

# Feature subset selection in unsupervised learning via multiobjective optimization

**Julia Handl**

Manchester Interdisciplinary Biocentre  
University of Manchester  
j.handl@postgrad.manchester.ac.uk

**Joshua Knowles**

Manchester Interdisciplinary Biocentre  
University of Manchester  
j.knowles@manchester.ac.uk

**Abstract-** In this paper, the problem of unsupervised feature selection and its formulation as a multiobjective optimization problem are investigated. Two existing multiobjective methods from the literature are revisited and used as the basis for an algorithmic framework, encompassing both wrapper and filter methods of feature selection. A number of alternative algorithms implemented within this framework are then evaluated using an extensive data test suite; the main effect investigated is that of the choice of a primary objective function (a secondary objective function is used only to militate against an inherent cardinality bias affecting all methods of feature subset evaluation). Particular attention is paid in the study to high-dimensional data sets in which the number of features is much larger than the number of data items.

## 1 Introduction

Feature selection, or subset selection, is a process commonly used for dimensionality reduction in machine learning. Dimensionality reduction in learning tasks can be crucial for a number of reasons. First, for large feature sets, the processing of all available features may be computationally infeasible. Second, many of the available features may be redundant, noise-dominated or irrelevant to the classification task at hand. Consequently, the inclusion of all features will be detrimental and the subset most relevant for the learning task at hand needs to be identified. Third, high-dimensionality is also a problem if the number of variables is much larger than the number of data points available. In such a scenario, dimensionality reduction is crucial in order to overcome the curse of dimensionality [2] and allow for meaningful data analysis.

For the above reasons, feature selection is important both in supervised and unsupervised data analysis. The problem has been well-studied in the supervised scenario but only little research to date has dealt with the unsupervised problem (for a recent overview of research efforts in both areas, see [24]). Yet, several of the challenges faced in the unsupervised problem are very different to those encountered in a supervised scenario: in particular the assessment of the quality of an individual feature or a feature subset becomes even more intricate in unsupervised classification.

In this paper, the benefits of treating unsupervised feature selection as a multiobjective optimization problem are discussed and variations of this formulation are considered. In particular, the main contributions of this work are as follows.

1. A critical review of two seminal evolutionary multi-objective approaches to unsupervised feature selection [22, 31] is given. Experimental examples are used to highlight some of the limitations of these approaches.
2. A common algorithmic framework is developed to enable us to test different choices of objective function(s) whilst maintaining the same optimizer, pre-processing, initialization, encoding and clustering modules. We also devise a method to select solutions from the Pareto front by comparison with control distributions so that no external knowledge is needed.
3. The framework is then used to undertake an extensive experimental analysis using a total of 125 different data sets. Altogether, four alternative MOEA approaches are evaluated, together with three baseline methods: two provide estimates of an upper and lower bound on feature selection performance and the third shows the performance of a greedy optimizer. Our analysis pays particular attention to the performance of the algorithms on high-dimensional data, for which the number of dimensions greatly exceeds the number of data items. Such data is encountered in many data-mining tasks (for example in biological and chemical data analysis [19]), but has not been considered in previous work on unsupervised feature selection [13, 22, 31].

The remainder of this paper is structured as follows. Section 2 introduces a definition of feature selection and gives an overview of previous research related to both supervised and unsupervised versions of the problem. It also sets out the motivation for tackling feature selection from a multi-objective perspective and summarizes previous research efforts in this respect. Next, Section 3 gives a more detailed analysis of two existing multiobjective approaches to the unsupervised problem. In Section 4, we introduce the algorithmic framework that allows us to analyze several alternative objective function choices, while maintaining many common components. Section 5 describes the experimental setup used and gives results on the performance of a number of algorithms implemented within our framework. Section 6 discusses the findings of the study, and, finally, Section 7 concludes.

## 2 Related work on feature selection

As outlined in the introduction, dimensionality reduction is an important processing step in many data-mining scenarios. In general, two different types of dimensionality reduction can be distinguished.

The first group consists of methods that are based on a transformation of the original feature space. In these methods, features in the transformed feature space consist of linear or non-linear combinations of the original variables; usually, a limited number of these transformed features are selected in order to obtain a representation of the data in a lower dimensional data manifold. Both unsupervised and supervised methods of this type exist and examples include principal component analysis (which is unsupervised) [27] and discriminant function analysis (which is supervised) [27]. These transformation-based approaches are potentially powerful, as they can capture complex relationships between variables, although the main focus of most of these methods is not on retrieving cluster structure. Further disadvantages include their computational expense and the difficulty to interpret results in terms of the original input variables.

The second group comprises those methods that are based on subset selection, also referred to as feature selection [17]. Here, a lower dimensional projection of the data is obtained by selecting a subset of the original features and discarding the remaining ones. Hence, in this approach all features in the reduced feature space directly correspond to a single feature in the original high-dimensional feature space. Both unsupervised and supervised methods for feature subset selection exist and we will survey these in the following sections. The main advantage of the approach lies in its ease of interpretation: the method directly returns the variables that are relevant for a given classification task.

### 2.1 Definition of feature selection

The general feature selection problem  $(\Omega, P)$  can be defined formally as a single-objective optimization problem: determine the feature set  $F^*$  for which

$$P(F^*) = \min_{F \in \Omega} P(F, E), \quad (1)$$

where  $\Omega$  is the set of feasible feature sets,  $F$  is a feature subset and  $P : \Omega \times \Psi \rightarrow \mathbb{R}$  is a criterion function that assesses the quality of a given feature subset in terms of its utility for classifying the set of data points,  $E \in \Psi$ . That is to say, the elements of  $E$ , which are vectors in a metric space of dimension  $d$  are projected into the subspace of dimension  $d_F = |F| \leq d$  defined by  $F$  and the quality of this subspace is estimated by  $P$ .<sup>1</sup>

In a supervised scenario, the correct class memberships for all data points within  $E$  are additionally known and the utility of  $F$  is usually measured in terms of the performance of a classifier at predicting the memberships of all data points within  $E$  when applied to their projections

in feature space  $F$ . This can either be measured directly with respect to a given classifier or it can be estimated using ‘proxy’ measures that consider how well the known classes are reflected by the distribution of feature values.

In an unsupervised scenario, utility is usually measured in terms of the performance of a clustering method when applied to  $E$  in feature space  $F$ . Analogously to the supervised case, this can either be measured directly with respect to a given clustering algorithm or it can be estimated using proxy measures that consider the degree to which the distribution of feature values exhibits cluster structure in the subspace  $F$ .

Feature selection methods that assess feature utility with respect to a given classifier or clustering method, are referred to as ‘wrapper’ approaches. In contrast, those feature selection methods that make use of a proxy measure to estimate utility are termed ‘filter’ approaches.

The search space of the feature selection problem is of size  $2^d$  (each feature can be either selected or not). Therefore, exhaustive search becomes infeasible even for moderate  $d$  and heuristic search strategies need to be employed. Possible choices for such search strategies include branch-and-bound [4], sequential search methods (such as forward selection [1], backward selection [1] or floating search [35]) and randomized search strategies (such as simulated annealing [25] or evolutionary algorithms [39]). An alternative and computationally cheaper approach is that of feature ranking: here, the utility of individual features is studied in isolation only and those features with the highest utility (e.g. those exceeding a certain threshold level) are selected.

### 2.2 Supervised feature selection

Feature selection is a well-studied problem in the area of supervised classification [17]. Performance in supervised classification is typically measured as the ability of a classifier to classify all data within a data set  $E$  correctly, even when only a subset of  $E$  is used for ‘training’ the classifier. This is termed cross-validation and the performance of a classifier is quantified by the cross-validation error.

In the training of a classifier, the most straightforward choice of input variables is the set of all available features. While this may appear to be the most powerful and general approach, as all available information is taken into account, research in the machine learning literature has shown that this is usually not the case in practice [3]. Real data frequently contain variables that are redundant or have a low information content; their consideration during the classification process introduces noise and may cause high cross-validation errors. The exclusive use of the most discriminative variables may therefore yield significant gains in terms of classification performance.

Wrapper methods for supervised classification [23] interact directly with (are ‘wrapped around’) a specific classifier. The usefulness of a particular feature set can then be assessed directly by the performance (cross-validation error) of a classifier that is trained on this set of variables only. Wrapper methods are very effective at decreasing the dimensionality of the feature space and increasing classifi-

<sup>1</sup>Note: since we are minimizing, strictly,  $P$  is a cost or error term; if a utility measure is used for  $P$ , we shall minimize its negative value.

cation accuracy. Their disadvantages include their computational expense and their susceptibility to overtraining [25].

Filter approaches to supervised classification typically select variables based on their discriminatory power with regard to the target classes. Popular methods in this respect include distance, dependency, information and consistency measures [24]. As filter methods are independent of the classifier applied subsequently, they have good generalization properties, but may be less effective at decreasing the dimensionality of the feature space and boosting classification accuracy. Generally, they are computationally cheaper than wrapper approaches, albeit their computational expense varies largely dependent on whether feature ranking or a search for feature subsets is used.

### 2.3 Unsupervised feature selection

Unsupervised feature selection has been addressed only relatively recently in the literature [8, 13, 16, 22, 30, 31, 34, 41, 42]. Performance in unsupervised classification is typically measured as the ability of a clustering to reveal ‘interesting’ groupings (clusters) in a given data set  $E$ . This is where a fundamental difference between unsupervised and supervised feature selection lies: while the overall objective is relatively clearly defined in a supervised scenario (usually as the minimization of the cross-validation error), the definition of a suitable objective for unsupervised feature selection is more ambiguous and involved. This arises from the difficulty of defining the notion of a good cluster and of objectively evaluating clustering quality in the absence of known class labels [29].

Wrapper approaches to unsupervised classification evaluate feature subsets in the context of a specific clustering algorithm. More specifically, a clustering algorithm is applied to a given feature subspace and the quality of the resulting clustering solution is evaluated using internal cluster validation techniques [19]. Here, the term “internal” signifies the fact that no external information, that is, information about the class memberships of individual data items, is used for the evaluation of individual clustering solutions. A range of such internal validation techniques exist in the clustering literature, each of which has its own biases, strengths and limitations. The choice of this cluster validation technique may potentially have a major impact on the overall performance of an unsupervised feature selection scheme. Furthermore, the best choice for the validation technique is not necessarily independent of the clustering algorithm used and the combination of the two must therefore be chosen carefully. A third choice faced in wrapper approaches to unsupervised feature selection concerns the number of clusters that the data are partitioned into: most clustering algorithms require the specification of the number of clusters as an input parameter and this parameter can be kept constant or dynamically determined within a wrapper approach. Previous research by Dy and Brodley [13] has demonstrated that a dynamic number of clusters is preferable because different feature subspaces may contain different numbers of clusters.

Filter approaches typically select variables based on the

distribution of their values across the set of data vectors available. In contrast to wrapper approaches, the most common filter strategies are based on feature ranking. In this context, two opposite strategies have been proposed in the literature: those that aim at the removal of redundant features [30] and those that focus on the removal of irrelevant features [41, 34, 42]. Both, redundancy and irrelevance can be determined, for example, by the mutual information or the correlation between pairs of features. However, redundancy-based approaches are based on the assumption that dependent features should be discarded, whereas irrelevance-based approaches stipulate the opposite (i.e. the preservation of dependent features and the removal of independent ones). This apparent contradiction demonstrates some of the difficulty of deriving ranking techniques that are appropriate in all data-mining scenarios. Recently, a number of filter approaches have been introduced that select feature subsets containing areas of high density. This can be measured, for example, by the entropy of the distribution of data points [16] or the entropy of the distribution of dissimilarities between data items [8].

In supervised feature selection, wrapper approaches are thought to offer performance advantages although there is also a danger of overfitting with them. In the unsupervised context, however, the advantages and disadvantages of filter versus wrapper approaches are less well understood, and we consider both in the experimental sections that follow.

