Feature Selection in Data Mining

YongSeog Kim, W. Nick Street, and Filippo Menczer, University of Iowa, USA

INTRODUCTION

Feature selection has been an active research area in pattern recognition, statistics, and data mining communities. The main idea of feature selection is to choose a subset of input variables by eliminating features with little or no predictive information. Feature selection can significantly improve the comprehensibility of the resulting classifier models and often build a model that generalizes better to unseen points. Further, it is often the case that finding the correct subset of predictive features is an important problem in its own right. For example, physician may make a decision based on the selected features whether a dangerous surgery is necessary for treatment or not.

Feature selection in supervised learning has been well studied, where the main goal is to find a feature subset that produces higher classification accuracy. Recently, several researches (Dy and Brodley, 2000b, Devaney and Ram, 1997, Agrawal *et al.*, 1998) have studied feature selection and clustering together with a single or unified criterion. For feature selection in unsupervised learning, learning algorithms are designed to find natural grouping of the examples in the feature space. Thus feature selection in unsupervised learning aims to find a good subset of features that forms high quality of clusters for a given number of clusters.

However, the traditional approaches to feature selection with single evaluation criterion have shown limited capability in terms of knowledge discovery and decision support. This is because decision-makers should take into account multiple, conflicted objectives simultaneously. In particular no single criterion for unsupervised feature selection is best for every application (Dy and Brodley, 2000a) and only the decision-maker can determine the relative weights of criteria for her application. In order to provide a clear picture of the (possibly nonlinear) tradeoffs among the various objectives, feature selection has been formulated as a multi-objective or Pareto optimization.

In this framework, we evaluate each feature subset in terms of multiple objectives. Each solution s_i is associated with an evaluation vector $F = F_1(s_i),..., F_C(s_i)$ where C is the number of quality criteria. One solution s_1 is said to *dominate* another solution s_2 if $\forall c$: $F_C(s_1) \geq F_C(s_2)$ and $\exists c$: $F_C(s_1) > F_C(s_2)$, where F_C is the c-th criterion, $c \in \{1,...,C\}$. Neither solution dominates the other if $\exists c_1,c_2$: $F_{C1}(s_1) > F_{C2}(s_2)$, $F_{C2}(s_2) > F_{C2}(s_1)$. We define the *Pareto front* as the set of nondominated solutions. In feature selection as a Pareto optimization, the goal is to approximate as best possible the Pareto front, presenting the decision maker with a set of high-quality solutions from which to choose.

We use Evolutionary Algorithms (EAs) to intelligently search the space of possible feature subsets. A number of multi-objective extensions of EAs have been proposed (VanVeldhuizen, 1999) to consider multiple fitness criteria effectively. However, most of them employ computationally expensive selection mechanisms to favor dominating solutions and to maintain diversity, such as Pareto domination tournaments (Horn, 1997) and fitness sharing (Goldberg and Richardson, 1987). We propose a new algorithm, Evolutionary Local Selection Algorithms (ELSA), where an individual solution is allocated to a local environment based on its criteria values and competes with others to consume shared resources only if they are located in the same environment.

The remainder of the chapter is organized as follows. We first introduce our search algorithm, ELSA. Then we discuss the feature selection in supervised and unsupervised learning, respectively. Finally, we present a new two-level evolutionary environment, Meta-Evolutionary Ensembles (MEE) that uses feature selection as the mechanism for boosting diversity of a classifier in an ensemble.

EVOLUTIONARY LOCAL SELECTION ALGORITHMS (ELSA)

Agents, Mutation and Selection

ELSA springs from artificial life models of adaptive agents in ecological environments (Menczer and Belew, 1996). In ELSA, an agent may die, reproduce, or neither based on an endogenous energy level that fluctuates via interactions with the environment. Figure 1 outlines the ELSA algorithm at a high level of abstraction.

```
initialize population of agents, each with energy $\theta/2$
while there are alive agents and for $T$ iterations
    for each energy source $c$
        for each $v$ (0 .. 1)
            $E_{envt}^{c}(v) \leftarrow 2 v E_{tot}^{c}$;
    for each agent $a$
        $a' \leftarrow mutate(crossover(a, random\ mate))$;
        for each energy source $c$
            $v \leftarrow Fitness(a',c)$;  $\Delta E \leftarrow min(v, E_{envt}^{c}(v))$;
            $E_{envt}^{c}(v) \leftarrow E_{envt}^{c}(v) - \Delta E$;  $E_a \leftarrow E_a + \Delta E$;
        $E_a \leftarrow E_a - E_{cost}$;
        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;
endwhile
```

Figure 1: ELSA pseudo-code.

