Intelligent Data Analysis 6 (2002) 531–556
IOS Press

Evolutionary model selection in unsupervised learning

YongSeog Kim*, W. Nick Street and Filippo Menczer
Department of Business Information Systems, Utah State University, Logan, UT 84322-3515, USA

Received 8 March 2002
Revised 28 April 2002
Accepted 5 May 2002

Abstract. Feature subset selection is important not only for the insight gained from determining relevant modeling variables but also for the improved understandability, scalability, and possibly, accuracy of the resulting models. Feature selection has traditionally been studied in supervised learning situations, with some estimate of accuracy used to evaluate candidate subsets. However, we often cannot apply supervised learning for lack of a training signal. For these cases, we propose a new feature selection approach based on clustering. A number of heuristic criteria can be used to estimate the quality of clusters built from a given feature subset. Rather than combining such criteria, we use ELSA, an evolutionary local selection algorithm that maintains a diverse population of solutions that approximate the Pareto front in a multi-dimensional objective space. Each evolved solution represents a feature subset and a number of clusters; two representative clustering algorithms, K-means and EM, are applied to form the given number of clusters based on the selected features. Experimental results on both real and synthetic data show that the method can consistently find approximate Pareto-optimal solutions through which we can identify the significant features and an appropriate number of clusters. This results in models with better and clearer semantic relevance.

1. Introduction

Feature selection is the process of choosing a subset of the original predictive variables by eliminating redundant and uninformative ones. By extracting as much information as possible from a given data set while using the smallest number of features, we can save significant computing time and often build models that generalize better to unseen points. Further, it is often the case that finding a predictive subset of input variables is an important problem in its own right.

Feature selection has primarily been studied in a supervised learning context, where predictive accuracy is commonly used to evaluate feature subsets. Feature selection can be done through two different models, the wrapper model and the filter model [37]. The conceptual difference between these two models is based on the underlying mechanism for feature selection and evaluation. The wrapper model uses the same learning algorithm for the purpose of selecting and evaluating feature subsets. However, the filter model selects feature subsets using intrinsic properties of the data and uses an independent induction algorithm to evaluate subset quality. Both models require a search algorithm that explores the combinatorial space of feature subsets. Let D represent the original feature dimension of a given data set. The whole search space is $O(2^D)$, making exhaustive search impractical for data sets with even moderate dimensionality.

*Corresponding author: YongSeog Kim, Tel.: +1 435 797 2271; Fax: +1 435 797 2351; E-mail: ykim@b202.usu.edu.

Most feature selection algorithms have focused on heuristic search approaches, such as sequential search [36], nonlinear optimization [9], and genetic algorithms [54]. Recent reviews of these methods can be found in [13,40]. Regardless of the search algorithm employed, these methods evaluate potential solutions in terms of predictive accuracy. Specifically, the data set could be divided into training and test sets, with the error rate on the test set used to estimate the true error rate of classifiers. However, in many situations we don't have information about the class to which each data point belongs, and thus we cannot apply supervised learning to estimate subset quality.

When we do not have prior information to evaluate candidate solutions, we instead wish to find natural grouping of the examples in the feature space via *clustering* or *unsupervised learning* and utilize the clustering results to evaluate solutions. The idea is to represent groups of points by a cluster center after determining the inherent number of clusters in the given data set. Once the clusters have been formed based on some given features, we must evaluate how well this model represents the complexity of the data. Clustering may be performed using methods such as K-means [18], expectation maximization (EM) [16], or optimization models [8]. Recently a set of novel clustering algorithms have been proposed in the database community [26,55]. For instance, Agrawal *et al.* [1] present an order-independent clustering algorithm, CLIQUE, that forms clusters in large data sets.

The problem of determining an appropriate model in unsupervised learning has gained popularity in the machine learning, pattern recognition, and data mining communities. Unsupervised model selection addresses either how to identify the optimal number of clusters K or how to select feature subsets while determining the correct number of clusters. The latter problem is more difficult because of the inter-dependency between the number of clusters and the feature subsets used to form the clusters [49]. To this point, most research on unsupervised model selection has considered the problem of identifying the right number of clusters using all available features [38,49].

Other researchers [1,51] have studied feature selection and clustering together. In particular, Devaney and Ram [17] combined a sequential forward and backward search algorithm with two concept learning algorithms, COBWEB [21] and AICC, an improved variant of COBWEB. The category utility score was employed as a quality measurement of feature subsets and the number of clusters was one of the factors used for the computation of utility score. In [52], a Bayesian framework with a unified objective function considering both the number of clusters and the feature subset was applied to the problem of document clustering. Recently, Dy and Brodley [19] proposed a wrapper approach that uses an EM algorithm to form clusters. Feature subsets are evaluated in terms of clustering quality based on either scatter separability or maximum likelihood. In this study, we propose a new wrapper approach that considers feature selection and clustering simultaneously.

The model we propose is very flexible so that any clustering algorithm can be easily combined with our feature selection algorithm. We demonstrate this flexibility by applying our algorithm with the two most popular clustering algorithms, K-means and EM. Note that it is not our intention to compare the performance of two clustering algorithms. Since each algorithm is evaluated on a different set of heuristic metrics, it might not be possible to draw general conclusions. However, we can still observe relative performance from this comparative study in terms of speed, the significance of selected features, and so on.

Our model also differs from other approaches in two main aspects of methodology: the evaluation of candidate solutions along multiple independent heuristic criteria, and the use of a local evolutionary algorithm to effectively cover the space of feature subsets and of cluster numbers.

First, we consider multiple fitness criteria simultaneously for evaluating clustering models. A number of heuristic criteria, such as cluster compactness, inter-cluster separation, and maximum likelihood have

been proposed, and attempts have been made to combine some or all of these into a single objective [14, 22]. Previous research on unsupervised model selection considered only one (single or combined) criterion. We claim that our approach is a generalization of such previous work, in the sense that it could capture both linear and non-linear relationships among the criteria.

From the perspective of knowledge discovery, our goal is to provide a clear picture of the (possibly nonlinear) tradeoffs among the various objectives. This is important because no single criterion for unsupervised feature selection is best for every application [20] and only the decision maker can determine the relative weights of criteria for her application. In such situations we must use *multi-objective* or *Pareto* optimization. Formally, each solution s_i is associated with an evaluation vector $F(s_i) = (F_1(s_i), \ldots, F_C(s_i))$ where C is the number of quality criteria. One solution s_1 is said to *dominate* another solution s_2 if $\forall c : F_c(s_1) \geqslant F_c(s_2)$ and $\exists c : F_c(s_1) > F_c(s_2)$, where F_c is the c-th criterion, $c \in \{1 \ldots C\}$. Neither solution dominates the other if $\exists c_1, c_2 : F_{c_1}(s_1) > F_{c_1}(s_2), F_{c_2}(s_2) > F_{c_2}(s_1)$.

The *Pareto front* is defined as the set of nondominated solutions. Our goal in Pareto optimization is to approximate as best possible the Pareto front, presenting the decision maker with a set of high-quality compromise solutions from which to choose. Non-Pareto solutions will not be considered because they are inferior to those in the Pareto front by definition. By providing a set of alternative solutions to the decision maker, our approach helps her to choose the *right* solution at the right time. This could present a big advantage over other decision support systems that provide the decision maker with a single solution, given that she might not be familiar with how the algorithm reached such solution.

Secondly, as a search process, we turn to evolutionary algorithms (EAs) to intelligently search the space of possible feature subsets and to determine the appropriate number of clusters. Our choice of EAs as a search algorithm is reasonable because of their potential capability to search through spaces in a more global fashion than many other machine learning algorithms. EAs have also been used for clustering, using an adjacency-based representation [47] or in conjunction with other algorithms [24,34].

A number of multi-objective extensions of evolutionary algorithms have been proposed in recent years [15]. Most of them, such as the Niched Pareto Genetic Algorithm [29], employ computationally expensive selection mechanisms like fitness sharing [24] and Pareto tournaments to favor dominating solutions and to maintain diversity. Others [30,46,54] combine multiple objectives, such as measurements of model complexity and model accuracy, into one evaluation criterion in a subjective manner. Instead, we use a new evolutionary algorithm that maintains diversity over multiple objectives by employing a *local* selection scheme. This Evolutionary Local Selection Algorithm (ELSA) considers each objective separately and works well for Pareto optimization problems [34,45].

The remainder of this article is organized as follows. We motivate our approach by illustrating possible application areas in Section 2. In Section 3 we review the K-means clustering algorithm and heuristic metrics to evaluate the quality of clusters constructed by K-means. In Section 4 we present the EM algorithm and justify our clustering quality metrics. We discuss our approach in detail in Section 5, illustrating the evolutionary algorithm and describing how ELSA is combined with K-means or EM. Sections 6 and 7 present some experimental results with a synthetic data set and a real data set, and discuss the interpretation of the ELSA output to select a subset of good features. Finally Section 8 addresses directions of future research and concludes the paper.

2. Motivation for clustering-based feature selection

Let us consider some possible applications of our approach in marketing and other domains. The main idea of cluster analyses is to represent groups of points that are similar. For example, cluster analysis can

be used for grouping species of plants or animals or classifying new species. Certain methods for data compression and encryption are also based on similarity among data points. In the marketing research community, clustering and its variants [48,53], neural networks [4] and conjoint analysis [25] have been widely used for market structure analysis and market segmentation. It is well known that manufacturers use different marketing strategies based on customer behavior such as brand loyalty, price sensitivity, or quality sensitivity. Furthermore, they can save time and expense by restricting their concern to a group of customers who are most likely to buy their goods.

Standard application of cluster analysis uses the complete set of features or a pre-selected subset of features. For instance, a market survey data contains various types of questions with regard to respondents' demographic and psychographic information, attitudes toward products and benefits sought. Commonly, separate clustering analyses are implemented to find respondent segments and decide the number of segments based on each different type of variables. A market manager might choose to cluster using only demographic variables of the customers, to offer different campaign options to different customer segments based on their age, sex, education level or income level. Or, the manager might consider customer responses to changes in price, display style, or advertisements to define market segments. Therefore clustering analysis has been implemented in a top-down fashion, dependent on the prior knowledge of market managers who pre-determine the features to be used to segment customers.

However, this top-down approach could not find and exploit interactions among various types of features on segments. Further, some segments can be discovered only if different types of variables are considered together. Therefore, the utility of such a top-down approach is limited from the perspective of knowledge discovery, because it cannot provide new marketing models that could be effective but have not been considered. Our data-driven approach remedies this limitation by searching the space of models, varying the feature subsets and the number of clusters. This way we can present the decision maker with a set of high-quality solutions from which to choose.

As an example, consider the application of our approach to datasets like those collected by insurance companies, containing customer information on both socio-demographic characteristics and ownership of various types of insurance policies (see, for instance, the CoIL data sets [33].) When insurers try to identify customers that are likely to buy a new policy, they consider only a few models dependent on the prior knowledge and past experience of the market managers. Our data-driven approach searches a much broader space of models and provides a compact summary of solutions over possible feature subset sizes and numbers of clusters. Among such high-quality solutions, the manager can select a specific model after considering the model's complexity and accuracy. Further, newly-discovered feature subsets that form well-differentiated clusters can affect the way new marketing campaigns should be implemented. Let us suppose that an insurance company uses our data-driven approach to campaign a new recreational vehicle policy. Let us also assume that our model selects as a solution a set of features including ownership of moped and car policies. The market manager notes that moped policy features are included in a final solution, even though she has never used this information before to identify customer segments. However, further investigation reveals that many people who purchase a moped policy might also purchase a recreational vehicle policy because they often carry their mopeds or bicycles on the back of their vehicles [35].