#### 2.3.1 Feature cardinality bias

We have mentioned above that wrapper methods for unsupervised feature selection rely on the use of an internal technique of cluster validation. Cluster validation techniques have been designed specifically in order to allow for the selection of the best clustering solution out of a set of partitionings obtained on the same set of data but generated by different algorithms or corresponding to different numbers of clusters. However, they are not originally aimed at accurately comparing partitionings obtained on different sets of data or in different feature subspaces.

In fact, internal cluster validation techniques are generally based on some form of distance computation in feature space and this is problematic for their use in feature selection, as it automatically induces a bias of these measures with respect to the dimensionality of the feature space. The existence of this bias is related to the fact that, when moving to high dimensions, the histogram of distances between items in data space changes: the mean of the histogram tends to increase and the variance of the histogram tends to decrease. In other words, the distances between all pairs of points tend to become highly similar and (dependent on the specific form of the validation technique) this causes a bias to low or high dimensions. For example, many cluster validation techniques consider ratios between intra-cluster compactness and inter-cluster separation, the values of which draw closer for high dimensions. Consequently, these validation techniques are biased towards low dimensions and clustering solutions in higher-dimensional space that are actually better than solutions in lower-dimensional

space, may be overlooked.

Such biases complicate the validation of clustering results across subspaces of different dimensionalities. If the natural dimensionality-bias is not accounted for, a wrapper-based feature selection method will always favour extreme feature spaces (i.e. the lowest- or highest-dimensional feature spaces available). Similar limitations hold for filter approaches that compare subspaces of different dimensionality and whose evaluation is based on distance computations.

In the literature, three different approaches have been proposed to tackle the issue of bias. The first approach is a simple ad-hoc normalization of the evaluation function by means of an appropriate scaling factor (usually expected to be a function of the feature cardinality  $d_F$ ) [13, 22]. This type of normalization may reduce the bias or overcompensate for it, but will not usually remove it cleanly. An alternative approach is the cross-projection technique proposed by Dy and Brodley [13], which attempts to reduce the cardinality-specific bias by considering pairs of clustering solutions, each derived in a different feature subspace, and comparing each of them in *both* of these subspaces. This relation can be used for pairwise comparisons between features sets but it is not transitive, which makes its use in global optimization techniques problematic. Finally, two recent papers [22, 31] have suggested dealing with the bias by considering feature cardinality as a separate objective and applying a Pareto multiobjective optimization algorithm. Of the three approaches, we consider the latter the most general and promising and will focus on it in the remainder of this paper.

## 2.4 Multiobjective approaches to feature selection

A number of papers in the literature have investigated multiobjective approaches to feature selection. It is worth noting that the motivations for the use of multiobjective techniques in supervised and unsupervised feature selection are quite different.

In the supervised problem, the literature has focused predominantly on wrapper approaches, where the performance of a classifier is to be maximized, while minimizing the cardinality of the feature set [12, 14, 32]. This approach is largely motivated by the problem of overtraining. In a supervised context, larger feature sets will usually result in a higher classification accuracy on the training data. Yet, these solutions will be prone to overfitting and will have a low generalization performance. The simultaneous maximization of the accuracy of a classifier and minimization of the cardinality of the feature space is therefore favoured in order to explore different bias-variance trade-offs: given two feature sets of different cardinality that result in the same classification accuracy, the smaller of the two is expected to result in a better generalization (this is also known as Occam’s Razor [38, 40, 44]). Implementations of multiobjective supervised feature selection directly capture this intuition (and discard the larger solution) by means of the concept of Pareto optimality.

The above type of overfitting is not an issue in an unsupervised scenario. Instead, in unsupervised feature se-

lection, multiobjective approaches are of interest as a flexible approach to counter-balance the dimensionality-bias of cluster validation techniques. Note that this represents a major difference between supervised and unsupervised feature selection: while obtaining a low-dimensional feature set is often desirable (as it facilitates the interpretation of the final clustering result), a preference for smaller solutions should not *generally* be used: given two feature sets of different cardinality that result in the same clustering performance (as measured by a cluster validation technique/filter method with a bias to low dimensionalities), the larger of the two is expected to correspond to the better clustering result. Implementations of multiobjective unsupervised feature selection, we suggest, should therefore minimize *or* maximize the number of features as appropriate to counteract the specific bias of the cluster validation technique/filter method employed.

Multiobjective approaches to unsupervised feature selection have been studied in only two papers [22, 31] to our knowledge, both of these suggesting wrapper-approaches. In the following section, we will discuss and analyze this work in some detail.

## 3 Analysis of previous work on multiobjective unsupervised feature selection

### 3.1 Kim *et al.*’s algorithm

In 2002, Kim *et al.* [22] presented the first wrapper approach to multiobjective unsupervised feature selection, which is based on the multiobjective algorithm ELSA (Evolutionary Local Selection Algorithm, [28]). The  $k$ -means algorithm [26] is used to obtain a clustering for a given feature subset, and both good feature subsets and the corresponding numbers of clusters are evolved. Four different clustering objectives are optimized simultaneously, which measure (some function of) the number of features, the number of clusters, intra-cluster compactness and between-cluster separation.

The approach was evaluated on one synthetic (‘Kim’) and one real world data set (Wisconsin Prognostic Breast Cancer, ‘WPBC’), which can both be characterized as relatively low-dimensional in the sense that the number of data items is much larger than the number of features. For the synthetic data, the correct solution is known, so we will use it here to discuss certain aspects of Kim *et al.*’s approach. This data set consists of 500 data points described by 30 features each. Out of these, only the first 10 dimensions are designed to contain structure in the form of five Gaussian clusters that are embedded in this 10-dimensional space. The remaining 20 dimensions are noise dimensions: features 11 to 20 are each sampled from two Gaussian distributions (independently generated for each feature); features 21 to 30 are each sampled from a Uniform distribution. This data contains a number of interesting patterns, as shown by several two-dimensional projections in Figure 1.

While the general idea proposed by Kim *et al.* seems very promising, their results on the ‘Kim’ data, as well as our own analysis, suggest a number of pitfalls of their par-

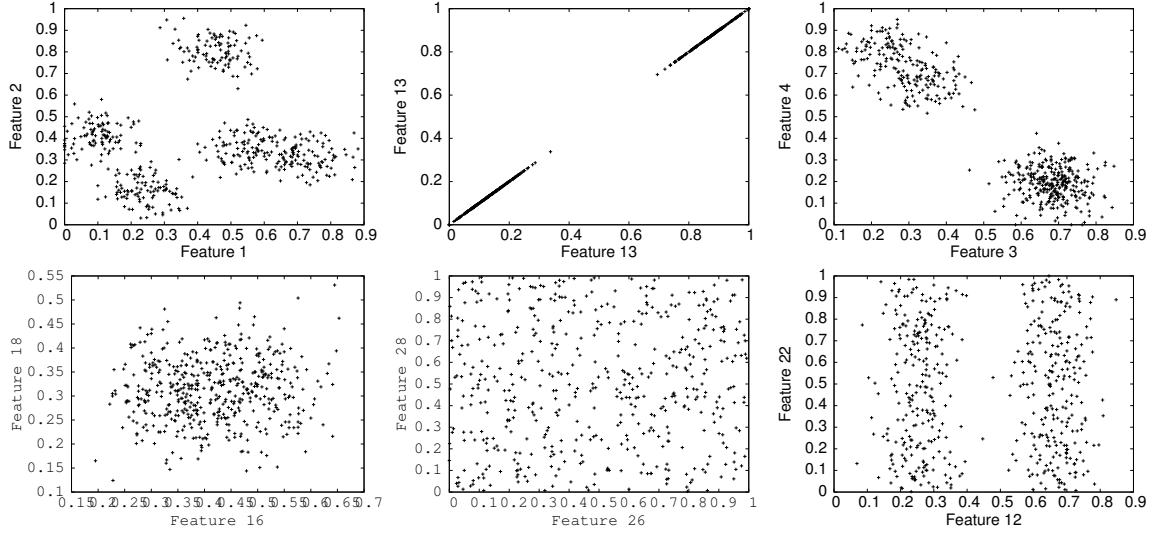


Figure 1: Two-dimensional projections of the ‘Kim’ data set. From left to right, top to bottom: (a) Dimension 1  $\times$  2. The five 10-dimensional Gaussian clusters can be discerned in this projection to two dimensions. (b) Dimension 13  $\times$  13. The two Normal Distributions used for noise generation in this dimension happen to be spatially separated. This results in an artefact in this subspace. (c) Dimension 3  $\times$  4. Only two clusters can be discerned in this projection to two dimensions. (d) Dimension 16  $\times$  18. The Normal Distributions used for noise generation in dimensions 16 and 18 overlap. This subspace therefore contains no cluster structure. (e) Dimension 26  $\times$  28. Dimensions 26 and 28 are sampled from uniform distributions. This subspace therefore contains no cluster structure. (f) Dimension 12  $\times$  22. Dimension 22 is sampled from a uniform distribution. However, the two Normal Distributions used for noise generation in dimension 12 happen to be spatially separated. This results in an artefact in this subspace.

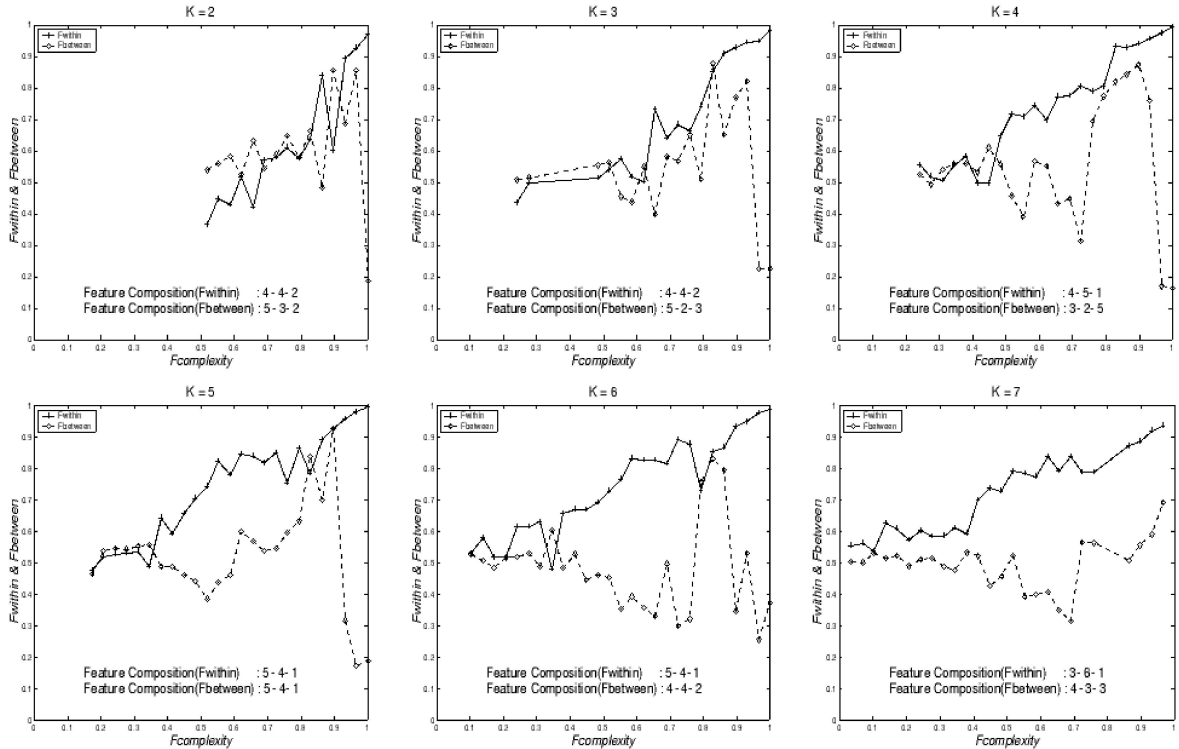


Figure 2: Slices of the Pareto front identified by Kim *et al.* (reprinted from [22] with permission from IOS Press).  $F_{complexity}$  denotes the objective related to the cardinality of the feature space.  $F_{within}$  denotes the objective related to intra-cluster compactness.  $F_{between}$  denotes the objective related to between-cluster separation. These three objectives are to be maximized.  $k$  denotes the number of clusters. These plots are quite difficult to interpret for a decision maker. Kim *et al.* suggest that the slices for  $k = 5$  and  $k = 6$  can be identified as promising due to the high scores obtained for  $F_{within}$  and the large range of  $F_{complexity}$  values covered.

ticular approach.