The representation of an agent consists of D bits and each of D bits is an indicator as to the corresponding feature is selected or not (1 if a feature is selected, 0 otherwise). Each agent is first initialized with some random solution and an initial reservoir of *energy*, and competes for a scare resource, energy, based on multi-dimensional fitness and the proximity of other agents in solution space. The mutation operator randomly selects one bit of the agent and flips it. Our commonality-based crossover operator (Chen *et al.*, 1999) makes the offspring inherit all the common features of the parents.

In the selection part of the algorithm, each agent compares its current energy level with a constant reproduction threshold θ. If its energy is higher than θ, 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 θ, only the agent itself joins the new population. If an agent runs out of energy, it is killed. The population size is maintained dynamically over iterations and is determined by the carrying capacity of the environment depending on the costs incurred by any action, and the replenishment of resources (Menczer *et al.*, 2000b).

Energy Allocation and Replenishment

In each iteration of the algorithm, an agent explores a candidate solution similar to itself. The agent collects ΔE from the environment and is taxed with E_{cost} for this action. The net energy intake of an agent is determined by its offspring's fitness and the state of the environment that corresponds to the set of possible values for each of the criteria being optimized.[1] 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 $E_{envt}^{c}(v)$ corresponding to the value $F_c = v$. Further, the environment keeps a count of the number of agents $P_c(v)$ having $F_c = v$. The energy corresponding to an action (alternative solution) a for criterion F_c is given by

$$Fitness(\mathrm{a}, c) = F_c(\mathrm{a}) \,/\, P_c(F_c(\mathrm{a})). \tag{1}$$

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. Ecost for any action is a constant (Ecost < θ). When the environment is replenished with energy, each criterion c is allocated an equal share of energy as follows:

$$E_{tot}^{c} = p_{max} E_{cost} / C \tag{2}$$

where C is the number of criteria considered. 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.

Advantages and Disadvantages

One of the major advantages of ELSA is its minimal centralized control over agents. By relying on local selection, ELSA minimizes the communication among agents, which makes the algorithm efficient in terms of computational time and scalability (Menczer *et al.*, 2000a). Further, the local selection naturally enforces the diversity of the population by evaluating agents based on both their quality measurements and the number of similar individuals in the neighborhood in objective space. Note also that ELSA can be easily combined with any predictive and clustering models.

In particular, there is no upper limit of number of objective functions that ELSA can accommodate. Noting that no single criterion is best for every application, we consider all (or at least some) of them simultaneously in order to provide a clear picture of the (possibly nonlinear) tradeoffs among the various objectives. The decision-maker can select a final model after determining her relative weights of criteria for application.

ELSA can be useful for various tasks in which the maintenance of diversity within the population is more important than a speedy convergence to the optimum. Feature selection is one such promising application. Based on the well-covered range of feature vector complexities, ELSA is able to locate most of the Pareto front (Menczer *et al.*, 2000a). However, for problems requiring effective selection pressure, local selection may not be ideal because of its weak selection scheme.

FEATURE SELECTION IN SUPERVISED LEARNING

In this section, we propose a new approach for the customer targeting that combines evolutionary algorithms (EAs) and artificial neural networks (ANNs). In particular, we want to address the multi-objective nature of the customer targeting applications -- maximizing hit rate and minimizing complexity of the model through feature selection. We use the ELSA to search the possible combinations of features and ANNs to score the probability of buying new services or products using only the selected features by ELSA.

Problem Specification and Data Sets

Direct mailings to potential customers have been one of the most common approaches to market a new product or service. With a better understanding of who their potential customers were, the company would know more accurately whom to target, and they could reduce expenses and the waste of time and effort. In particular, we are interested in predicting potential customers who would be interested in buying a recreational vehicle (RV) insurance policy[2] while reducing feature dimensionality.

Suppose that one insurance company wants to advertise a new insurance policy based on socio-demographic data over a certain geographic area. From its first direct mailing to 5822 prospects, 348 purchased RV insurance, resulting in a hit rate of 348/5822 = 5.97%. Could the company attain a higher response rate from another carefully chosen direct mailings from the top x% of a new set of 4000 potential prospects?

In our experiment, we use two separate data sets, a training (5822 records) and an evaluation set (4000 records). Originally, each data set had 85 attributes, containing socio-demographic information (attributes 1--43) and contribution to and ownership of various insurance policies (attributes 44--85). The socio-demographic data was derived using zip codes and thus all customers living in areas with the same zip code have the same socio-demographic attributes. We omitted the first feature (customer subtype) mainly because it would expand search space dramatically with little information gain if we represented it as a 41-bit variable. Further we can still exploit the information of customer type by recording the fifth feature (customer main type) as a 10-bit variable. The other features are considered continuous and scaled to a common range (0--9).