Similarly, our approach can be useful for the analysis of finance and accounting data. For instance, forecasting corporate bankruptcy has been studied extensively in the accounting, economics, and finance community [2]. However, we are more interested in finding common and unknown factors that affect the financial structure and eventually lead companies to go bankrupt. Our approach provides a number of clustering results from different sets of selected features. If, say, profitability-related features form well-separated clusters in terms of how soon companies go bankrupt, credit analysts can build a model that

```
assign each data point to a randomly chosen cluster
calculate the centroid γ_k of each cluster k
do
   for each point x_n, n ∈ {1,···,N}
      move x_n to nearest cluster arg min_k distance(x_n, γ_k)
   endfor
   for each cluster k with changed membership
      update γ_k
   endfor
while at least one point changed cluster assignment
```

Fig. 1. K-means clustering algorithm.

predicts bankruptcy time more accurately. Or, if clustering analysis reveals that market-driven variables such as market size have serious effects on the performance of the small-sized companies, forecasters of corporate mergers should pay more attention to the changes in these variables than in other variables.

3. K-means algorithm

3.1. Algorithm detail

K-means is one of the most often used nonhierarchical clustering methods [7,22]. Nonhierarchical clustering algorithms are designed to group items into a collection of K clusters that can be specified in advance or determined as part of the clustering procedure. Nonhierarchical methods start from either an initial partition of items into groups or an initial set of seed points, which form the centroid or medoid[1] of clusters.

The K-means algorithm employs a squared error criterion and implicitly assumes that clusters are represented by spherical Gaussian distributions located at the K cluster means [6]. Starting with a random initial partition, it iteratively assigns each data point to the cluster whose centroid is located nearest to the given point, and recalculates the centroids based on the new set of assignments until a convergence criterion is met. Some variants of K-means have been suggested in order to improve the efficiency of the algorithm, avoid initial seed value effects, or find the global optimum [3,39]. However, in our study we use the standard K-means algorithm [31] as summarized in Fig. 1.

3.2. Heuristic metrics for clustering

A number of numerical measurements are available to evaluate clustering quality [14,28]. Most of them are based on geometric distance metrics to measure cluster cohesiveness or inter-cluster separateness. Even though we find that they capture important properties to be measured, these metrics are not directly applicable in our study because they are computed based on the whole dimensionality of feature space. Note that the dimensionality of the space is variable in our study. Therefore any metrics without appropriate adjustment of dimensionality can be biased and misleading. In our study we use four heuristic fitness criteria, described below. Two of the criteria are inspired by statistical metrics and two by Occam's razor [5]. Each objective, after being normalized into the unit interval, is to be maximized by the EA.

[1] A medoid is the most centrally located point in a cluster.

F_{within}: This objective is meant to favor dense clusters by measuring cluster cohesiveness. It is inspired by the total within-cluster sum of squares (TWSS) measure. Formally, let $x_{\text{n}}, n = 1, \cdots, N$, be data points and x_{nj} be the value of the j-th feature of x_n. Let d be the dimension of the *selected* feature set, J, and K be the number of clusters. Now, define the cluster membership variables α_{nk} as follows:

$$\alpha_{\text{nk}} = \begin{cases} 1 \text{ if } x_{\text{n}} \text{ belongs to cluster } k \\ 0 \text{ otherwise} \end{cases}$$

where $k = 1, \cdots, K$ and $n = 1, \cdots, N$.
The centroid of the k-th cluster, γ_k, can be defined by its coordinates:

$$\gamma_{kj} = \frac{\sum_{n=1}^{N} \alpha_{nk} x_{nj}}{\sum_{n=1}^{N} \alpha_{nk}},\ j \in J.$$

F_{within} can be computed as follows:

$$F_{\text{within}} = 1 - \frac{1}{Z_{\text{within}}} \frac{1}{d} \sum_{k=1}^{K} \sum_{n=1}^{N} \alpha_{nk} \sum_{j \in J} (x_{nj} - \gamma_{kj})^2 \tag{1}$$

where the normalization by the number of selected features, d, is meant to compensate for the dependency of the distance metric on the dimensionality of the feature subspace. Z_{within} is a normalization constant meant to achieve F_{within} values spanning the unit interval. Its value is set empirically for each data set.

F_{between}: This objective is meant to favor well-separated clusters by measuring their distance from the global centroid. It is inspired by the total between-cluster sum of squares (TBSS) measure. We compute F_{between} as follows:

$$F_{\text{between}} = \frac{1}{Z_{\text{between}}} \frac{1}{d} \frac{1}{K-1} \sum_{k=1}^{K} \sum_{n=1}^{N} (1 - \alpha_{nk}) \sum_{j \in J} (x_{nj} - \gamma_{kj})^2 \tag{2}$$

where, as for F_{within}, we normalize by the dimensionality of the selected feature subspace and by the empirically derived constant Z_{between}.

F_{clusters}: The purpose of this objective is to compensate for the previous metrics' bias towards increasing the number of clusters. For example, $F_{\text{within}} = 1$ in the extreme case when we have the same number of clusters as the number of data points, with each point allocated to its own cluster. Clearly such overfitting makes the model more complex than can be justified by the data, and thus less generalizable. Therefore, other things being equal, we want fewer clusters:

$$F_{\text{clusters}} = 1 - \frac{K - K_{\min}}{K_{\max} - K_{\min}} \tag{3}$$

where $K_{\max}$ ($K_{\min}$) is the maximum (minimum) number of clusters that can be encoded into a candidate solution's representation.

$\boldsymbol{F}_{\text{complexity}}$**:** The final objective is aimed at finding parsimonious solutions by minimizing the number of selected features:

$$F_{\text{complexity}} = 1 - \frac{d-1}{D-1}. \tag{4}$$

Note that at least one feature must be used. Other things being equal, we expect that lower complexity will lead to easier interpretability and scalability of the solutions as well as better generalization.

4. EM algorithm for mixture models

4.1. Algorithm detail

The expectation maximization algorithm [16] is based on the well-established theory of probability and is one of the most often used statistical modeling algorithms [11,23]. The EM algorithm often significantly outperforms other clustering methods [42] and is superior to the distance-based algorithms (e.g. K-means) in the sense that it can handle categorical data. The EM algorithm for mixture models assumes that the patterns are drawn from one of several given distributions, and the goal is to identify the parameters of each distribution. In the EM framework, the parameters of the clusters are unknown, and these are estimated from the given data.

The EM algorithm starts with an initial estimate of the parameters and iteratively recomputes the likelihood that each pattern is drawn from a particular density function, and then updates the parameter estimates. Formally, let $x_n, n = 1, \cdots, N$, be a data point and x_{nj} be the value of the j-th feature of x_n. Let d be the dimension of the *selected* feature set, J, and K be the number of clusters. If we model each cluster with a d-dimensional Gaussian distribution, we can approximate the data distribution by fitting K density functions c_k, $k = 1, \cdots, K$, to the data set $\{x_n | n = 1, \cdots, N\}$. The probability density function evaluated at x_n is the sum of all densities:

$$P(x_n) = \sum_{k=1}^{K} p_k \cdot c_k(x_n|\theta_k) \tag{5}$$

where the *a priori* probability p_k is the fraction of the data points in cluster k and $\sum_{k=1}^{K} p_k = 1,\ p_k \geqslant 0$. The functions $c_k(x_n|\theta_k)$ are the density functions for patterns of the cluster k and θ_k represents the parameters of the density function. For Gaussian distributions, the parameters are the mean μ_k and covariance matrix Σ_k. For greater efficiency and reduced overfitting, we ignore cross terms and represent Σ_k as a vector of the variances for each dimension. The membership probability of pattern x_n in cluster k is computed as follows:

$$p_k(x_n) = \frac{p_k \cdot c_k(x_n|\theta_k)}{\sum_{i=1}^{K} p_i \cdot c_i(x_n|\theta_i)}. \tag{6}$$

Now, the original problem of finding clusters is reduced to the problem of how to estimate the parameters $\Theta = \{\theta_1, \cdots, \theta_K\}$ of the probability density [10]. Under the independence assumption among attributes within a given cluster, we can represent each density function as a product of density functions over each selected attribute $j = 1, \cdots, d$:

$$c_k(x_n|\theta_k) = \prod_{j \in J} c_{kj}(x_{nj}|\theta_{kj}) \tag{7}$$

where θ_{kj} represents the parameters of the j-th feature of cluster k.

Finally, the multivariate Gaussian distribution for cluster $k = 1, \cdots, K$ is parameterized as follows:

$$c_k(x_n|\mu_k, \Sigma_k) = \prod_{j \in J} \frac{1}{\sqrt{2\ \pi\ \sigma_{kj}^2}} \exp\left(\frac{(x_{nj} - \mu_{kj})^2}{-2\ \sigma_{kj}^2}\right) \tag{8}$$

where μ_{kj} and σ_{kj}^2 represent the mean and variance of the j-th feature of cluster k, respectively.[2] Now we can quantify the quality of a given set of parameters Θ using the Eq. (8). At this point, our only problem is to find the mixture parameters μ_k and Σ_k along with p_k. The maximum likelihood (ML) method [18] is used to maximize the probability of the data set given a particular mixture model, and often the log-likelihood is maximized for analytical purposes as follows:

$$L(\Theta) = \sum_{n=1}^{N} \log P(x_n) = \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} p_k \cdot c_k(x_n|\mu_k, \Sigma_k)\right).$$

The EM algorithm begins with an initial estimation of Θ and iteratively updates it in such a way that the sequence of $L(\Theta)$ is non-decreasing. In our implementation, EM iterates until $|L(\Theta^{t+1}) - L(\Theta^t)| \leqslant \epsilon$, $\epsilon > 0$ or up to $maxIteration$ iterations. We choose the somewhat loose convergence criteria, $\epsilon = 1.0$ and $maxIteration = 15$ because the marginal likelihood gain per additional computing resource for more restrictive criteria is negligible. We outline the standard EM algorithm in Fig. 2.

4.2. Heuristic metrics for clustering

In order to evaluate the quality of the clusters formed by the EM algorithm, we use three heuristic fitness criteria, described below. One of the criteria is inspired by statistical metrics and two by Occam's razor. Each objective is again normalized into the unit interval and maximized by the EA. Note that we exclude two distance-based metrics, F_{within} and F_{between}, but include one new likelihood-based metric. This is mainly because data points can belong to multiple clusters in the EM algorithm, and thus distance-based metrics are not intuitive to evaluate the quality of clustering.

F_{accuracy}: This objective is meant to favor cluster models with parameters whose corresponding likelihood of the data given the model is higher. With estimated distribution parameters μ_k and Σ_k, F_{accuracy} is computed as follows:

$$F_{\text{accuracy}} = \frac{1}{Z_{\text{accuracy}}} \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} p_k \cdot c_k(x_n|\mu_k, \Sigma_k)\right) \tag{9}$$

where Z_{accuracy} is an empirically derived, data-dependent normalization constant meant to achieve F_{accuracy} values spanning the unit interval.

F_{clusters}: This criterion is defined in the same way as in Section 3 (Eq. (3)).

$F_{\text{complexity}}$: This is another criterion defined as in Section 3 (Eq. (3)).

[2] Since small values of σ_{kj}^2 can cause overflow in our computations, we set a lower bound value of σ_{kj}^2 to 10^{-10}.