### 1. Efficiency of the optimization

Four objectives are a large number of objectives, in particular, if Pareto-based ranking is used. Recent results by Hughes [21] suggest that the performance of Pareto-based ranking MOEAs can be quite poor for this number of objectives and we therefore would expect deficiencies in the optimization performance of Kim *et al.*'s method. This expectation seems to be confirmed by the results provided by Kim *et al.* for the 'Kim' data set: the feature sets identified by their method include a significant fraction of noise features and the embedded clustering algorithm does not succeed in correctly retrieving the five clusters inherent to the data.

### 2. Dependencies between objectives

The values returned by the two internal clustering validation techniques (i.e. compactness and separation) are biased with respect to the cardinality of the feature space and the number of clusters in the partitioning. Kim *et al.* attempt to counter-balance the latter of these biases through a division of both measures by the cardinality of the feature space. Given that the cardinality of the feature space is also an objective, this division may be problematic: a transformation of this kind will usually affect the Pareto optimal set. We will discuss the implications of this type of transformation in more detail in Section 3.2.

### 3. Interpretability of the results

Kim *et al.*'s algorithm generates a four-dimensional Pareto front. Consequently, an interpretation of this front and the discovery of promising solutions from within it, becomes highly involved. This is illustrated in Figure 2, which shows results reproduced from Kim *et al.*'s paper. These results are evidently difficult to interpret, which is caused by two factors: first, it is generally harder to interpret Pareto fronts in four dimensions, as only two- or three-dimensional slices of the Pareto front can be investigated at a time; secondly, the interpretation is exacerbated by the different biases of the objectives as well as the dependencies between individual objectives.

Problems (1) and (3) can best be appreciated if contrasted with the performance of an alternative method on the same data set. Figure 3 shows the result obtained using our algorithmic framework (described later in Section 4) on the 'Kim' data set. The method optimizes two objectives only: a measure of cluster validation (Silhouette Width, see Section 4.2) and the cardinality of the feature subspace  $d_F$ . As the output of this method, a simple two-dimensional Pareto front is obtained, which is straightforward to interpret. The method identifies the correct 10-dimensional feature subspace and the correct number of clusters. Furthermore, this best solution is located at a clearly discernible 'knee' in the Pareto front and can therefore be found easily by a decision maker.

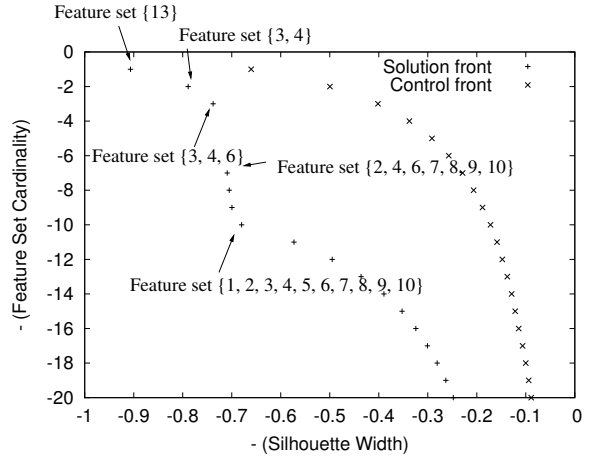


Figure 3: Pareto front identified on the 'Kim' data set using Silhouette Width as the primary objective (both objectives being minimized). This two-dimensional Pareto front reflects the main trends in the data and is straightforward to interpret. Specifically, the correct  $d_F = 10$  solution is situated at a 'knee' in the Pareto front and is easy to detect by a decision maker. A control front obtained for random data is shown to indicate the dimensionality bias of the primary objective used (see Section 4.4).

In addition, this simple two-dimensional Pareto front reveals some interesting insights into the structure of the data set that have not been highlighted in [22]. In particular, a Pareto optimal solution is identified for  $d_F = 1$ . It consists of a feature (feature 13) that is not originally considered 'significant' and is also not included in the larger feature subsets (i.e. the solutions for  $d_F \in \{2, 3, 7, 8, 9, 10\}$ ). A plot of the subspace spanned by feature 13 (see Figure 1) reveals the reasons for this result: as an artefact of the method of noise-generation (in particular the use of *two* Gaussian clusters per dimension), this subspace contains very clear cluster structure, which, for this feature cardinality, is more distinct than any of the one-dimensional projections of the five Gaussian clusters. However, due to the fact that this artefact is not correlated with any of the structure in the other features, it is discarded when higher-dimensional subspaces are considered. Furthermore, four Pareto-optimal solutions can be identified in the range  $d_F \in \{7, \dots, 10\}$ , for which  $k$ -means identifies the correct  $k = 5$  clustering solution. This indicates that not all of the first ten features are required to correctly discern the five-cluster structure of the data.

### 3.2 Morita *et al.*'s algorithm

Morita *et al.* [31] present an alternative multiobjective wrapper approach to unsupervised feature selection. Like Kim *et al.* [22] they use the  $k$ -means algorithm for the generation of candidate partitionings and keep the number of clusters dynamic. A different multiobjective evolutionary algorithm (the Non-dominated Sorting GA-II, NSGA-II, [11]) is used and just two clustering objectives are employed, namely the number of features and the Davies-Bouldin-Index (DB-Index, [9]).

The Davies-Bouldin-Index is a popular internal valida-

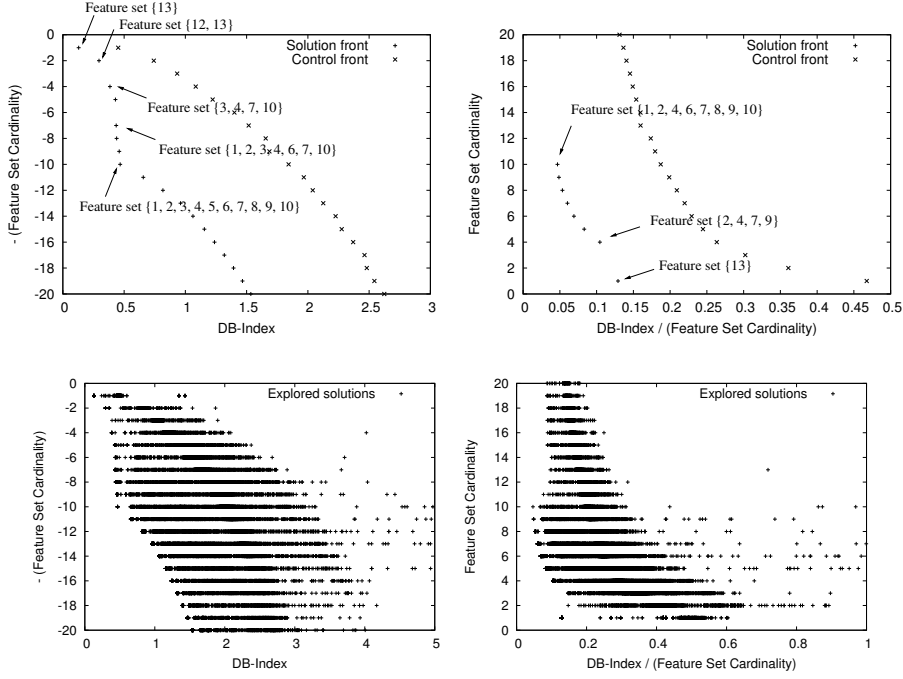


Figure 4: Solutions explored and Pareto fronts obtained for two different bi-objective optimization tasks. (Left) Minimization of the DB-Index and maximization of the number of features. (Right) Minimization of the normalized DB-Index and minimization of the number of features. Control fronts obtained for random data are shown to indicate the dimensionality bias of the primary objectives used (see Section 4.4).

tion technique from the literature that captures both intra-cluster compactness and between-cluster separation (in a non-linear combination). It is defined as

$$I_{DB} = \frac{1}{k} \sum_{i=1}^k R_i, \quad (2)$$

where

$$R_i = \max_{j, j \neq i} \frac{S_i + S_j}{B_{ij}}$$

is the ratio of the sum of within-cluster scatter

$$S_i = \frac{1}{|C_i|} \sum_{x \in C_i} \|x - Z_i\| \quad (3)$$

to the between-cluster separation

$$B_{ij} = \|Z_i - Z_j\|. \quad (4)$$

Here,  $Z_i$  denotes the cluster centre of cluster  $C_i$ .  $I_{DB}$  takes values in the interval  $\mathbb{R}_0^+$  and is to be minimized.

The use of the DB-Index has some strong advantages over Kim *et al.*'s approach [22]. First, it reduces the number of objectives that need to be considered. Second, the DB-Index is unbiased with respect to the number of clusters. Values returned by the DB-Index are therefore easier to interpret, which makes the final Pareto front easier to analyze. Also, it removes the need for the use of the number of clusters as an additional objective.

However, the DB-Index is biased with respect to feature cardinality and improves for small feature subspaces. As outlined previously, a possible approach to deal with this bias is to treat feature cardinality as an individual objective. This can be done in two different ways:

1. Minimization of the DB-Index and maximization of the number of features. The bias of the DB-Index can be counterbalanced by maximizing the number of features instead of minimizing it. This approach has not been considered in the literature, presumably for the reason that a maximization of the number of features seems somewhat counter-intuitive to the idea of feature selection.
2. Normalization of the DB-Index to reverse the bias toward small feature subspaces. Once the DB-Index has been 'normalized' in this way, a minimization of both the normalized DB-Index and the number of features is possible. This is the approach chosen by Morita *et al.*, who divide the DB-Index by the cardinality of the feature space

$$I_{DB}^* = \frac{1}{d} \frac{1}{k} \sum_{i=1}^k R_i. \quad (5)$$

The resulting index is biased towards high-dimensional feature spaces and this is then counter-balanced through the minimization of feature cardinality as a second objective. While this minimization of the number of features seems more 'natural', there are two potential pitfalls with this approach. First, the exact form of the bias is not known and the division by  $d_F$  is a heuristic normalization only. It is not clear whether this normalization properly removes the bias for all feature cardinalities. Second, the DB-Index (which is the real objective) is divided by the feature cardinality, which is, at

the same time, used as the second objective. The division of one objective by another is not usual in Pareto optimization and may yield a different Pareto optimal set than the one desired.

In previous work [22, 31], strategy (2) has been applied without any investigation of its possible side-effects. Here, we will try to highlight some of the effects of this transformation step and discuss the advantages and disadvantages of both strategies. For this purpose, Figure 4 shows the Pareto fronts obtained on the ‘Kim’ data set for the two different strategies (which, essentially, correspond to different optimization problems). Some distinct differences between the Pareto optimal sets obtained can be identified. Evidently, strategy (1) obtains Pareto optimal solutions for the range of feature set cardinalities up to  $d_F = 20$ , with the exception of  $d_F = 3$  and  $d_F = 6$ . In contrast, strategy (2) obtains Pareto optimal solutions for  $d_F = 1$  and  $d_F \in 4, \dots, 10$  only. The solutions for  $d_F = 2$ ,  $d_F = 3$  and  $d_F > 10$  are dominated and do not form a part of the Pareto optimal set.

This example highlights one potential advantage of Morita *et al.*’s approach: if very good low-dimensional clustering solutions exist, these solutions tend to dominate the solutions in all higher-dimensional subspaces. For data sets with very clear structures, the exploration of the solution space can therefore focus on low-dimensional feature subspaces, which may enable a more efficient optimization. However, the example also highlights a potential pitfall of the approach: the Pareto optimal set is changed and, in the process, interesting solutions may be lost. For the ‘Kim’ data set, an example of this is the  $d_F = 2$  solution, which corresponds to a very clear cluster structure (see Figure 3), but, nevertheless, becomes dominated by the  $d_F = 1$  solution. More generally, all solutions that are situated upstream of any distinct ‘knees’ in the Pareto front (in the optimization problem corresponding to strategy (1)), are lost in the transformed optimization problem. Apart from the loss of these solutions, this has a second serious implication: the resulting Pareto front becomes less structured and no clear ‘knees’ are discernible. This complicates the analysis and interpretation of the final Pareto fronts and the selection of good solutions.