ELSA/ANN Model Specification

Structure of the ELSA/ANN Model

Our predictive model is a hybrid model of the ELSA and ANN procedures, as shown in Figure 2. ELSA searches for a set of feature subsets and passes it to an ANN. The ANN extracts predictive information from each subset and learns the patterns using a randomly selected 2/3 of the training data. The trained ANN is then evaluated on the remaining 1/3 of the training data, and returns two evaluation metrics, $F_{accuracy}$ and

$F_{complexity}$ (described below), to ELSA. Note that in both the learning and evaluation procedures, the ANN uses only the selected features. Based on the returned metric values, ELSA biases its search to maximize the two objectives until the maximum number of iterations is attained.

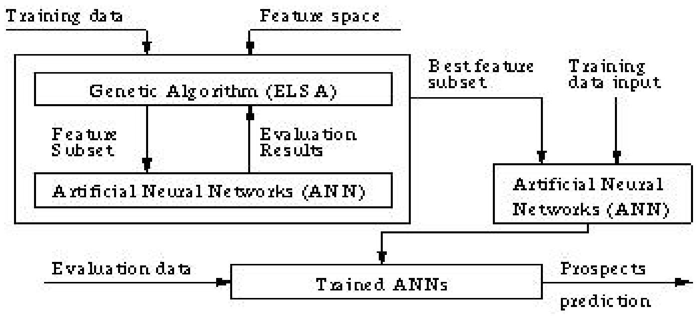

Figure 2: The ELSA/ANN model.

Among all evaluated solutions over the generations, we choose for further evaluation the set of candidates that satisfy a minimum hit rate threshold. With chosen candidates, we start a 10-fold cross validation. In this procedure, the training data is divided into 10 non-overlapping groups. We train an ANN using the first nine groups of training data and evaluate the trained ANN on the remaining group. We repeat this procedure until each of the 10 groups has been used as a test set once. We take the average of the accuracy measurements over the 10 evaluations and call it an *intermediate* accuracy. We repeat 10-fold cross validation procedure five times and call the average of the five intermediate accuracy estimates *estimated* accuracy.

We maintain a superset of the Pareto front containing those solutions with the highest accuracy at every Fcomplexity level covered by ELSA. For evaluation purposes, we subjectively decided to pick a best solution with the minimal number of features at the marginal accuracy level.[3] Then we train the ANN using all the training data with the selected features only and the trained model is used to select the top *x*% of the potential customers in the evaluation set based on the estimated probability of buying RV insurance. We finally calculate the actual accuracy of our model.

Evaluation Metrics

We use two heuristic evaluation criteria, $F_{accuracy}$ and $F_{complexity}$, to evaluate selected feature subsets. Each objective, after being normalized into 25 intervals to allocate energy, is to be maximized by ELSA.

$\boldsymbol{F_{accuracy}}$: The purpose of this objective is to favor feature sets with a higher hit rate. We define two different measures, $F_{accuracy}^{1}$ and $F_{accuracy}^{2}$ for two different experiments. In experiment 1, we select the top 20% of potential customers in descending order of the probability of purchasing the product and compute the ratio of the number of actual customers, *AC*, out of the chosen prospects, *TC*. We calculate $F_{accuracy}^{1}$ as follows:

$$F_{accuracy}^{1} = (1 / Z_{accuracy}^{1})\ (AC / TC) \tag{3}$$

where $Z_{accuracy}{}^{1}$ is an empirically derived normalization constant.

In experiment 2, we measure accuracy at the first m intervals[4] after dividing the range of customer selection percentages into 50 intervals with equal width (2%). At each interval $i \leq m$, we select the top $(2 \cdot i)$% of potential customers in descending order of the probability of purchasing the product and compute the ratio of the number of actual customers, AC_i, out of the total number of actual customers in the evaluation data, Tot. We multiply the width of interval and sum those values to get the area under the lift curve over m intervals. Finally we divide it by m to get our final metric, $F_{accuracy}{}^{2}$. We formulate it as follows:

$$F_{accuracy}{}^{2} = \frac{1}{Z_{accuracy}{}^{2}} \frac{1}{m} \sum_{i=1}^{m} \frac{AC_i}{Tot} \cdot 2 \qquad (4)$$

where $Tot = 238$, $m = 25$ and $Z_{accuracy}{}^{2}$ is an empirically derived normalization constant.

$F_{complexity}$: This objective is aimed at finding parsimonious solutions by minimizing the number of selected features as follows:

$$F_{complexity} = 1 - (d - 1)/(D - 1). \qquad (5)$$

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

Experimental Results

Experiment 1

In this experiment, we select the top 20% of customers to measure the hit rate of each solution as in (Kim *et al.*, 2000). For comparison purpose, we implement the PCA/logit model by first applying PCA on the training set. We select 22 PCs --- the minimum required to explain more than 90% of the variance in the data set --- and use them to reduce the dimensionality of the training set and the evaluation set.

