formulation for this clustering criterion is

$$\text{minimize } L_1(C) = \sum_{i=1}^{n} \mathrm{EUCLID}(\vec{x}_i, \mathrm{REP}[\vec{x}_i, C]). \quad (3)$$

While Equation (2) and Equation (3) only differ on the squaring of the error, we will see in the next section that they constitute radically different computational problems. [1] For example, in the case of just one cluster ($k = 1$), the global minimum of Equation (2) can be obtained in $\Theta(n)$ time while for more than one dimension, Equation (3) is theoretically intractable.

Aldenderfer and Blanshfield state that "*clustering is structure imposing*"[3]. The universe of structures we are prepared to believe are suitable to reflect the variations in density on a set of data constitute our models. An inductive principle will make explicit a criterion to select a best-fit for a given data set. For a given inductive principle, each new data set will posse an optimization problem. Namely, find the model that optimizes the criteria given the data. Since the resulting optimization problems are typically intractable or their algorithmic complexity is beyond what would be acceptable in practice for the large data sets that KDD has in mind, the optimization problem is solved approximately. For the same optimization problem there are usually may heuristic algorithms that will trade-off the quality of the optimization (the quality of the local optima) for their computational effort.

[1] Note that L_1 and L_2 are used in the statistical literature for loss functions that measure error; we use them here in this sense. However, the literature in metric spaces uses L_p for the p-th Minkowsky metric; with L_1 usually know as the Manhattan metric and L_2 used as the Euclidean metric. We do not use L_p here in this sense.

3. THE PROBLEMS

The notions of induction principle, model and clustering algorithm, apply in a general manner to clustering algorithms. We say that a *context* for a clustering algorithm is the pair (model, induction principle) that applies to the algorithm. Because the context around a clustering algorithm is left to implicit interpretation, rather than explicitly described, and because research efforts have been focused on other desiderata (like scalability or applicability to data formats), confusion has emerged whose direct effect is the difficulty in comparing algorithms and contrasting clustering results. We elaborate on what are some of these sources of confusion.

3.1 Model or Induction principle

Representative-based clustering typically finds a prototype item for each cluster. Algorithms for this approach include EXPECTATION MAXIMISATION [9], k-MEANS [40], FUZZY-c-MEANS [5], and many others [21]. The terminology to denominate the representatives of clusters as "centers", "means", "medians" and even "medoids" has proliferated freely to the point that its difficult to identify if the name refers more to the model or to the induction principle. It is well established in the statistical literature that the *observed mean* is the point $\vec{\mu}$ that minimizes $L_2(\vec{\mu}) = \sum_{i=1}^{n} \text{EUCLID}(\vec{x}, \vec{\mu})^2$. This is the case $k = 1$ for Equation (2). It can be solved analytically using multidimensional calculus and results in $\vec{\mu} = \sum_{i=1}^{n} \vec{x}_i/n$. The observed mean can be computed in $\Theta(n)$ time and no faster algorithm is possible. This is an unbiased estimator of the mean of a multivariate normal distribution under the assumption that this is the probabilistic model for the data. However, the median of a cloud of points is obtained from a different induction principle that removes the squaring. Namely, minimize $\text{FW}(\vec{\mu}) = L_1(\vec{\mu}) = \sum_{i=1}^{n} \text{EUCLID}(\vec{x}, \vec{\mu})$. This problem is intractable for dimensions $d > 1$ [4]. The solution is the famous Fermat-Weber center [38; 39]. Thus, the distinction between "mean" and "median" is a distinction on the induction principle. While the median is statistically superior on notions like robustness and resistance [46], the mean is algorithmicly easy since its computation requires $O(n)$ time. Researchers have restricted the universe of models because of the algorithmic complexity of using medians in more than one dimension (the case $d > 1$). That is, the universe where the solution is searched for is all points in d dimensional space (for the case of k representatives, Equation (2) and Equation (3) remain continuous optimization problems where the universe is the sets of k points in d-dimensional space). By adding the restriction $\vec{\mu} \in \mathcal{D}$, these problems become discrete optimization problems. The discrete mean is the solution to

$$\begin{aligned} \text{minimize } L_2(\mu) &= \sum_{i=1}^{n} \text{EUCLID}(\vec{x}, \vec{\mu})^2 \\ \text{restricted to } & \quad \vec{\mu} \in \mathcal{D}. \end{aligned}$$

This is also solved in linear time, since it is the closest point to the observed mean. The discrete median problem is

$$\begin{aligned} \text{minimize } \text{FW}(\vec{\mu}) &= L_1(\vec{\mu}) \\ &= \sum_{i=1}^{n} \text{EUCLID}(\vec{x}, \vec{\mu}) \\ \text{restricted to } & \quad \vec{\mu} \in \mathcal{D}. \end{aligned}$$

This has a trivial $O(n^2)$ algorithm which essentially consist of exhaustively evaluating $\text{FW}(\vec{x})$ for all $\vec{\mu} \in \mathcal{D}$.

It seems that the term "medoid" is used interchangeably to refer to the discrete median and the discrete mean [41; 30]; although, it certainly would be more appropriate to use it only for the discrete median. The term "medoid" was originally introduced to indicate discrete median [36, Chaper 4] in the statistical literature and imported [41] to KDD with exactly the same meaning of representatives that are data points and minimize the absolute error.

In any case, this reflects the interest in the KDD community to distinguish clustering algorithms more on the models (which is more about the computability) than on the induction principle (which is certainly an issue on the quality of the clustering). The difference in induction principle is the more fundamental. Solving L_2 with continuous or discrete centers is sensitive to outliers and noise [46]. The statement "*the k-medoids method is more robust than the k-means method in the presence of noise and outliers*" [30, page 353] is misleading since the algorithms described in [30] are both using L_2 as the cost function.

In the appendix, we illustrate that the quality of the clustering in the presence of noise would not be comparable even if we were to find the global optimum to Equation (2) (by an algorithm far better that k-MEANS in approximating the solution). A good approximation of Equation (3) with the model restricted to $(C \subset \mathcal{D}) \wedge (||C|| = k)$ [19] is superior. The latter is an inductive principle whose solutions are robust to noise (in the same statistical sense in which medians are robust estimators of location and means are not [52]). Thus, many of the disadvantages attributed to k-MEANS are really inherited from its context. The k-Medoids methods in [30] and its variants (like CLARANS) will not alleviate these problems if they use a less robust inductive principle.