$t = 0$
`initialize` p_k^t, μ_k^t, `and` Σ_k^t `for each cluster` $k \in \{1, \cdots, K\}$
`compute` $memberProb(t)$
`compute` $L(\Theta^t)$
do
  for each `cluster` $k \in \{1, \cdots, K\}$

$$p_k^{t+1} = \sum_{n=1}^{N} p_k^t(x_n)$$

$$\mu_k^{t+1} = \frac{\sum_{n=1}^{N} p_k^t(x_n) \cdot x_n}{\sum_{n=1}^{N} p_k^t(x_n)}$$

$$\Sigma_k^{t+1} = \frac{\sum_{n=1}^{N} p_k^t(x_n)(x_n - \mu_k^{t+1})(x_n - \mu_k^{t+1})^T}{\sum_{n=1}^{N} p_k^t(x_n)}$$

  endfor
  `compute` $memberProb(t+1)$
  `compute` $L(\Theta^{t+1})$
  $t = t + 1$
while $|L(\Theta^t) - L(\Theta^{t-1})| > \epsilon$ `and` $t < maxIteration$

`compute` $memberProb(t)$
{
  for each `pattern` x_n, $n \in \{1, \cdots, N\}$

$$p_k^t(x_n) = \frac{p_k^t \cdot c_k(x_n | \mu_k^t, \Sigma_k^t)}{\sum_{i=1}^{K} p_i^t \cdot c_i(x_n | \mu_i^t, \Sigma_i^t)}$$

  endfor
}

Fig. 2. Summary of the EM algorithm where $\epsilon > 0$ is a stopping tolerance and p_k^t, μ_k^t, and Σ_k^t represent the mixture model parameters of cluster k at iteration t. In our implementation, we set $\epsilon = 1.0$ and $maxIteration = 15$ for fast convergence.

5. Evolutionary local selection algorithm

ELSA springs from algorithms originally motivated by artificial life models of adaptive agents in ecological environments [43]. Modeling reproduction in evolving populations of realistic organisms requires that selection, like any other agent process, be locally mediated by the environment in which the agents are situated. In these models an agent's fitness results from local individual interactions with the environment, which contains other agents as well as finite shared resources. This naturally yields a diverse set of agents [44], which is essential to achieve a broad coverage of the search space as required for the feature selection problem and for Pareto optimization in general.

Below we describe the ELSA implementation for the feature selection problem discussed in this paper. Further discussion of the algorithm and its application to Pareto optimization problems, including feature selection in supervised learning, can be found elsewhere [33–35].

We first show the wrapper model of ELSA with clustering algorithms in Fig. 3. In our proposed wrapper model, there are three relevant spaces: *search space*, *data space*, and *objective space*. In search space, ELSA searches the space of feature subsets and number of clusters K. Once a specific feature subset (e.g. c) and number of clusters (e.g. $K = 3$) is selected, this information is encoded into a chromosome of an agent. In data space, three clusters are formed via K-means or EM. In objective space, clusters are evaluated in terms of the evaluation criteria and an agent is rewarded energy from each

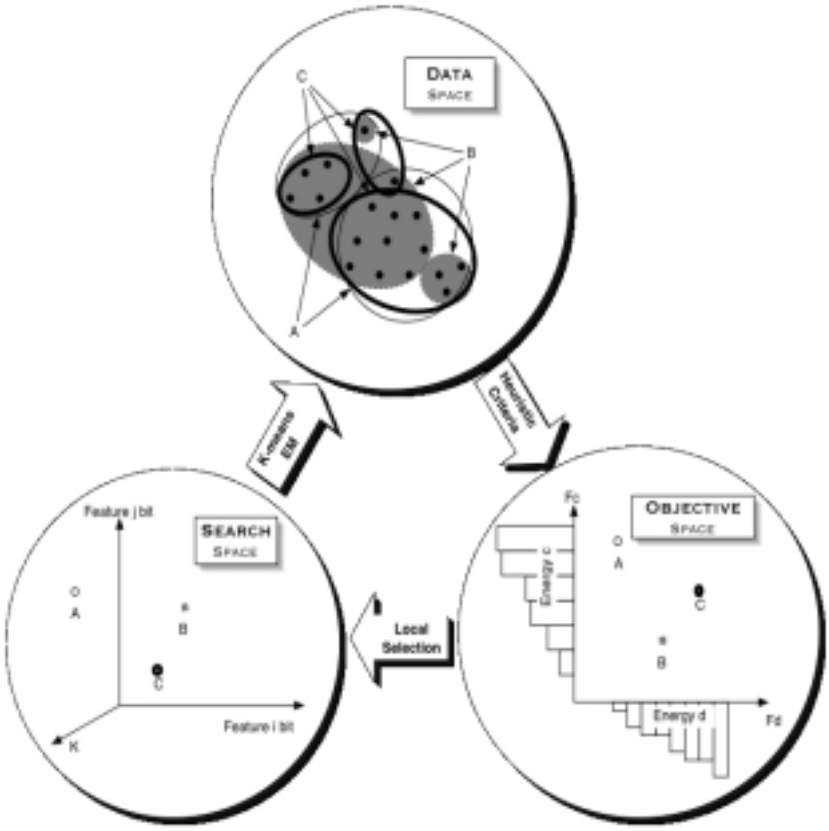

Fig. 3. The wrapper model of ELSA with clustering algorithms.

objective based on its fitness and the local environment to which it belongs. Each agent will survive, reproduce, or die depending on its energy level, and ELSA biases its search in the direction of high energy levels. This process is repeated for a fixed number of iterations, T.

We outline the ELSA algorithm in Fig. 4. Each agent (candidate solution) in the population is first initialized with some random solution and an initial reservoir of energy. The representation of an agent consists of $D + K_{\max} - 2$ bits. D bits correspond to the selected features (1 if a feature is selected, 0 otherwise). The remaining bits are a unary representation of the number of clusters.[3] This representation is motivated by the desire to preserve the regularity of the number of clusters under the genetic operators: changing any one bit will change K by one. Mutation and crossover operators are used to explore the search space. The mutation operator randomly selects one bit of an agent and flips it. Our crossover operator follows the commonality-based crossover framework [12]. It takes two agents, a parent a and a random mate, and scans through every bit of the two agents. If it locates a different bit, it flips a coin to determine the offspring's bit. In this process, the mate contributes only to construct the offspring's bit string, which inherits all the common features of the parents.

In each iteration of the algorithm, an agent a explores a candidate solution a' similar to itself (offspring). The agent collects ΔE from the environment and is taxed with E_{cost} for this action. E_{cost} for any action

[3]The cases of zero or one cluster are meaningless, therefore we count the number of clusters as $K = \kappa + 2$ where κ is the number of ones and $K_{\min} = 2 \leq K \leq K_{\max}$.

```
initialize p_max agents, each with energy η/2
while there are alive agents in Pop^g and t < T
   Replenishment()
   for each agent a in Pop^g
      Search & Evaluation()
      Selection()
      t = t + 1
   endfor
   g = g + 1
endwhile

Replenishment()
{
   for each energy source c ∈ {1, ···, C}
      for each v ∈ {1/B, 2/B, ···, 1} where B is number of bins
         E^c_envt(v) ← 2vE^c_tot
      endfor
   endfor
}

Search & Evaluation()
{
   a' ← mutate(crossover(a, randommate))
   for each energy source c ∈ {1, ···, C}
      v ← Fitness(a')
      ΔE ← min(v, E^c_envt(v))
      E^c_envt(v) ← E^c_envt(v) − ΔE
      E_a ← E_a + ΔE
   endfor
   E_a ← E_a − E_cost
}

Selection()
{
   if (E_a > η)
      insert a, a' into Pop^{g+1}
      E_a' ← E_a/2
      E_a ← E_a − E_a'
   else if (E_a > 0)
      insert a into Pop^{g+1}
   endif
}
```

Fig. 4. ELSA pseudo-code. In each iteration, the environment is replenished and then each alive agent executes the main loop. The program terminates after T solutions (agents) are evaluated. The details of the algorithm are illustrated in the text.

is a constant ($E_{\text{cost}} < \eta$). The net energy intake of an agent is determined by its offspring's fitness. This is a function of how well the candidate solution performs with respect to the criteria being optimized. But the energy also depends on the state of the environment. The environment corresponds to the set of possible values for each of the criteria being optimized.[4] We have an energy source for each criterion, divided into bins corresponding to its values. So, for criterion fitness F_c and bin value v, the environment keeps track of the energy $E^c_{\text{envt}}(v)$ corresponding to the value $F_c = v$. Further, the environment keeps

[4]Continuous objectives are discretized.

a count of the number of agents $P_c(v)$ having $F_c = v$. The energy corresponding to a solution a for criterion c is given by

$$\text{Fitness}(a, c) = \frac{F_c(a)}{P_c(F_c(a))}. \tag{10}$$

Agents receive energy only inasmuch as the environment has sufficient resources; if these are depleted, no benefits are available until the environmental resources are replenished. Thus an agent is rewarded with energy for its high fitness values, but also has an interest in finding unpopulated niches in objective space, where more energy is available. The result is a natural bias toward diverse solutions in the population.

In the selection part of the algorithm, an agent compares its current energy level with a constant reproduction threshold η. If its energy is higher than η, the agent reproduces: the agent and its mutated clone that was just evaluated become part of the new population, each with half of the parent's energy. If the energy level of an agent is positive but lower than η, only the agent itself joins the new population. If an agent runs out of energy, it is killed. The population size is independent of the reproduction threshold; η only affects the energy stored by the population at steady-state.

When the environment is replenished, each criterion c is allocated an equal share of energy as follows:

$$E^c_{\text{tot}} = \frac{E_{\text{total}}}{C} = \frac{p_{\max} E_{\text{cost}}}{C} \tag{11}$$

where $C = 4$ for K-means or $C = 3$ for EM in this study. This energy is apportioned in linear proportion to the values of each fitness criterion, so as to bias the population toward more promising areas in objective space. Note that the total replenishment energy that enters the system at each iteration, $E_{\text{total}} = p_{max} E_{\text{cost}}$ is determined in such a way that the population size p will be bound by $p_{\max}$. This is because energy is conserved and the energy leaving the system ($p\ E_{\text{cost}}$) is always less than the total energy (E_{total}).

In order to assign energy to a solution, ELSA must be informed of clustering quality. In the experiments described here, the clusters to be evaluated are constructed based on the selected features using a standard K-means or EM algorithm (cf. Fig. 3). Each time a new candidate solution is evaluated, the corresponding bit string is parsed to get a feature subset J and a cluster number K. The clustering algorithm is given the projection of the data set onto J, uses it to form K clusters, and returns the fitness values.

6. Experiments on synthetic data set

6.1. Data description and baseline algorithm

It is difficult to evaluate the quality of an unsupervised learning algorithm, and feature selection problems present the added difficulties that the clusters depend on the dimensionality of the selected features and that any given feature subset may have its own clusters, which may well be incompatible with those formed from different subsets. In order to evaluate our approach, we construct a moderate-dimensional synthetic data set, in which the distributions of the points and the significant features are known, while the appropriate clusters in any given feature subspace are not known. We evaluate the evolved solutions by their ability to discover five pre-constructed clusters in a ten-dimensional subspace.