Despite these differences, it may be argued that both approaches still succeed in finding the ‘best’  $d_F = 10$  solutions on the ‘Kim’ test set. In our more extensive experimental section to follow, we revisit this issue in greater depth and find that this choice has a significant effect on the outcome for other data sets.

## 4 An algorithmic framework for multiobjective unsupervised feature selection

In the previous sections, we have discussed the need for unsupervised feature selection and have highlighted why a multiobjective approach to the problem seems particularly promising. Our review of previous research in this respect has indicated that the two algorithms developed to date seem generally promising, but it has also pinpointed a number of potential limitations of that work. In this section,

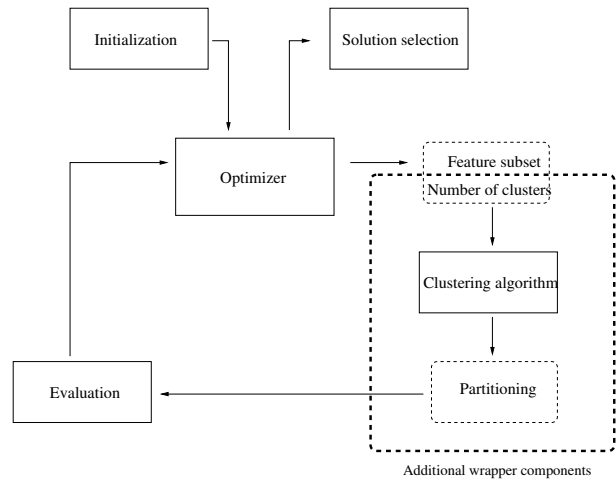


Figure 5: The main components of our framework for multi-objective unsupervised feature selection. After an initialization phase, the main loop of the algorithm is started. A search algorithm/optimizer constructs candidate solutions that specify a feature subset and the number of clusters. In a wrapper approach, each solution serves as the input to a clustering algorithm, which, in the given feature subspace, partitions the data into the number of clusters specified. The resulting partitioning is evaluated and the resulting objective values are fed back to the optimizer. In a filter approach, the feature set is evaluated independently of a clustering algorithm and the resulting objective values are fed back to the optimizer. The main cycle is iterated for a pre-specified number of iterations. The final output of the algorithm is the set of Pareto optimal solutions, which is subjected to a final post-processing step that selects promising solutions from the Pareto front.

we describe a framework for multiobjective unsupervised feature selection, within which some of the limitations we have identified can be isolated and analyzed and which will also allow us to investigate other alternatives. The architecture of this framework is illustrated in Figure 5.

### 4.1 MOEA optimizer

Evolutionary algorithms are well-suited for multiobjective optimization as their use of a population enables the whole Pareto front to be approximated in a single algorithm run. Thus, in recent years there has been a growing effort in applying evolutionary computing methods in multiobjective optimization, giving rise to many different algorithms (see, e.g. [6, 10, 15] for reviews of the state of the art and [5] for an extensive list of references on this field). One example of a multiobjective evolutionary algorithm (MOEA), the Pareto envelope-based selection algorithm version 2 (PESA-II) [7], forms the basis of our feature selection method. The choice of this particular MOEA is motivated by our familiarity with the algorithm and is not believed to yield any particular advantage compared to other state-of-the-art MOEAs.

PESA-II follows the standard principles of an evolutionary algorithm with the difference that *two* populations of solutions are maintained: an internal population (*IP*) of fixed size and an external population (*EP*) of non-fixed but limited size. The purpose of the external population is to *exploit*

good solutions: to this end it implements a form of elitism by maintaining a large and diverse set of the nondominated solutions discovered during search. The internal population’s job, on the other hand, is to *explore* new solutions and achieves this by the standard EA processes of reproduction and variation (i.e., recombination and mutation). Selection occurs at the interface between the two populations, both in the update of the external population and in the construction of the internal population, which is done anew each generation. This means that difficult parameter tuning is avoided, and objective functions that have very different ranges can be readily used. PESA-II can also handle any number of objective functions. For further details on PESA-II, the reader is referred to [7].

For the internal population size of PESA-II we use a standard setting of 10. In our experiments, we then set the number of evaluations effected by the MOEA (including the initialization phase) equal to the number of evaluations taken by a greedy forward selection algorithm described in Section 5.1. Thus, PESA-II is run for a total of  $\frac{d_{max} \cdot 16 \cdot d - d}{10}$  generations, where  $d$  is the number of features available and  $d_{max}$  is the maximum feature set cardinality that the MOEA considers (see Section 4.1.2).

#### 4.1.1 Encoding and variation operators

The application of PESA-II to feature selection requires the choice of an appropriate encoding and operators. In a wrapper approach, there are two components of a solution that need to be coded: the actual feature subset, and the number of clusters. In a filter approach, only the feature subset is needed.

A simple binary encoding is used to select/deselect features: the genome comprises one bit for every feature, with a value of 1 indicating the activation of a feature and a value of 0 indicating its deactivation. The variation operators applied to this part of the genome are uniform crossover (with a standard crossover probability of 0.7) and bit-flip mutation (with a mutation probability of  $\frac{1}{d}$  where  $d$  is the total number of features available).

Four-bit Gray coding is used to encode the number of clusters, constrained to the range  $k \in \{2, \dots, 17\}$ . The variation operator applied to this part of the genome is bit-flip mutation (with a mutation probability of 0.25).

#### 4.1.2 Constraints

We have seen previously that the size of the full search space of the feature selection problem grows exponentially with the numbers of features. Yet, in most applications, researchers are predominantly interested in finding partitionings in feature subspaces that involve a relatively small number of variables only. In order to allow for an efficient search by the algorithm through these low-dimensional subspaces, a constraint on the maximum cardinality of the feature subspaces considered is imposed, which reduces the size of the search space to  $O(d^{d_{max}})$ . In all of the experiments presented in this paper, this constraint is set to  $d_{max} = \min(20, d)$ .

## 4.2 Objective functions: wrapper and filter methods

Our framework enables us to exchange the objective functions used to evaluate the candidate feature subsets. We have three alternative primary objective functions that are based on a wrapper method and one primary objective function based on a filter method. The secondary objective in each case is always the feature set cardinality  $d_F$  and is minimized or maximized depending on the bias of the primary objective. For the wrapper methods, the same clustering algorithm,  $k$ -means is always used; for the filter method no clustering algorithm is needed. Details of all the alternative objective functions follow, but first we detail the  $k$ -means algorithm.

**Clustering algorithm (for all wrapper methods):** In certain respects, the ideal choice for the clustering algorithm would be a powerful clustering method that is capable of detecting clusters of very different types (such as cluster of arbitrary shape, clusters with overlapping clusters or unequally sized clusters). Unfortunately, the clustering algorithm needs to be run for every single evaluation and the use of an algorithm with high computational complexity is therefore undesirable. In our experiments, we decide on the use of  $k$ -means, which seeks compact clusters, but whose time complexity is only linear in the number of data items.

The  $k$ -means algorithm starts from a random partitioning of the data into  $k$  clusters (where  $k$  is an input parameter). It repeatedly (i) computes the current cluster centres (that is, the average vector of each cluster in data space) and (ii) re-assigns each data item to the cluster whose centre is closest to it. It terminates when no more reassignments take place. By this means, the intra-cluster variance, that is, the sum of squares of the differences between data items and their associated cluster centres, is locally minimized.

Our implementation of the  $k$ -means algorithm is based on the batch version of  $k$ -means, that is, cluster centres are only recomputed after the reassignment of all data items. Random initialization (which is known to be an effective initialization method [33]) is used.

#### 4.2.1 Wrapper method: Silhouette Width

The clustering literature provides a number of internal clustering validation techniques that can be used to evaluate the quality of a candidate partitioning for a given feature space. The DB-Index, used by Morita *et al.* [31] has the advantage of linear time complexity (in the number of data items). However, a disadvantage of this index is its use of cluster centres for the computation of intra-cluster compactness and between-cluster separation. The use of cluster representatives can affect the accuracy of the resulting quality estimates, in particular if small, high-dimensional data sets are tackled. We therefore decide to investigate an alternative, popular validation technique from the literature: the Silhouette Width [37].

The Silhouette Width can potentially provide more accurate and smoother ‘search guidance’, as it does not use cluster representatives, but considers dissimilarities between in-

dividual data items. Its increased accuracy comes at a computational cost, however: the Silhouette Width has quadratic complexity in the number of data items.

The Silhouette Width for a partitioning is computed as the average Silhouette value over all data items. The Silhouette value for an individual data item  $i$ , which reflects the confidence in this particular cluster assignment, is computed as

$$S(i) = \frac{b_i - a_i}{\max(b_i, a_i)}, \quad (6)$$

where  $a_i$  denotes the average distance between  $i$  and all data items in the same cluster and  $b_i$  denotes the average distance between  $i$  and all data items in the closest other cluster (which is defined as the one yielding the minimal  $b_i$ ). The Silhouette Width returns values in the interval  $[-1, 1]$  and is to be maximized.

Similarly to the DB-Index, the Silhouette Width is largely unbiased with respect to the number of clusters and it is commonly used to select the best clustering out of a set of partitionings containing different numbers of clusters. Yet, like the DB-Index, the Silhouette Width is also biased towards low-dimensional subspaces, that is, it increases (i.e. improves) for a decreasing number of features. In order to counter-balance this bias, we chose feature cardinality as a second objective, which we maximize.

#### 4.2.2 Wrapper method: DB-Index

The DB-Index described in Section 3.2 is used as the primary objective (minimized). Because this index is biased toward smaller feature subsets, it is used with a secondary objective to maximize feature subset cardinality,  $d_F$ .

#### 4.2.3 Wrapper method: Normalized DB-Index

The primary objective here follows Morita *et al.* [31], that is, the normalized version of the DB-Index (see Equation 5) is minimized and simultaneously the number of features is also *minimized*.

#### 4.2.4 Filter method: Entropy

There are few objective functions in the literature that can be used to evaluate feature subsets of arbitrary cardinality, without resorting to the use of a clustering algorithm. The filter method we choose is an entropy measure [8], which considers the distribution of the dissimilarities between data items. For a set of  $N$  data items, this entropy measure is given as

$$E = - \sum_{i=1}^N \sum_{j=1}^N (s_{ij} \cdot \log s_{ij} + (1 - s_{ij}) \cdot \log(1 - s_{ij})) \quad (7)$$

where

$$s_{ij} = e^{-\alpha \cdot dist_{ij}} \quad (8)$$

and

$$\alpha = \frac{-\log 0.5}{\overline{dist}}. \quad (9)$$

Here  $dist_{ij}$  is the Euclidean distance between data items  $i$  and  $j$  for a given feature subspace and  $\overline{dist}$  is the mean dissimilarity between items in the data set for a given feature subspace.

Like the DB-Index and the Silhouette Width, this entropy measure is also biased towards small feature spaces. Consequently, the optimization task at hand requires the minimization of the entropy measure and the simultaneous maximization of the number of features.

Note that this method only returns feature sets, but does not indicate the corresponding number of clusters. In order to obtain clustering results for each solution within the final Pareto optimal set (for the purpose of our experimental analysis), the  $k$ -means algorithm is applied using the known correct number of clusters.

### 4.3 Initialization

A heuristic initialization scheme is implemented that aims to seed the optimization method with good initial feature sets. This initialization phase works as follows. First, all possible feature sets of cardinality 1 are constructed. All of these singleton feature sets are evaluated (for the wrapper approaches  $k = 2$  clusters are used) and sorted by the value of the primary objective in decreasing order. Initially,  $d_{max}$  solutions (recall that  $d_{max}$  is the constraint on the maximum cardinality of the feature space) are then generated as follows: For  $i = 1, \dots, d_{max}$ , the  $i$ th solution within this population is constructed by combining the  $i$  features with the highest individual scores under the primary objective (again, for the wrapper approaches  $k = 2$  clusters are used). The nondominated solutions are identified and are stored in the external population.