3.2 Categorization of algorithms

Despite the quality of the results is primarily dependent of the induction principle rather than on the model, the emphasis on computability/scalability of the KDD literature has shaped categorization of clustering algorithms on the model with little emphasis on the induction principle. These categorizations of clustering methods are along dimensions that include the type of algorithm used in the optimization and under close scrutiny, their boundaries are blurry. The final result are misleading categorizations of clustering algorithms. Han & Kamber [30] offers one of these many categorizations. (but others could be scrutinized similarly, another example of similar categorization is by Jain et al [33]). For example, in [30], the category named "Partitioning Methods" corresponds more accurately to representative-based clustering (again, k-MEANS, EXPECTATION MAXIMISATION, FUZZY-c-MEANS, etc) that are clearly methods based on mathematical models (probabilistic models, and fuzzy membership models). In fact, all the examples supplied for this category are examples of representative-based clustering, ignoring examples of partitioning clustering not based in representatives (like methods based on the inductive principle of Total within Group Distance [17] or methods based on finding cuts on graphs [14; 35]).

There is no reason to have another category for "Model-based methods". The methods comprising this category are AUTOCLASS and CLASSIT, also based on mathematical models of distributions from probability theory and statistics. Perhaps AUTOCLASS is closer to Bayesian statistics while Expectation Maximisation is closer to parametric statistics. Nevertheless, the boundary between these two categories of clustering algorithms is certainly insufficient to justify their separate naming.

But, are other categories (like Hierarchical Methods, Density-Based methods and Grid-Based methods) not based on models? The position of this paper is that they are certainly based on models. Not continuous mathematical models like probability distributions, but on discrete, structural models. For hierarchical methods, their context includes an induction principle as well. Hierarchical methods search the tree that best fits the data among the family of all trees with the data at the leafs. And in fact, several criteria have been proposed as explicit formulations of the induction principle. In almost all cases, the corresponding optimization problems have been shown to be NP-Hard or NP-complete [8; 25]. Consider why the dendrograms produced by these algorithms are binary. That is, why do divisive algorithms split a cluster into two (and not k parts) and why do agglomerative algorithms merge just the next two closest clusters and not consider merging more? Simply because criteria, like complete linkage and so an are NP-complete for split numbers larger than 2 [25; 8, page 281].

Thus, the family of hierarchical clustering algorithms is also a series of heuristics to find good approximate solutions.

Structural models also are the type of models used by density-based algorithms. Just think what type of output this algorithms produce[2]. Algorithms like CHAMELEON [35] and DBSCAN [15] produce as output a graph G. This graph G represents (encodes) the relation $\vec{x}_i$ is in the same cluster as $\vec{x}_j$ with the predicate "there is a path in G from $\vec{x}_i$ to $\vec{x}_j$". That is, these algorithms search for the graph that best fits the data. The graph may be represented by its adjacency matrix (although this may require $\Omega(n^2)$ time and space, which may be infeasible for Data Mining). Also, many implicit mathematical models are hidden under the algorithms (for example, a simple model to eliminate outliers is inside CURE [26], while a model for the types of distributions is implicit in BIRCH [56] which maintains a tree of nodes with estimators of location and scatter). Even methods like DENCLUE [31] are based on kernel models and selecting these kernels is crucial. The convergence of neural network algorithms for unsupervised learning is based on showing that some measure of error (perhaps the total squared error) is improved, and again this is a reflection on an iterative optimization (gradient descent) converging on a local optima.

However, the presentation of clustering algorithms in the database literature becomes more and more imprecise about the inductive principle, and fundamentally, proponents of these methods leave undefined what are "good clusters". They delegate this responsibility to the user by making the algorithms depend on arguments supplied by the user (so called parameters of the algorithm). As a consequence, the results vary widely with changes to these user-supplied arguments. Because of this, comparisons between these algorithms seem impossible or at best inconclusive. The positive side is that the types of representations for clusters is extended. We have more flexible representations, and thus, other intuitive notions of clusters seem supported (like non-convex clusters).

[2] Categorizing Data Mining and Machine Learning algorithms by their type of output format is not unusual [54].

The point is that differences that impact heavily on the quality of results are relative to their context (model, induction principle) and not only to the working of the algorithm. One more illustration is the distinction between partition methods and hierarchical methods. Partitioning algorithms result directly into so called *Divisive hierarchical clustering*. Simply, the top-down strategy applies the partitioning algorithm at each node of the hierarchy. Hierarchical algorithms with a 'criteria' for stoppage (an induction principle) result in partitioning algorithms. Similarly, crisp membership of items to a cluster versus non-crisp membership is simply and issue of modeling. The actual working of algorithms like Fuzzy-c-Means, Expectation Maximisation, k-Means, Harmonic-k-Means is extremely similar [16; 22].

Some researchers have created a direction that considers the size or complexity of the model in the formulation of the induction principle. For example, MDL [45] and MML [51] are pioneer approaches in this regard.

3.3 Trade-offs in optimization

So why do we insist on using algorithms whose context (models, induction principle) is known to be inferior at fundamental levels than others. The simple answer is that the computation of the objective function (the evaluation of a feasible solution for optimization) may be much less costly. Consider criteria like the Total Within Group Distance. Given a partition $P = \{C_1|C_2|\ldots|C_k\}$ of the data into k non-empty, not overlapping ($C_i \cap C_j = \emptyset$, $i \neq j$) sets, the criteria is formalized as follows.

$$\text{Minimize } TWGD(P) = \sum_{j=1}^{k} \sum_{\vec{x}_i, \vec{x}_{i'} \in C_j} dist(\vec{x}_i, \vec{x}_{i'}). \qquad (4)$$

Evaluating this criterion in one model (a partition of the data into k clusters) requires $\Theta(n^2)$ time because $\Theta(n^2)$ distance evaluations are involved. However, evaluation of Equation (2) on a partition can easily be performed with $O(kn)$ distance evaluations. Therefore, a hill-climber type of algorithm like k-Means will be extremely fast with respect to an equivalent naive hill-climber for Total Within Group Distance.