The data set has $N = 500$ points and $D = 30$ features. The feature set consists of "significant" features, "Gaussian noise" features, and "white noise" features. It is constructed so that the first 10

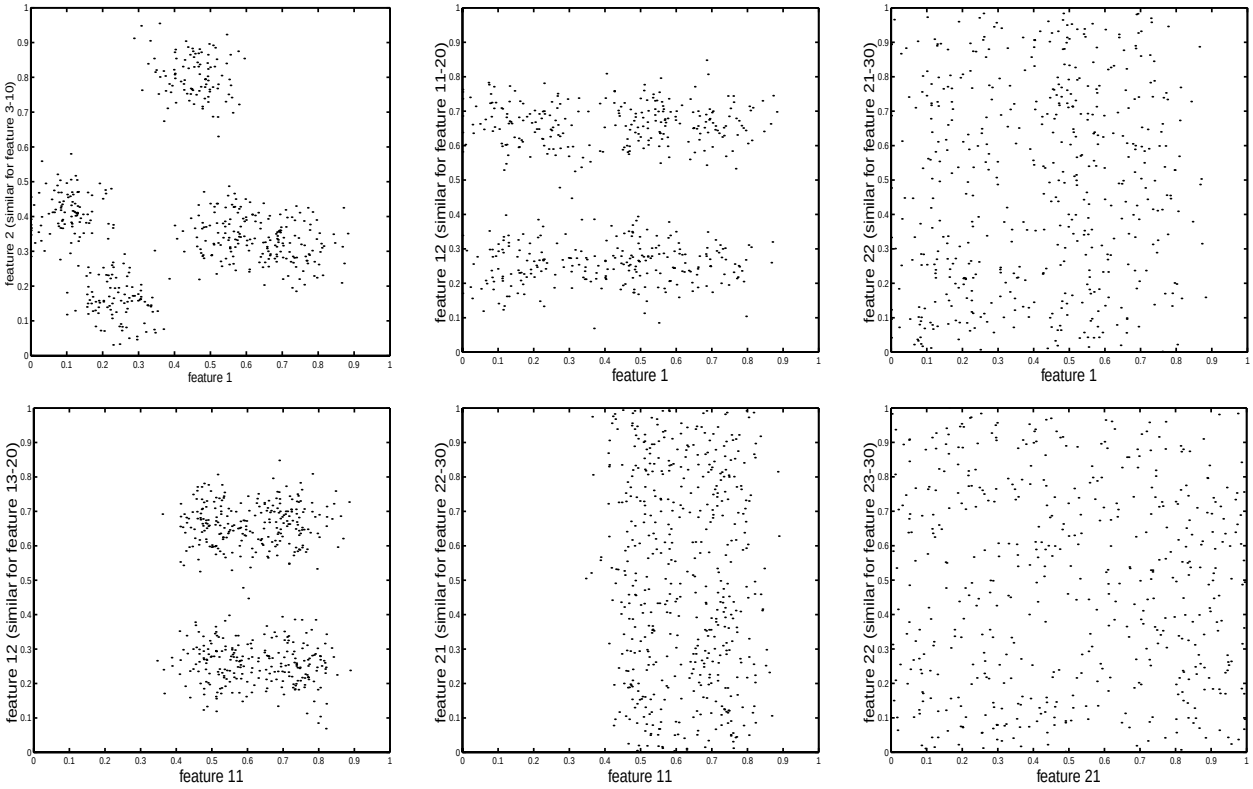

Fig. 5. A few 2-dimensional projections of the synthetic data set.

features are significant, with 5 "true" normal clusters consistent across these features. The next 10 features are Gaussian noise, with points randomly and independently assigned to 2 normal clusters along each of these dimensions. The remaining 10 features are white noise in which points are drawn from uniform distributions. The standard deviation of the normal distributions is $\sigma \approx 0.06$ and the means are themselves drawn from uniform distributions in the unit interval, so that the clusters may overlap. We present some 2-dimensional projections of the synthetic data set in Fig. 5.

For further comparisons we have implemented a greedy heuristic algorithm known as the *plus 2-take away 1 sequential selection* algorithm [36]. This is a reasonable choice for a comparative algorithm because we want our algorithm to outperform most commercial statistical programs (e.g. SAS and SPSS) that implement simpler search algorithms, such as sequential forward and backward selection, for feature selection. Since the greedy algorithm we have implemented allows limited backtracking, it performs better than feature selection algorithms typically used in commercial programs. Our implementation of this algorithm for clustering requires a set value of K and uses F_{within} and F_{between} for K-means, and F_{accuracy} for EM as the optimization criteria. It begins by finding the single dimension along which the objective is optimized. At each successive step, the algorithm adds an additional feature that, when combined with the current set, forms the best clusters. It then checks to see if the least significant feature in the current set can be eliminated to form a new set with superior performance. This iteration is continued until all the features have been added. We ran the algorithm for each of the values of K considered by ELSA.

Individuals are represented by 36 bits, 30 for the features and 6 for K ($K_{\max} = 8$). There are 15

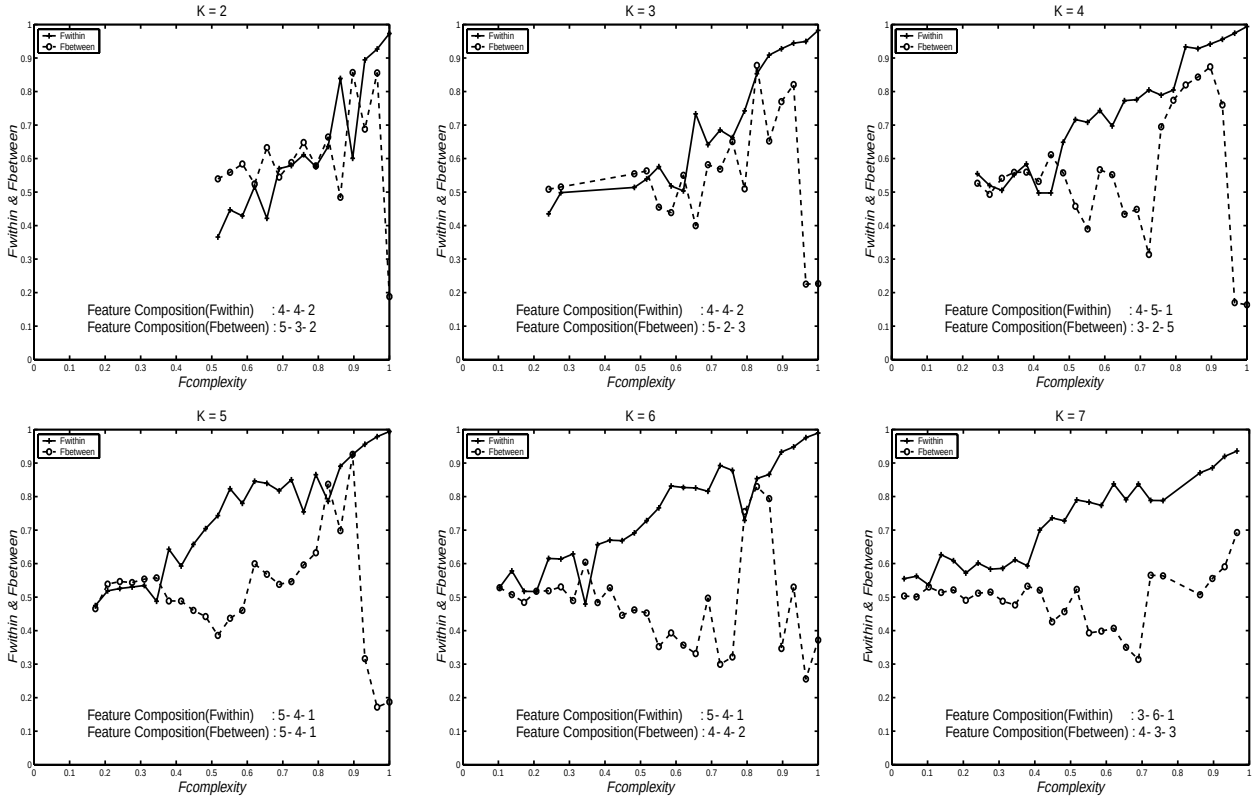

Fig. 6. The ELSA/K-means fronts with composition of features selected for $F_{\mathrm{complexity}}$ corresponding to 10 features (see text). We omit the candidate fronts of $K = 8$ because of its incomplete coverage of the search space.

energy bins for all energy sources, F_{clusters}, $F_{\mathrm{complexity}}$, F_{within}, F_{between}, and F_{accuracy}. The values for the various ELSA parameters are: $\Pr(\mathrm{mutation})$ = 1.0, $\Pr(\mathrm{crossover})$ = 0.8, $p_{\max}$ = 100, E_{cost} = 0.2, E_{total} = 40, η = 0.3, and T = 30,000.

6.2. Results using K-means

We first show two different types of Pareto approximation evolved by ELSA/K-means in Fig. 6, one based on F_{within} and the other one based on F_{between}, in order to observe the usefulness of our two clustering quality metrics. Recall that both are used in ELSA to evaluate the quality of clusters. We use the term *candidate front* for the set of solutions with the highest measured clustering quality at every $F_{\mathrm{complexity}}$ value for each K. We construct candidate fronts based on all solutions evaluated during the evolutionary process with two different clustering quality measures, F_{within} and F_{between}. In order to show the candidate fronts of each different number of clusters K, we sort all the non-dominated solutions by K.

We expect the candidate front based on F_{within} for any reasonable K to typically decrease from higher values of $F_{\mathrm{complexity}}$ (lower complexity) to lower values of $F_{\mathrm{complexity}}$ (higher complexity). This is because we normalize F_{within} by the number of selected features d. Selecting more features make it more likely to select less relevant features, deteriorating the clustering quality. The fronts based on F_{within} in Fig. 6 show the trend that we expect. The clustering quality in terms of F_{within} improves as

the number of clusters approaches the true number of clusters, K = 5. In particular, the fronts for K = 5 and K = 6 not only explore most $F_{\text{complexity}}$ values but also show high clustering quality. A decision maker would determine the correct number of clusters to be either 5 or 6.

The candidate fronts based on F_{between} are less stable than those based on F_{within}. We attribute this to the fact that F_{between} is more sensitive to outliers than F_{within}. F_{between} is affected explicitly by both d and K in its computation, while F_{within} is affected explicitly by d but implicitly by K via clustering quality. However, the fronts become stable with more than half of features selected because many features neutralize the effects of outliers from certain features. Although we feel that F_{between} captures useful information about the quality of the clusters, its instability makes it inappropriate as a single metric to determine the best solution to be presented to a decision maker.

We also show in Fig. 6 the composition of selected features, i.e., the number of significant-Gaussian noise-white noise features selected at $F_{\text{complexity}}$ = 0.69 (10 features). Note that the selected features at this value of $F_{\text{complexity}}$ are not necessarily all the "significant" features that we constructed. We attribute this finding to the fact that if one or more Gaussian noise features form good clusters with the previously selected significant features, the clustering quality can be improved by adding these features. This is also consistent with the notion that not all strongly relevant features are selected and some weakly relevant features could be selected as "relevant" features [37]. Though even mixes of significant and Gaussian features are selected at $F_{\text{complexity}}$ = 0.69, ELSA/K-means found a better composition of selected features at values of $F_{\text{complexity}}$ near 0.69. For example, the composition of selected features for K = 5 based on F_{within} were 8-3-1, 7-3-1, 5-4-1, and 6-3-0 over $0.62 \leq F_{\text{complexity}} \leq 0.73$ (9–12 features), respectively.

Figure 7 shows snapshots of the candidate fronts with K = 5 based on F_{within} at intervals of every 3,000 solution evaluations. It is evident that ELSA/K-means identifies better solutions and explores an increasingly broad space of feature subsets as it evaluates more solutions.[5] We show the improvement of the candidate fronts by computing the $\text{coverage}_{\text{KM}}$ as follows:

$$\text{coverage}_{KM} = \sum_{i \in F_{\text{complexity}}} F^i_{\text{within}} \tag{12}$$

where F^i_{within} is the F_{within} value at $F_{\text{complexity}} = i$. As ELSA finds new and better solutions (with higher F_{within}), the coverage increases.

