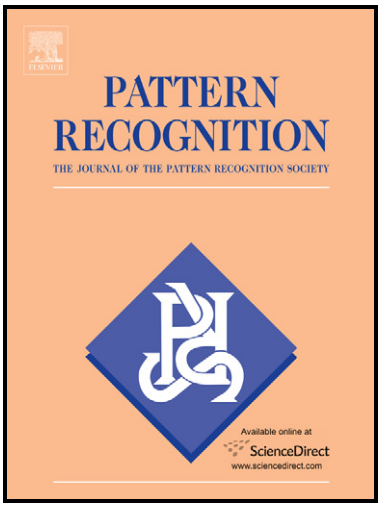

A unifying criterion for unsupervised clustering and feature selection

Mihaela Breaban, Henri Luchian

PII: S0031-3203(10)00490-5
DOI: doi:10.1016/j.patcog.2010.10.006
Reference: PR 3991

To appear in: *Pattern Recognition*

Received date: 20 December 2009
Revised date: 21 July 2010
Accepted date: 6 October 2010

Cite this article as: Mihaela Breaban and Henri Luchian, A unifying criterion for unsupervised clustering and feature selection, *Pattern Recognition*, doi:10.1016/j.patcog.2010.10.006

A Unifying Criterion for Unsupervised Clustering and Feature Selection

Mihaela Breaban*, Henri Luchian

Faculty of Computer Science, Alexandru Ioan Cuza University, Iasi, Romania
{*pmihaela, hluchian*}*@infoiasi.ro*

Abstract

Exploratory data analysis methods are essential for getting insight into data. Identifying the most important variables and detecting quasi-homogenous groups of data are problems of interest in this context. Solving such problems is a difficult task, mainly due to the unsupervised nature of the underlying learning process. Unsupervised feature selection and unsupervised clustering can be successfully approached as optimization problems by means of global optimization heuristics if an appropriate objective function is considered. This paper introduces an objective function capable of efficiently guiding the search for significant features and simultaneously for the respective optimal partitions. Experiments conducted on complex synthetic data suggest that the function we propose is unbiased with respect to both the number of clusters and the number of features.

Keywords: unsupervised feature selection, unsupervised clustering, global optimization

1. Introduction

Clustering is the task of identifying *natural* groups in data. The problem can be stated more formally as follows:

*Corresponding Author: Breaban Mihaela, Facultatea de Informatica, Universitatea "Al. I. Cuza", General Berthelot, 16, IASI 700483, ROMANIA
Phone: +40 744 821303 Fax: +40 232 201490 Email: pmihaela@infoiasi.ro

Given a set S of n data items each of which is described by m numerical attributes: $S = \{d_1, d_2, ..., d_n\}$ where $d_i = \{f_{i1}, f_{i2}, ..., f_{im}\} \in \Im_1 \times \Im_2 \times ... \times \Im_m \subset \Re^m \forall i = \overline{1..n}$,

find

$$C^* = argmax_{C \in \Omega} F(C)$$

where

- Ω is the set of all possible hard partitions C of the data set S, where each C is a hard partition if $C = \{C_1, C_2, ..., C_k\}$, $\bigcup_{i=1}^{k} C_i = S$ and $C_i \bigcap C_j = \emptyset \forall i, j = \overline{1..n}$, $i \neq j$, $k \in \{1, 2, ..., card(C)\}$.
- F is a function which measures the quality of each partition $C \in \Omega$ with respect to the requirement implicitly described above by the word *natural*: similar data items should belong to the same cluster and dissimilar items should reside in distinct clusters.

The notion of *similarity* is seldom given in the problem statement.

If the number of clusters p is known in advance the problem is called *supervised clustering*; otherwise, it is called *unsupervised clustering*.

This definition leaves space to a wide choice of objective functions and similarity functions, depending strongly on the domain under investigation. The choice is rarely straightforward. The literature records a lot of comparative studies regarding the impact of various objective functions on the solution especially in the case of unsupervised clustering. As for the similarity function, if extra-information is available in the form of pairwise constraints of data items that must reside in the same cluster (the case of semi-supervised clustering and supervised classification) then an optimal distance metric can be learned. For unsupervised clustering, metric learning is usually performed in a pre-processing step, using methods that reduce data dimensionality through statistical analysis.

Dimensionality reduction is a problem intensively studied in both supervised and unsupervised clustering. The main goal is to reduce the size of the representation of data items in order to decrease the computational cost of subsequent

steps, with minimal alterations in terms of descriptive accuracy. Dimensionality reduction is approached in two distinct ways: Feature Selection (FS) and Feature extraction (FE). The feature selection approach searches for irrelevant original features (attributes) and excludes them; additionally, feature weighting may be performed. Feature extraction methods create new features from the original ones. The points in the original D-dimensional feature space are mapped into new points in a d-dimensional feature space, d < D. Compared to FS methods, FE methods provide an improved lower-dimensional representation for the full data set; however, an important drawback of FE methods is that the relationship between the original and the reduced space is more difficult to interpret.

Feature selection plays different roles in the supervised and respectively the unsupervised scenario. In both situations, in a pre-processing step redundant features may be eliminated by means of statistical analysis. Further, in classification feature selection aims at identifying those features that predict with highest accuracy the appropriate class labels, while in clustering feature selection aims at identifying the features which generate good partitions.

Unsupervised feature selection is performed by means of:

- filter approaches, which compute some entropy measure in order to asses the grouping tendency of data items in different feature subspaces. The subsequent unsupervised learning method is completely ignored.
- wrapper approaches, which actually search for partitions in different feature subspaces using a clustering algorithm. These approaches give better results but at higher computational costs.

In view of the definition of clustering, feature selection can be stated as an optimization problem:

find

$$w^* = argmax_w Q(S')$$

where

- $w = \{w_1, w_2, ...w_m\} \in \{0,1\}^m$ is a binary string;
- S' is the data set constructed from the original set S and the string w as follows: $S' = \{d'_1, d'_2, ..., d'_n\}$, $d'_i = \{w_1 \cdot f_{i1}, w_2 \cdot f_{i2}, ..., w_m \cdot f_{im}\}, \forall i = \overline{1..n}$;
- $Q(S')$ is a function which measures the tendency of data items in set S' to group into well-separated clusters; it can be expressed by means of the entropy (filter approaches) or of a fitness function which measures the quality of a partition detected by a clustering algorithm (wrapper approaches). In the latter case feature weighting is akin to solving the clustering problem in different feature spaces.

Our study approaches unsupervised feature selection in a wrapper manner. In this regard, a new optimization criterion largely unbiased with respect to the number of clusters is introduced in section 2. Section 3 discusses the normalization of the clustering criterion with respect to the number of features. Section 4 presents a framework for performing unsupervised feature selection in conjunction with unsupervised clustering and summarizes the experimental results. Section 5 draws conclusions and points to future work.

2. Unsupervised clustering: searching for the optimal number of clusters

Classical clustering methods, such as k-means and hierarchical algorithms, are designed to use prior knowledge on the number of clusters. In k-means, an iterative process reallocates data items to the clusters of a k-class partition in order to minimize the within-cluster variance. Hierarchical clustering adopts a greedy strategy constructing trees/dendrograms based on the similarity between data items; each level in these dendrograms corresponds to partitions with a specific number of clusters and the method offers no guidance regarding the level where the optimal partition is represented (hence, the optimal number of clusters).

The algorithms mentioned above are local optimizers. In order to design a global optimizer for the clustering problem, a criterion for ranking all partitions, irrespective of the number of clusters, is needed. The problem is far for being trivial: with no hint on the number of clusters, common-sense clustering criteria like minimizing the variance within clusters and/or maximizing the distance between clusters guide the search towards the extreme solution - the n-class partition with each class containing exactly one point.

Existing studies in the literature propose and experiment with various clustering criteria [3, 22, 24, 13]. The main concern is the bias these criteria introduce towards either lower or higher numbers of clusters. Since this bias proved to be hard to eliminate, multi-objective algorithms were proposed [9], which evaluate the quality of a partition against several criteria. The main drawback remains the fact that identifying the optimal solution within the final Pareto front is not straightforward.