Therefore, we observe that it is natural to prefer optimizing an induction principle $\mathcal{P}_1$ whose objective function is easier to evaluate than to optimize an induction principle $\mathcal{P}_2$ whose objective function is much costly to evaluate. Although not proved formally, much of this approach is current practice and hopes that the optimization of the simpler objective function results in a good approximation to the complex objective function. The point of this paper is that this calls for a comparison between induction principles, not between clustering algorithms. Clustering algorithms should be compared in a first instance for a common context (model, induction principle).

For example, the Fuzzy-c-Means algorithms is an iterative algorithm (a la k-Means) to optimize the following problem.

$$\begin{aligned} &\text{minimize } Fuzzy_b(C) \\ &= \sum_{j=1}^{k} \sum_{i=1}^{n} [\text{MEM}_j(\vec{x}_i)]^b \, \text{Euclid}^2(\vec{x}_i, \vec{c}_j), \end{aligned} \qquad (5)$$

restricted to $\sum_{j=1}^{k} \text{MEM}_j(\vec{x}_i) = 1$ (for $i = 1, \ldots, n$) where the parameter $b \geq 1$ regulates the degree of fuzziness. However, the same optimization problem has been attempted recently with Genetic Algorithms [29]. In this case, the Genetic Algorithms and the iteratively local search can be compared rather objectively as to which obtain the best value for Equation (5) given the same computational resources (function evaluations).

The point here is that clustering with genetic algorithms, simulated annealing or other search/optimization techniques besides local search (and hybrids of these) offer better clustering algorithms if they are more effective in the trade-off of quality of the answer versus computational resources with respect to the same clustering criteria [16].

3.4 Cluster validity

The popularized definition of KDD [24] postulates it as the "*non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data*". We are unsure if valid is listed amongst the first characteristics in proportion to its importance, but certainly, patterns in data will be far from useful if they were invalid. A more cynical view would say that trivial processes can certainly deliver invalid, understandable and novel patterns. So, the validity of results is of up-most importance. But, how is this achieved in clustering algorithms?

Validity is a certain amount of confidence that the clusters found are actually true [11]. That is, the hypothetical structure postulated as the result of a clustering algorithm must be tested to gain confidence that it actually exists in the data. Kloesgen and Zytkow state that "*validation checks a pattern instance (or component of an instance) referring to the subset of data in the sample set which is connected with the instance. To check an instance, usually a statistical test or some other criteria are validated. Additionally to the decision, whether an instance is valid or not, often also an evidence measure (the statistical significance or some other kind of conspicuousness of a pattern instance) is calculated*" [37]. We would be required to test if the clusters found by the algorithm are "*real*" [page 16] [3]. This question has faced many difficulties, and Bezdek [5] dedicated a full chapter to it. Bezdek realized that it seemed impossible to formulate a theoretical null hypothesis used to substantiate or repudiate the validity of algorithmically suggested clusters. The simplest of questions seemed to be, are there truly k groups. Bezdek suggested that indexes of error with respect to already clustered data are "*useful expedient for performance comparisons of different algorithms*" [5, page 95]. He considered extending those indexes to be used to select the number c of clusters in FUZZY-c-MEANS. This question of how many clusters are truly in the data was also critical for cutting dendrograms or stopping hierarchical clustering algorithms [23]. However, there is no definition of "cluster" and as such, the "*concept of no structure in the data set (one possible null hypothesis) is far from clear*" [3] (although three types of reference population under the null hypothesis are used for small data sets [32; 49]).

Unfortunately, some of the literature on "cluster validity" has rapidly evolved after Bezdek's work (specially in the area around FUZZY-c-MEANS [6; 42; 53; 44]). The many developed indexes are now used [43] to plug-in clustering results for algorithms that require the number k as an user supplied argument (typically, for partitioning algorithms like k-MEANS, EXPECTATION MAXIMISATION, FUZZY-c-MEANS). Clustering is repeated with the argument k tested for a range of values $k \in \{2, \ldots, K_{\max}\}$. The value of k resulting in the smallest value for the index is considered the "valid" clustering. While some researchers may be seduced by a perception that results are trustworthy this is far from actually providing a real test for validity.

In an attempt to allow the subjective nature of what constitutes a cluster, the identification of clusters may involve the end-user [2]. This creates potentially useful exploratory tools [1]. Placing the user in the loop allows flexible and application oriented clustering but because of its incorporation of human bias the question of validity becomes even more critical. The exploratory analysis may not discover new true clusters but allowing the user to select views/projections of the data reflecting preconceived notions. Thus, a subsequent confirmatory phase with sound statistical methodology is essential after the exploratory phase.

Clear and comprehensive descriptions of the statistical tools available for cluster validity are Chapter 4 in the book by Jain and Dubes [32] and Chapter 16 in the book by Theodoridis and Koutroumbas [49]. Dubes [11] offers a uniform view of the methodology towards validity, but clearly, what constitutes data with no structure (and is simulated generation) are central to the entire process. Consider two dimensional datasets layered in the pattern of chess-board where parameters like the granularity of the grid and the discrepancy in density on black and white regions varies. Despite the regular structure in such a family of data sets, any clustering algorithm will be able to test only a finite set of hypothesis in finite time. So there are many datasets of these family that will appear to have no structure.

We will show here that this so called family of *internal validity* techniques is simply a larger family of induction principles that perhaps guides the optimization of the user-supplied arguments in a family of clustering algorithms. These internal validity measures are based solely on the data to be clustered (without any class labels). In that sense, we agree with Dom [10]: "*Why not just use this measure itself as an objective function for clustering?" This may be in fact possible in some cases where the objective function used does exactly capture what is desirable in a particular application and there is a feasible algorithm for finding the optimal clustering. In such cases the (quality [sic]) evaluation is moot.*"

We illustrate internal validity with some induction principles discussed here earlier. Formal clustering criteria have been provided (Equation (2), Equation (3) and Equation (4)). However, how do we select the value k for the number of clusters required by the algorithms that approximately solve this optimization problems? The difficulty is that these criteria (and many others) are monotonically decreasing in k. Thus, minimizing the criteria set by the induction principles along the argument k results in the trivial clustering that places each point on a cluster by itself.