We finally evaluated ELSA/K-means in terms of classification accuracy. We compute accuracy by assigning a class label to each cluster based on the majority class of the points contained in the cluster, and then computing correctness on *only those classes*, e.g., models with only two clusters are graded on their ability to find two classes. ELSA results represent individuals chosen from fronts based on $\widetilde{F}_{\text{accuracy}} = F_{\text{within}} \cdot F_{\text{between}}$.[6] This criterion is based on the fact that neither F_{within} nor F_{between} truly represents the quality of the clusters. The classification accuracy of candidate solutions based on either F_{within} or F_{between} was inferior to that based on $\widetilde{F}_{\text{accuracy}}$. Table 1 shows the classification accuracy with standard error of various models formed by both ELSA/K-means and the greedy feature search. ELSA/K-means results represent individuals with less than eight features from the candidate fronts. All accuracy measures are averaged over five different runs of ELSA/K-means and of the greedy search.

The overall performance of ELSA/K-means is superior to that of the greedy search on models with few features and few clusters–exactly the sort of models the algorithm was designed to find. The last row and

[5] Similar results were obtained for different number of clusters K and for F_{between}.

[6] This new measurement is used only for selecting a final solution. In ELSA, F_{within} and F_{between} are considered separately.

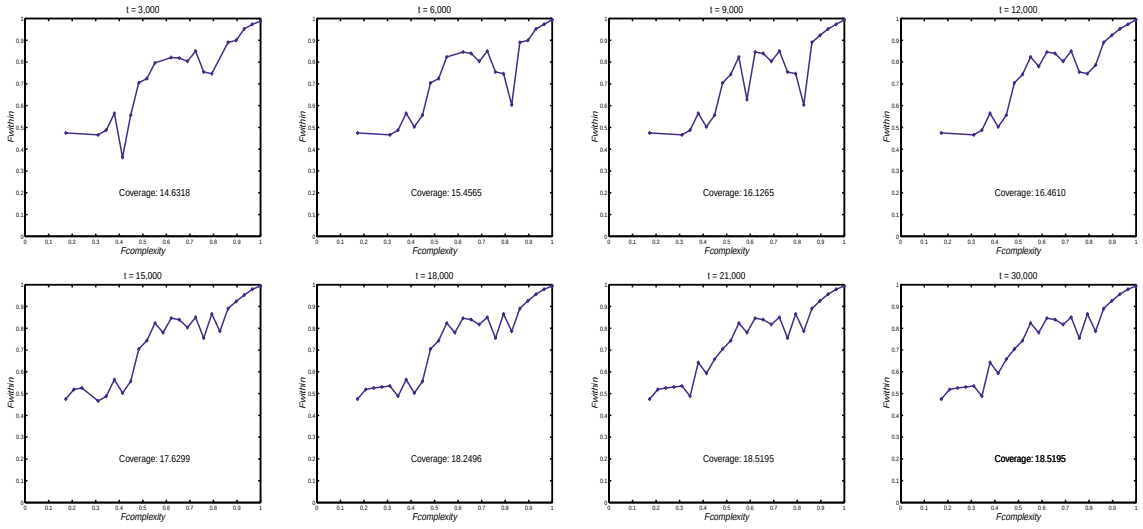

Fig. 7. The candidate fronts for $K = 5$ based on F_{within} evolved in ELSA/K-means. It is captured every 3,000 evaluated solutions. There is no further improvement in coverage after the 7th interval.

Table 1
The average classification accuracy (%) with standard error of five runs of ELSA/K-means and greedy search. The last row and column show the number of win-lose-tie cases of ELSA/K-means compared with greedy search. A tie is assumed when error bars overlap

K		Number of selected features						
		2	3	4	5	6	7	W-L-T
2	ELSA/KM	100 ± 0.0	100 ± 0.0	100 ± 0.0	100 ± 0.0	59 ± 0.0	100 ± 0.0	5-0-1
	Greedy	59 ± 0.0	59 ± 0.0	59 ± 0.0	59 ± 0.0	59 ± 0.0	59 ± 0.0	
3	ELSA/KM	93.2 ± 5.2	39.4 ± 0.4	98.6 ± 1.4	100 ± 0.0	100 ± 0.0	100 ± 0.0	3-1-2
	Greedy	40.6 ± 0.3	40.8 ± 0.2	40.2 ± 0.2	63.6 ± 3.9	100 ± 0.0	100 ± 0.0	
4	ELSA/KM	85.2 ± 4.1	31.4 ± 0.3	92 ± 4.9	100 ± 0.0	100 ± 0.0	100 ± 0.0	5-1-0
	Greedy	30.8 ± 0.2	55 ± 0.0	55 ± 0.0	55 ± 0.0	55 ± 0.0	55 ± 0.0	
5	ELSA/KM	62 ± 0.6	48.4 ± 1.5	75 ± 2.1	58.4 ± 1.9	63.4 ± 0.4	79.4 ± 0.6	4-1-1
	Greedy	25.6 ± 0.3	53.4 ± 0.6	53.4 ± 0.6	55 ± 0.0	63 ± 0.0	66.4 ± 3.4	
	W-L-T	4-0-0	1-3-0	4-0-0	4-0-0	1-0-3	3-0-1	17-3-4

column shows the number of win-loss-tie cases of ELSA/K-means compared with greedy search. The performance of ELSA/K-means for $d = 3$ across different K is slightly inferior, although the difference is small for $K = 3$ and $K = 5$. For more complex models with more than 10 selected features (not shown), the greedy method is often better able to reconstruct the original classes. This is reasonable, since ELSA by design does not concentrate on this part of the search space.

6.3. Results using EM

We show the candidate fronts found by the ELSA/EM algorithm for each different number of clusters K in Fig. 8. In contrast with the ELSA/K-means model, we have a single measurement of clustering quality F_{accuracy} in ELSA/EM. We did the same analysis to see whether our ELSA/EM model is able to identify the correct number of clusters based on the shape of the candidate fronts across different values of K and F_{accuracy}. A different characteristic shape of the Pareto fronts is observed in ELSA/EM because of the different measurement of clustering quality: an ascent in the range of higher values of

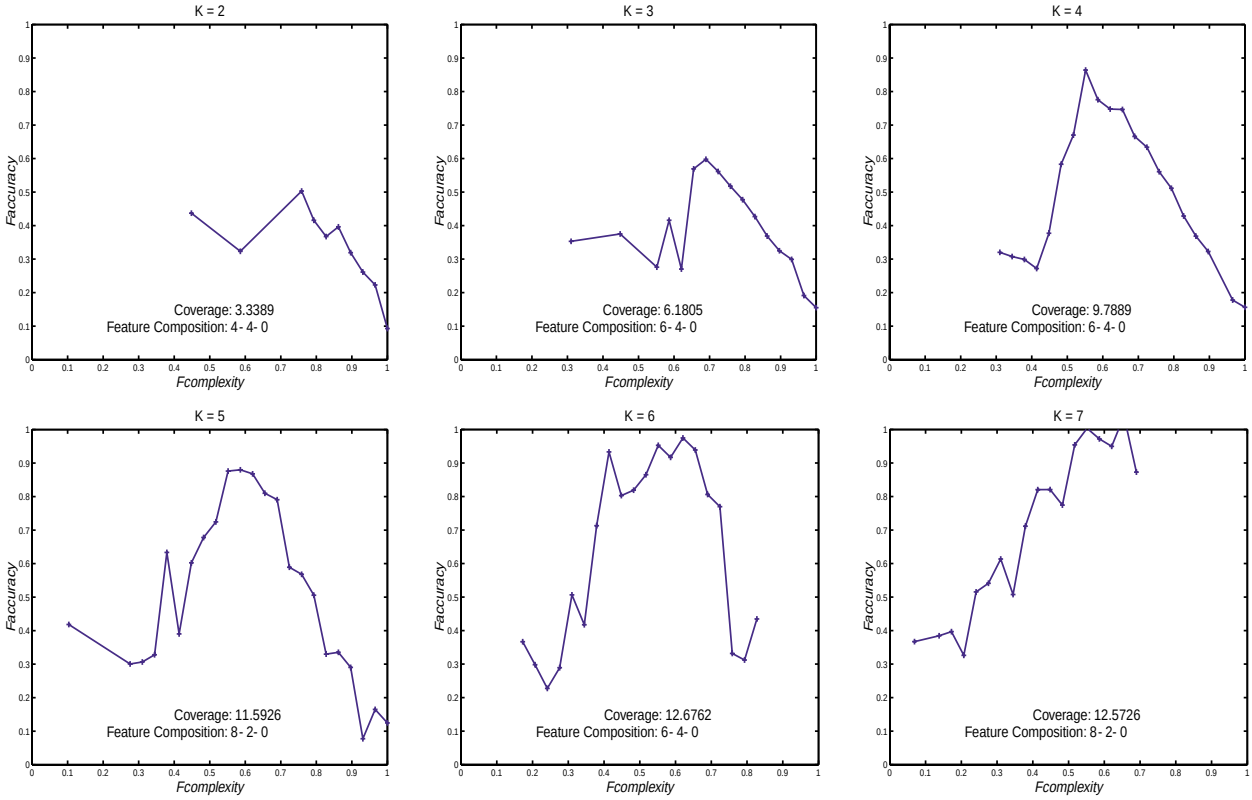

Fig. 8. The candidate fronts of ELSA/EM model. We omit the candidate front for $K = 8$ because of its inferiority in terms of clustering quality and incomplete coverage of the search space. Composition of selected features is shown for $F_{complexity}$ corresponding to 10 features (see text).

$F_{\text{complexity}}$ (lower complexity), and a descent for lower values of $F_{\text{complexity}}$ (higher complexity). This is reasonable because adding additional significant features will have a good effect on the the clustering quality with few previously selected features. However, adding noise features will have a negative effect on clustering quality in the probabilistic model, which, unlike Euclidean distance, is not affected by dimensionality. Hence the curve forms a shape similar to the supervised learning curve, with a global maximum indicating the optimal number of features. The coverage of the ELSA/EM model shown in Fig. 8 is defined as:

$$\text{coverage}_{EM} = \sum_{i \in F_{\text{complexity}}} F^i_{\text{accuracy}} \tag{13}$$

We note that the clustering quality and the search space coverage improve as the evolved number of clusters approaches the "true" number of clusters, K = 5. The candidate front for K = 5 not only shows the typical shape we expect but also an overall improvement in clustering quality. The other fronts do not cover comparable ranges of the feature space either because of the agents' low F_{clusters} (K = 7) or because of the agents' low F_{accuracy} and $F_{\text{complexity}}$ (K = 2 and K = 3). A decision maker again would conclude the right number of clusters to be 5 or 6.