### 4.4 Solution selection

The final output of the optimizer is a two-dimensional Pareto front corresponding to feature subspaces of different cardinality that contain cluster structures of different qualities. The size of the Pareto optimal set is constrained by  $d_{max}$ , that is, at most one solution can be found for every feature cardinality investigated. In the post-processing phase we are interested in the possibility of automatically assessing the quality of individual solutions in the Pareto front and selecting the most promising solutions.

Cluster validation techniques are subject to several biases, which distort their results and hamper the comparison and interpretation of clustering results. A standard approach to this problem is the normalization of such measures by the expected performance on random ‘control data’ (i.e. data with no structure). While the expected performance can be computed exactly for some types of external measures (e.g. the Rand Index), it can only be estimated (by Monte-Carlo simulation) for most internal measures.

This type of normalization has been successfully used to abstract from  $k$ -specific biases in the choice of the best number of clusters [43] to use in a standard clustering scenario.

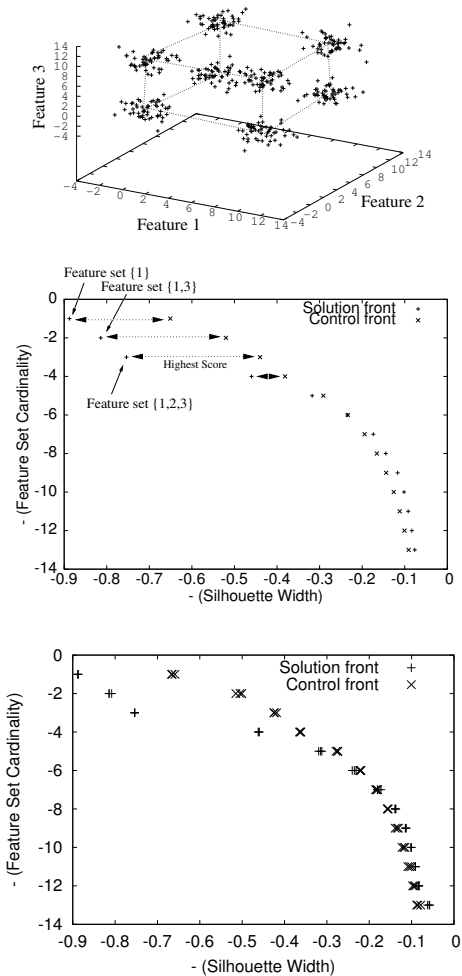


Figure 6: (Top) Plot of Square3d, a 13-feature data set, containing eight clusters arranged in a cube pattern in the first three dimensions, and Gaussian noise in the remaining 10 dimensions. (Centre) Solution front and control front obtained on this data. The distance between the solution and the control point obtained for a given feature cardinality can serve as an indicator of quality. In our method of solution selection, the solution point with the maximum distance from its control point is selected as the best solution. (Bottom) Comparison of the solution and control fronts generated in three independent runs. The variation between the control fronts is relatively small and the maximum distance between solution and control front is consistently located at  $d_F = 3$ . Note that for  $d_F \geq 7$ , the solution front distinctly lies behind the control front. This is due to differences in the distribution of the original and the control data: the noise features in the original data are normally distributed, whereas the features in the control data are uniformly distributed.

However, to the best of our knowledge, the use of random control data has not been used previously to abstract from feature-space cardinality biases, which is the approach we investigate.

The methodology used for solution selection is detailed in the following. First, the feature selection algorithm is applied to the data of interest in order to obtain a Pareto front, which we term the ‘solution front’. Second, the min-

imum and maximum bounds of the original data (in each feature) are determined and uniformly random data is generated within these bounds of the original data. This ‘control data’ is then subjected to the same procedure of feature selection as the original data. The resulting Pareto front is referred to as the ‘control front’. The solutions obtained in the solution and in the control front for a given feature cardinality are then directly comparable: we can therefore score solutions in the solution front by their distance to the corresponding solution in the control front. This score is then plotted as a function of the cardinality of the feature set and the maximum value (often corresponding to a ‘knee’ in the solution front) is selected as the best solution. This methodology is illustrated in Figure 6.

Note that, for reasons of computational expense, only a single control front is used in this paper. A statistically more rigorous approach would be the use of a larger number of control fronts and the use of the mean of these fronts to obtain a more accurate estimate of the expected performance. In Figure 6 we illustrate the variation in the control fronts.

## 5 Experimental comparison

In our experiments, we are interested in investigating a number of hypotheses related to multiobjective unsupervised feature selection.

1. We anticipate that feature selection based on the DB-Index may potentially outperform an equivalent method based on the normalized DB-Index. This is due to the fact that, using the normalized version of the DB-Index, the Pareto front is transformed and valid solutions may be overlooked (as discussed in Section 3).
2. Furthermore, feature selection based on the Silhouette Width may potentially outperform an equivalent method based on the DB-Index. This is because the Silhouette Width uses no cluster representatives and therefore provides a more accurate estimate of clustering quality. However, this potential increase in quality comes at a definitive increase in time complexity, which, for the Silhouette Width, is quadratic in the size of the data set.
3. Despite  $k$ -means’ failure to correctly retrieve elongated clusters, our method of feature selection may potentially work well on data with elongated cluster shapes. The reason for this is the fact that feature selection only requires the perception of structure (by means of the cluster validation technique), which in turn does not necessarily require the correct retrieval of all individual clusters. For example, elongated clusters may be subdivided into several spherical clusters by the  $k$ -means algorithm and the resulting clustering solution may still score highly under an appropriate validation technique. Such a property would have significant implications for the development of an efficient wrapper approach to feature selection: it would allow for a cheap clustering method

(such as the  $k$ -means algorithm) to be used during feature selection without major losses in terms of feature quality. A more powerful but expensive clustering method could then still be applied at a later stage (e.g. to the final feature set), in order to improve the final classification performance.

4. In supervised feature selection, the effectiveness of wrapper approaches has been repeatedly demonstrated. However, it is not intuitively clear whether their advantage over filter methods directly carries over to unsupervised feature selection. In our experiments, we would like to investigate whether a wrapper approach to unsupervised classification has advantages over comparable filter approaches.

### 5.1 Comparison

In order to investigate the above issues, we compare the different primary objective functions available in our framework (see Section 4.2). This gives four different multiobjective algorithms in all.

To provide a baseline comparison we also run three further feature selection methods, described below.

#### 1. Greedy search strategy

Forward selection is a popular sequential search algorithm in the feature selection literature. It is a greedy strategy which starts with the best feature set of size one and, in every iteration, adds the feature that will result in the highest value under the given objective. In our implementation the  $k$ -means algorithm is applied to the candidate feature set and the resulting clustering solution is evaluated by means of the Silhouette Width (comparably to one of our MOEAs). We repeat this evaluation step for a range of different numbers of clusters ( $k \in \{2, \dots, 17\}$ ) in order to determine the number of clusters best suited for a given feature cardinality (note that this is different to the MOEA, where  $k$  is encoded as part of the genotype). Forward selection is run to obtain feature sets of up to a size of  $d_{max} = \min(20, d)$ . Thus, the total number of evaluations effected by forward selection is  $d_{max} \cdot 16 \cdot d$ , where  $d$  is the number of available features. Forward selection is a very effective search strategy for data sets with a simple cluster structure. Comparison to this alternative search strategy therefore permits us to draw conclusions on the difficulty of the data sets considered and provides us with a realistic assessment of the need for the use of more advanced search methods such as an MOEA.

#### 2. Lower bound method

In order to demonstrate the necessity and the advantages of feature selection, results are compared to the results obtained when applying  $k$ -means to the full set of variables, that is, we consider the performance of  $k$ -means without any kind of feature selection. Ideally, this should give us a lower bound on the expected performance of any feature selection algorithm. In

this method, we provide  $k$ -means with the correct number of clusters.

#### 3. Upper bound method

Furthermore, we compare to the results obtained when applying  $k$ -means to the set of significant variables only, that is, we consider the performance of  $k$ -means after ‘perfect’ feature selection. Given that our method uses  $k$ -means as the embedded clustering algorithm (and assuming that all of the significant variables are indeed helpful for the clustering task), we would not expect our algorithm to exceed this performance. In this method,  $k$ -means is again provided with the known correct number of clusters.

### 5.2 Test suite

Three different types of data sets are used in our experiments and these are designed to study specific aspects of the algorithms’ performance. All of these data sets are comprised of two types of variables: significant variables, which are relevant to the clustering task at hand and insignificant variables, which correspond to Gaussian or uniformly random noise variables.

1. The first two data sets are synthetic data sets from the literature, which have been introduced by Kim *et al.* [22] (‘Kim’) and Morita *et al.* [31] (‘Morita’). These data sets can be considered as relatively simple data sets, as all of the significant variables reveal clear cluster structure even when considered in isolation.
2. The second group of data consists of three data sets (‘Smile’, ‘Spiral’ and ‘Long’), which contain elongated cluster shapes that cannot be correctly detected by the  $k$ -means algorithm. Using these data sets, we aim to investigate whether a wrapper-based method of feature selection can indeed perform well, even if the base clusters cannot be detected by the clustering algorithm (here,  $k$ -means). These data sets have 400 data points and consist of two significant features and 100 additional Gaussian noise variables. Plots of the first two dimensions containing the actual cluster structure are shown in Figure 7.

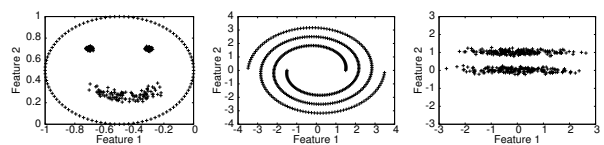


Figure 7: The second group of experimental data contains data sets with highly elongated cluster shapes. From left to right: (a) The ‘Smile’ data set. (b) The ‘Spiral’ data set. (c) The ‘Long’ data set.

3. The third group of data is designed to capture some of the features of data typically encountered in real-world applications. This data is high-dimensional, contains many more dimensions than data points and

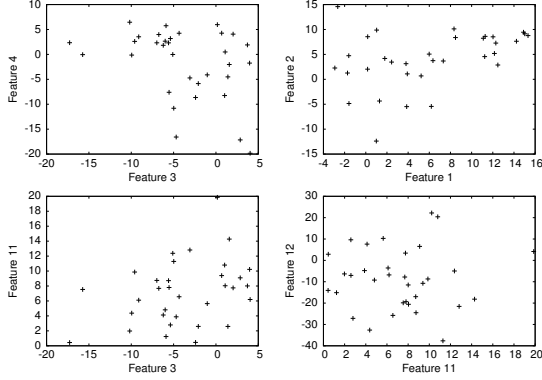


Figure 8: Two-dimensional projections of a 100-dimensional data set with ten significant features and containing two clusters. The data set contains very few data items and very little data structure can be discerned in two-dimensional projections of the data. (Top) Significant features. (Bottom left) Mixture of insignificant and significant features. (Bottom right) Insignificant features.

only few of the features are actually relevant to the classification task at hand. We obtain this data using a data generator for multivariate Gaussian clusters whose data sets have been shown to be hard to solve for a variety of different algorithms (including  $k$ -means) [18]. The generator is applied to produce a number of small data sets with  $k \in \{2, 4, 10\}$  and  $d_1 \in \{2, 10\}$  (individual cluster sizes are uniformly distributed within the interval  $\{10, \dots, 50\}$ ). We produce ten instances of each type. In our experiments, all ten data sets that are of dimensionality  $d_1$  and contain  $k$  clusters are then grouped and referred to as the group of data  $d_1 d$ - $k$ - $c$ . Finally, a number of Gaussian noise variables  $d_2 \in \{100, 1000\}$  are added to all types of data, resulting in a total dimensionality  $d = d_1 + d_2$ . Hence, in total we obtain 12 different groups of data sets, which consist of 10 individual instances each. An example of such a data set is given in Figure 8.