We set the values for the ELSA parameters in the ELSA/ANN and ELSA/logit models as follows: $Pr(mutation) = 1.0$, $p_{max} = 1{,}000$, $E_{cost} = 0.2$, $\theta = 0.3$, and $T = 2{,}000$. In both models, we select the single solution with the highest expected hit rate among those solutions with fewer than 10 features selected. We evaluated each model on the evaluation set and summarized our results in Table 1.

Model (# Features)	Training set	Evaluation set	
	Hit Rate ± s.d	# Correct	Hit Rate
PCA/logit (22)	12.83 ± 0.498	109	13.63
ELSA/logit (6)	15.73 ± 0.203	115	14.38
ELSA/ANN (7)	15.92 ± 0.146	120	15.00

Table 1: Results of experiment 1. The column marked "# Correct" shows the number of actual customers who are included in the chosen top 20%. The number in parenthesis represents the number of selected features except for the PCA/logit model, where it represents the number of PCs selected.

In terms of the actual hit rate, ELSA/ANN returns the highest actual hit rate. Feature selection (the difference in actual hit rate between PCA/logit and ELSA/logit) and non-linear approximation (the difference in actual hit rate between ELSA/logit and ELSA/ANN) contribute about half of the total accuracy gain respectively. The improvement of the ELSA/ANN model in actual hit rate could make a meaningful difference in profit as the number of targeted prospects increases.

The resulting model of ELSA/ANN is also easier to interpret than that of PCA/logit. This is because, in the PCA/logit model, it is difficult to interpret the meaning of each of PCs in high dimensional feature spaces. Further the ELSA/ANN model makes it possible to evaluate the predictive importance of each features. The chosen seven features by the ELSA/ANN model are: Customer main type (average family),

Contribution to 3rd party policy, car policy, moped policy and fire policy, and number of 3rd party policies and social security policies. Among those features, we expected at least one of the car insurance related features to be selected. Moped policy ownership is justified by the fact that many people carry their mopeds or bicycles on the back of RVs. Those two features are selected again by the ELSA/logit model.[5] Using this type of information, we were able to build a potentially valuable profile of likely customers (Kim *et al.*, 2000).

The fact that the ELSA/ANN model used only seven features for customer prediction makes it possible to save a great amount of money through reduced storage requirements ($86/93 \approx 92.5\%$) and through the reduced labor and communication costs for data collection, transfer, and analysis. By contrast, the PCA/logit model needs the whole feature set to extract PCs. We also compare the lift curves of the three models. Figure 3 shows the cumulative hit rate over the top x% of prospects ($2 \le x \le 100$).

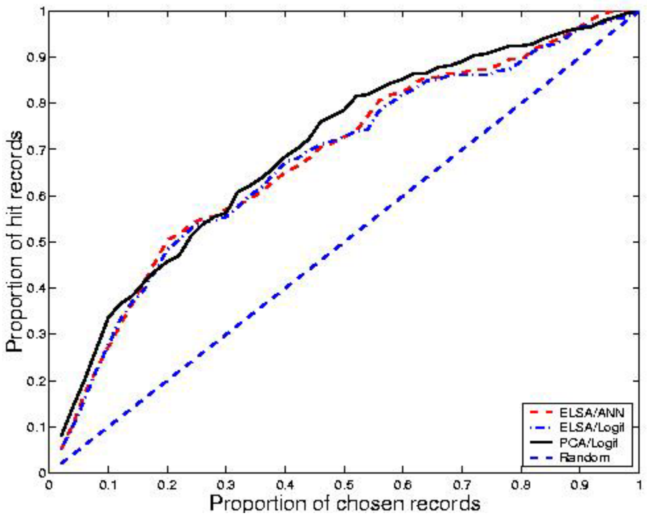

Figure 3: Lift curves of three models that maximize the hit rate when targeting the top 20% of prospects.

As expected, our ELSA/ANN model followed by ELSA/logit is the best when marketing around the top 20% of prospects. However, the performance of ELSA/ANN and ELSA/logit over all other target percentages was worse than that of PCA/logit. This is understandable because our solution is specifically designed to optimize at the top 20% of prospects while PCA/logit is not designed for specific selection points. This observation leads us to do the second experiment in order to improve the performance of ELSA/ANN model over all selection points.

Experiment 2

In this experiment, we search for the best solution that maximizes the overall accuracy up to the top 50% of potential customers. ELSA/ANN and ELSA/logit models are adjusted to maximize the overall area under the lift curve over the same intervals. In practice, we optimize over the first 25 intervals which have the same width, 2%, to approximate the area under the lift curve. Because this new experiment is computationally expensive, we use 2-fold cross validation estimates of all solutions. We, however, set the values of the ELSA parameters identically with the previous experiment except p_{max} = 200 and T = 500. Based on the accuracy estimates, we choose a solution that has the highest estimated accuracy with less than half of the original features in both models. We evaluate the three models on the evaluation set and summarize the results in Table 2 and in Figure 4.