Researchers have suggested to create an alternative criteria to decide on the value of k. In the KDD literature, the *silhouette* coefficient is mentioned [30] because it was imported by Ng and Han [41] from the work of Kaufman and Roussseeuw [36]. This is one of the many indexes that follow the generic scheme consisting of evaluating the ratio of inter-cluster separation and intra-cluster similarity (like Dunn's partition coefficient and its variants [13]). However, Kaufman and Roussseeuw [36, page 121] admit that "*like most validity coefficients, the average silhoute $\bar{s}$ could be used as an objective function for the clustering itself (that is, one might want to find a clustering that maximizes $\bar{s}$)*".

A simple example of this family of validity coefficients [43] is to study a specific trade-off in cluster cohesion and separation. Letting $L_2(C)/n$ be a measure of cohesion (the average squared representation error), we can define a measure of separation as $Sep(C) = \min_{\vec{c}_i, c_j \in C} \text{EUCLID}^2(\vec{c}_i, \vec{c}_j)$; that is, the smallest squared distance between two representatives. An algorithm is executed many times with its argument k set to many values in $[2, n]$. Then, the value of k selected is the value that minimizes the objective function $L_2(C)/(nSep(C))$ among the results produced in the many runs. The objection put forward here is the use of these types of objective functions as "validity indexes" while in fact they are just clustering criteria. If we directly optimize $L_2(C)/(nSep(C))$, and obtain a lower value (rather that optimizing for each given k), would the results be more valid? Of course not. There is a large list of proposed indexes for validity that essentially correspond to formulations of clustering criteria [11]. They do not seem to be used as the objective function for a clustering algorithm for apparently two reasons.

- Some of them do not have a minimum, and are rather used with the help of visualization to search for a change in the plot generated by the index.
- They are very expensive to evaluate, and any optimization heuristic would have to pay the performance penalty of costly evaluation of the objective function.

The point is that in the absence of anything else, (like a Monte Carlo procedure [3] and an effective way for generating data sets with no structure [32; 49]) these indexes are merely more elaborate mathematical formulations of clustering criteria reflecting an induction principle.

Another very common approach for internal validity has been named *cophenetic correlations*. Although believed to be applicable only to hierarchical clustering algorithms, the idea is simple enough to be applicable in general. The approach assumes that we have the dissimilarity matrix $D = [d_{ij}]$ where $d_{ij} = \text{dissimilarity}(\vec{x}_i, \vec{x}_j)$. This is the input to the clustering algorithm and it is usually supplied as a function (because if represented explicitly as a matrix, it demands $\Theta(n^2)$ space). The output of the clustering creates an implied similarity matrix $M = [m_{ij}]$. One simple implied similarity matrix is

$$m_{ij} = \begin{cases} 0 & \text{if } \vec{x}_i, \vec{x}_j \text{ in same cluster} \\ 1 & \text{if otherwise.} \end{cases} \tag{6}$$

A representation of such a matrix is the result of many algorithms (like DBSCAN [15]). The validity then consists of assessing the distance (correlation) between the matrix D and the matrix M. As a validation technique, this approach is regarded as one of the weakest [3] (although others consider suitable under carefully chosen assumptions [32]). In some cases, it is obvious why this just represents an induction principle. Consider the following clustering procedure for applications where we have a dissimilarity matrix D with all the values between 0 and 1. Construct a matrix A simply rounding D. Values larger that 0.5 in D are rounded to 1 in A and value smaller than 0.5 in D are rounded to 0 in A. Then, find connected components in A to produce clusters. In other words, M is the transitive closure of A. The fact that M is almost a direct function of D is certainly going to indicate high correlation between M and D, but are the clusters found by this method valid?

An approach to clustering based on a simple matrix M given by Equation (6) has been used for clustering WEB documents [55]. So let M the simple implicit matrix derived from a clustering result (from a partition) and given by Equation (6), and consider two of these indexes that compare matrix D and the matrix M [32].

1. The Hubert Γ statistic is given by
$$\frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} m_{ij}. \tag{7}$$
2. The Normalized Γ statistic is given by
$$\frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{(d_{ij} - \hat{\mu}(D))(m_{ij} - \hat{\mu}(M))}{\sigma(D)\sigma(M)}, \tag{8}$$
where $\hat{\mu}(X)$ is the observed mean in the values of matrix X and $\sigma(X)$ is the observed variance in the values of matrix X.

Equation (7) and Equation (8) are both functions of the partition of the data. The Normalized Γ has range [-1.1]. It is possible to imagine a clustering algorithm that performs a hill-climbing of the space of all partitions into k clusters optimizing either of these functions. This optimization should not be interpreted as obtaining valid clusters, but it would constitute interesting clustering algorithms. Algorithms are a product of optimizing an induction principle, validity is a function of the data set.

Typically, these indexes have to be used inside some more sophisticated procedures for validity (like Monte Carlo methods [3] and their suitability depends on the application, the randomness hypotheses and the internal/external/relative methodology [32]). But naturally, this also implies a proliferation of indexes for validity, as they reflect different types of structure that are sought for in the data.

How can we compare clustering algorithms if we cannot tell if the results are valid? We believe researchers can approach this issue in two ways. First, use external validity measures. That is, use datasets with known structure and evaluate how much structure is recovered by the clustering algorithm. Then, publicize and distribute freely those data sets and detail the process of production so others can replicate the results. Later, validity indexes can be used to asses if algorithms (not designed with the induction principle of the validity index in mind) can also work to obtain good solutions on a different context. The comparison would now be on the context of the validity index. This second alternative will not allow researchers to conclude that one algorithm is superior to another beyond the context of the induction principle reflected by the particular validity index used. An algorithm may be considered better than another if it surpasses the performance of another across a large range of validity indexes.

However, it is unlikely that a set of validity indexes is accepted as a benchmark on which clustering algorithms are evaluated. This will restrict the possible points of view on what constitutes "good clusters" to the induction principles mathematically formulated as validity indexes. Nevertheless, I encourage researchers who postulate new clustering algorithms to be more explicit about what is to be regarded as good clusters. I suggest to perform and report empirical evaluations on those clustering criteria (induction principles or validity indexes) they consider suitable contexts for the proposed algorithm.