### 5.3 Data pre-processing

Previous work by Dy and Brodley [13] has shown that the standardization of all features is crucial in unsupervised feature selection. Our own preliminary experiments confirmed this, and we use variables normalized to a mean of zero and a standard deviation of one for all of the feature selection algorithms discussed in this paper.

### 5.4 Performance assessment

Results are evaluated using two different aspects: the quality of the partitioning discovered and the quality of the feature subset retrieved. Our evaluation is external, that is, we make use of the known correct clustering solution and the known significant features in order to quantify the performance of the different algorithms. The advantage of this type of evaluation is that it can be considered objective and is unbiased with respect to the different algorithms.

#### 5.4.1 Adjusted Rand Index

Clustering quality is objectively evaluated using the *Adjusted Rand Index*, an external measure of clustering quality that is a generalization of the Rand Index. The Rand indices are based on counting the number of pair-wise co-assignments of data items. The Adjusted Rand Index additionally introduces a statistically induced normalization in order to yield values close to 0 for random partitions. This normalization removes the bias of the Rand Index with respect to different numbers of clusters, which is of particular importance in our application, as results across a range of cluster numbers are compared. Using a representation based on contingency tables, the Adjusted Rand Index [20] is given as

$$R(U, V) = \frac{\sum_{lk} (n_{lk} \binom{n_{lk}}{2}) - [\sum_l \binom{n_{l.}}{2}] \cdot \sum_k \binom{n_{.k}}{2}}{\frac{1}{2} [\sum_l \binom{n_{l.}}{2} + \sum_k \binom{n_{.k}}{2}] - [\sum_l \binom{n_{l.}}{2}] \cdot \sum_k \binom{n_{.k}}{2}} / \binom{n}{2}, \quad (10)$$

where  $n_{lk}$  denotes the number of data items that have been assigned to both cluster  $l$  and cluster  $k$ .

The Adjusted Rand Index returns values in the interval  $[\sim 0, 1]$  and is to be maximized.

#### 5.4.2 F-Measure

The fraction of significant features identified by a given method can be referred to as its *Sensitivity*. Sensitivity is computed as

$$Sensitivity = \frac{\#(\text{significant features identified})}{\#(\text{significant features})}. \quad (11)$$

The quality of a feature subset identified by a given method can also be referred to as its *Specificity*. Specificity is computed as

$$Specificity = \frac{\#(\text{significant features identified})}{\#(\text{of features identified})}. \quad (12)$$

Both Sensitivity and Specificity return values in the interval  $[0, 1]$  and are to be maximized.

Specificity and Sensitivity describe certain qualitative properties of the feature sets, but, individually, they do not provide comprehensive information on the quality of a feature set. This is because both Sensitivity and Specificity obtain maximal values for trivial solutions, e.g. Sensitivity is 1, if all features have been selected and Specificity is 1, if a single feature has been selected that happens to be significant. In order to recognize genuinely good feature set we need to simultaneously consider both Specificity and Sensitivity, and we do so using the *F-Measure* [36].

The F-Measure has first been introduced in information retrieval and tries to capture the trade-off between precision and recall. The concepts of precision and recall directly correspond to those of Specificity and Sensitivity and the use of the F-Measure for the assessment of feature set quality is therefore straightforward:

$$F\text{-Measure} = \frac{2 \cdot \text{Specificity} \cdot \text{Sensitivity}}{\text{Specificity} + \text{Sensitivity}}.$$

The F-Measure returns values in the interval  $[0, 1]$  and is to be maximized.

### 5.4.3 Selection of solutions from the Pareto front

All of the five feature selection methods in our study return a range of solutions corresponding to different feature cardinalities. Here, we do not analyze the entire distribution of solutions, but are predominantly interested in the quality of the best solution identified by each method. In order to analyze this ‘peak’ performance, we therefore need to devise a way to select the best solution from the Pareto front and we use the F-Measure for this purpose. Using the F-Measure, most of the results in the following section are obtained in the following way: for a given Pareto front, the feature set with the best F-Measure is selected. This solution is then analyzed in terms of its feature cardinality, its F-Measure and the number of clusters and Adjusted Rand Index of the corresponding clustering solution. Note that this way of selecting the best solution from the Pareto front is external, that is, we use our knowledge of the significant features to select the best solution from the Pareto front. This allows us to do an objective comparison of the quality of the best solutions generated by the different algorithms.

Evidently, external knowledge of this type is not usually available in a data-mining scenario. When applying a feature selection method to real data, we therefore need to find some other ways to select good solutions from the Pareto front. The post-processing method introduced in Section 4.4 is a possible way to do this selection. In an additional experiment, we will assess how well this ‘internal’ way of solution selection performs in comparison to the ‘external’ way based on the F-Measure.

## 5.5 Results

### 5.5.1 Results obtained on synthetic data sets from the literature

Tables 1 and 2 summarize the experimental results on the two synthetic data sets taken from the literature [22, 31]. Here, we compare the solutions corresponding to the highest F-Measure value. On both the ‘Kim’ and the ‘Morita’ data set, a very similar performance of all the MOEA-based algorithms can be observed. Evidently, these two data sets are relatively easy to analyze and more complex data sets are needed in order to identify differences and trends in the performance of the different validation techniques. However, the structure of the ‘Kim’ data is sufficiently complex to highlight one of the fundamental limitations of the use of greedy search strategies in feature selection: in forward selection, once a feature has been included during the search process (for a given feature cardinality), it remains included for all subsequently explored feature sets of larger cardinality. On data sets containing a variety of different cluster structures (in different, mutually non-inclusive feature

Method	$d_F$	$k$	Rand	F-Measure
Silhouette	5	3	1.0	1.0
DB-Index	5	3	1.0	1.0
DB-Index/ $d_F$	5	3	1.0	1.0
Forward selection	5	3	1.0	1.0
Filter approach	5	3	1.0	1.0
Lower bound	10	3	1.0	0.6667
Upper bound	5	3	1.0	1.0

Table 1: Results on the ‘Morita’ data set, as described by the cardinality and F-Measure of the obtained feature set and the number of clusters  $k$  and Adjusted Rand Index (Rand) of the corresponding clustering solution. All values shown are averages over 21 runs. For the five feature selection methods, the Pareto optimal solution scoring highest under the F-Measure is shown. For all five algorithms, this solution consistently corresponds to the correct feature set of cardinality  $d_F = 5$  and the correct clustering. Notably, the correct clustering is also obtained by the lower bound method.

Method	$d_F$	$k$	Rand	F-Measure
Silhouette	10	5	1.0	1.0
DB-Index	10	5	1.0	1.0
DB-Index/ $d_F$	10	5	1.0	1.0
Forward selection	12	5	1.0	0.9091
Filter approach	10	5	1.0	1.0
Lower bound	30	5	1.0	0.2906
Upper bound	10	5	1.0	1.0

Table 2: Results on the ‘Kim’ data set, as described by the cardinality and F-Measure of the obtained feature set and the number of clusters  $k$  and Adjusted Rand Index (Rand) of the corresponding clustering solution. All values shown are averages over 21 runs. For the five feature selection methods, the Pareto optimal solution scoring highest under the F-Measure is shown. For all four MOEA-based algorithms, this solution consistently corresponds to the correct feature set of cardinality  $d_F = 10$  and the correct clustering. The best solution for forward selection has a cardinality of  $d_F = 12$ : it includes two insignificant features, but nevertheless corresponds to the correct partitioning. Notably, the correct clustering is also obtained by the lower bound method.

spaces), this results in a sub-optimal performance. The set of solutions identified by forward selection on the ‘Kim’ data set is shown in Figure 9 and compared to those identified by an MOEA.

### 5.5.2 Results obtained on data sets with elongated cluster structures

Tables 3 to 5 show the results obtained on the second group of experimental data. The results returned by the upper bound method confirm that on these data sets with elongated cluster structures, the  $k$ -means algorithm does indeed fail to capture all individual clusters (c.f. the low values of the Adjusted Rand Index). However, the results also indicate that, despite this apparent failure, the combination of the  $k$ -means algorithm with an appropriate validation technique still works efficiently at recognizing the presence of structure in the data and at identifying the most significant features in the data.

The majority of algorithms identify the correct feature

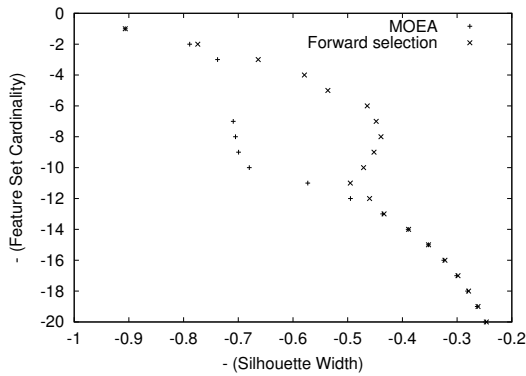


Figure 9: Solution sets returned by an MOEA and forward selection on the ‘Kim’ data set, both using Silhouette Width as their validation technique.

Method	$d_F$	$k$	Rand	F-Measure
Silhouette	2	8	0.669677	1.0
DB-Index	2	3	0.597809	1.0
DB-Index/ $d_F$	1	2	0.329469	0.6667
Forward selection	2	3	0.5978	1.0
Filter approach	2	4	0.536464	1.0
Lower bound	100	4	-0.0014	0.0039
Upper bound	2	4	0.5445	1.0

Table 3: Results on the ‘Smile’ data set, as described by the cardinality and F-Measure of the obtained feature set and the number of clusters  $k$  and Adjusted Rand Index (Rand) of the corresponding clustering solution. All values shown are averages over 21 runs. For the five feature selection methods, the Pareto optimal solution scoring highest under the F-Measure is shown. None of the algorithms identifies the correct clustering. Nevertheless, four out of the five algorithms consistently succeed in identifying the two significant features. The MOEA using the normalized version of the DB-Index fails to identify any solutions with  $d_F > 1$ .

Method	$d_F$	$k$	Rand	F-Measure
Silhouette	2	4	0.0513	1.0
DB-Index	2	4.3333	0.0437	1.0
DB-Index/ $d_F$	2	4.1905	0.0471	1.0
Forward selection	2	4.7143	0.0353	1.0
Filter approach	1	2	-0.0020471	0
Lower bound	100	2	4.8046e-05	0.0039
Upper bound	2	2	0.0417	1.0

Table 4: Results on the ‘Spiral’ data set, as described by the cardinality and F-Measure of the obtained feature set and the number of clusters  $k$  and Adjusted Rand Index (Rand) of the corresponding clustering solution. All values shown are averages over 21 runs. For the five feature selection methods, the Pareto optimal solution scoring highest under the F-Measure is shown. None of the algorithms identifies the correct clustering. Nevertheless, four out of the five algorithms consistently succeed in identifying the two significant features. The entropy-based MOEA fails to identify any of the significant features.

sets on the ‘Spiral’ and the ‘Smile’ data sets. The only exceptions are the MOEA based on the normalized DB-Index, which identifies only the first feature in the ‘Smile’ data set (all other solutions are dominated), and the entropy-based MOEA, which fails to identify any of the significant features in the ‘Spiral’ data. On the Long data set, all algorithms fail to identify the second of the two significant features. This is caused by the fact that the second dimension in this data set contains no more structure than a random dimension (c.f. Figure 7). Notably, all algorithms in the study succeed in correctly retrieving the first significant dimension and the corresponding two-cluster solution.

### 5.5.3 Results obtained on complex, sparse data

Figure 10 summarizes the experimental result on the third category of test data. On this sparse data, differences between the different validation techniques become more pronounced. The main trends observable are outlined in the following.

#### Differences between the original and the normalized version of the DB-Index

Clear differences can be observed between the performance of the two MOEAs based on the original and the normalized version of the DB-Index. On the majority of data sets, the MOEA using the original DB-Index seems to perform better in terms of the F-Measure of the feature set identified.