The clustering criterion used in the present work originates in the analogy with the Huygens' theorem from mechanics, analogy introduced in [6] and used further in [19]. Considering the data set S in the above definitions, the following notations are used:

$W = \sum_{i=1}^{k} \sum_{d \in C_i} \delta(c_i, d)$ is the within-cluster inertia computed as the sum of the distances between all data items d in cluster C_i and their cluster center c_i;

$B = \sum_{i=1}^{k} |C_i| \cdot \delta(c_i, g)$ is the between-cluster inertia computed as the sum of the distances between the cluster centers c_i and the center of the entire data set g weighted with the size of each cluster $|C_i|$.

$T = \sum_{i=1}^{n} \delta(d_i, g)$ is the total inertia of the data set computed as the sum of the distances between the data items and the center g of the data set.

In the above *center* is the gravity center.

The above-mentioned analogy with mechanics can only be applied as an approximation. The simplest approximation of the Huygens theorem is then

$$W + B \approx T$$

According to the above formula, for any partition of the data set, regardless the number of clusters, the sum W+B is merely constant. Figure 1 illustrates this for the case of a data set with 10 random Gaussian features/variables: W, B, and W+B are computed for locally optimal partitions of the data set obtained by the k-means algorithm with the number of clusters varying between 2 and 50.

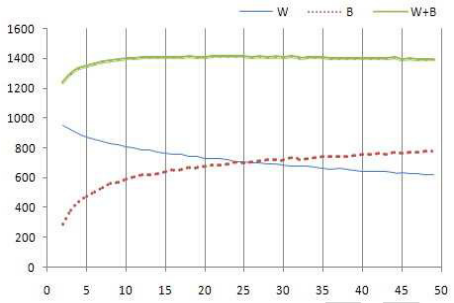

Figure 1: The within-cluster inertia W, between-cluster inertia B and their sum plotted for locally optimal partitions obtained with k-means over different numbers of clusters

In view of the Huygens theorem, if the number of clusters is fixed, minimizing W or maximizing B are equivalent clustering criteria which can be used in general heuristics [6]. Note that the within-cluster variance is a widely used clustering criterion in supervised clustering. The Huygens theorem provides an equivalent clustering criterion (namely B), at a lower computational cost, which can be used in a nearest-neighbor assignment scenario [19].

When the number of clusters is unknown both these criteria are useless: they direct the search towards the extreme n-class partition. However, a corollary of the Huygens theorem in conjunction with penalties against the increase of the number of clusters proved to work in unsupervised clustering: $\left(\frac{B}{T}\right)^k$ is used in [21]; an equivalent (in view of the Huygens' theorem) function $\left(\frac{1}{1+W/B}\right)^k$ is used in [20] in order to use local Mahalanobis distances. Unfortunately, extensive experiments we conducted recently with these fitness functions, showed that they are appropriate only for data sets with a small number of features; other penalization factors may therefore be necessary for higher-dimensional spaces.

Both within-cluster inertia and between-cluster inertia are necessary for a reliable comparison and evaluation of partitions in different feature subspaces. In this regard, we minimize the within-cluster inertia and maximize the between-cluster inertia simultaneously, through maximizing

$$F = \left(\frac{1}{1 + W/B}\right). \tag{1}$$

In order to study the bias this function induces on the number of clusters in unsupervised clustering, we used the k-Means algorithm to derive partitions for data sets consisting of between 2 and 20 random Gaussian features. As shown in figure 2-left, the function F is monotonically increasing with respect to the number of clusters, taking smaller values for higher dimensional data sets. Figure 2-right penalizes the increase in the number of clusters: F^k is represented by the dotted lines and $F^{log_2(k+1)+1}$ is represented by the continuous lines.

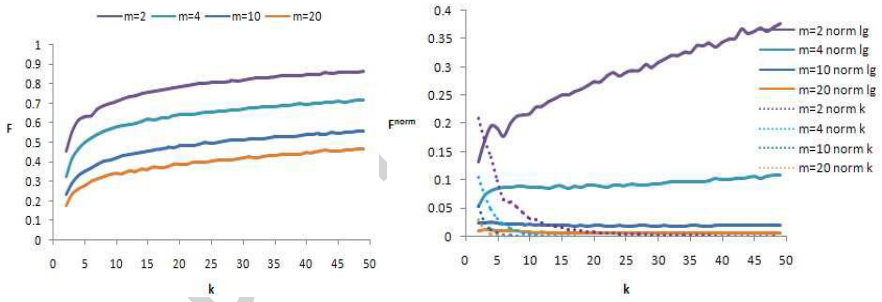

Figure 2: Left: function F plotted for partitions obtained with k-Means over different numbers of clusters, for data sets with 2, 4, 10 and 20 features; Right: function F is penalized introducing at exponent k (dotted lines) and $le(k) = log_2(k+1) + 1$ (continuous lines).

Figure 2 presents the behavior of function F when the Euclidean metric is used as distance function: $\delta(x, y) = (\sum_{i=1}^{m} |x_i - y_i|^q)^{1/q}$ with $q = 2$. Experiments showed that using the Manhattan metric ($q = 1$) and Chebyshev metric ($q = \infty$) - the extreme cases of the Minkovski metric, identical plots as the ones in Figure 2 are obtained; this suggests that function F records the same behavior under all Minkovski metrics. Moreover, experimental studies with fractional norms gave the same results: function F is biased towards lower numbers of features

and higher numbers of clusters. For unsupervised clustering, its use would fail to identify the optimal number of clusters and would favor the partition with the maximum allowed number of clusters; for feature selection its use would fail to identify all relevant features and would favor the subset with the minimum allowed cardinality.

Exponent k reverses the bias towards low numbers of clusters in all cases, while the logarithmic exponent is able to eliminate any bias for the data sets with more than 10 features. For lower dimensions, the logarithmic factor is, however too weak. In order to make it work in low dimensional spaces, a new factor which penalizes F linearly in the number of features m is introduced.

The new criterion we introduce for measuring the quality of a partition is

$$CritC = (a \cdot F)^{le(k)} \tag{2}$$

where $a = \frac{2 \cdot m}{(2 \cdot m+1)}$ and $le(k) = log_2(k+1) + 1$ ($l$ogarithmic $e$xponent).

$CritC$ takes values in range [0,1] and should be maximized.

This function is studied in the sequel in the context of unsupervised clustering; we test its capacity of detecting simultaneously the optimal partition and the optimal number of clusters (see section 4).

3. Unsupervised feature selection: searching for the optimal number of features

Since the search space is exponential in the number of features, unsupervised feature selection has been initially approached using greedy heuristics in the filter manner. One such approach is *sequential selection* which was implemented in two ways [17]: *sequential forward selection*, which starts with the empty set and iteratively adds the most rewarding feature among the unselected ones and *sequential backward selection*, which at the beginning considers all features as being selected and iteratively removes the least rewarding feature among the selected ones,until the stopping criterion is met. Two strategies are used to compute the merit of each feature: one that aims at removing redundant features and one that scores the relevance of features. Redundancy-based

approaches hold that mutually-dependent features should be discarded. On the contrary, approaches in the second category [18, 26] compute the relevance of features based on the assumption that relevant features are kept dependent to each other by the structure of the clusters. Pairwise dependence scores are consequently computed using *mutual information* and *mutual prediction* [26]. Other approaches rank the features according to their variances or according to their contribution to the entropy calculated on a leave-one-out basis [27]. However, most of these methods deliver feature rankings and leave to the user the decision regarding the number of features.

Wrapper methods for feature selection evaluate subsets of features based on the quality of the best partition generated by each subset. In this scenario, an unsupervised clustering criterion unbiased with respect to the number of clusters and able to compare different partitions is required in order to asses the quality of feature subsets. However, existing unsupervised clustering criteria are not appropriate/fair evaluators in the context of feature subsets of different cardinalities: they are based on computing some distance function for every pair of data items. Since dimensionality influences the distribution of the distances between data items, it induces a bias in the objective function with respect to the size of the feature space. To illustrate this, consider the case of Minkowski distance functions: the mean of the distribution increases with the size of the feature space because one more feature introduces one more positive term into the sum; combined with an objective function which minimizes the between-cluster variance, feature selection will be strongly biased towards low dimensionality. This example is not unique: it is also the case of the most popular unsupervised clustering criteria - Davies- Bouldin Index [5], Sillhouette Width [25] - which are also biased towards low dimensionality.

The influence of the dimensionality of the data set on distance-based data analysis methods - including clustering techniques, was thoroughly investigated in [1]. The authors show that in high-dimensional spaces fractional norms are more appropriate to discriminate between data items. As a consequence, clustering algorithms using fractional norms to measure the distance between data

items of large dimensionality, are more successful. However, fractional norms are not a solution when a large number of noisy features are present in data.