4. CONCLUSIONS

The nature of clustering is exploratory, rather than confirmatory. The nature of data mining is that we are to find novel patterns, that is, previously unknown and unsuspected patterns. "*The strategy of cluster analysis is structure seeking although its operation is structure-imposing*" [3]. There are many ways in which the potential structure could be imagined and represented. Already in 1964, Bonner argued that there could not be a universal definition of cluster and that it was to late to impose one [7]. Much more recently, we find that clusters and outliers are in the eye of the beholder: "*one person's noise could be another person's signal*" [30]. This is an element that contributes to the diversity. Out of all those possible models for the patterns, we need to extract only those that are intuitively good clusters. Again, there are many ways in which the inductive principle can be formalized into a clustering criterion.
We conclude then with a summary of recommendations.

1. Do not forget that clusters are, in large part, on the eye of the beholder.

2. Different researchers make explicit what are good clusters by different mathematical formula. There is a richness in this diversity.

3. While it is natural that surveys or compilations of clustering algorithms categorize (or build taxonomies) based on models more than induction principles, these categorizations should not imply the non-existence of models or induction principles. The strongest distinction between clustering algorithms is between the adoption of mathematical (continuous) models (common in statistical inference) and the structural (discrete) models (common in machine learning).

4. Clustering translates into optimization problem whose computational complexity is typically intractable and which are solved by approximation algorithms.

5. The first level of comparisons between two clustering algorithms, is in terms of the quality of the solution as measured by the **same** objective function and on equal computational resources (for example, number of evaluations of the objective function).

6. There are objective functions (inductive principles) that are less costly than others. It is natural to seek to optimize those that are less costly in the hope that solutions to these would be good solutions for the costly objective functions. However, this implies that one must investigate the relationship between the inductive principles (and at a second stage, between the clustering algorithms).

7. There is a way to compare inductive principles P_1 and P_2 by obtaining formal results that indicate how the good solutions for P_1 rank among the good solutions for P_2. Some of these can be clear and formal notions, like statistical robustness.

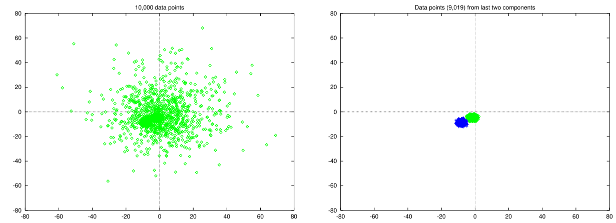

Figure 1: (a) A data set. (b) The data of last two components.

8. Researchers should try to make (mathematically) explicit what are their models and their inductive principle when proposing new or improved clustering algorithms. This should facilitate further investigations and comparison with previous and emerging methods.

9. Indexes of clustering validity are direct mathematical formulations of induction principles. Comparing algorithms on those can provide some insight about the contexts in which one algorithm performs better than another. However, this shall not imply than one algorithm produces more valid results than another. Validity depends on the data set where a claim of existence of structure is made. The two algorithms on a set with no structure will both produce invalid results.

10. Quality of clustering can be demonstrated by external criteria of validity and associated measures. How much of the (known) structure can the algorithm recover. In this sense, data sets for supervised learning can be used for evaluating clustering algorithms. We measure the discrepancy between the clustering results and the known labels.

11. An algorithm designed for some universe of models has no chance if the data sets have a structure that is actually representable by a radically different family of models (k-MEANS can not find non-convex clusters).

5. ACKNOWLEDGMENTS

The support during most of this research by the Department of Computer Science, University of Victoria, Canada, is much appreciated. Special mention to Dale Olesky and Daniel Germán for assisting and supplying many tools towards the preparation of the manuscript and plots.

APPENDIX

A. AN EXPERIMENTAL ILLUSTRATION

We present experimental results that illustrate some of the points made earlier. We use low dimensional examples for illustration and visual clarity, but the arguments can only be more severe in larger dimensions.
Our first experiment will show that the inductive principle formulated by Equation (2) is less effective than the induction principle formulated by Equation (3).
The data used in the experiment are 10,000 2-dimensional points generated by a mixture of 4 bi-variate normal distributions with circular level curves (diagonal covariance matrices). The mixture $\sum_{i=1}^{4} \pi_i N_{\vec{\mu}_i,\Sigma_i}(\vec{x})$ has proportions

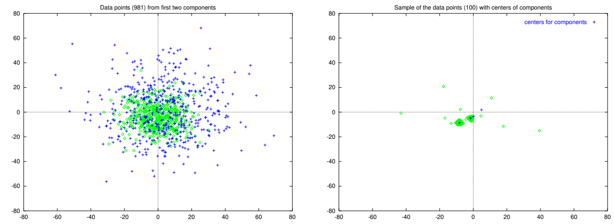

Figure 2: (a)The data of first two components. (b) Location of components.

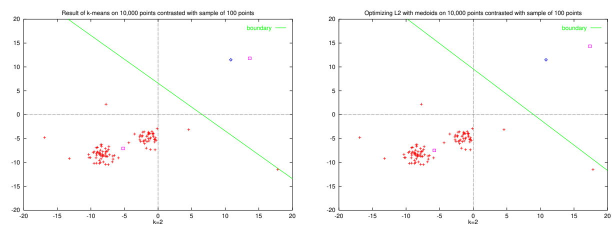

Figure 3: (a) Result with k-MEANS. (b) Optimizing L_2 with TAB.

given by $\pi^T = (0.05, 0.05, 0.4, 0.5)$. The centers of the components are $\vec{\mu}_1^T = (0.10, -3.60)$, $\vec{\mu}_2^T = (4.90, 1.82)$, $\vec{\mu}_3^T = (-1.59, -4.83)$, and $\vec{\mu}_4^T = (-8.11, -8.69)$. While the covariance matrices Σ_3 and Σ_4 of the last two components are both equal to the identity matrix, the covariance matrices of the first two have larger elements ($\Sigma_1 = diag[10^2, 10^2]$ and $\Sigma_2 = diag[20^2, 20^2]$).
Thus, the first two have large dispersion and small proportion and can potentially be interpreted as "noise". They represent about 10% of the data (981 data points). Figure 1 (a) displays our data set. In this figure, the last two components of the mixture appear amongst the cloud of "noise" of the first two components. Figure 1 (b) displays the data of the last two components. These two components appear as well separated circular clusters. Figure 2 (a) shows the data of the first two components labeled with their component membership (those points generated on the first component are marked $\diamond$ while those of the second component are marked $+$). This graph allows us to verify that the dispersion of the second component is much larger than the dispersion of the first. Figure 2 (b) displays the true centers of the 4 components against a subset of the data. This allows to locate the components while not over-crowding the plot.
We look first at typical results obtained by k-MEANS with $k = 2$ (refer to Figure 3 (a)). The resolution in this figure and those that follow is increased with respect to the previous figures (see scale on axes' labels). The centers of clusters proposed by the algorithm are indicated with $\square$. Clearly, the solution is very poor, resulting in merging the two large components as a single cluster and placing one center almost arbitrarily far towards the top-right corner of the figure. Figure 3 (a) displays the boundary separating the classification by k-MEANS. This line corresponds to the bisector of the two centers found by this algorithm. The only plus is that this algorithm requires $O(n)$ time to approximately optimize L_2! But what if we use medoids to optimize the L_2 criteria (total squared error) as suggested in [30]. With