#### Performance of the Silhouette Width

The Silhouette Width outperforms both versions of the DB-Index on these sparse data sets. Across the range of data sets, a substantial performance gain in terms of both the F-Measure and the Adjusted Rand Index can be observed. On several of the data sets including 100 noise variables, the performance of the MOEA based on the Silhouette Width is close to that of the upper-bound method. Notably, on the 10d-2c data sets, the values obtained under the Adjusted Rand Index are indeed better than those returned by the upper bound method. This can be explained by the removal of certain ‘significant’ features that do not amplify the cluster structure, but render the clustering task more difficult to solve for the  $k$ -means algorithm. An illustrative example of such a feature is the second (horizontal) feature in the Long data set, which introduces an elongated cluster structure: consequently,  $k$ -means performs much better in this clustering tasks if only applied to the first (vertical) feature.

#### Performance on data with elongated cluster structures

In accordance with the results on the second group of experimental data, the results on this third group also confirm that a  $k$ -means based wrapper method can indeed perform well even if the underlying clusters cannot be identified by the  $k$ -means algorithm. This is illustrated, for example, by the results obtained on the 2d-10c data sets. On this data, most of the feature selection methods considered obtain close to optimal F-Measure values despite scoring relatively poorly under the Adjusted Rand Index.

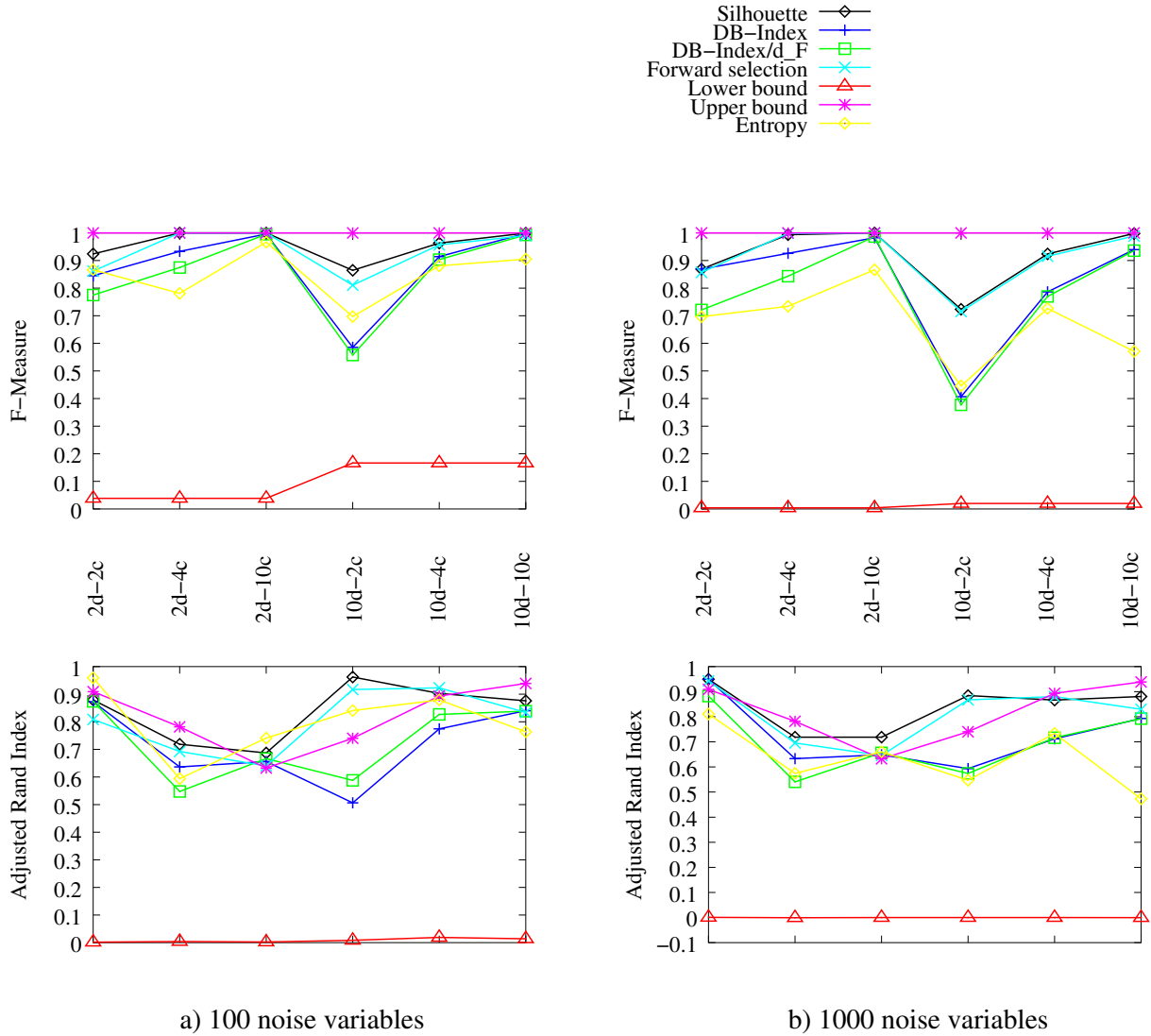


Figure 10: Comparison of the best solution generated by the five different methods of feature selection on the third group of experimental data, as well as the upper and lower bound solutions. All values shown are averages over 21 runs. The Silhouette Width emerges as the strongest performer out of the five contestants.

Method	$d_F$	$k$	Rand	F-Measure
Silhouette	1	2	1	0.6667
DB-Index	1	2	q	0.6667
DB-Index/ $d_F$	1	2	1	0.6667
Forward selection	1	2	1	0.6667
Filter approach	1	2	1	0.6667
Lower bound	100	2	-0.00047	0.00392
Upper bound	2	2	-0.0025	1.0

Table 5: Results on the ‘Long’ data set, as described by the cardinality and F-Measure of the obtained feature set and the number of clusters  $k$  and Adjusted Rand Index (Rand) of the corresponding clustering solution. All values shown are averages over 21 runs. For the five feature selection methods, the Pareto optimal solution scoring highest under the F-Measure is shown. All five algorithms consistently identify the correct clustering and the first significant feature. They fail to identify the second significant feature, which, however, contains no discernible structure.

### Scalability to large embedding subspaces

Regarding the algorithms’ scalability to the dimensionality of the subspace that contains the actual cluster structure, this scalability depends on the type of data tackled. If individual features reflect clear signs of structure (and can thus be identified in the initialization phase, as on the ‘Kim’ data set), the algorithm will easily scale to larger subspaces. Yet, the third group of data in our experiments is of a different kind: in this data, clusters have been constructed as multivariate Gaussians with strong covariance between variables. Consequently, the one-dimensional projections of this data do not usually reveal much structure and clusters only emerge as a result of the correlations between several variables. Identifying the significant variables in this type of data is much harder and increases in difficulty with increasing dimensionality of the multivariate Gaussians (approaching ‘needle-in-a-haystack’ search scenarios for high dimensionalities). In our experiments, this increase in difficulty can be appreciated when comparing the results obtained for

the two and the ten-dimensional Gaussians.

### Scalability to large numbers of noise variables

Regarding the algorithm’s scalability to the number of noise variables, a comparison of the results for 100 and 1000 noise variables confirms that it is generally more difficult to retrieve the relevant features with a greater number of noise variables and that the performance of all methods suffers slightly. The effect is more pronounced for those data sets with cluster structures embedded in ten-dimensional subspaces, which present the harder search tasks.

### Effect of the number of clusters

The experimental results reveal two opposite trends with respect to the number of clusters in the data sets: the F-Measure values obtained tend to increase with an increasing number of clusters, while the Adjusted Rand Index values tend to decrease. This effect is specific to the test data used in our experiments and can be explained as follows: due to its random generation, this data may contain dimensions that are marked as significant, but do not actually contribute significantly to the cluster structure (and may even be inhibitive by giving rise to elongated cluster shapes). Therefore, a feature selection method may discard these features, but still succeed in identifying a very good (and even improved) clustering. Evidently, this is more likely to happen for data sets with low numbers of clusters. In addition, a larger number of clusters may generally facilitate the task of identifying the significant dimensions, as it increases the likelihood of some degree of structure being discernible along a given single dimension.

### Time complexity

With the restriction on the feature set cardinality that we use, all methods scale linearly in the number of features. Differences in the runtime of the algorithms arise due to differences in the complexity of the validation techniques used. While the the DB-Index is linear in the number of data items, the Silhouette Width and the entropy measure are quadratic and this results in a significant increase in runtime for large data sets. Representative runtimes illustrating this trend are shown in Figure 11.

### Performance of our method of solution selection

The performance of our method of solution selection shows some variation dependent on the primary optimization criterion used within the MOEA. In particular, the method is most accurate for the MOEA based on the Silhouette Width, as the knees in the Pareto fronts resulting from this method are the most pronounced. In contrast, the Pareto fronts resulting from the optimization of the normalized DB-Index contain no significant knee structures and the identification of the best solution becomes more involved.

Figure 12 indicates the performance of the method when applied to the output of the MOEA using the Silhouette Width. In order to assess the relative loss of solution quality, the solution identified by the post-processing phase is compared to the best and the worst solution in the Pareto front

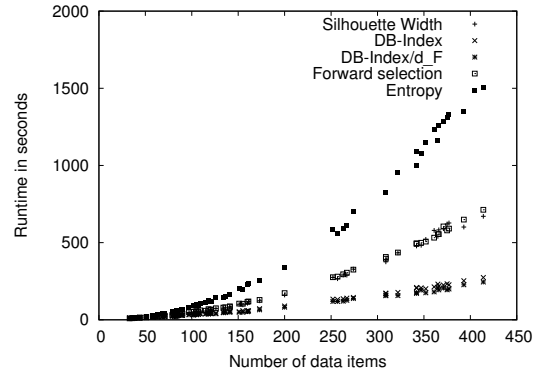


Figure 11: Representative runtimes of the four MOEAs and forward selection as a function of the number of data items (for 100 noise variables).

(as identified by the F-Measure).

Results under Sensitivity indicate a certain tendency of the approach to underestimate the number of significant features in a data set, but, overall, the results obtained under the F-Measure and the Adjusted Rand Index show a satisfactory performance.

## 6 Discussion

### 6.1 Observed performance differences

In our experimental study we have put a particular emphasis on the performance of the different feature selection methods when applied to data in which the number of features is much higher than the number of data points. The structure of such data cannot easily be perceived within individual variables or the original data space and it is in such a scenario that we would expect the choice of evaluation technique to be of particular importance. Our experimental results confirm this and indicate clear performance differences between the techniques evaluated. In particular, the two wrapper approaches based on the Silhouette Width (forward selection and the MOEA) clearly emerge as the two strongest performers.