A few strategies were proposed in order to reduce the dimensionality bias. Dy and Brodley [8] use *sequential forward search* to search for feature subsets in conjunction with the Expectation Maximization algorithm searching for the best partition. The search for the number of clusters is performed for each feature subspace starting with a high number of clusters and iteratively decrementing by one this number, merging at each step the clusters which produce the least difference in the objective function. Two feature subset selection criteria are tested in [8]: the *Maximum Likelihood* criterion which is biased towards lower-dimensional spaces of features and the *scatter separability* which favors higher-dimensional spaces. In order to counteract these biases, cross-projection is introduced: given a pair of feature subsets, for each feature subset the best partition is determined and the resulting partitions are evaluated as well in the other subspace. The fitness of each feature subspace is computed with regard to the quality of the optimal partition it produces, measured in both feature subspaces. This cross-projection normalization can be used for pairwise comparisons between features sets. However, it is not transitive which raises obstacles against its use in global optimization techniques.

Multi-objective optimization algorithms are a more straightforward way to deal with biases: the bias introduced in the primary objective function is counterbalanced by a second objective function. The approach was initially proposed in [14] where four objectives are used by the Evolutionary Local Selection Algorithm (ELSA). In [23] only two objectives are used by a multi-objective genetic algorithm: the Non-dominated Sorting GA-II. A more extensive study on the use of multi-objective optimization for unsupervised feature selection is carried out in [10]: some drawbacks of the existing methods are outlined and several objective functions are thoroughly tested on a complex synthetic benchmark. Furthermore, a strategy for automated solution extraction from the Pareto front is proposed. However, the entire method is time-consuming and somewhat awkward to use.

In [29] clustering and feature selection are performed iteratively: a recently proposed Gaussian mixture clustering algorithm is used to derive partitions; subsequently, a refined feature subset in terms of relevance and non-redundance under the current data partition is identified. As a result, a more accurate partition is to be found using the selected feature subset in the next iteration. A new score is proposed to quantify the relevance of each feature based on the intra-cluster variance reported to the total variance. A Markov Blanket filter is used to identify redundant features. Both procedures - the one computing relevancy and the one computing redundancy - return rankings over features. Therefore, two numerical parameters are introduced, one for each procedure, in order to perform feature selection. This parameters are optimized empirically.

Recent work in clustering is focused on ensemble learning methods. In [11] ensemble clustering is used in order to leverage the consensus across multiple clustering solutions. Then, a population based incremental learning algorithm is used to search for a subset of all features such that the clustering algorithm trained on this feature subset can achieve the most similar clustering solution to the one obtained by the ensemble clustering method. The bias with respect to the cardinality of the feature subsets is eliminated, as the feature selection problem is now reformulated within a supervised scenario: given a partition of the data set, a subset of features which predicts with highest accuracy the cluster assignments must be identified.

There are methods which push feature selection directly into a specific clustering procedure [16, 4].

In the available literature we did not come across an objective function (ranking criterion) to provide a ranking of partitions with regard to their quality, **irrespective of the number of features**. It is the goal of this paper to propose a function which, in the search space defined by all possible subsets of features in conjunction with a variable number of clusters, assigns a ranking score to each partition that may be defined. The function we propose may be used by any heuristic searching for the best partition when both the number of features and the number of clusters vary during the search.

As shown in Figure 2, the function $CritC$ (proposed in section 2), monotonically decreases with the number of features even if factor a penalizes small feature spaces. This function would point the search towards small feature subsets. In order to eliminate this bias with respect to the number of features m, we use the factor $le(m) = log_2(m+1)+1$ with the goal of penalizing small values of m. Our new optimization function is:

$$CritCF = CritC^{\frac{1}{le(m)}} = (a \cdot F)^{\frac{le(k)}{le(m)}} \tag{3}$$

Studies we undertook on datasets containing Gaussian features, show that the function CritCF removes considerably the bias with regard to both the number of clusters and the number of features, but not completely. Yet, this function is the winner of a contest with only a few hand-made competitors; a safe assumption would be that better candidates may exist. We tested this assumption through an automated search process using Genetic Programming. this search is described in the rest of this section.

Because F is influenced by both the number of features m and the number of clusters k, we search for a function expressing this dependency. The problem is formulated as follows: given tuples (m, k, F), an equation satisfied by them must be determined. We deal in fact with symbolic regression: the optimization process has to work simultaneously on the analytical form of the function, the variables involved, and the coefficients.

Using datasets containing only standard Gaussian features, optimal partitions were constructed with the k-Means algorithm varying the number of clusters k in the range 2-49 and the number of features m in the range 2-20. Then, using formula (1) the values of F were computed for all resulted partitions.

Tuples (m, k, F), generated in this way, are given as input to a genetic programming algorithm [15, 2]. The chromosomes are trees which are decoded into functions over the two variables m and k. The crossover and mutation operators are similar to those described by Koza in [15]. The fitness function computes the ability of each chromosome to predict the values of F in terms of absolute

error of the encoded solution relative to the input data set. The set of operators is {+, -, ·, / }; additionally, the natural logarithm, the base 2 logarithm and the power function are used.

A chromosome derived from CritCF function was introduced in the initial population consisting of random generated individuals. CritCF should deliver a constant value for all the partitions derived in this part of the experiments; however, slight variations exist and an average denoted by *const* is computed over all values. Then, F is extracted from formula (3) as shown below and encoded as a GP chromosome.

$$F = \frac{1}{a} \cdot (const)^{\frac{le(d)}{le(m)}} \tag{4}$$

Several runs of the GP algorithm with different settings were performed, each run consisting of evaluating about 20 000 new chromosomes derived from the application of genetic operators. None of the chromosomes generated throughout the run outperformed the chromosome encoding formula (4). The absolute error obtained by the best chromosome recorded during the runs of the algorithm was 41% higher compared to the absolute error recorded by the chromosome encoding formula (4).

These first experiments strongly suggest that the function CritCF we propose is a near-optimal solution; it is largely unbiased with respect to both the number of clusters and the number of features. It takes values in range [0,1] and should be maximized in order to simultaneously obtain the best feature subset and partition. This function is further studied on synthetic complex data sets.

4. Experiments

4.1. The unsupervised clustering criterion

In order to study the criterion CritC, we undertook comparative studies on complex synthetic data sets against the most widely used criteria for unsupervised clustering.

4.1.1. Currently used unsupervised clustering criteria

Several studies [9] indicate **Silhouette Width** (SW) [25] as the best partition ranking criterion in unsupervised clustering. SW for a partition is computed as the average *silhouette* over all data items in the data set. The *Silhouette* for a data item i is computed as follows:

$$S(i) = \frac{b_i - a_i}{\max(b_i, a_i)}$$

where

$a_i = avg_{d \in C_i} \delta(d_i, d)$ where $d_i \in C_i$ denotes the average distance between i and all data items in the same cluster;

$b_i = min_{C \neq C_i} avg_{d \in C} \delta(d_i, d)$ where $d_i \in C_i$ denotes the average distance between i and all data items in the closest other cluster (defined as the one yielding the minimal b_i).

SW takes on values in the range [-1,1] and is to be maximized in search of the optimal clustering.

Another widely used clustering criterion is **Davies-Bouldin Index** (DB) [5] which, as in the case of CritC above, makes use of cluster representatives to compute the within-cluster compactness and between-cluster separation in a partitioning:

$W_{DB}(j) = \frac{1}{|C_j|} \cdot \sum_{d \in C_j} \delta(c_j, d)$ is the intra-cluster compactness for cluster C_j;

$B_{DB}(j, l) = \delta(c_j, c_l)$ is the separation between clusters C_j and C_l.

The DB Index is defined as:

$$I_{DB} = \frac{1}{k} \cdot \sum_{j=1}^{k} \max_{j,l \neq j} \left(\frac{W_{DB}(j) + W_{DB}(l)}{B_{DB}(j, l)} \right)$$

and is to be minimized in order to seek for the optimum clustering.

4.1.2. The search method

In order to search for the best clustering in a fixed feature space, the K-means algorithm is run over the given data set with the number of clusters k ranging from 2 to 50. In order to avoid suboptimal solutions due to unfavorable

initialization, K-means is run 10 times and only the best solution with regard to the within-cluster inertia is reported. This is what we further call one k-Means run.