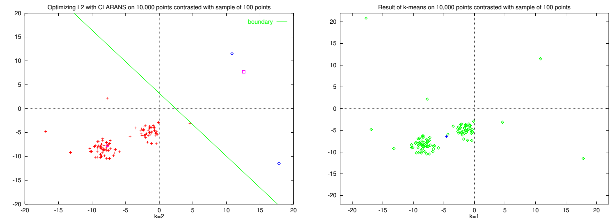

Figure 4: (a) Optimizing L_2 with CLARANS. (b) Result with k-MEANS.

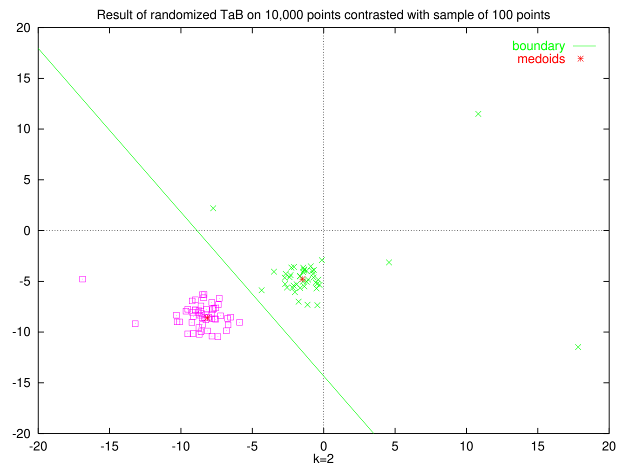

Figure 5: L_1 with randomized TAB.

an interchange heuristic this requires $O(n^2)$ time. k-MEANS required under 3 seconds of CPU time, while this algorithm required 7 hours of CPU time. Although it is expected that an algorithm that requires this much more time should do better than the linear algorithm there is no improvement (refer to Figure 3 (b)). By contrast, CLARANS [41] performs a restricted window interchange to ensure faster processing, but its results to approximately solve L_2 are almost as bad as k-MEANS(refer to Figure 4 (a)). Although the results seem better for CLARANS, CLARANSwill always reach an inferior solution with respect to L_2 than the interchange heuristic.
Using medoids in the original sense of discrete medians (optimization) and an $O(n\sqrt{n})$ (subquadratic) randomized algorithm [22; 18] to improve on the Tab interchange heuristic [48], we obtain essentially perfect results (refer to Figure 5). The optimization of a more robust induction principle makes all the difference. Although some computational cost must be paid with respect to k-MEANS, if one is prepared to pay for the cost of CLARANS(or an interchange heuristic for L_2), one may as well optimize L_1.
Our second point is now with respect to the use (or misuse) of validity indexes. Suppose we use k-MEANS on the data described earlier, with $k = 1, \ldots, 4$. And we compute the validity index $L_2(C)/nSep(C)$ for $k = 2, \ldots, k$. This validity index for $k = 1$ is undefined, and this is natural since in this case we are after clusters and not declaring the entire data one group (although this is also an issue sometimes studied under cluster tendency [32]). Simple examples to show

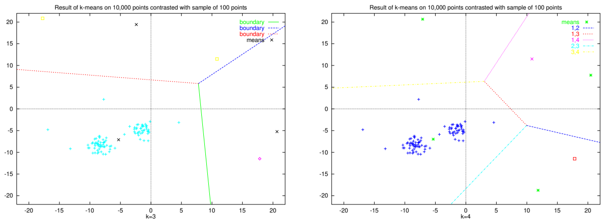

Figure 6: Result with k-Means; (a) $k = 3$ (b) $k = 4$.

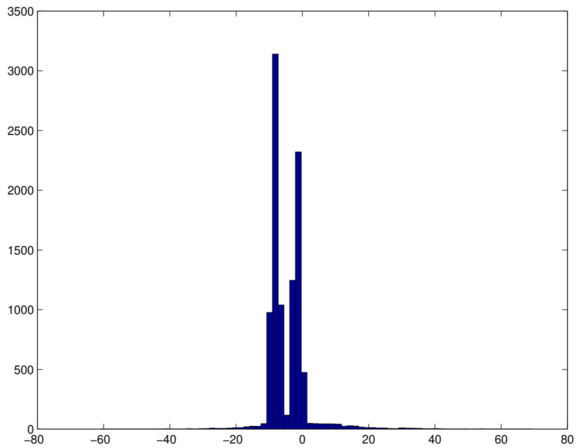

Figure 7: Histogram of x-values.

the problems of validity indexes can also be constructed for other validity indexes of this type. Figure 4 (b) shows the results by k-Means with $k = 1$, this places the unique center at the center of mass for the data. Figure 6 (a) and Figure 6 (b) shows the results of k-Means for $k = 3$ and $k = 4$, respectively. The values of $L_2(C)/nSep(C)$ are 0.072 for $k = 2$, 0.060 for $k = 3$ and 0.084 for $k = 4$. A miss-interpretation of validity indexes would indicate 3 clusters in the data and the results of Figure 6 as "valid". But we know that the actual truth is 4 components and perhaps, to the eye of some researchers, two clusters and noise.

No validity index is useful for all situations. Each validity index is surrounded by assumptions about the nature of the data and what the researcher is willing to accept as a null hypothesis. Thus, there are as many validity indexes as induction principles. However, optimizing directly the validity index does not necessarily lead to valid (more accurate results). We illustrate this with a visual example. This time the data will be one dimensional, but again, the situation can only be worse with an increase in dimension. The one dimensional data we use is the first coordinate of the data in Figure 1. Figure 7 displays a histogram of this one dimensional data set. The last two components are clearly visible with their true centers at $\mu_3 = -1.59$, and $\mu_4 = -8.11$.