To some degree, the performance of forward selection on this high-dimensional and relatively complex data is surprising: forward selection performs close to optimally, and the design of an efficient MOEA with comparable accuracy and scalability becomes very intricate. However, while this outstanding performance of a greedy technique may raise questions regarding the benefits of the use of global optimization methods for unsupervised feature selection, our experimental results on the ‘Kim’ data set also alert us to the general pitfalls of greedy approaches: on data containing noise or multiple cluster structures, greedy approaches to feature selection can perform poorly and may be significantly outperformed by global optimization techniques. We expect that the incorporation of more ‘structured’ noise features into our high-dimensional data sets would also cause the greedy approach more difficulty; we also believe that such structured noise is typical of real data, which may exhibit multiple structured but disjoint subspaces. Such pat-

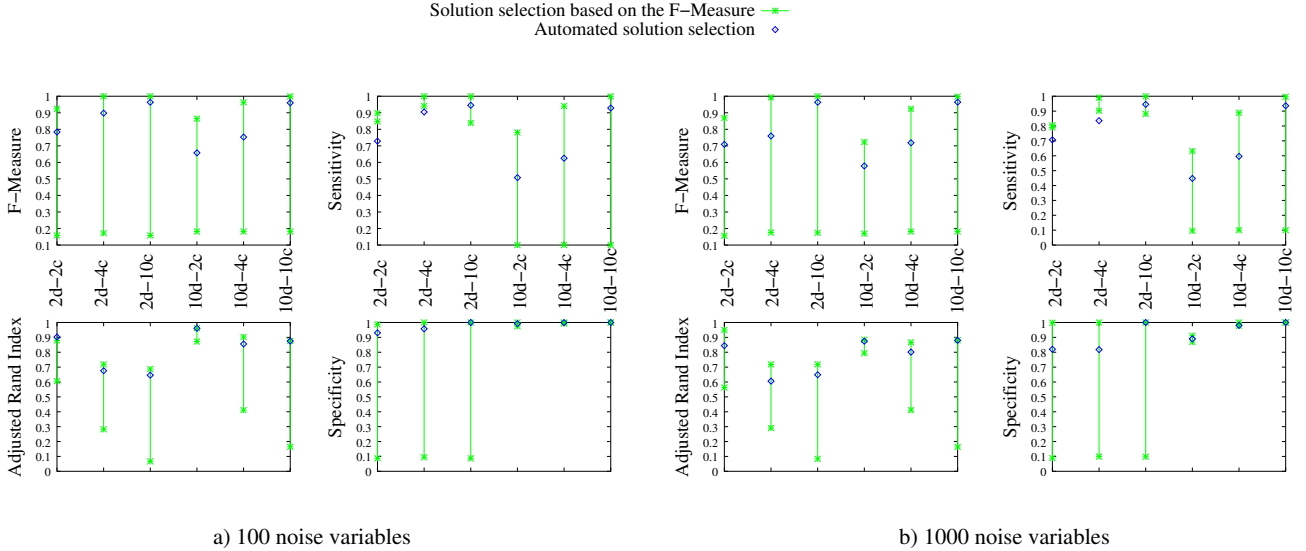


Figure 12: Analysis of the performance of our method of solution selection for the Silhouette Width-based MOEA on the third group of experimental data. We compare the solution selected by our post-processing method to the worst and the best solution in the Pareto front (as determined by the F-Measure). Shown are Sensitivity, Specificity and F-Measure of the selected feature set and the Adjusted Rand Index of the corresponding clustering solution. All values shown are averages over 21 runs. Note that the solution scoring best/worst under the F-Measure does not necessarily correspond to the best/worst solution under the individual measures.

terms may arise due to the impact of different environmental or instrumental noise sources, but may also impart information regarding different aspects of a classification problem. The identification and juxtaposition of disjoint sets of features within the Pareto front (corresponding to these patterns) may therefore be an instructive source of information for a decision maker.

The MOEA using Silhouette Width outperforms forward selection on data with 100 noise variables, however, the implementation is currently slightly less scalable towards high dimensionalities. In future work, we hope to overcome this limitation through a more efficient use of the large number of available evaluations, for example, by introducing random restarts and more efficient initialization schemes. In this paper, we have chosen to run the MOEA for the same number of evaluations as taken by forward selection, which facilitates an objective comparison between the methods, but may not be the most effective approach.

On this line, it is also worth noting that stopping at an earlier stage is evidently possible for the MOEA-based approaches, but not so for forward selection. In the presence of time constraints, the use of an MOEA may therefore turn out to be the only feasible option. Regarding the time complexity of the algorithms, it is also interesting to consider the differences observed between the filter and the wrapper approaches in our comparison. For very large feature sets, filter approaches are commonly thought to be a computationally more realistic option. However, this is not necessarily accurate when referring to filter approaches that are not restricted to feature ranking, but search for good feature subsets: while the filter approach in our study avoids the use of the  $k$ -means algorithm, its runtime performance remains comparable to that of the wrapper approaches.

## 6.2 Differences to traditional Pareto optimization

Traditionally, Pareto optimization is employed in the optimization of a set of decision variables with respect to two (or more) incommensurable objectives. In such a scenario we usually have a clear idea whether to maximize or minimize each individual objective.

In some ways, the application discussed in this paper is quite different to this ordinary setting of Pareto optimization. In particular, we have a single primary objective, the score returned by either an internal cluster validation technique or filter method, which we would like to optimize. The necessity of using a second objective only arises as a result of the bias of this validation technique and the fact that the exact form of the bias cannot be specified. However we do know that for subspaces containing clusters of equal quality, the score returned by the primary objective function deteriorates monotonically for either increasing or decreasing dimensionality of the subspace. Assuming a primary objective function with a bias towards low dimensions, the following order relations serve to define the problem more formally:

- We are given two feature sets of the same cardinality. In this case there is no dimensionality-bias, so the judgement returned by the primary objective function can be seen as objective. The feature set that scores better under the primary objective function should be preferred.
- We are given two feature sets of different cardinality, the smaller one of which scores better under the primary objective function. In this case, we cannot judge whether the larger feature set scores worse due to the dimensionality-bias or due to a weaker cluster structure. The two feature sets should therefore be treated

as incomparable.

- We are given two feature sets of different cardinality, the larger one of which scores better than or equal to the smaller one under the primary objective function. In this case we know that the high score of the large feature set must be due to a stronger cluster structure. The larger feature set should therefore be preferred.

The above description is correctly captured by means of a standard Pareto optimization formulation when optimizing the cluster validation score and maximizing the number of features.

### 6.3 Advantages of the use of Pareto optimization

Given the discrete nature of the second objective (the number of features), the use of Pareto optimization is not the only possible choice. An alternative option would be the single-objective optimization of the cluster primary objective function for a range of different feature set cardinalities. However, several of the results presented in this paper show that some of the feature sets identified for different feature cardinalities are closely related. Given this structure in the decision space, the identification of all solutions in a single run should be more efficient than using individual runs of a single-objective optimization method.

## 7 Conclusion

This paper has investigated the formulation and implementation of unsupervised feature selection as a multiobjective optimization problem. We have discussed the reasoning behind a multiobjective approach to the problem and have analyzed the strengths and limitations of two seminal algorithms from the literature. Our analysis has led to the specification of a number of alternative approaches, which have then been compared across a range of benchmark data sets. Different to previous work on unsupervised feature selection, a particular focus in our evaluation has been on data sets in which the number of features largely exceeds the number of data items. In this context, a wrapper approach based on the Silhouette Width emerges as the strongest overall performer.

### Acknowledgments

We would like to thank Yong Seog Kim, Nick Street and Luiz Oliveira for providing their synthetic test data and answering questions related to their work. JH gratefully acknowledges support of a doctoral scholarship from the German Academic Exchange Service and the Gottlieb Daimler and Karl Benz-Foundation, Germany. JK is supported by a David Phillips Fellowship from the Biotechnology and Biological Sciences Research Council (BBSRC), UK.

## Bibliography

- [1] D. W. Aha and R. L. Bankert. A comparative evaluation of sequential feature selection algorithms. In *Learning from Data*, pages 199–206. Springer-Verlag, New York, NY, 1996.
- [2] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, NJ, 1961.
- [3] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.
- [4] X. Chen. An improved branch and bound algorithm for feature selection. *Pattern Recognition Letters*, 24(12):1925–1933, 2003.
- [5] C. A. Coello Coello. List of references on evolutionary multiobjective optimization. <http://delta.cs.cinvestav.mx/~ccoello/EMOO/EMOObib.html>.
- [6] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, NY, 2002.
- [7] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 283–290. Morgan Kaufmann Publishers, San Francisco, CA, 2001.
- [8] M. Dash and H. Liu. Handling large unsupervised data via dimensionality reduction. In *Proceedings of the ACM SIGMOD Workshop on Research Numbers in Data Mining and Knowledge Discovery*. <http://www.almaden.ibm.com/cs/dmkd/>, 1999.
- [9] J. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:224–227, 1979.
- [10] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001.
- [11] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858. Springer-Verlag, Berlin, Germany, 2000.
- [12] K. Deb and A. R. Reddy. Classification of two-class cancer data reliably using evolutionary algorithms. Technical Report KanGALReport No. 2003001, Kanpur Genetic Algorithm Laboratory, Kanpur, India, 2003.
- [13] J. G. Dy and C. E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5(5):845–889, 2004.
- [14] C. Emmanouilidis, A. Hunter, and J. MacIntyre. A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In *Proceedings of the 2000 Congress on Evolutionary Computation*, pages 309–316. IEEE Press, New York, NY, 2000.
- [15] X. Gandibleux, M. Sevaux, K. Sörensen, and V. T’Kindt, editors. *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, Germany, 2004.
- [16] D. Guo, M. Gahegan, D. Peuquet, and A. MacEachren. Breaking down dimensionality: An effective feature selection method for high-dimensional clustering. In *Proceedings of the Third SIAM International Conference on Data Mining*, pages 29–42. SIAM Press, San Francisco, CA, 2003.
- [17] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(3):1157–1182, 2002.

- [18] J. Handl and J. Knowles. Improvements to the scalability of multiobjective clustering. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, pages 2372–2379. IEEE Press, Anaheim, CA, 2005.
- [19] J. Handl, J. Knowles, and D. B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 2005.
- [20] A. Hubert. Comparing partitions. *Journal of Classification*, 2:193–198, 1985.
- [21] E. Hughes. Evolutionary many-objective optimisation: Many once or one many? In *Proceedings of the 2005 Congress on Evolutionary Computation*, pages 460–465. IEEE Press, Anaheim, CA, 2005.
- [22] Y. Kim, W. N. Street, and F. Menczer. Evolutionary model selection in unsupervised learning. *Intelligent Data Analysis*, 6(6):531–556, 2002.
- [23] R. Kohavi and G. John. Wrapper for feature subset selection. *Artificial Intelligence*, 97(1–2):273–324, 1997.
- [24] H. Liu and L. Yu. Towards integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):1–12, 2005.
- [25] J. Loughrey and P. Cunningham. Using early-stopping to avoid overtraining in wrapper-based feature selection employing stochastic search. Technical Report TCD-CD-2005-37, Department of Computer Science, Trinity College Dublin, UK, 2005.
- [26] L. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, Berkeley, CA, 1967.
- [27] B. F. Manly. *Multivariate statistical methods: a primer*. Chapman & Hall, London, UK, 1986.
- [28] F. Menczer, M. Degeratu, and W. N. Street. Efficient and scalable pareto optimization by evolutionary local selection algorithms. *Evolutionary Computation*, 8(2):223–247, 2000.
- [29] G. W. Milligan. *Clustering validation: results and implications for applied analyses*, chapter 10, pages 341–367. World Scientific, New Jersey, NJ, 1996.
- [30] P. Mitra, C. A. Murthy, and S. K. Pal. Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):301–312, 2002.
- [31] M. Morita, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pages 666–671. IEEE Press, New York, NY, 2003.
- [32] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. A methodology for feature selection using multiobjective genetic algorithms for handwritten digit string recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(6):903–929, 2003.
- [33] J. M. Pena, J. A. Lozana, and P. Larranaga. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Letters*, 20(10):1027–1040, 1999.
- [34] J. M. Pena, J. A. Lozano, P. Larranaga, and I. Inza. Dimensionality reduction in unsupervised learning of conditional Gaussian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):590–603, 2001.
- [35] P. Pudil, J. Novovicov’a, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994.
- [36] C. V. Rijsbergen. *Information Retrieval, 2nd edition*. Butterworths, London, UK, 1979.
- [37] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, 1987.
- [38] J. Schmidhuber. Discovering neural nets with low Kolmogorov complexity and high generalization capability. *Neural Networks*, 10(5):857–873, 1997.
- [39] W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5):335–347, 1989.
- [40] R. Solomonoff. The Kolmogorov lecture: The universal distribution and machine learning. *Computer Journal*, 46(6):598–601, 2003.
- [41] N. Sondberg-Madsen, C. Thomsen, and J. M. Pena. Unsupervised feature subset selection. In *Proceedings of the Workshop on Probabilistic Graphical Models for Classification*, pages 71–82. <http://www.sc.ehu.es/ccwbayes/ecml-pkdd-03-workshop/call.htm>, 2003.
- [42] L. Talavera. Feature selection as a preprocessing step for hierarchical clustering. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 389–39. Morgan Kaufmann, San Francisco, CA, 1999.
- [43] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the Gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.
- [44] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY, 1995.