Model(# Features)	% of selected									
	5	10	15	20	25	30	35	40	45	50
PCA/logit (22)	20.06	20.06	16.04	13.63	12.44	11.20	10.81	10.22	9.87	9.38
ELSA/logit (46)	23.04	18.09	15.56	13.79	12.13	12.04	10.97	10.54	10.03	9.53
ELSA/ANN(44)	19.58	17.55	16.40	14.42	13.13	11.96	10.97	10.40	9.98	9.64

Table 2: Summary of experiment 2. The hit rates of three different models are shown over the top 50% of prospects.

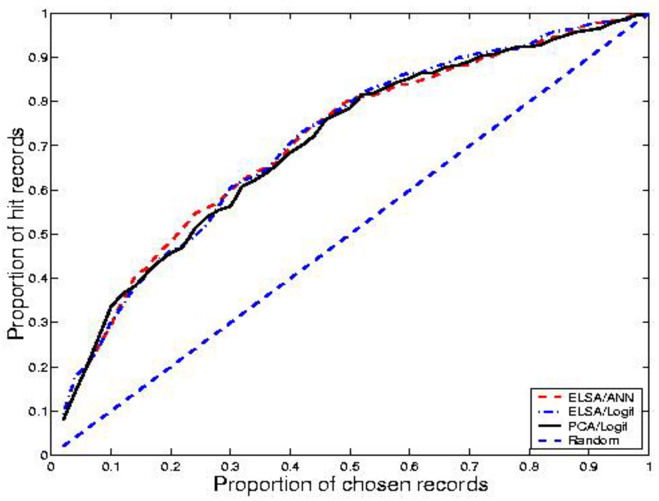

Figure 4: Lift curves of three models that maximize the area under lift curve when targeting up to top 50% of prospects.

The ELSA/ANN model works better than PCA/logit and ELSA/logit over the targeting range between 15% and 50%. In particular, ELSA/ANN is best at 15%, 20%, 25%, and 50% of targeted customers, and approximately equal to the best at 30-45%. The overall performance of ELSA/logit is better than that of PCA/logit. We attribute this to the fact that solutions from both ELSA models exclude many irrelevant features. PCA/logit, however, is competitive for targeting more than 50% of the customers, since ELSA/ANN and ELSA/logit do not optimize over these ranges. Though the well-established parsimony of the ELSA/ANN models in experiment 1 is largely lost in experiment 2, the ELSA/ANN model is still superior to PCA/logit model in terms of the parsimony of selected features since the PCA/logit model needs the whole feature set to construct PCs.

Conclusions

In this section, we presented a novel application of the multi-objective evolutionary algorithms for customer targeting. We used ELSA to search for possible combinations of features and an ANN to score customers based on the probability that they will buy the new service or product. The overall performance of ELSA/ANN in terms of accuracy was superior to the traditional method, PCA/logit, and an intermediate model, ELSA/logit. Further, the final output of the ELSA/ANN model was much easier to interpret because only a small number of features are used.

In future work we want to investigate how more general objectives affect the parsimony of selected features. We also would like to consider a marketing campaign in a more realistic environment where various types of costs and net revenue for additional customers are considered. We could also consider budget constraints and minimum/maximum campaign sizes. This way the number of targeted customers would be determined inside an optimization routine to maximize the expected profit.

FEATURE SELECTION IN UNSUPERVISED LEARNING

In this section, we propose a new approach to feature selection in *clustering* or unsupervised learning. This can be very useful for enhancing customer relationship management (CRM) because standard application of cluster analysis uses the complete set of features or a pre-selected subset of features based on the prior knowledge of market managers. Thus it cannot provide new marketing models that could be effective but have not been considered. 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.

Our model is also different from other approaches (Agrawal *et al.*, 1998, Dy and Brodley 2000b, Devaney and Ram, 1997) in two aspects: the evaluation of candidate solutions along multiple criteria, and the use of a local evolutionary algorithm to cover the space of feature subsets and of cluster numbers. Further, by identifying newly-discovered feature subsets that form well-differentiated clusters, our model can affect the way new marketing campaigns should be implemented.

EM Algorithm for Finite Mixture Models

The expectation maximization algorithm (Dempster *et al.*, 1977) is one of the most often used statistical modeling algorithms (Cheeseman and Stutz, 1996). The EM algorithm often significantly outperforms other clustering methods (Meila and

Heckerman, 1998) and is superior to the distance-based algorithms (*e.g.* K-means) in the sense that it can handle categorical 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. For Gaussian distributions, the parameters are the mean μ_k and covariance matrix Σ_k. Readers who are interested in algorithm detail refer to (Buhmann, 1995, Bradley *et al.*, 1998).