4.1.3. The data suite

The performance of our clustering criterion $CritC$ in finding the optimal clustering is evaluated under various scenarios: data sets with lower or higher dimensionality, some having optimal partitions with a small number of clusters and others having optimal partitions with a high number of clusters. In this regard the complex benchmark made available by Julia Handl [1] [9] is used; it represents a standard cluster model built using multivariate normal distributions. The clusters in a data set are built iteratively based on covariance matrices which need to be symmetric and positive definite. Overlapping clusters are rejected and regenerated, until a valid set of clusters has been found. The covariance matrices are built in such a way to encourage the production of elongated clusters; this is the reason why k-Means fail to identify in some test cases the correct partition, as the results in the experimental section show.

Ten data sets of dimensionality d and containing k clusters are created and referred to as the group of data dd-kc, with $d \in \{2, 10\}$ and $k \in \{4, 10, 20, 40\}$. The size of each cluster varies uniformly in the range [50,500] for the data sets with 4 and 10 clusters and in range [10,100] for the data sets with 20 and 40 clusters.

A total of 120 data sets were used in order to study our criterion in the context of unsupervised clustering.

[1] http://dbkgroup.org/handl/generators/

4.1.4. Validation measures: the Adjusted Rand Index

Each partition returned by k-means is evaluated against the optimal clustering using the Adjusted Rand Index (ARI) [12]:

$$ARI = \frac{\sum_{ij} C_{N_{ij}}^2 - \left[\sum_i C_{N_{i-}}^2 \cdot \sum_j C_{N_{-j}}^2\right] / C_N^2}{\frac{1}{2}\left[\sum_i C_{N_{i-}}^2 + \sum_j C_{N_{-j}}^2\right] - \left[\sum_i C_{N_{i-}}^2 \cdot \sum_j C_{N_{-j}}^2\right] / C_N^2}$$

where N_{ij} is the number of data items of optimal cluster i within cluster j in the evaluated clustering, N_{i-} is the number of data items in the optimal cluster i and N_{-j} is the number of data items in cluster j under evaluation.

4.2. The unsupervised feature selection criterion

4.2.1. Search methods

Wrapper feature selection methods usually involve two distinct heuristics: one for searching the feature space for the optimal feature subset and the other one for searching the optimal partition, given a feature set. In our approach the latter is performed using the k-Means algorithm. The former is conducted with two heuristics: a greedy method named sequential forward selection which is widely used in the context of feature selection and a global optimization heuristic. Extensive experiments employing two versions of the greedy algorithm and a multi-modal optimization algorithm provide insight into the fitness landscape under our criterion.

(A) Sequential forward Selection

Forward selection is a greedy algorithm widely used for feature selection [17, 8, 10]. In our implementation it starts with the empty set and iteratively adds the feature which, added to the already selected features gives the highest value for the CritCF function. For each candidate feature the k-Means algorithm is run repeatedly with the number of clusters ranging from 2 to 17 and the best partition is chosen using the CritCF function. The range of values used for the number of clusters was chosen in order to be consistent with the genetic algorithm employed in subsection B)as the global search method.

Two versions of this algorithm corresponding to different halting conditions are considered. In a first scenario, the algorithm is stopped when none of the

remaining features brings any improvement when added to the already selected ones.

A second version of the greedy algorithm is inspired from the experiments described in [10] where a fixed number of features are selected and the best solution is eventually chosen from a Pareto front. As in [10], the algorithm selects iteratively up to 20 features which is akin to ranking the most relevant 20 features. The best solution is chosen further based on the fitness values computed with CritCF for each group of the first i features.

The first algorithm has a reduced time complexity due to the halting condition but it gets more easily trapped in local optima.

The number of evaluations required by the greedy forward feature selection algorithm with variable number of clusters is

$$(k_{max} - 1) \cdot d_{max} \cdot d$$

where

- d is the dimensionality of the data set;
- d_{max} is the maximum cardinality of the feature subsets;
- k_{max} is the maximum number of clusters allowed while searching for the best partition.

For Handl's data sets with 100 noisy variables used in this paper, the second version of the greedy algorithm searching for at most 20 relevant features would thus require at least 32 000 evaluations. The number of evaluations required by the first version of the algorithm depends on the number of relevant features identified.

(B) A Genetic Algorithm

The Multi-Niche Crowding Genetic Algorithm (MNC GA) [28] is a steady-state GA which implements the crowding mechanism in order to maintain diversity in population during the search. It typically converges to multiple local optima. Since in various feature subspaces different numbers of clusters and consequently different groupings would be (locally) optimal, the algorithm searches

for both the optimal feature subset and the respective optimal number of clusters. Therefore, a chromosome is a binary string encoding both the features ('1' for 'selected', '0' for 'unselected') and the number of clusters (4 bits under Gray coding); this representation was used in several papers [10, 23, 14] and reduces the computational cost as it eliminates the need for searching exhaustively for the optimal number of clusters in a given feature space.

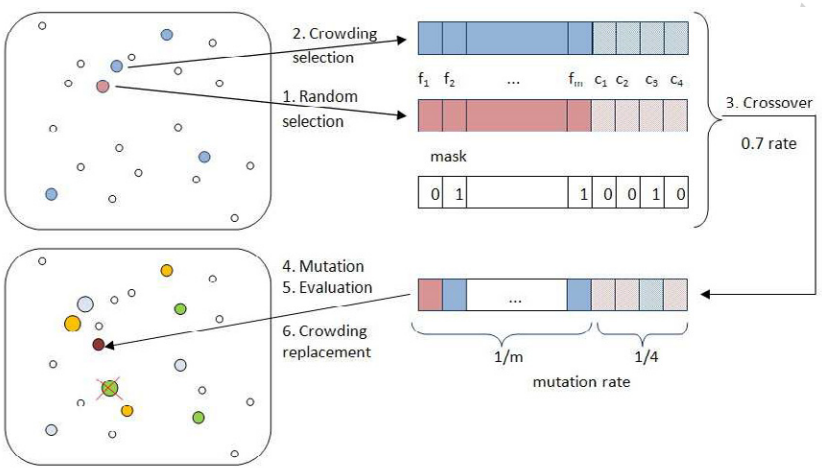

Figure 3: One iteration in MNC GA for unsupervised feature selection.

In MNC GA, both selection and replacement operators implement a crowding mechanism. Mating and replacement among members of the same niche are encouraged while allowing at the same time some competition among the niches for the population slots. Selection for recombination has two steps: 1. one individual is selected randomly from the population 2. its mate is the most similar individual from a group of size s which consists of randomly chosen individuals from the population; one offspring is created. The individual to be replaced by the offspring is chosen according to a replacement policy called *worst among most similar*:

- f groups are created by randomly picking from the population g (crowding group size) individuals for each group ;

- one individual from each group that is most similar to the offspring is identified;
- the individual with the lowest fitness value among most similar ones is replaced by the offspring.

In the original MNC algorithm the replacement is always performed, even if the fitness of the offspring is lower than the fitness of the individual chosen to be replaced. We adopted a strategy inspired from Simulated Annealing: an individual having lower fitness survives with a probability that decreases during the run of the algorithm. Lower fitness survivals have a probability of 0.5 at the beginning of the run; it decreases exponentially during the run by multiplying it with 0.9995 at each iteration.

One iteration of the MNC algorithm is illustrated in Figure 3. The evaluation step (step 5. in Figure 3) has two sub-steps:

- the k-Means algorithm is run in the feature space consisting only of the features selected and with the number of clusters encoded in the chromosome. K-Means is restarted 10 times and the best partition with regard to the within-cluster inertia score is chosen;
- the criterion CritCF, calculated for the partition obtained above, gives the fitness of the chromosome.

Recombination is performed using uniform crossover and binary mutation (p_m is $\frac{1}{numberOfFeatures}$ for genes encoding features and $\frac{1}{4}$ for genes encoding the number of clusters).

The similarity between two individuals is measured using the Hamming distance; only the substring which encodes features is considered.

For comparison purposes, many parameter values in our experiments reproduce those reported in [10]. The maximum number of clusters allowed during the search is $k_{max} = 17$ (only 4 bits encode the number of clusters). The search space is restricted to solutions with a maximum of $d_{max} = min\{20, d\}$ features

from a total of d features. In addition, in order to speed up the process of identifying the optimal feature subset, the maximum number of features selected within a chromosome is gradually increased throughout the run from a maximum of 5 features during the first generations to 10, then 15 and finally 20 at the end of the run. The risk of getting trapped in local optima due to this incremental search (specific to greedy algorithms) is tackled by the crowding GA through maintaining multiple niches in the search space.

