# Feature Selection in Unsupervised Learning via Evolutionary Search

YongSeog Kim
Management Sciences Dept.
University of Iowa
Iowa City, IA 52242 USA
yong-s-kim@uiowa.edu

W. Nick Street
Management Sciences Dept.
University of Iowa
Iowa City, IA 52242 USA
nick-street@uiowa.edu

Filippo Menczer
Management Sciences Dept.
University of Iowa
Iowa City, IA 52242 USA
filippo-menczer@uiowa.edu

## ABSTRACT

Feature subset selection is an important problem in knowledge discovery, 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. In this paper we consider the problem of feature selection for unsupervised learning. 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; a standard K-means algorithm is applied to form the given number of clusters based on the selected features. Preliminary results on both real and synthetic data show promise in finding Pareto-optimal solutions through which we can identify the significant features and the correct number of clusters.

## Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database Management—*Database Applications*[Data Mining]

## General Terms

Feature selection, evolutionary search, clustering

## 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. We adopt the wrapper model [12] of feature selection which requires two components: a search algorithm that explores the combinatorial space of feature subsets, and one or more criterion functions that evaluate the quality of each subset based directly on the predictive model. Most feature selection research has focused on heuristic search approaches, such as sequential search [13], nonlinear optimization [5], and genetic algorithms (GAs) [17]. A recent review of these methods can be found in [6]. These methods considered feature selection in a supervised learning context, evaluating potential solutions in terms of predictive accuracy. We instead wish to find natural grouping of the examples in the feature space via *clustering* or *unsupervised learning*. Clustering may be performed using methods such as K-means [10], expectation maximization (EM) [9], or optimization models [4]. We use the standard K-means algorithm with each solution's selected subset of features. Recent research has focused on forming clusters in large data sets [3, 1]. We take the view that an effective way to scale a clustering algorithm is to reduce the dimensionality of the data by using a subset of the points to select a subset of the features.

A number of heuristic criteria, such as cluster compactness and inter-cluster separation, have been used to estimate the quality of the clusters, and attempts have been made to combine these into a single objective [7]. This is a difficult problem to solve in the general case, since each data set has it's own characteristics and each decision maker has her own priorities. In such situations we must use *multi-objective* or *Pareto* optimization. Informally, a solution is said to *dominate* another if it has higher values along all the objective functions. We define the *Pareto front* as the set of nondominated solutions. The goal 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.

We use evolutionary algorithms (EAs) to intelligently search the space of possible feature subsets (and values of $K$). Standard EAs assume a single fitness function to be optimized. A number of multi-objective extensions of evolutionary algorithms have been proposed in recent years [8]. Most of them, such as the Niched Pareto Genetic Algorithm [11], employ computationally expensive selection mechanisms to favor dominating solutions and to maintain diversity. 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) works well for Pareto optimization problems [16].

In Section 2 we discuss our approach in detail, justifying our heuristic clustering quality metrics, illustrating the evolutionary algorithm, and describing how ELSA is combined with K-means. Section 3 presents some experiments with synthetic and real data sets, and discusses the interpretation of the ELSA output to select a subset of good features. Section 4 concludes the paper.

# 2. FEATURE SELECTION ALGORITHM

## 2.1 Heuristic metrics for clustering

Most measurements to evaluate cluster quality are based on geometric distance metrics and are therefore not directly applicable because they are biased by the dimensionality of the space, which is variable in feature selection problems. 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 [2]. Each objective is normalized into the unit interval and maximized by the EA.

**$F_{within}$ :** 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 $x_i, i = 1, \cdots, n$, be data points and $x_{ij}$ be the value of the $j$-th feature of $x_i$. Let $d$ be the dimension of the *selected* feature set, $J$, and $K$ be the number of clusters. Define the cluster membership variables $\alpha_{ik}$ as follows:

$$\alpha_{ik} = \begin{cases} 1 & \text{if } x_i \text{ belongs to cluster } k \\ 0 & \text{otherwise} \end{cases}$$

where $k = 1, \cdots, K$ and $i = 1, \cdots, n$. The centroid of the $k$-th cluster, $\gamma_k$, is defined by its coordinates:

$$\gamma_{kj} = \frac{\sum_{i=1}^{n} \alpha_{ik} x_{ij}}{\sum_{i=1}^{n} \alpha_{ik}}, \; j \in J.$$