Now, consider applying k-Means with $k = 2$ in this data set ($n = 10,000$). The k-Means algorithm seeks the optimum of Equation (2). That is, it look for two values c_3 and c_4 such that L_2 is minimum. Figure 8 (a) displays the level curves of the function $L_2(c_3, c_4)$. If k-Means was fortunate enough to actually find the optimum of L_2 it would find

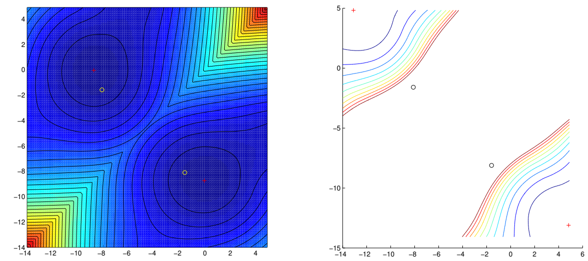

Figure 8: (a) Optimum values of L_2 (indicated with $+$) versus true values for the means of the clusters (indicated with $\circ$). (b) Optimum values of $L_2/(n\,Sep)$ (indicated with $+$) versus true values for the means of the clusters (indicated with $\circ$).

$\hat{c}_3 = -0.05$ and $\hat{c}_4 = -8.75$ (or $\hat{c}_3 = -8.75$ $\hat{c}_4 = -0.05$ because L_2 is symmetric, that is $L_2(c_3, c_4) = L_2(c_4, c_3)$). These optima are marked with $+$. Note that this suggest far more separation between the prototypes for the clusters that there actually exists. This is usually referred in the literature as "k-Means is statistically biased". The point we are trying to make is that it is not a problem of the algorithm, but of the induction principle formalized in L_2. Any algorithm for Equation 2, even one that could find the optima of L_2 would be biased for mixtures of normals.

Now, turning to the issue of validity. Suppose for a moment that somebody presents the solution $C = \{c_3 = -13.1, c_4 = 4.83\}$. If we apply the validity index $L_2(C)/(nSep(C))$ to our data set, we would obtain a very small value because this C is the minimum. The preference for well separated representatives for clusters drives the two values further apart. Figure 8 (b) shows the level curves of this simple validity index. This misuse of the validity index could potentially suggest that this person is presenting us with valid results. How would people obtain such results? They could use an optimization algorithm (perhaps quadratic or cubic) to closely approximate the minimum of $L_2(C)/(nSep(C))$. Note that the same could be done for more sophisticated validity indexes. They can all be evaluated by computer, and although the minimization may be very computationally expensive, close approximations to the minimum are possible. So then the question is, how do we verify the validity of results supplied by others? Note that 'others' could be an algorithm (machines), a person (people), or both. The opinion here is that validity needs very careful analysis of the nature of the data, the hypothesis one is prepared to select as "no-structure" (random data) and the researchers belief that there is some type of structure in the data. Thus, even for validity, clusters are in the eye of the beholder and there will also be many validity indexes.

B. REFERENCES

[1] C. Aggarwal. A human-computer cooperative system for effective high dimensional clustering. In *Proceedings of the KDD Conference*, pages 221–226, San Francisco, CA, August 26-29 2001. ACM-SIGKDD, ACM Press.

[2] R. Agrawal, R. Bayardo, and R. Srikant. Athena: Mining-based interactive management of text

databases. In C. Zaniolo, P. Lockemann, M. Scholl, and T. Grust, editors, *Extending Database Technology EDBT, 7th International Conference*, volume 1777 of *Lecture Notes in Computer Science*, Konstanz, Germany, March 27-31, 2000. Springer.

[3] M. Aldenderfer and R. Blashfield. *Cluster Analysis*. Sage Publications, Beverly Hills, USA, 1984.

[4] C. Bajaj. Proving geometric algorithm non-solvability: An application of factoring polynomials. *Journal of Symbolic Computation*, 2:99–102, 1986.

[5] J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[6] J. Bezdek and N. Pal. Some new indexes of cluster validity. *IEEE Transactions on System, Man and Cybernetics, Part B*, 28:301–315, 1998.

[7] R. Bonner. On some clustering techniques. *IBM Journal of Research and Development*, 8:22–32, 1964.

[8] P. Brucker. On the complexity of clustering problems. In R. Henn, B. Korte, and W. Oetti, editors, *Optimization and Operations Research: Proceedings of the workshop held at the University of Bonn*, pages 45–54, Berlin, 1978. Springer Verlag Lecture Notes in Economics and Mathematical Systems 157.

[9] A. Dempster, N. Laird, and D. Rubin. Maximum likehood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[10] B. Dom. An information-theoretic external cluster-validity measure. IBM Research Report RJ 10219, IBM's Almaden Research Center, San Jose, CA, October 5th 2001.

[11] R. Dubes. Cluster analysis and reated issues. In C. Chen, L. Pau, and P. Wnag, editors, *Handbook of Pattern Recognition and Computer Vision*, pages 3–32, River Edge, NJ, 1993. World Scientific Publiching Co. Chapter 1.1.

[12] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, NY, USA, 1973.

[13] J. Dunn. Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4:95–104, 1974.

[14] U. Elsner. Graph partitioning: A survey. Technical Report 97-27, Technische Universit"at Chemnitz, December 1997.

[15] M. Ester, H. Kriegel, S. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. Fayyad, editors, *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231, Menlo Park, CA, 1996. AAAI, AAAI Press.

[16] V. Estivill-Castro. Hybrid genetic algorithms are better for spatial clustering. In R. Mizoguchi and J. Slaney, editors, *Proceedings Sixth Pacific Rim International Conference on Artificial Intelligence PRICAI 2000*, pages 424–434, Melbourne, Australia, 2000. Springer-Verlag Lecture Notes in Artificial Intelligence 1886.

[17] V. Estivill-Castro and M. Houle. Data structures for minimization of total within-group distance for spatio-temporal clustering. In L. De Raedt and A. Siebes, editors, *5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01)*, pages 91–102, Freiburg, Germany, September 3-7 2001. Springer Verlag Lecture Notes in Artificial Intelligence 2168.