In order to evaluate the quality of the clusters formed by the EM algorithm, we use three heuristic fitness criteria, described below. Each objective is normalized into the unit interval and maximized by the EA.

$\boldsymbol{F_{accuracy}}$: 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, $\boldsymbol{F_{accuracy}}$ is computed as follows:

$$F_{accuracy} = \frac{1}{Z_{accuracy}} \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} p_k \cdot c_k(x_n \mid \mu_k, \Sigma_k) \right) \tag{6}$$

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

$\boldsymbol{F_{clusters}}$: The purpose of this objective is to favor clustering models with fewer clusters, if other things being equal.

$$\boldsymbol{F_{clusters}} = 1 - (K - K_{\boldsymbol{min}}) / (K_{\boldsymbol{max}} - K_{\boldsymbol{min}}) \tag{7}$$

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

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

$$F_{complexity} = 1 - (d - 1)/(D - 1). \tag{8}$$

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.

The Wrapper Model of ELSA/EM

We first outline the model of ELSA/EM in Figure 5. In ELSA, 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 ($D + K_{\boldsymbol{max}} - 2$) 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.[6] 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.

```
initialize $p_{max}$ agents, each with energy $\eta/2$;
while there are alive agents in $Pop^g$ and $t < T$
    Replenishment();
    for each agent $a$ in $Pop^g$
        Search & Evaluation();  Selection();  $t = t+1$;
    $g = g+1$;
endwhile

Replenishment(){
    for each energy source $c \in \{1, ..., C\}$
        for each $v \in (1/B, 2/B, ..., 1)$  where $B$ is number of bins
            $E_{envt}^c(v) \leftarrow 2vE_{tot}^c$;  }

Search & Evaluation() {
    $a' \leftarrow mutate(crossover(a, random\ mate))$;
    for each energy source $c \in \{1, ..., C\}$
        $v \leftarrow Fitness(a')$;      $\Delta E \leftarrow min(v, E_{envt}^c(v))$;
        $E_{envt}^c(v) \leftarrow E_{envt}^c(v) - \Delta E$;  $E_a \leftarrow E_a + \Delta E$;   $E_a \leftarrow E_a - E_{cost}$;  }

Selection(){
    if ($E_a > \eta$)
insert $a,a'$ into $Pop^{g+1}$;  $E_{a'} \leftarrow E_a / 2$;  $E_a \leftarrow E_a - E_{a'}$;
    else if ($E_a > 0$)
        insert $a$ into $Pop^{g+1}$;  }
```

Figure 5: The pseudo-code of ELSA/EM.

Mutation and crossover operators are used to explore the search space and are defined in the same way as in previous section. 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 EM algorithm. 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.

Experiments on the Synthetic Data

Data Set and Baseline Algorithm

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. 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" 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. We evaluate the evolved solutions by their ability to discover five pre-constructed clusters in a ten-dimensional subspace.

We present some 2-dimensional projections of the synthetic data set in Figure 6. In our experiments, Individuals are represented by 36 bits, 30 for the features and 6 for K ($K_{max} = 8$). There are 15 energy bins for all energy sources, $F_{clusters}$, $F_{complexity}$, and $F_{accuracy}$. The values for the various ELSA parameters are: Pr(*mutation*) = 1.0, Pr(*crossover*) = 0.8, $p_{max} = 100$, $E_{cost} = 0.2$, $E_{total} = 40$, $\eta = 0.3$, and $T = 30{,}000$.

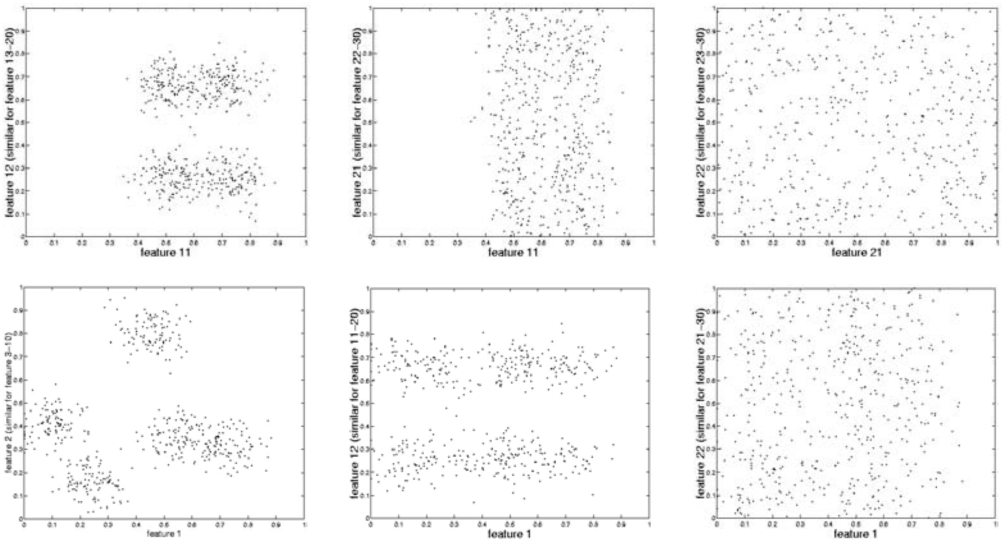

Figure 6: A few 2-dimensional projections of the synthetic data set.