The size of the population in MNC-GA is set to $pop\_size = 100$ individuals. The following values give an appropriate balance between exploration/exploitation and a moderate fitness pressure at replacement: $s = 0.10 \cdot pop\_size$, $g = 0.15 \cdot pop\_size$, $f = 0.10 \cdot pop\_size$.

The most time-consuming part of the algorithm is the evaluation step which mainly consists of finding the best partition for a given feature set and a given number of clusters. Each run of the MNC-GA algorithm consisted of 10.000 iterations which correspond to 10.000 evaluations.

At the end of each run of the MNC-GA algorithm a local search was performed around the best individual in order to avoid sub-optimal numbers of clusters: for the best feature subset, the k-Means algorithm was run with the number of clusters varying from $k-2$ to $k+2$ where k is the number of classes encoded by the best chromosome. The best partition is chosen to be the one with the best (highest) CritCF value.

4.2.2. The data suite

In order to validate our criterion in the context of unsupervised wrapper feature selection, a total of 40 data sets were used in this part of the experiments.

Using the multivariate Gaussian cluster model from Section 4.1.3, Handl and Knowles designed small data sets for validating feature selection methods [10]. They created ten data sets of dimensionality d and containing k clusters, with $d \in \{2, 10\}$ and $k \in \{2, 4, 10\}$. The size of each cluster is uniformly distributed within the set{10,....,50}.

Handl and Knowles introduced 100 Gaussian noise variables in each data

set in order to create high-dimensional data, which contain in some cases more dimensions than data points, a few features being relevant for the classification task. We denote these data sets by dd-kk-100-gaussian.

We also evaluate the performance of criterion CritCF when other kind of noise than Gaussian is involved. Therefore, we replaced the Gaussian noisy variables introduced within the datasets designed by Handl and Knowles with 100 uniform noisy variables and thus 40 new data sets were created and denoted in the experimental section as dd-kk-100-uniform.

Tests were also conducted on some real data sets form UCI Repository. The datasets used represent hand-written digits and letters. The Digits data set has 64 attributes representing pixels on a 8x8 grid and the Letters data set consist of 16 attributes corresponding to a 4x4 grid. Their values express the intensity of the color.

Since features may be expressed on different scales, all datasets are normalized such that each feature has mean 0 and standard deviation 1.

4.2.3. Validation measures

Results regarding feature selection can be validated from two perspectives: the quality of the best found partition reported to the (known) optimal partition, and the quality of the feature set reported to the (known) relevant features.

The quality of a partition is measured by the Adjusted Rand Index (ARI, described in section 4.1.4).

The quality of a feature subset is computed with respect to the known relevant feature set by means of two indexes from information retrieval:

$$precision = \frac{\#(significant\ features\ identified)}{\#(features\ identified)}$$

$$recall = \frac{\#(significant\ features\ identified)}{\#(significant\ features)}$$

Based on these indexes the F-Measure is computed as:

$$FMeasure = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

4.3. Results

4.3.1. Results for unsupervised clustering

One observation is mandatory for the sake of further discussion: the two criteria CritC in formula (2) and CritCF in formula (3) deliver the same ordering on the set of all partitions in the context of unsupervised clustering in a fixed feature space. Therefore, the two criteria can be considered as equivalent for the first task investigated in this paper which is unsupervised clustering in a fixed feature space; this discussion is conducted further for the CritCF function.

In order to verify that the clustering criterion does not suffer from any bias with regard to the number of clusters, tests are made within a larger range. For each data set, k-Means is run with the number of clusters varying in the range [2, 50].

In case of the data sets dd-kk-100 designed for feature selection, k-Means is run on the reduced data consisting only of the relevant features. For these test cases, k-Means is run with the number of clusters varying in range [2, 17], in order to be able to report these results as upper bounds for feature selection.

Subsequently, the three unsupervised clustering criteria (Davies Bouldin Index, Silhouette Width and CritCF) are computed for each of the 49 (respectively 16) partitions returned by k-Means, in order to select the partition with the optimal number of clusters. The ARI score and the number of clusters k are recorded for the winning partition. The best partition for each dataset is designated to be the one with the highest ARI score.

The procedure described above was applied 20 times to each data set. Table 1 presents averages and standard deviations over the ARI scores and the numbers of clusters for best partitions with respect to: 1. the ARI score recorded by k-Means (Best), 2. Davies-Bouldin Index (DB), 3. Silhouette-Width (SW) and 4. our criterion.

Statistical tests were conducted to verify if the differences of the ARI scores are significant enough to extract a winner for each type of problem. In this regard, the 3 groups of ARI values corresponding to the 3 clustering criteria,

obtained through repeated independent runs of the algorithm on instances of a given type of problem (dd-kk) are analyzed. We skip the Kruskal-Wallis test for testing equality of population medians among the 3 groups because it assumes identically-shaped distributions for all group. This condition does not hold: the standard deviation is significantly larger in case of Davies-Bouldin and Silhouette-Width criteria. One reason for the high variance is a bad performance on one or several test instances. For example, in case of the problem 2d-4c all clustering criteria select partitions with a number of clusters ranging from 3 to 5, except in case of the instance 2d-4c-no5 for which DB and SW criteria select the partition with only 2 clusters. An opposite example is the problem 10d-4c-100: for the instance 10d-4c-100-no5 CritCF selects partitions with a number of clusters ranging from 3 to 5, while SW and mostly DB select partitions with a higher number of clusters, even up to the maximum allowed value - 17 clusters.

The Wilcoxon Signed-Rank non-parametric test was applied for each problem on the group of ARI scores determined with CritCF function and the group with the highest average ARI score among the other two criteria. Where differences are significant (at the level 1%) the winner is marked in bold.

Table 1 shows that the new criterion CritCF achieves the best results in unsupervised clustering in most test cases. In a few cases it is outperformed by the Silhouette Width Index However, compared to Silhouette Width Index which has quadratic complexity in the number of data items, CritCF has linear time complexity. The highest difference in performance between CritCF and the two opponents can be observed in case of smaller (sparser) data sets dd-4c-100 due to the poor estimation of the number of clusters by Davies-Bouldin Index and Silhouette Width Index. The high values of the standard deviation for the number of clusters and, implicitly, for the ARI scores in case of these latter criteria show that they are more sensitive to minor changes in the structure of clusters. For a given problem instance, in some experiments these criteria were able to identify the optimal clustering but in others they showed a bias towards a higher number of clusters (each experiment consisting of running k-Means with varying number of clusters). In case of the Silhouette Width

index, this sensitivity can be explained by the fact that it is dependent on each particular assignment (to clusters) of the data items rather than on cluster representatives. In the case of the k-Means algorithm, which is highly dependent on the initialization step and yields near-optimal partitions, this sensitivity is a drawback. CritCF proved to be more robust: in repeated runs on the same problem instance, it chose partitions with the same number of clusters.

4.3.2. Results for unsupervised wrapper feature selection

Table 2 presents the **average** performance of the genetic algorithm MNC-GA making use of CritCF as fitness function on the datasets with Gaussian noise: the ARI score for the best partition obtained in the selected feature space and its number of clusters k, the size m, the recall and the precision of the selected feature subset. The average performance over 20 runs of our method is reported for these datasets in order to make comparisons with the algorithms investigated in [10]. Figure 4 shows the comparative results on the performance of CritCF and the algorithms investigated in [10] for feature selection:

- the three red lines correspond to the MNC-GA and Forward Selection algorithms both employing CritCF as evaluation criterion
- the two blue lines correspond to the multi-objective algorithm investigated in [10] with Silhouette Width and Davies Bouldin criteria used as the primary objective;
- the performance of a filter method is represented in yellow. The method was also investigated in [10]. It returns a Pareto front of solutions over two objectives: the minimization of an entropy measure and the maximization of the number of features in order to balance the bias introduced by the first objective. After the optimal feature subset is extracted the corresponding ARI value is obtained using k-Means with the optimal number of clusters. This procedure is quite unfair since the other methods receive no input regarding the number of clusters.

The gray line corresponds to the best partition that can be obtained using k-Means with the exact number of clusters, on the optimal standardized feature subset consisting of the known relevant features. These values provide an upper bound for the methods under investigation.