$F_{within}$ can finally be computed:

$$F_{within} = 1 - \frac{1}{Z_W} \frac{1}{d} \sum_{k=1}^{K} \sum_{i=1}^{n} \alpha_{ik} \sum_{j \in J} (x_{ij} - \gamma_{kj})^2$$

where the normalization by the number of selected features $d$ compensates for the dependency of the distance metric on the feature subspace dimensionality. $Z_W$ 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:

$$F_{between} = \frac{1}{Z_B} \frac{1}{d} \frac{1}{k-1} \sum_{k=1}^{K} \sum_{i=1}^{n} (1 - \alpha_{ik}) \sum_{j \in J} (x_{ij} - \gamma_{kj})^2$$

where, as for $F_{within}$, we normalize by the dimensionality of the selected feature subspace and by the empirically derived, data-dependent constant $Z_B$.

```
initialize $p_{max}$ agents, each with energy $\theta/2$
while there are alive agents and for $T$ iterations
  for each energy source $c$ (1 .. $C$)
    for each $v$ (0 .. 1)
      $E^c_{envt}(v) \leftarrow 2vp_{max}E_{cost}/C$
    endfor
  endfor
  for each agent $a$
    $a' \leftarrow mutate(clone(a))$
    for each energy source $c$ (1 .. $C$)
      $v \leftarrow F_c(a')/P_c(F_c(a'))$
      $\Delta E \leftarrow \min(v, E^c_{envt}(v))$
      $E^c_{envt}(v) \leftarrow E^c_{envt}(v) - \Delta E$
      $E_a \leftarrow E_a + \Delta E$
    endfor
    $E_a \leftarrow E_a - E_{cost}$
    if ($E_a > \theta$)
      insert $a'$ into population
      $E_{a'} \leftarrow E_a/2$
      $E_a \leftarrow E_a - E_{a'}$
    else if ($E_a < 0$)
      remove $a$ from population
    endif
  endfor
endwhile
```

**Figure 1: ELSA pseudo-code. See text for details.**

**$F_{clusters}$ :** Other things being equal, fewer clusters make the model more understandable and avoid possible overfitting. We implement this with the criterion

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

where $K_{max}$ ($K_{min}$) is the maximum (minimum) number of clusters the user chooses to consider.

**$F_{complexity}$ :** This objective is aimed at minimizing the number of selected features:

$$F_{complexity} = 1 - \frac{d-1}{D-1},$$

where $D$ is the dimensionality of the full data set. Again, we expect that lower complexity will lead to easier interpretability of solutions as well as better generalization.

## 2.2 Evolutionary local selection algorithm

ELSA springs from algorithms originally motivated by artificial life models of adaptive agents in ecological environments [15]. In these models an agent's fitness results from individual interactions with the environment, which contains other agents as well as finite shared resources. A more extensive discussion of the algorithm and its application to Pareto optimization problems can be found elsewhere [16]. Figure 1 outlines the ELSA algorithm.

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.[1] This representation is motivated by the desire to preserve the regularity of the number of clusters under the mutation operator. Mutation is the only genetic operator used to explore the search space in these experiments.

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

The environment corresponds to the set of possible values for each of the criteria being optimized.[2] We have an energy source for each criterion, divided into bins corresponding to its values. When the environment is replenished, each criterion is allocated an equal share of energy, apportioned in proportion to the fitness values in order to bias the population toward more promising areas in objective space. Note that the total replenishment energy that enters the system at each iteration is such that we can maintain a population size of $p_{max}$ on average.

In each iteration of the algorithm, an agent explores a candidate solution similar to itself; it is rewarded with some energy from the environment and taxed with a constant cost. To compute the energy intake of an agent, for each objective function, the environment scales the agent's fitness value by the number of agents sharing the corresponding bin. Candidate solutions receive energy only inasmuch as the environment has sufficient resources; if these are depleted, no benefits are available until the environment is replenished. Thus an agent is rewarded with energy for its high objective 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 to decide whether the mutated clone that was just evaluated should become part of the population. If an agent runs out of energy, it is killed.