Experimental Results

We show the candidate fronts found by the ELSA/EM algorithm for each different number of clusters K in Figure 7.

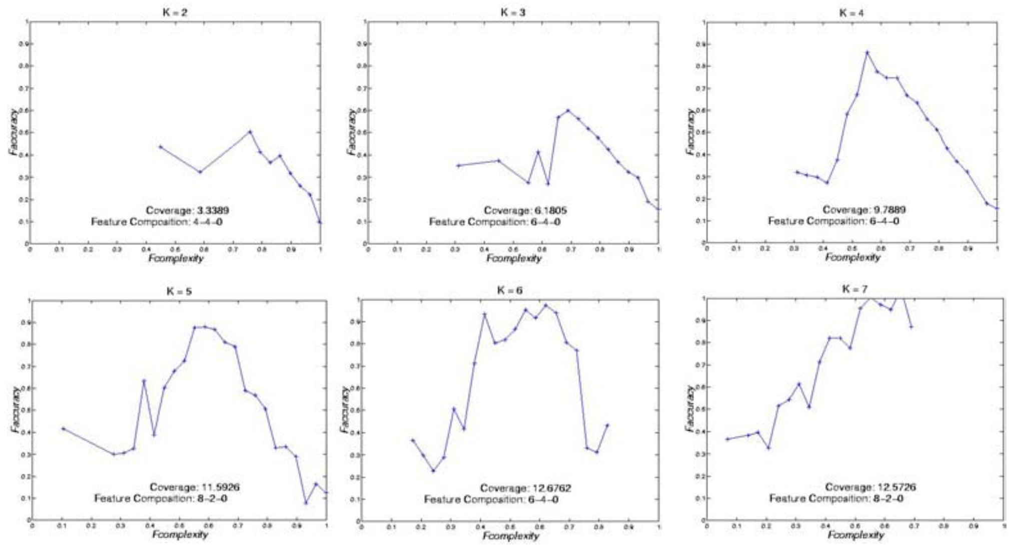

Figure 7: The candidate fronts of ELSA/EM model. We omit the candidate front for $K = 8$ because of its inferiority in terms of clustering quality and incomplete coverage of the search space. Composition of selected features is shown for $F_{complexity}$ corresponding to 10 features (see text).

We analyze 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 $F_{accuracy}$. The shape of the Pareto fronts observed in ELSA/EM is as follows: an ascent in the range of higher values of $F_{complexity}$ (lower complexity), and a descent for lower values of $F_{complexity}$ (higher complexity). This is reasonable because adding additional significant features will have a good effect on 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. The coverage of the ELSA/EM model shown in Figure 7 is defined as:

$$\operatorname{cov}erage_{EM} = \sum_{i \in F_{complexity}} F_{accuracy}^{\ \ i} \tag{9}$$

We note that the clustering quality and the search space coverage improve as the evolved number of clusters approaches the "true" number of cluster $K = 5$. 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 $F_{complexity}$ ($K = 2$ and $K = 3$). A decision-maker again would conclude the right number of clusters to be 5 or 6.

We note that the first 10 selected features, $0.69 \le F_{complexity} \le 1$, are not all significant. This notion is again quantified through the number of significant-Gaussian noise-white noise features selected at $F_{complexity} = 0.69$ (10 features) in Figure 7.[7] None of the "white noise" features is selected. We also show snapshots of the ELSA/EM fronts for $K = 5$ at every 3,000 solution evaluations in Figure 8. ELSA/EM explores a broad subset of the search space, and thus identifies better solutions across $F_{complexity}$ as more solutions are evaluated. We observed similar results for different number of clusters K.