All methods implemented in [10] return a range of solutions corresponding to different feature cardinalities. The results presented in [10] and cited in our paper are obtained in the following way: for a given Pareto front, the feature set with the best F-Measure is selected. This procedure requires supplementary input and thus makes the comparison unfair for the methods which use the CritCF criterion and work in an entirely unsupervised manner.

Table 3 presents the results obtained with the two versions of the Forward Selection algorithms on the same datasets containing Gaussian noise.

To estimate the real performance of criterion CritCF, an exhaustive search method should be employed. Because under an exhaustive search the problem becomes intractable, we present in Figure 5 the best results (and not averages) obtained by the genetic algorithm. In this way we obtain a lower bound for the performance of our criterion if an exhaustive search should be employed. For each problem instance, from 5 runs of MNC-GA on each problem instance, the solution having the highest CritCF fitness value is retained. For each class of problems (each class consisting of 10 problem instances) the results are summarized as boxplots over the Adjusted Rand Index. Experiments are performed both for gaussian and uniform noise. For each problem, Figure 5 presents the following:

- the performance of KMeans over the datasets consisting only of relevant features and given the correct number of clusters (supervised clustering),
- the performance of KMeans over the datasets consisting only of relevant features and employing CritCF to determine the correct number of clusters (unsupervised clustering),
- the performance of KMeans over the datasets containing 100 gaussian noisy features and employing CritCF to determine simultaneously the cor-

rect number of clusters and the relevant features (unsupervised wrapper FS)

- item the performance of KMeans over the datasets containing 100 noisy features with uniform distribution and employing CritCF to determine simultaneously the correct number of clusters and the relevant features (unsupervised wrapper FS).

As upper bound

Comparing the ARI scores obtained in the original feature space consisting only of relevant features (Table 1) with the ARI scores obtained in the selected normalized feature space (Table 2), a decrease in the computed partition quality can be observed. Part of this loss can be related to the data normalization performed before feature selection which, as shown in [7], can decrease considerably the separability between clusters making thus more difficult for the algorithm to identify proper groupings.

In case of the data sets 2d-4c containing gaussian noise, our algorithms using the criterion CritCF identified the 2 relevant attributes in 7 out of 10 test instances and delivered partitions with the number of clusters varying between 3 and 5. The poor average performance reported in Table 2 is due to the misleading behavior of our criterion on the remaining 3 test instances. The MNC-GA either selected more features or chose partitions with a larger number of clusters for these instances. This behavior also explains the high value of the standard deviation. However, analyzing the chromosomes in the last generation of the algorithm, we discovered individuals encoding the relevant features and the right number of clusters; their presence shows that the right configuration constitutes a local optimum in the landscape designed by the CritCF function. This distorted behavior of our criterion constitutes a drawback for global search methods; however, the experiments show that it is much reduced in the greedy context: the two versions of the Forward Selection algorithm added only one irrelevant feature in the case of these 3 instances determining only on one of them a higher number of clusters which justifies the high average value for k in

Table 3.

Regarding the class of problems 2d-10c with gaussian noise, all algorithms selected the two relevant features for all problem instances as shown by the value of the F-Measure. The high variance of the number of clusters is due to a higher number of clusters chosen for two out of ten problem instances. Compared to the results presented in Table 1 for CritCF corresponding to the original relevant feature subset, a loss in performance is observed because of the normalized features. However, the results are still better than those obtained with Silhouette Width and Davies-Bouldin criteria in the original relevant feature space (see Table 1).

For the problems with 10 relevant features and gaussian noise, all three algorithms selected only relevant features but discarded some of them. For the problems with 10 clusters, this seems to be an advantage: the quality of the partitions derived in the reduced feature space is better, compared to the quality of the partitions derived in the feature space containing all relevant features (see Table 2 vs. Table 1).

Even if the F-Measure values in Figure 4 bottom corresponding to the criterion CritCF are lower compared to those reported in [10], which were obtained as described above, the ARI scores obtained by our methods are higher for some problem instances. For example, in case of the problems 10d-4c our methods obtain the lowest values for the F-measure but outperform most of the algorithms with regard to the ARI scores. Value 1 for the precision in Tables 2 and 3 shows that CritCF manages to remove all the gaussian noisy features. On the other hand, the recall values show that some relevant features are also discarded. All these observations suggest the hypotheses that some of the features known to be relevant actually may be redundant. This hypotheses is suggested as well by experiments reported in [10]: when the procedure for automated extraction of the optimal solution from the Pareto front was used, the results concerning F-Measure were significantly worst while the ARI scores were relatively close.

Unfortunately, analyzing Figure 4 no definite winner can be identified: if an algorithm outperforms the others on one class of test instances, there exists

an algorithm which beats it on a different problem. However, when comparing the methods, one must take into account that the methods based on CritCF function work completely unsupervised while the results reported for the other methods were obtained in a supervised manner as described above. Moreover, the methods using the function CritCF win with regard to time-complexity against the other methods presented in the experimental section. The first version of the Forward Selection algorithm required an average of 5,000 fitness evaluations for the problem instances with only 2 relevant features and about 17,000 fitness evaluations for the data sets with 10 relevant features. The Multi-Niche GA was run for only 10,000 fitness evaluations while the second version of the Forward Selection algorithm and all the methods from [10] (except the one based on entropy) employed more than 32,000 fitness evaluations. The quadratic complexity of the Silhouette Width criterion and the computational effort which must be payed for post-processing the Pareto front in [10] must also be considered.

Regarding the performance of our criterion on datasets with uniform noise, the same results as for the case of gaussian noise are obtained for the datasets with 10 relevant features. On the datasets with 2 relevant features and 10 clusters, the algorithm behaved impeccable on nine out of ten problem instances and selected one irrelevant feature along with the two relevant features for one problem instance. A significant decrease in performance can be observed in Figure 5 for the class of problems with uniform noise consisting of 2 relevant features and 4 clusters. Only on two problem instances out of ten in this class, the algorithm selected correctly only the two relevant features. On six problem instances the algorithm selected the 2 relevant features but also added 1 noisy features which led to an increasing number of clusters in the selected partition. On the remaining two problem instances, the algorithm did not identify the relevant features. However, this class of problem instances seems to be the most difficult one even for the case of supervised clustering, when k-Means is run on the dataset consisting only of the relevant features and is supplied with the correct number of clusters; this may be one reason for the bad performance of

the wrapper feature selection method: the clusters formed in the relevant feature space can not be correctly separated with k-Means, and the noisy uniform features mislead our criterion towards selecting smaller clusters.

For the Letters and Digits data sets we selected only 2 classes in order to interpret the results in terms of relevant features: classes A (789 data items) and B (766 data items) for Letters and classes 5 (376 data items) and 6 (377 data items) for Digits. For both test cases the MNC-GA was run 5 times each run consisting of 500 iterations; the best solution under CritCF criterion is reported. The results are presented in Table 4. For the Digits data set the best partition returned with MNC-GA accordingly to CritCF consisted of 3 clusters with one of the clusters consisting of only 5 data items; a k-Means run on the selected features with the number of clusters set to 2 returns an ARI score of 0.9476. The selected features are marked in gray in Figure 6.

As shown by the experimental results, CritCF can be used to search for both the most significant feature subspace and the best partition. The results reported are obtained on data sets containing more than 90 data items. However, for very small data sets, the between-cluster inertia and within-cluster inertia computed for optimal partitions with varying number of clusters do not follow the same distribution as the one illustrated in Figure 1, but a more linear one. For this reason, our function was unable to determine the right number of clusters on most of the test instances in class 2d-2c-100 and 10d-2c-100. For example, in case of the instances 10d-2c-100-no0 and 10d-2c-100-no8 (which consist of 118 and respectively 89 data items) CritCF was able to identify the optimum partition while for the rest of the test instances in the class 10d-2c-100 (which consist of less than 68 data items) CritCF biased the search process towards higher numbers of clusters. This phenomenon is common for a wide range of computational problems: there exist thresholds in the parameter space where certain characteristics of the problem change dramatically (phase transitions). Therefore, different algorithms may be appropriate for different instances of the problem. Experiments strongly suggest that the criterion we propose for unsupervised feature selection and clustering is appropriate for problem instances

(data sets) with more than 70 data items.

5. Conclusion and further work

This paper introduces a new clustering criterion which is in most cases unbiased with respect to the number of clusters and which provides at the same time a ranking of partitions in feature subspaces of different cardinalities. Therefore, this criterion is able to provide guidance to any heuristic that simultaneously searches for both relevant feature subspaces and optimal partitions.