As noticed in ELSA/K-means, the first 10 selected features, $0.69 \leq F_{\text{complexity}} \leq 1$, are not all significant. This notion is again quantified through the number of significant-Gaussian noise-white noise

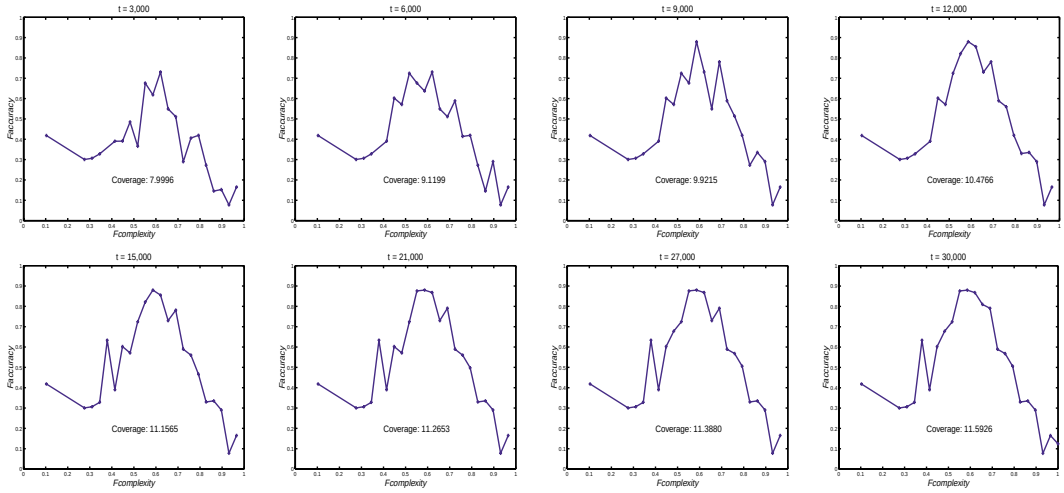

Fig. 9. Candidate fronts for $K = 5$ based on F_{accuracy} evolved in ELSA/EM. It is captured at every 3,000 solution evaluations and two fronts ($t = 18,000$ and $t = 24,000$) are omitted because they have the same shape as the ones at $t = 15,000$ and $t = 21,000$, respectively.

features selected at $F_{\text{complexity}}$ = 0.69 (10 features) in Fig. 9.[7] None of the "white noise" features is selected and the overall composition of selected features is better in ELSA/EM than in ELSA/K-means.

We also show snapshots of the ELSA/EM fronts for K = 5 at every 3,000 solution evaluations in Fig. 9. Similarly to the ELSA/K-means model, ELSA/EM explores a broad subset of the search space, and thus identifies better solutions across $F_{\text{complexity}}$ as more solutions are evaluated. We observed similar results for different number of clusters K.

Table 2 shows classification accuracy of various models formed by both ELSA/EM and the greedy feature search. We compute accuracy in the same way that we did in ELSA/K-means. ELSA results represent individuals selected from candidate fronts with less than eight features. ELSA/EM consistently outperforms the greedy search on models with few features and few clusters. As we noticed in the ELSA/K-means case, for more complex models with more than 10 selected features, the greedy method often shows higher classification accuracy.

7. Experiments on WPBC data

In addition to the artificial data set discussed in Section 6, we also tested our algorithm on a real data set, the Wisconsin Prognostic Breast Cancer (WPBC) data [41]. This data set records 30 numeric features quantifying the nuclear grade of breast cancer patients at the University of Wisconsin Hospital, along with two traditional prognostic variables–tumor size and number of positive lymph nodes. This results in a total of 32 features for each of 198 cases. For the experiment, individuals are represented by 38 bits, 32 for the features and 6 for K ($K_{\max} = 8$). Other ELSA parameters are the same as those used in the previous experiments.

[7] For K = 2, we use $F_{\text{complexity}}$ = 0.76, which is the closest value to 0.69 represented in the front.

Table 2
The average classification accuracy (%) with standard error of five runs of ELSA/EM and greedy search. The "–" entry indicates that no solution is found by ELSA/EM. The last row and column show the number of win-loss-tie cases of ELSA/EM compared with greedy search

K		Number of selected features						
		2	3	4	5	6	7	W-L-T
2	ELSA/EM	52.6 ± 0.3	56.6 ± 0.6	92.8 ± 5.2	100 ± 0.0	100 ± 0.0	100 ± 0.0	5-0-1
	Greedy	51.8 ± 1.3	52.8 ± 0.8	55.4 ± 1.1	56.6 ± 0.4	62.8 ± 3.2	80.2 ± 8.5	
3	ELSA/EM	83.2 ± 4.8	52 ± 6.6	91.6 ± 5.7	93.8 ± 6.2	99 ± 1.0	100 ± 0.0	4-0-2
	Greedy	40.6 ± 0.3	40.8 ± 0.2	40.2 ± 0.2	63.6 ± 3.8	100 ± 0.0	100 ± 0.0	
4	ELSA/EM	46.2 ± 2.2	–	50.6 ± 0.6	89.6 ± 5.9	52 ± 1.0	60.6 ± 5.1	4-2-0
	Greedy	27.8 ± 0.8	27.8 ± 0.4	29 ± 0.4	29.6 ± 0.9	38 ± 4.4	74.2 ± 3.5	
5	ELSA/EM	44.6 ± 2.0	32.6 ± 3.8	72 ± 3.8	62.4 ± 1.9	66.4 ± 3.7	88 ± 4.9	5-0-1
	Greedy	23 ± 0.4	22.2 ± 0.8	24.2 ± 0.9	23.8 ± 0.5	29.6 ± 1.7	81.2 ± 3.0	
	W-L-T	3-0-1	3-1-0	4-0-0	4-0-0	3-0-1	1-1-2	18-2-4

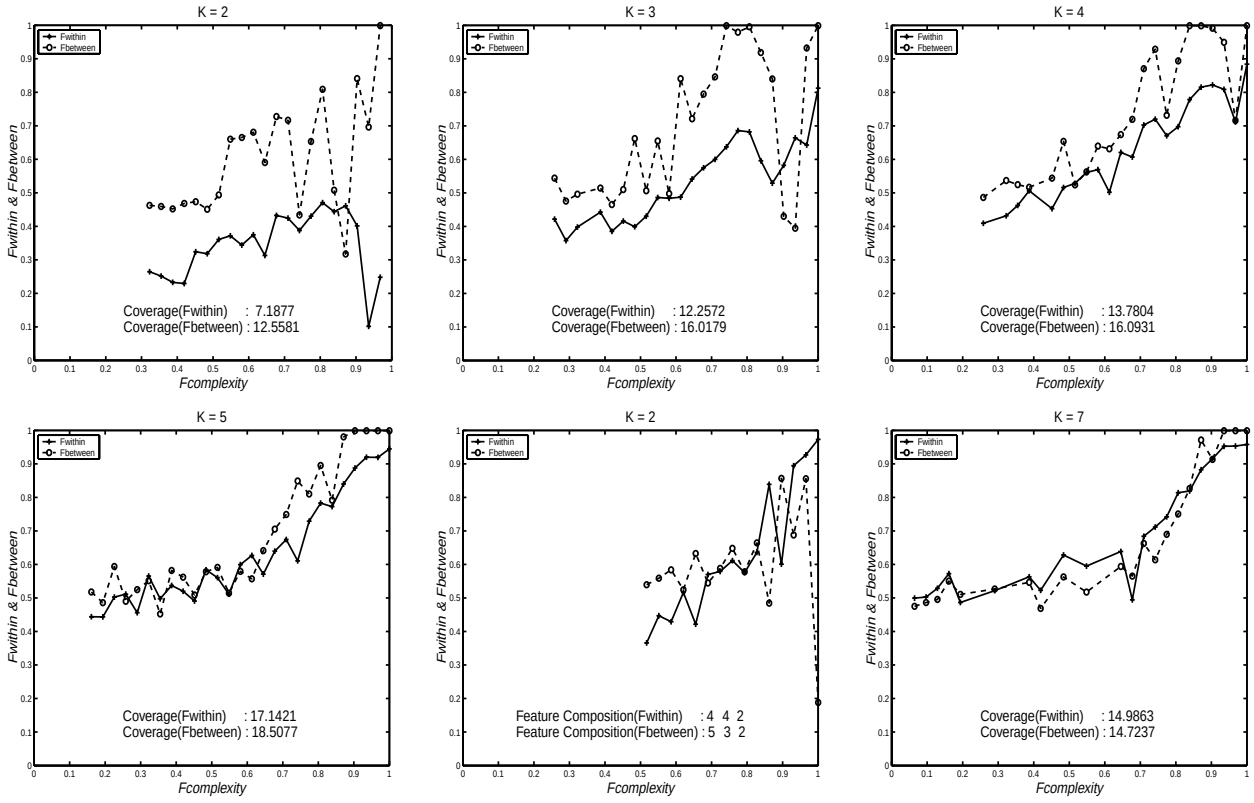

Fig. 10. Candidate fronts evolved by ELSA/K-means on the WPBC data. The front for $K = 8$ is omitted because of its incomplete coverage of the search space.

7.1. Clustering analysis

In this experiment, we assume that we have no prior knowledge about the clusters and the relevant features. We first show two different types of fronts evolved by ELSA/K-means in Fig. 10, one based on F_{within} and the other based on F_{between}.

The candidate fronts based on F_{within} in Fig. 10 again show a typical decrease in quality from

higher values of $F_{\text{complexity}}$ (lower complexity) to lower values of $F_{\text{complexity}}$ (higher complexity). It is interesting to note that the fronts based on F_{between} not only show much more stable patterns than those in Fig. 6 but also become almost identical to the fronts based on F_{within}, as K increases. We attribute this partially to the composition of correlated features in the WPBC data. The correlation among features comes from the fact that the mean, the standard error and the largest value of the 10 measurements that quantify the nuclear grade of breast cancer were recorded into the WPBC data, resulting in a highly correlated set of 30 features. Further, none of these features is regarded as white noise because each feature reflects some aspect of nuclear grade.

A decision maker might pick K = 5 as the correct number of clusters because the candidate front for K = 5 not only explores most of $F_{\text{complexity}}$ values but also shows a stable pattern with high clustering quality in terms of both F_{within} and F_{between}. However, let us select a model with K = 3 in order to compare our approach to previous research in which three clusters have been used to analyze this data set. In addition, the smaller number of clusters makes it easier to understand clustering results and satisfies one of our criteria, the preference for parsimonious models.

We also select a "best" solution (feature set) for prognostic analysis based on the value of $\tilde{F}_{\text{accuracy}}$. In particular, we chose the solution with three clusters and the highest value of $\tilde{F}_{\text{accuracy}}$ among solutions that have between five and ten features. These minimum and maximum limits on the number of features are used to find a robust but simple solution, respectively. The chosen solution has seven features and its implication for prognostic analysis is discussed in Section 7.2.

Our findings by ELSA/K-means are confirmed in the candidate fronts evolved by ELSA/EM, shown in Fig. 11. The correlation among features and the absence of white noise features result in a different characteristic shape of the candidate fronts from those in Fig. 8. The fronts show a steady increase from the range of higher values of $F_{\text{complexity}}$ (lower complexity) to the range of lower values of $F_{\text{complexity}}$ (higher complexity). However, the curves peak at a certain point (e.g., $F_{\text{complexity}}$ = 0.26 for $K \geqslant$ 4) because most of the information in the feature set is already extracted through previously selected features.

A decision maker might determine the correct number of clusters to be K = 4 or K = 5 because those models not only explore most of the $F_{\text{complexity}}$ values but also show a stable pattern with high clustering quality in terms of F_{accuracy}. For prognostic analysis, however, we again will consider solutions with three clusters, in order to be consistent with previous research. We note that the F_{accuracy} values of solutions with up to 10 features are steadily improving, which makes it difficult to choose any one of them as our final solution. This makes us turn to the gradient information in the candidate front for K = 3. We choose a solution that causes the greatest improvement in clustering quality in terms of F_{accuracy}. The chosen solution has 11 features ($F_{\text{complexity}}$ = 0.68) and we discuss its prognostic implication in the following section.