In order to assign energy to a solution based on the fitness criteria, ELSA must form the given number of clusters based on the selected features. In the experiments described here, the clusters to be evaluated are constructed using a standard K-means algorithm. 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, repeating until no points are reassigned. 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 K-means algorithm is given the projection of the data set onto $J$, uses it to form $K$ clusters, and returns the four fitness criteria $F_{within}$, $F_{between}$, $F_{clusters}$, and $F_{complexity}$.

## 3. EVALUATION

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. Therefore we take a gradual approach to evaluate the proposed method. First, we use a small-dimensional synthetic data set with well-defined distributions and clusters along each feature dimension. This allows us to determine whether the given solutions formed by ELSA represent a sensible compromise between the conflicting heuristic quality objectives. Second, we use a high-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 pre-constructed clusters in a five-dimensional subspace. Finally, we use a real data set for which we have knowledge about the clusters and the relevant features. In this case, we can evaluate the solutions both by examining the selected features and by judging the semantics of the resulting clusters.

For further comparisons we have implemented a greedy heuristic algorithm known as two-way sequential selection [13]. This algorithm requires a set value of $K$ and uses $F_{within}$ as the optimization criterion. 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 repeated the algorithm for the same values of $K$ considered by ELSA.

### 3.1 Experiment 1

The first synthetic data set has $n = 300$ points and $D = 6$ features, and is constructed as follows. One cluster is formed along feature 1 and two clusters are formed randomly along feature 2. Along feature 3, we randomly reassign the points to two independent clusters. We repeat the process for feature 4. Finally, for features 5 and 6, the points are distributed uniformly. All the clusters are formed by generating points from a pseudo-Gaussian distribution with standard deviation $\sigma \approx 0.06$. Figure 2 illustrates this data set by projecting the points onto some of the feature subspaces with $d = 2$. The motivation for this data set is to have an understanding of the relationships between the different features, and at the same time a realistic mixture of significant, less significant, and insignificant features.

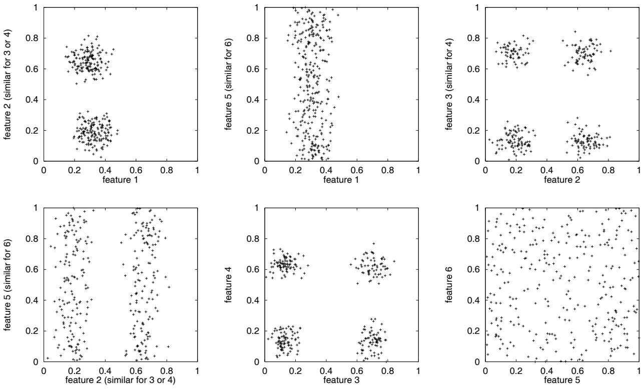

**Figure 2: Some 2-dimensional projections of the data set of Experiment 1.**

The individuals are represented by strings with 12 bits, 6 for the features and 6 for $K$, so that $K_{max} = 8$. There are 7 energy bins for $F_{clusters}$, 6 for $F_{complexity}$, and 10 each for $F_{within}$ and $F_{between}$. The values for the various ELSA parameters are: $\Pr(mutation) = 0.1$ (per bit), $p_{max} = 100$, $E_{cost} = 0.2$, $\theta = 0.3$, and $T = 40{,}000$.

The best solution with four clusters in more than one dimen-

[2] Here, $C = 4$ criteria; continuous objectives are discretized.

sion included features 2 and 3. The best solution with $K = 2$ and more than one dimension included features 1 and 4. As depicted in Figure 2, both of these solutions describe the data very well. The final population was dominated by solutions with one feature, which typically look extremely good along two criteria: complexity, and either $F_{within}$ (many centers inside one true cluster) or $F_{between}$ (well-separated centers along a random dimension).