[18] V. Estivill-Castro and M. Houle. Fast minimization of total within-group distance. In J. Fong and M. Ng, editors, *Proceedings of the International Workshop on Mining Spatial and Temporal Data in conjunction with the fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD-2001*, pages 72–81, Hong Kong, April 15-18 2001. City University of Hong Kong.

[19] V. Estivill-Castro and M. Houle. Robust distance-based clustering with applications to spatial data mining. *Algorithmica*, 30(2):216–242, June 2001.

[20] V. Estivill-Castro and A. Murray. Discovering associations in spatial data - an efficient medoid based approach. In X. Wu, R. Kotagiri, and K. Korb, editors, *Proceedings of the 2nd Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-98)*, pages 110–121, Melbourne, Australia, 1998. Springer-Verlag Lecture Notes in Artificial Intelligence 1394.

[21] V. Estivill-Castro and J. Yang. A fast and robust general purpose clustering algorithm. In R. Mizoguchi and J. Slaney, editors, *Proceedings Sixth Pacific Rim International Conference on Artificial Intelligence PRICAI 2000*, pages 208–218, Melbourne, Australia, 2000. Springer-Verlag Lecture Notes in Artificial Intelligence 1886.

[22] V. Estivill-Castro and J. Yang. Non-crisp clustering web visitors by fast, convergent and robust algorithms on access logs. In L. De Raedt and A. Siebes, editors, *5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01)*, pages 103–114, Freiburg, Germany, September 3-7 2001. Springer Verlag Lecture Notes in Artificial Intelligence 2168.

[23] B. Everitt. *Cluster Analysis*. Halsted Press, New York, USA, 2nd. edition, 1980.

[24] U. Fayyad and R. Uthurusamy. Data mining and knowledge discovery in databases. *Communications of the ACM*, 39(11):24–26, Nov. 1996. Special issue on Data Mining.

[25] M. Garey and D. Johnson. *Computers and Intractability — A guide to the Theory of NP-Completeness*. Freeman, NY, 1979.

[26] S. Guha, R. Rastogi, and K. Shim. CURE: An efficient clustering algorithm for large databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, volume 27, pages 73–84, New York, 1998. ACM, ACM Press.

[27] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *KDnuggets:News*, page Numver 19 item 16, September 2001. www.db-net.aueb.gr/mhalk/papers/validity_survey.pdf.

[28] M. Halkidi, M. Vazirgianis, and Y. Batistakis. Quality scheme assessment in the clustering process. In H. Zighed, D.A. Komorowski and J. Zytkow, editors, *Principles of Data Mining and Knowledge Discovery, 4th European Conference, PKDD*, pages 265–276, Lyon, France, September, 13-16 2000. Springer Verlag Lecture Notes in Computer Science 1920.

[29] I. Hall, L.O. Özyurt and J. Bezdek. Clustering with a genetically optimized approach. *IEEE Transactions on Evolutionary Computation*, 3(2):103–112, July 1999.

[30] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Mateo, CA, 2000.

[31] A. Hinneburg and D. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proc. 4rd Int. Conf. on Knowledge Discovery and Data Mining*, pages 58–65, New York, August 1998. AAAI Press.

[32] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1988. Advanced Reference Series: Computer Science.

[33] M. Jain, A.K. nd Murty and P. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–320, September 1999.

[34] J. Kalbfleisch. *Probability and Statistical Inference — Volume 2: Statistical Inference*. Springer-Verlag, NY, US., second edition, 1985.

[35] G. Karypis, E.-H. Han, and V. Kumar. Chameleon: Hierachical clustering using dynamic modeling. *Computer*, 32(8):68–75, August 1999.

[36] L. Kaufman and P. Rousseuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, NY, USA, 1990.

[37] W. Kloesgen and J. Zytkow. Machine discovery terminology. KDnuggets Publicatiosn and References http://www.kdnuggets.com/publications/index.html. http://orgwis.gmd.de/projects/explora/terms.html.

[38] H. Kuhn. A note on Fermat's problem. *Mathematical Programming*, 4(1):98–107, 1973.

[39] H. Kuhn and E. Kuenne. An efficient algorithm for the numerical solution of the generalized Weber problem in spatial economics. *Journal of Regional Science*, 4(2):21–33, 1962.

[40] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. Le Cam and J. Neyman, editors, *5th Berkley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967. Volume 1.

[41] R. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In J. Bocca, M. Jarke, and C. Zaniolo, editors, *Proceedings of the 20th Conference on Very Large Data Bases (VLDB)*, pages 144–155, San Francisco, CA, 1994. Santiago, Chile, Morgan Kaufmann Publishers.

[42] N. Pal and J. Bezdel. On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy Systems*, 3(3):370–379, August 1995.

[43] S. Ray and R. Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. In N. Pal, D. A.K., and J. Das, editors, *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT'99)*, pages 137–143, New Delhi, India, December 27-29 1999. Narosa Publishing House.

[44] R. Rezaee, B. Lelieveldt, and J. Reiber. A new cluster validity index for the fuzzy c-mean. *Pattern Recognition Letters*, 19:237–246, 1998.

[45] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society, Series B*, 49(3):223–239, 1987.

[46] P. Rousseeuw and A. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, NY, USA, 1987.

[47] M. Tanner. *Tools for Statistical Inference*. Springer-Verlag, NY, US., 1993.

[48] M. Teitz and P. Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16:955–961, 1968.

[49] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, NY, USA, 1999.

[50] D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley & sons, UK, 1985.

[51] C. Wallace and D. Boulton. An information measure for classification. *Computer Journal*, 11:185–195, 1968.

[52] R. Wilcox. *Introduction to Robust Estimation and Hypothesis Testing*. Academic Press, San Diego, CA, 1997.

[53] M. Windham. Cluster validity for the fuzzy c-means clustering algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(4):357–363, 1982.

[54] I. Witten and E. Frank. *Data Mining — Practical Machine Learning Tools and Technologies with JAVA implementations*. Morgan Kaufmann Publishers, San Mateo, CA, 2000.

[55] O. Zamir and O. Etzioni. Web document clustering: a feasibility demonstration. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR'98)*, pages 46–54, 1998.

[56] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient data clustering method for very large databases. *SIGMOD Record*, 25(2):103–114, June 1996. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data.