7.2. Prognostic analysis

We analyzed performance on this data set by looking for clinical relevance in the resulting clusters. Specifically, we observe the actual outcome (time to recurrence, or known disease-free time) of the cases in the three clusters. Figure 12 shows Kaplan-Meier estimates [32] of the true disease-free survival times for patients in the clusters found by ELSA/K-means.

Figure 12 displays well-separated survival characteristics of three prognostic groups: good (88 patients), intermediate (83 patients), and poor (27 patients). The good prognostic group was significantly different from the intermediate group ($p < 0.01$) and the intermediate group was well-differentiated from the poor group ($p < 0.06$).

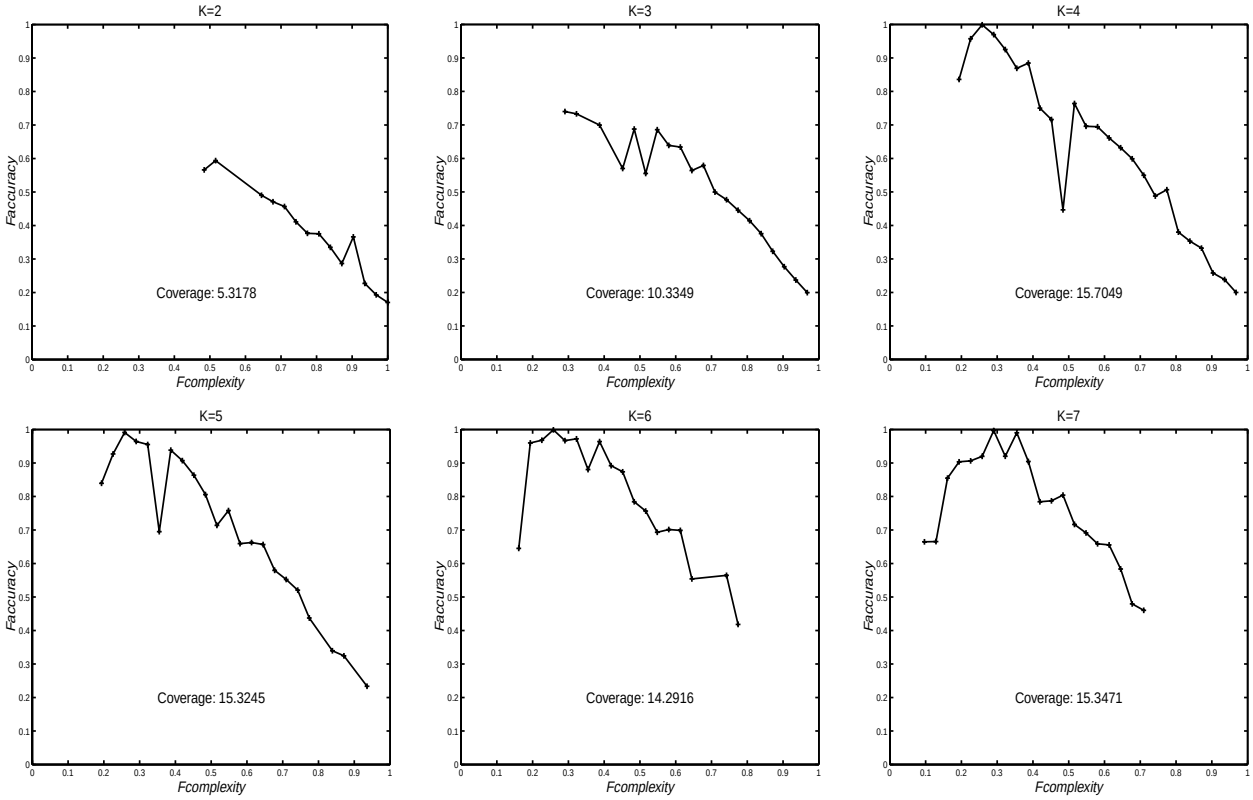

Fig. 11. Candidate fronts evolved by ELSA/EM on the WPBC data. The front for $K = 8$ is omitted because of its incomplete coverage of the search space.

Five-year recurrence rates of these groups were 11.28%, 35.91%, and 47.96%, respectively. The chosen solution used to cluster patients into three groups has seven dimensions including the mean nuclear radius and area, the standard error of the radius and area, and the largest value of the radius, perimeter and area. It is interesting that neither of the traditional medical prognostic factors, tumor size and lymph node status, is selected by ELSA/K-means.

Similarly, Fig. 13 shows the survival characteristics of three prognostic groups found by ELSA/EM. The three groups showed well-separated survival characteristics and more balanced clustering quality in the sense that patients are more evenly distributed. Out of 198 patients, 59 patients belong to the good prognostic group, and 54 patients and 85 patients belong to intermediate and poor prognostic groups, respectively. The good prognostic group was well-differentiated from the intermediate group ($p < 0.076$) and the intermediate group was significantly different from the poor group ($p < 0.036$). Five-year recurrence rates were 12.61%, 21.26%, and 39.85% for the patients in the three groups.

The chosen dimensions by ELSA/EM included a mix of nuclear morphometric features such as the mean and the standard error of the radius, perimeter and area, and the largest value of the area and symmetry along three other features. We note that again neither of the traditional medical prognostic factors is chosen, which is consistent with the result of ELSA/K-means. This finding is potentially important because one of the traditional prognostic factors, the lymph node status, can be determined by microscopic examination of lymph nodes only after they are surgically removed from the patient's armpit [50]. Our experiments tend to support the hypothesis that prognostic groups with significantly

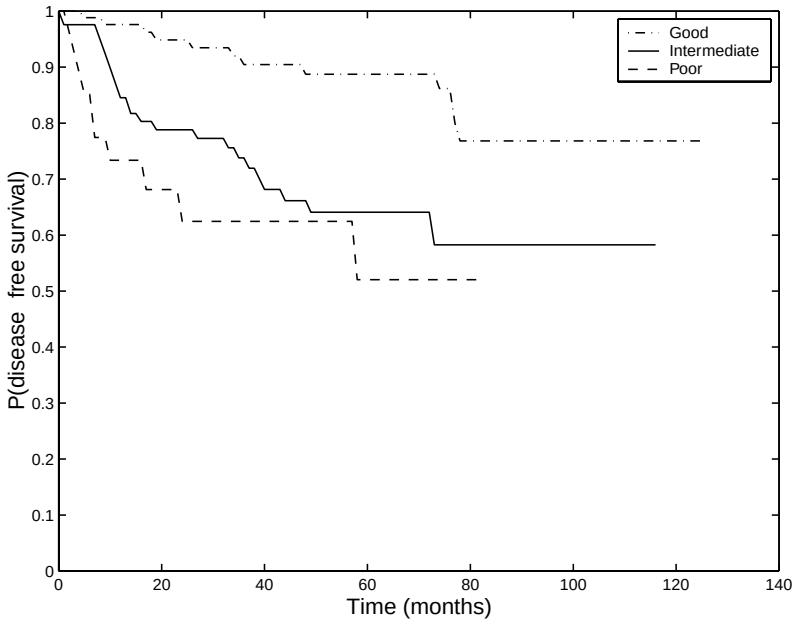

Fig. 12. Estimated survival curves for the three groups found by ELSA/K-means.

different expected outcomes can be formed without this data.

In order to address this matter, we investigate whether other solutions with lymph node information can form three prognostic groups as good as our EM chosen solution. For this purpose, we selected Pareto solutions across all different K values that have fewer than 10 features including lymph node information and formed three clusters using these selected features, disregarding the evolved value of K. The survival characteristics of the three prognostic groups found by the best of these solutions was very competitive with our chosen solution. The good prognostic group was well-differentiated from the intermediate group ($p < 0.10$), and the difference between the intermediate group and the poor group was significant ($p < 0.026$). This suggests that lymph node status may indeed have strong prognostic effects, even though it is excluded from the best models evolved by our algorithms.

8. Conclusions

We presented a novel evolutionary multi-objective local selection algorithm for unsupervised feature selection. ELSA, an evolutionary local selection algorithm, was used successfully in previous work in conjunction with supervised learning [33,45]. As an extension of our previous work [34], we used ELSA to search for possible combination of features and numbers of clusters, with the guidance of two representative clustering algorithms, K-means and EM. The combination of a multi-objective search algorithm with unsupervised learning provides a promising framework for feature selection. We summarize our findings as follows.

– ELSA covers a large space of possible feature combinations while simultaneously optimizing the multiple criteria separately. In particular, as ELSA evaluates more solutions, it finds new and better solutions, improving the candidate fronts of both ELSA/K-means and ELSA/EM models.

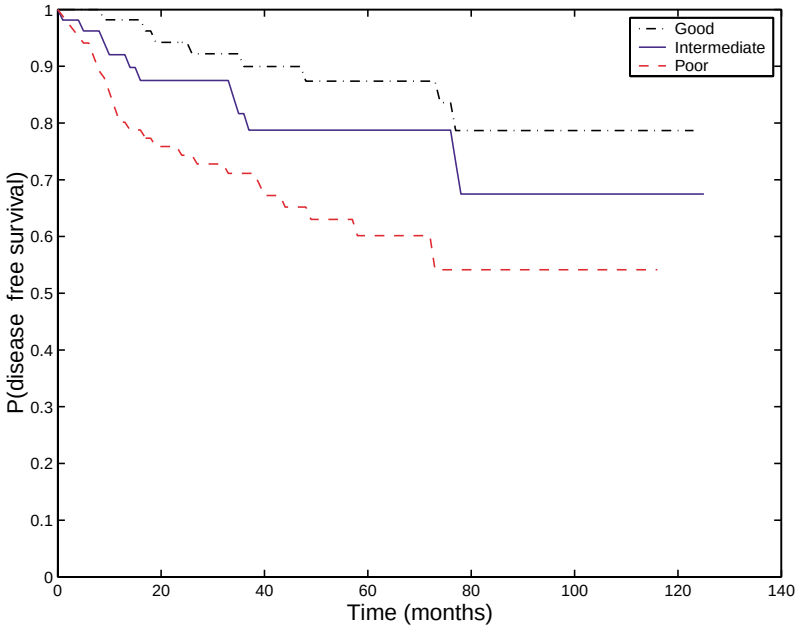

Fig. 13. Estimated survival curves for the three groups found by ELSA/EM.

– A number of possibly conflicting heuristic metrics can be plugged into the algorithm, while remaining agnostic about their relative worth or their relationships.
– Our algorithm, both ELSA/K-means and ELSA/EM, outperforms a greedy algorithm in terms of classification accuracy. On the artificial data, the number of win-lose-tie cases of ELSA/K-means and ELSA/EM compared to a greedy algorithm were 17-3-4 and 18-2-4, respectively.
– Most importantly, in the proposed framework we can reliably select an appropriate clustering model, including significant features and the number of clusters. For example, ELSA/K-means correctly identifed the number of clusters $K = 5$ based on F_{within}, and the composition of significant-Gaussian noise-white noise features were 6-3-0, 5-4-1, 7-3-1, and 8-3-1 when 9–12 features were selected.
– Our algorithm is more interpretable and scalable due to the reduced dimensionality. On the WPBC data, ELSA/K-means and ELSA/EM (with 11 and 7 chosen features, respectively) showed well-separated survival characteristics of for three different groups of patients.

In future work we would like to compare the performance of ELSA on the unsupervised feature selection task with other multi-objective EAs, using each in conjunction with clustering algorithms.