As expected, the greedy search method performed very well on this simple data set. With $K = 2$, features were added in the order 1, 3, 2, 4, 5, 6; with $K = 4$, the order was 1, 3, 4, 2, 5, 6. As it happens, the two-dimensional clusters along features 1 and 3 are somewhat better (in terms of $F_{within}$) than those along features 1 and 2.

## 3.2 Experiment 2

With the second data set we pose a problem with higher dimensionality, retaining the realistic flavor of the smaller data set. We again have some "significant" features (in which points belong to correlated normal clusters), some "Gaussian noise" features (in which values are drawn from single or bimodal normal distributions along each dimension, but the distributions along different features are uncorrelated), and some "white noise" features (in which points are drawn from uniform distributions). The data set has $n = 500$ points and $D = 30$ features. It is constructed so that the first 10 features are significant, with 5 "true" 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. 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.

Individuals are represented by 38 bits, 30 for the features and 8 for $K$ ($K_{max} = 10$). There are 9 bins for $F_{clusters}$ and 10 each for $F_{complexity}$, $F_{within}$, and $F_{between}$. The parameters for ELSA are the same as those used in Experiment 1, except that T = 800,000 iterations.

| | ELSA | | | | | Greedy | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $K$<br>$d$ | 2 | 3 | 4 | 5 | 6 | 2 | 3 | 4 | 5 | 6 |
| 2 | – | 68 | 71 | 24 | – | 56 | 40 | 29 | 24 | 24 |
| 3 | – | 62 | 74 | 62 | – | 57 | 39 | 29 | 25 | 51 |
| 4 | – | 40 | 34 | 44 | – | 58 | 40 | 31 | 43 | 76 |
| 5 | – | – | 53 | 61 | 38 | 57 | 38 | 76 | 70 | 80 |
| 6 | – | – | – | 35 | – | 100 | 67 | 99 | 80 | 100 |

**Table 1: Classification performance (%) of various solutions found by ELSA and by greedy search. The "–" entries indicate that no solution with those parameters existed in the final ELSA population.**

Table 1 shows the classification accuracy of various models formed by both ELSA and the greedy feature search. 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 from the final population. ELSA consistently outperforms the greedy search on models with few features, exactly the sort of models the algorithm was designed to find. For more complex models, the greedy method is better able to reconstruct the original classes. This is reasonable, since ELSA does not concentrate on this part of the search space.

## 3.3 Experiment 3

In addition to the artificial data sets discussed above, we also test our algorithm on a real data set, the Wisconsin Prognostic Breast Cancer (WPBC) data [14]. This data set records 30 numeric features quantifying the nuclear grade of breast cancer patients at the University of Wisconsin Hospital, along with traditional prognostic variables tumor size and number of positive lymph nodes, and a binary variable indicating whether lymph status was recorded. This results in a total of 33 features for each of 227 cases.

Individuals are represented by 37 bits, 33 for the features and 4 for $K$ ($K_{max} = 6$), therefore there are 5 bins for $F_{clusters}$. Other ELSA parameters are the same as those used in Experiment 1 except that T = 10,000.

We analyze performance on this data set by looking for clinical relevance in the resulting clusters. We chose a solution with three clusters in 7 dimensions by picking the best individual (in terms of $F_{between}$ and $F_{within}$) with 3 clusters from the final population. We used the actual outcomes (time to recurrence, or known disease-free time) of the cases in each cluster to form the Kaplan-Meier maximum likelihood estimate of the true disease-free survival curve. The three groups displayed well-separated survival characteristics. Ten-year recurrence rates were 17.8%, 26.9%, and 45.6% for the patients in the three groups. Because of its small size (22 cases, 3 recurrences), the best prognostic group was not statistically significantly different from the intermediate group (p = .075). The intermediate group was well-differentiated from the poor group ($p < 0.01$).

The chosen dimensions included a mix of nuclear morphometric features such as symmetry, concavity and texture, along with lymph status and tumor size. We note that the inclusion of lymph status requires dissection of the ancillary nodes for staging purposes, leaving the patient at risk for painful complications. While we would prefer to make treatment decisions without this feature, the clustering results consistently indicated that it was relevant to the forming of prognostic groups.