CritCF, the new criterion, minimizes the within-cluster variance and simultaneously maximizes the between-cluster separation. It facilitates the use of local Mahalanobis metrics which allow for identification of clusters of more general, ellipsoidal shape. The impact of using Mahalanobis distances in conjunction with the new criterion for unsupervised feature selection will be investigated in future work.

6. Acknowledgements

We would like to thank Julia Handl and Joshua Knowles for supplying us with the data sets investigated in the experimental section and with the results they obtained, making thus possible all the reported comparisons with their extensive studies in unsupervised feature selection and clustering.

References

[1] Aggarwal, C. C., Hinneburg, A., Keim, D. A., 2001. On the surprising behavior of distance metrics in high dimensional space. In: Lecture Notes in Computer Science. Springer, pp. 420–434.

[2] Bautu, E., Bautu, A., Luchian, H., September 2005. Symbolic regression on noisy data with genetic and gene expression programming. In: Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms For Scientific Computing. IEEE Computer Society, pp. 25–29.

[3] Bezdek, J. C., Pal, N. R., 1998. Some new indexes of cluster validity. IEEE Transactions on Systems, Man, and Cybernetics 28 (3), 301–315.

[4] Borgelt, C., 2009. Fuzzy subspace clustering. In: Studies in Classification, Data Analysis, and Knowledge Organization. Springer, pp. 93–103.

[5] Davies, D. L., Bouldin, D. W., 1979. A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (2), 224–227.

[6] Diday, E., Lemaire, J., POuget, J., Testu, F., 1982. Elements d'analyse de données. Dunod.

[7] Duda, R. O., Hart, P. E., Stork, D. G., 2001. Pattern Classification 2nd ed. John Wiley & Sons.

[8] Dy, J., Brodley, C., 2004. Feature selection for unsupervised learning. Journal of Machine Learning Research 5, 845–889.

[9] Handl, J., Knowles, J., 2005. Improving the scalability of multiobjective clustering. In: Proceedings of the Congress on Evolutionary Computation. IEEE Press, pp. 2372–2379.

[10] Handl, J., Knowles, J., 2006. Feature subset selection in unsupervised learning via multiobjective optimization. International Journal of Computational Intelligence Research 2 (3), 217–238.

[11] Hong, Y., Kwong, S., Chang, Y., Ren, Q., 2008. Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. Pattern Recognition 41, 2742–2756.

[12] Hubert, A., 1985. Comparing partitions. Journal of Classification 2, 193–198.

[13] Kim, M., Ramakrishna, R. S., 2005. Some new indexes of cluster validity. Pattern Recognition Letters 26 (15), 2353–2363.

[14] Kim, Y., Street, W. N., Menczer, F., 2002. Evolutionary model selection in unsupervised learning. Intelligent Data Analysis 6 (6), 531–556.

[15] Koza, J. R., 1992. Genetic programming: On the Programming of Computers by Means of Natural Selection. MIT Press.

[16] Law, M. H. C., Figueiredo, M. A. T., Jain, A. K., 2004. Simultaneous feature selection and clustering using mixture models. IEEE Trans. Pattern Anal. Mach. Intell 26, 1154–1166.

[17] Liu, H., Motoda, H., 1998. Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, Norwell, MA, USA.

[18] L.Talavera, 1990. Feature selection as a preprocessing step for hierarchical clustering. In: Proceedings of the Sixteenth International Conference on Machine Learning. Morgan Kaufmann, San Francisco, CA, pp. 389–398.

[19] Luchian, S., 1995. An evolutionary approach to unsupervised automated classification. In: Proceedings of EUFIT'95 - European Forum of Intelligent Techniques. ELITE (European Laboratory for Intelligent Techniques) and Verlag Mainz.

[20] Luchian, S., Luchian, H., 1999. Three evolutionary approaches to classification problems. In: Evolutionary Algorithms in Computer Science. John Wiley & Sons, Chichester-New York-Toronto, pp. 351–380.

[21] Luchian, S., Luchian, H., Petriuc, M., 1994. Evolutionary automated classification. In: Proceedings of 1st Congress on Evolutionary Computation. pp. 585–588.

[22] Milligan, G., Cooper, M. C., 1985. An examination of procedures for determining the number of clusters in a data set. Psychometrika 50, 159–179.

[23] Morita, M., Sabourin, R., Bortolozzi, F., Suen, C. Y., 2003. Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition. In: Proceedings of the Seventh International Conference

on Document Analysis and Recognition. IEEE Press, New York, pp. 666–671.

[24] Raskutti, B., Leckie, C., 1999. An evaluation of criteria for measuring the quality of clusters. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers, pp. 905 – 910.

[25] Rousseeuw, P. J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20 (1), 53–65.

[26] Sndberg-madsen, N., Thomsen, C., Pena, J. M., 2003. Unsupervised feature subset selection. In: Proceedings of the Workshop on Probabilistic Graphical Models for Classification (within ECML 2003).

[27] Varshavsky, R., Gottlieb, A., Linial, M., Horn, D., 2006. Novel unsupervised feature filtering of biological data. Bioinformatics 22 (14), 507–513.

[28] Vemuri, V., Cedeo, W., 1995. Multi-niche crowding for multimodal search. Practical Handbook of Genetic Algorithms: New Frontiers, Ed. Lance Chambers 2.

[29] Zeng, H., Cheung, Y.-M., 2009. A new feature selection method for gaussian mixture clustering. Pattern Recognition 42, 243–250.

Problem	Best		DB		SW		CritCF	
	ARI	k	ARI	k	ARI	k	ARI	k
2d-4c	0.9235 ±0.04	4.02 ±0.14	0.8182 ±0.15	3.30 ±0.78	0.8710 ±0.13	3.70 ±0.78	0.8580 ±0.08	4.01 ±0.79
2d-10c	0.8359 ±0.07	10.93 ±2.09	0.6869 ±0.14	7.78 ±3.35	0.7800 ±0.10	11.24 ±3.37	0.7945 ±0.08	10.39 ±2.44
2d-20c	0.9133 ±0.02	19.63 ±2.13	0.8046 ±0.12	14.17 ±3.15	0.8700 ±0.07	16.71 ±2.90	**0.8902** ±0.04	17.45 ±2.53
2d-40c	0.8347 ±0.02	39.48 ±4.50	0.7070 ±0.10	25.39 ±6.37	**0.8023** ±0.04	34.84 ±5.27	0.7868 ±0.05	30.64 ±4.74
10d-4c	0.9711 ±0.02	3.99 ±0.07	0.9158 ±0.14	3.69 ±0.64	0.9044 ±0.14	3.59 ±0.66	0.9327 ±0.04	3.5 ±0.50
10d-10c	0.9246 ±0.02	9.21 ±0.82	0.8930 ±0.07	8.62 ±1.26	**0.9178** ±0.02	9.03 ±0.78	0.8958 ±0.03	8.36 ±1.25
10d-20c	0.9744 ±0.01	20.23 ±1.82	0.9189 ±0.06	17.08 ±1.84	0.9479 ±0.04	18.05 ±1.59	**0.9636** ±0.02	20.44 ±1.63
10d-40c	0.9582 ±0.0122	41.36 ±3.25	0.8849 ±0.0529	32.03 ±2.89	0.9282 ±0.0379	35.48 ±3.12	**0.9459** ±0.02	42.975 ±3.23
2d-4c-100	0.8090 ±0.13	3.9 ±0.83	0.6996 ±0.15	3.39 ±2.43	0.6419 ±0.15	2.96 ±1.14	**0.7467** ±0.12	4.00 ±1.00
2d-10c-100	0.7913 ±0.06	10.7 ±2.19	0.6608 ±0.15	8.44 ±4.02	0.6880 ±0.13	8.945 ±4.05	**0.7420** ±0.05	9.65 ±2.90
10d-4c-100	0.9610 ±0.05	3.88 ±0.32	0.7963 ±0.26	4.66 ±3.52	0.8600 ±0.22	4.06 ±2.53	**0.9263** ±0.12	3.93 ±0.44
10d-10c-100	0.8805 ±0.04	9.70 ±1.19	0.8392 ±0.06	8.37 ±1.30	**0.8600** ±0.06	9.43 ±1.53	0.8327 ±0.07	8.18 ±1.39

Table 1: Results on synthetic and real data sets - partitions obtained with the k-Means algorithm. The ARI score and the number of clusters k reported here, are computed as averages over 20 runs per data set. For each data set, four partitions are reported: the one with the highest ARI value (Best) and the partition found by Davis-Bouldin Index (DB), Silhouette Width (SW) and CritCF function, respectively.