Another promising future direction will be a direct comparison of different clustering algorithms. In the results presented in this paper, ELSA/EM shows better results than ELSA/K-means on the synthetic data in terms of the composition of selected features and prediction accuracy. Furthermore, EM allows for easier choice of best compromise solution because of single quality metric. However, ELSA/K-means shows very competitive results on the real data set in terms of well-separated survival curves. Further, ELSA/K-means is much faster (roughly by a factor of 3) than ELSA/EM to evaluate the fixed number of solutions. Thus, it is possible for ELSA/K-means to find better solutions given the same amount of computing time as ELSA/EM.

From a knowledge discovery perspective, our algorithm offers several advantages. Certainly the simplicity bias of Occam's Razor is well-established as a means for improving generalization on real-world data sets. Further, it is often the case that the user can gain insight into the problem domain by

finding the set of relevant features; consider, for example, the problem of finding relevant prognostic factors in breast cancer, or determining the variables that define relevant market segments.

Finally, a key problem in data mining is the scaling of predictive methods to large data sets. Our algorithm can easily be used as a preprocessing step to determine an appropriate set of features, allowing the application of iterative algorithms like K-means on much larger problems.

Acknowledgments

This work was supported in part by NSF grant IIS-99-96044.

References

[1] R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, in: *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data* 1998, pp. 94–105, Seattle, WA.
[2] E.I. Altman, *Corporate Financial Distress and Bankruptcy: A Complete Guide to Predicting and Avoiding Distress and Profitting from Bankruptcy*, John Wiley and Sons, New York, NY, 1993.
[3] G.P. Babu and M.N. Murty, A near-optimal initial seed value selection in K-means algorithm using a genetic algorithm, *Pattern Recognition Letters* **14**(10) (1993), 763–769.
[4] P.V.S. Balakrishnan, M.C. Cooper, V.S. Jacob and P.A. Lewis. Comparative performance of the FSCL neural net and K-means algorithm for market segmentation, *European Journal of Operation Research* **93**(10) 1996), 346–357.
[5] A. Blumer, A. Ehrenfeucht, D. Haussler and M.K. Warmuth, Occam's razor, *Information Processing Letters* **24** (1987), 377–380.
[6] L. Bottou and Y. Bengio, Convergence properties of the K-means algorithm, in: *Advances in Neural Information Processing Systems 7*, M.C. Mozer, M.I. Jordan and T. Petsche, eds, MIT Press, 1995.
[7] P.S. Bradley, U.M. Fayyad and C. Reina, Scaling clustering algorithms to large databases, in: *Proc. of 4th Int'l Conf. on Knowledge Discovery and Data Mining* (*KDD98*), R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, eds, 1998, pp. 9–15, Menlo Park, CA, AAAI Press.
[8] P.S. Bradley, O.L. Mangasarian and W.N. Street, Clustering via concave minimization, in: *Advances in Neural Information Processing Systems 9*, M.C. Mozer, M.I. Jordan and T. Petsche, eds, 1997, pp. 368–374, MA: Cambridge, MIT Press.
[9] P. S. Bradley, O. L. Mangasarian and W. N. Street. Feature selection via mathematical programming. *INFORMS Journal on Computing* **10**(2):209–217, 1998.
[10] J. Buhmann, Data clustering and learning, in: *Handbook of Brain Theory and Neural Networks*, M. Arbib, ed., Bradfort Books/MIT Press, 1995.
[11] P. Cheeseman and J. Stutz, Bayesian classification system (AutoClass): Theory and results, in: *Advances in Knowledge Discovery and Data Mining*, U. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, eds, 1996, pp. 153–180, San Francisco, CA, MIT Press.
[12] S. Chen, C. Guerra-Salcedo and S. Smith, Non-standard crossover for a standard representation – commonality-based feature subset selection, in: *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, 1999.
[13] M. Dash and H. Liu, Feature selection for classification. *Intelligent Data Analysis* **1**(3) (1997), 131–156.
[14] D.L. Davies and D.W. Bouldin, A cluster separation measure, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-1**(2) (1979), 224–227.
[15] K. Deb and J. Horn, Special issue on multi-criterion optimization,*Evolutionary Computation Journal* **8**(2) (2000).
[16] A.P. Dempster, N.M. Laird and D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society, Series B* **39**(1) (1977), 1–38.
[17] M. Devaney and A. Ram, Efficient feature selection in conceptual clustering, in: *Proc. 14th Int'l Conf. on Machine Learning*, 1997, pp. 92–97, Morgan Kaufmann.
[18] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.
[19] J.G. Dy and C.E. Brodley, Feature subset selection and order identification for unsupervised learning, in: *Proc. 17th Int'l Conf. on Machine Learning*, 2000, pp. 247–254, Morgan Kaufmann, San Francisco, CA.
[20] J.G. Dy and C.E. Brodley, Visualization and interactive feature selection for unsupervised data, in: *Proc. 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining* (*KDD-00*), 2000, pp. 360–364, 2000.
[21] D. Fisher, COBWEB: Knowledge acquisition via conceptual clustering, *Machine Learning* **2** (1987), 139–172.

[22] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, CA, 1990.
[23] C. Glymour, D. Madigan, D. Pregibon and P. Smyth, Statistical themes and lessons for data mining. *Data Mining and Knowledge Discovery* **1**(1) (1997).
[24] D.E. Goldberg and J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: *Proc. of 2nd Int'l Conf. on Genetic Algorithms*, 1987.
[25] P.E. Green and V. Srinivasan, Conjoint analysis in marketing: New developments with implications for research and practice, *Journal of Marketing* **54**(4) (1990), 3–19.
[26] S. Guha, R. Rastogi and K. Shim, CURE: An efficient clustering algorithm for large databases, in: *Proc. of ACM SIGMOD Int'l Conf. on Management of Data* (*SIGMOD98*), 1998, pp. 73–84.
[27] L.O. Hall, I.B. Ozyurt and J.C. Bezdek, Clustering with a genetically opitmized approach, *IEEE Transactions on Evolutionary Computation* **3**(2) (1999), 103–112.
[28] J.A. Hartigan, *Clustering Algorithms*. Wiley, New York, 1975.
[29] J. Horn, Multicriteria decision making and evolutionary computation, in: *Handbook of Evolutionary Computation*, Institute of Physics Publishing, London, 1997.
[30] H. Ishibuchi and T. Nakashima, Multi-objective pattern and feature selection by a genetic algorithm, in: *Proc. of Genetic and Evolutionary Computation Conference* (*GECCO'2000*), D. Whitley, D. Goldberg, E. Cant-Paz, L. Spector, I. Parmee and H.-G. Beyer, eds, 2000, pp. 1069–1076.
[31] R.A. Johnson and D.W. Wichern, *Applied Multivariate Statistical Analysis*, Prentice Hall, New Jersey, 3 edition, 1992.
[32] E.L. Kaplan and P. Meier, Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association* **53** (1958), 457–481.
[33] Y. Kim and W.N. Street, CoIL challenge 2000: Choosing and explaining likely caravan insurance customers. Technical Report 2000-09, Sentient Machine Research and Leiden Institute of Advanced Computer Science, June 2000. http://www.wi.leidenuniv.nl/ putten/library/cc2000/.
[34] Y. Kim, W.N. Street and F. Menczer, Feature selection in unsupervised learning via evolutionary search, in: *Proc. of 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining* (*KDD-00*), 2000, pp. 365–369.
[35] Y. Kim, W.N. Street and F. Menczer, An evolutionary multi-objective local selection algorithm for customer targeting, in: *Proc. of Congress on Evolutionary Computation* (*CEC-01*), accepted.
[36] J. Kittler, Feature selection and extraction, in: *Handbook of Pattern Recognition and Image Processing*, Y. Fu, ed., New York, 1986, Academic Press.
[37] R. Kohavi and G.H. John, Wrappers for feature subset selection, *Artificial Intelligence* **97**(1-2) (1997), 273–324.
[38] P. Kontkanen, P. Myllymaki and H. Tirri, Comparing Bayesian model class selection criteria by discrete finite mixtures. in: *Proc. of Information, Statistics and Induction in Science Conference*(*ISIS'96*) *in Melbourne, Australia*, D. Dowe, K. Korb and J. Oliver, eds, 1996, pp. 364–374, World Scientific, Singapore.
[39] K. Krishna and M.N. Murty, Genetic K-means algorithm. *IEEE Transactions on Systems, Man and Cybernetics - PartB: Cybernetics* **29**(3) (1999), 433–439.
[40] M. Kudo and J. Sklansky, Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition* **33** (2000), 25–41.
[41] O.L. Mangasarian, W.N. Street and W.N. Wolberg, Breast cancer diagnosis and prognosis via linear programming, *Operations Research* **43**(4) (July-August, 1995), 570–577.
[42] M. Meila and D. Heckerman, An experimental comparison of several clustering methods. Technical Report MSR-TR-98-06, Microsoft, Redmond, WA, 1998.
[43] F. Menczer and R.K. Belew, From complex environments to complex behaviors. *Adaptive Behavior* **4** (1996), 317–363.
[44] F. Menczer and R.K. Belew, Local selection, in: *Evolutionary Programming VII, LNAI 1447*, V.W. Porto, N. Saravanan, D. Waagen and A.E. Eiben, eds, Berlin, 1998, Springer Verlag.
[45] F. Menczer, M. Degeratu and W.N. Street, Efficient and scalable pareto optimization by evolutionary local selection algorithms. *Evolutionary Computation* **8**(2) (Summer 2000), 223–247.
[46] D. Opitz, Feature selection for ensembles, in: *16th National Conf. on Artificial Intelligence* (*AAAI*), 1999, pp. 379–384, Orlando, FL.
[47] Y.-J. Park and M.-S. Song, A genetic algorithm for clustering problems, in: *Proc. of 3rd Annual Conf. on Genetic Programming*, 1998, pp. 568–575.
[48] C.M. Schaffer and P.E. Green, Cluster-based market segmentation: Some further comparisons of alternative approaches, *Journal of the Market Research Society* **40**(2) (April, 1998), 155–163.
[49] P. Smyth, Clustering using Monte Carlo cross-validation, in: *Proc. of 2nd Int'l Conf. on Knowledge Discovery and Data Mining* (*KDD-96*), 1996, pp. 126–133.
[50] W.N. Street, O.L. Mangasarian and W.H. Wolberg, An inductive learning approach to prognostic prediction, in: *Proc. of 12th Int'l Conf. on Machine Learning*, A. Prieditis and S. Russell, eds, 1995, pp. 522–530, San Francisco, Morgan Kaufmann.

[51] L. Talavera, Feature selection as a preprocessing step for hierarchical clustering, in: *Proc. 16th Int'l Conf. on Machine Learning*, 1999, pp. 389–397, Morgan Kaufmann, San Francisco, CA.
[52] S. Vaithyanathan and B. Dom, Model selection in unsupervised learning with applications to document clustering, in: *Proc. 16th Int'l Conf. on Machine Learning*, 1999, pp. 433–443, Morgan Kaufmann, San Francisco, CA.
[53] M. Wedel and W.A. Kamakura, *Market Segmentation: Conceptual and Methodological Foundations*, chapter 6–7, Kluwer Academic Publishers, 2 edition, 1999.
[54] J. Yang and V. Honavar, Feature subset selection using a genetic algorithm, *IEEE Intelligent Systems and their Applciations* **13**(2) (1998), 44–49.
[55] T. Zhang, R. Ramakrishnan and M. Livny, BIRCH: A new data clustering algorithm and its applications, *Data Mining and Knowledge Discovery* **1**(2) (1997), 141–182.