# 4. CONCLUSIONS

We presented a novel approach for large-scale feature selection problems using unsupervised learning. ELSA, an evolutionary local selection algorithm, was used successfully in previous work in conjunction with supervised learning [16]. In this paper we used ELSA to search for possible combination of features and numbers of clusters, with the guidance of the K-means algorithm. While the search biases of ELSA and K-means may not be ideal for this application (a topic to be explored in future work), the combination of a multi-objective search algorithm with unsupervised learning provides a promising framework for feature selection. We summarize our findings as follows. *i*) ELSA covers a large space of possible feature combinations well while simultaneously optimizing the multiple criteria. *ii*) The standard K-means algorithm can be used to guide ELSA by evaluating the quality of a subset of features. *iii*) A number of possibly conflicting heuristic metrics can be plugged into the algorithm, while remaining agnostic about their relative worth or their relationships. *iv*) Most importantly, in the proposed framework we can select significant feature subsets without training examples, while at the same time identifying the inherent numbers of clusters.

In future work we will further analyze the interactions among our various optimization criteria. For instance, increasing the number of features dramatically affects both of our cluster quality metrics. We corrected for much of this with normalization terms, but further study is needed to decorrelate the effects of the various criteria. Further, well-separated but nearby clusters are judged harshly by the traditional TBSS measure on which $F_{between}$ is based. We will explore other objectives that implement the idea of forming well-separated clusters. Interactions among different features can also be studied in ELSA, by employing genetic operators such as crossover.

It would often be desirable to identify one single solution from the estimated Pareto front representing a "best compromise." Once the algorithm has identified a set of candidate solutions we might be able to apply some more expensive statistical or geometric method. For example, we might look along the approximate Pareto front for a point of maximal curvature.

From a knowledge discovery perspective, our algorithm offers several advantages. Certainly the simplicity bias of Occam's Razor is a well-established 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 such problems as prognostic factors in breast cancer, target marketing, or genetic analysis. 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 (and number of clusters), allowing the application of iterative algorithms like K-means on much larger problems.

## 5. ACKNOWLEDGMENTS

We are grateful to Melania Degeratu for developing the original ELSA software and for helpful discussions. This work was supported in part by NSF grant IIS-99-96044.

## 6. 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*, Seattle, WA, 1998.

[2] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987.

[3] P. S. Bradley, U. M. Fayyad, and C. Reina. Scaling clustering algorithms to large databases. In R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, editors, *Proc. 4th Int'l Conf. on Knowledge Discovery and Data Mining*, pages 9–15, Menlo Park, CA, 1998. AAAI Press.

[4] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Clustering via concave minimization. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 368–374, MA: Cambridge, 1997. MIT Press.

[5] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Feature selection via mathematical programming. *INFORMS Journal on Computing*, 10(2):209–217, 1998.

[6] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(3):131–156, 1997.

[7] D. Davies and D. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[8] K. Deb and J. Horn. Special issue on multi-criterion optimization. *Evolutionary Computation Journal*, 8(2), 2000.

[9] 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):1–38, 1977.

[10] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[11] J. Horn. Multicriteria decision making and evolutionary computation. In *Handbook of Evolutionary Computation.* Institute of Physics Publishing, London, 1997.

[12] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proc. 11th Int'l Conf. on Machine Learning*, San Mateo, CA, 1994. Morgan Kaufmann.

[13] J. Kittler. Feature selection and extraction. In Y. Fu, editor, *Handbook of Pattern Recognition and Image Processing*, New York, 1978. Academic Press.

[14] O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, July-August 1995.

[15] F. Menczer and R. K. Belew. Latent energy environments. In R. K. Belew and M. Mitchell, editors, *Adaptive Individuals in Evolving Populations: Models and Algorithms.* Addison Wesley, Reading, MA, 1996.

[16] F. Menczer, M. Degeratu, and W. Street. Efficient and scalable pareto optimization by evolutionary local selection algorithms. *Evolutionary Computation*, 8(2):223–247, Summer 2000.

[17] J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. In H. Motada and H. Liu, editors, *Feature Extraction, Construction, and Subset Selection: A Data Mining Perspective.* Kluwer, New York, 1998.