Problem	ARI	k	m	recall	precision	F-measure
2d-4c-100-Gaussian	0.5649 ±0.28	6.27 ± 4.77	3.85 ± 4.73	0.93	0.80	0.86
2d-10c-100-Gaussian	0.7281 ±0.07	9.52 ± 3.40	2 ± 0	1	1	1
10d-4c-100-Gaussian	0.9087 ±0.13	3.95 ± 0.45	7.36 ± 1.33	0.74	1	0.85
10d-10c-100-Gaussian	0.8528 ±0.07	8.51 ± 1.23	9.20 ± 0.71	0.92	1	0.95

Table 2: Results for feature selection obtained with the MNC-GA algorithm using CritCF, on data sets with gaussian noise (100 gaussian features). The ARI score for the best partition, the number of clusters k, the number of features m, the recall and the precision of the selected feature space are computed as averages over 20 runs on each data set.

Problem	Alg.	ARI	k	m	recall	precision	F-measure
2d-4c-100-gaussian	FS1	0.6321	5.60	2.30	1	0.90	0.94
	FS2	0.6391	5.4	2.3	1	0.90	0.94
2d-10c-100-gaussian	FS1	0.7464	9.3	2	1	1	1
	FS2	0.7491	9.4	2	1	1	1
10d-4c-100-gaussian	FS1	0.8580	3.80	7.20	0.72	1	0.83
	FS2	0.9055	4	7.5	0.75	1	0.85
10d-10c-100-gaussian	FS1	0.8084	7.70	9.00	0.90	1	0.9428
	FS2	0.8648	8.7	9.4	0.94	1	0.9678

Table 3: Results for feature selection obtained with the two versions of the Forward Selection algorithm using CritCF on data sets with gaussian noise (100 gaussian features). The ARI score for the best partition, the number of clusters k, the number of features m, the recall and the precision of the selected feature space are listed.

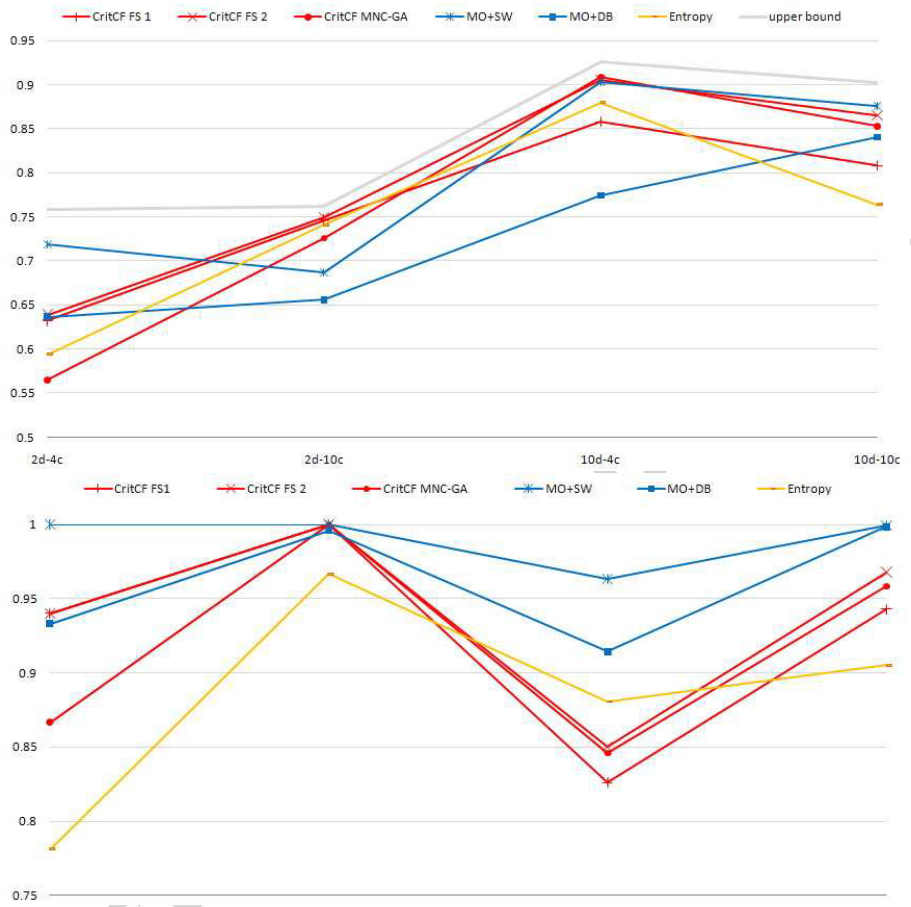

Figure 4: Results for the datasets containing Gaussian noise. Adjusted Rand Index (top) and F-Measure (bottom) for the best partition obtained in the feature subspace extracted with various methods: the three red lines correspond to the MNC-GA and the two versions of Forward selection algorithm using CritCF; the two blue lines correspond to the multi-objective algorithm investigated in [10] using Silhouette Widthand Davies Bouldin as the primary objective. The yellow line corresponds to a filter method investigated in [10] using an entropy measure. The gray line corresponds to the best partition that can be obtained with k-Means run on the optimal standardized feature subset

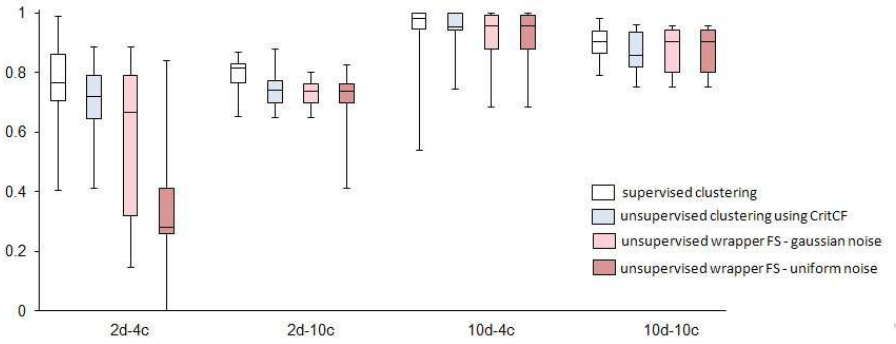

Figure 5: The Adjusted Rand Index for the partitions obtained as follows: supervised clustering on the relevant features, unsupervised clustering on the relevant features using CritCF, unsupervised wrapper feature selection using CritCF on datasets containing 100 gaussian features and on datasets containing 100 uniform features. Each boxplot summarizes 10 values corresponding to the 10 problem instances in each class.

Problem	Alg.	ARI	k	m
Letters AB	kMeans	0.7524	2	16
	MNC-GA	0.7704	2	8
Digits 56	k-Means	0.9475	2	64
	MNC-GA	0.9347	3	6

Table 4: Results for feature selection on real data sets. The first line for each data set presents the performance of k-Means on the initial data set with the correct number of clusters. The second line presents the performance of MNC-GA for unsupervised wrapper feature selection: the ARI score, the number of clusters k identified and the number of features m selected.

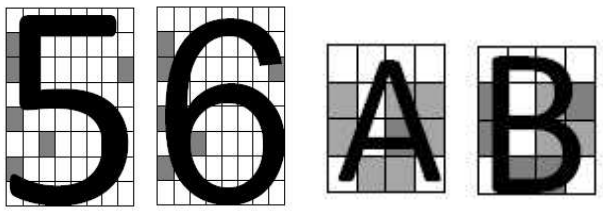

Figure 6: Results for MNC-GA on real data. The selected features are marked in gray

Mihaela Breaban - received the B.Sc. in 2003 and the M.Sc. in 2005, both in Computer Science from ”Alexandru Ioan Cuza” University of Iasi, Romania. She is currently a teaching assistant at the Faculty of Computer Science and she is studying for a PhD at ”Alexandru Ioan Cuza” University. The current research topic is unsupervised classification.

Henri Luchian - received the *MSC in Computer Science* at the Faculty of Mathematics, "Alexandru IoanCuza" University of Iasi, Romania. He received the Ph.D. *in Computer Science* from "Babes-Bolyai" University of Cluj-Napoca, Romania. Since 2000 he is professor at the Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iasi.