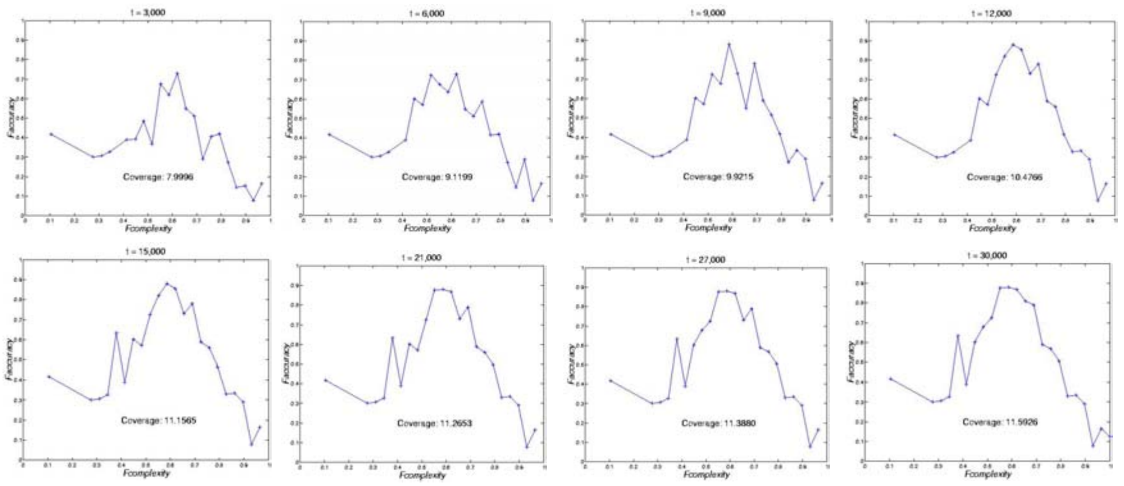

Figure 8: Candidate fronts for $K = 5$ based on $F_{accuracy}$ evolved in ELSA/EM. It is captured at every 3,000 solution evaluations and two fronts ($t = 18{,}000$ and $t = 24{,}000$) are omitted because they have the same shape as the ones at $t = 15{,}000$ and $t = 21{,}000$, respectively.

Table 3 shows classification accuracy of models formed by both ELSA/EM and the greedy 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 selected from candidate fronts with less than eight features. ELSA/EM consistently outperforms the greedy search on models with few features and few clusters. For more complex models with more than 10 selected features, the greedy method often shows higher classification accuracy.

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	52.0±6.6	91.6±5.7	93.8±6.2	99.0±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	52.0±1.0	60.6±5.1	4-2-0
	Greedy	27.8±0.8	27.8±0.4	29.0±0.4	29.6±0.9	38.0±4.4	74.2±3.5	
5	ELSA/EM	44.6±2.0	32.6±3.8	72.0±3.8	62.4±1.9	66.4±3.7	88.0±4.9	5-0-1
	Greedy	23.0±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

Table 3: 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.

Experiments on WPBC Data

We also tested our algorithm on a real data set, the Wisconsin Prognostic Breast Cancer (WPBC) data (Mangasarian *et al.*, 1995). 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.

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 9 shows the survival characteristics of three prognostic groups found by ELSA/EM.

The three groups showed well-separated survival characteristics. Out of 198 patients, 59, 54 and 85 patients belong to the good, 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.

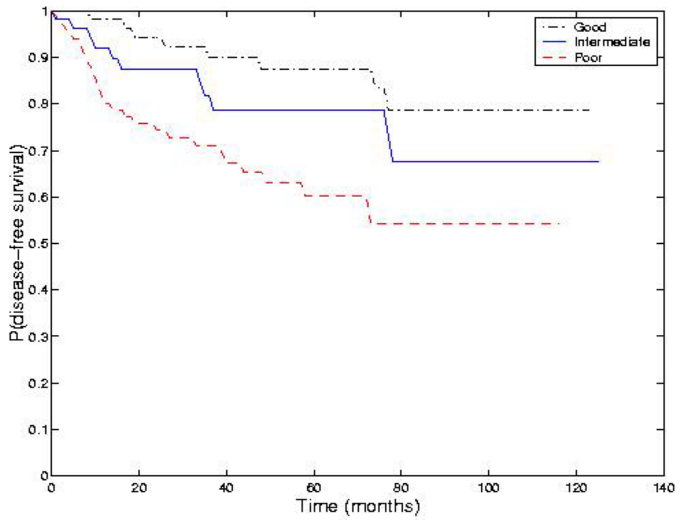

Figure 9: Estimated survival curves for the three groups found by ELSA/EM.

We note that neither of the traditional medical prognostic factors, tumor size and lymph node status, is chosen. This finding is potentially important because the lymph node status can be determined only after lymph nodes are surgically removed from the patient's armpit (Street *et al.*, 95). We further investigate whether other solutions with lymph node information can form three prognostic groups as good as our EM 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.

Conclusions

In this section, we presented a new ELSA/EM algorithm for unsupervised feature selection. Our ELSA/EM model outperforms a greedy algorithm in terms of classification accuracy while considering a number of possibly conflicting heuristic metrics. Most importantly, our model can reliably select an appropriate clustering model, including significant features and the number of clusters.

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 terms of the composition of selected features and prediction accuracy.

FEATURE SELECTION FOR ENSEMBLES

In this section, we propose a new meta-ensembles algorithm to directly optimize ensembles by creating a two-level evolutionary environment. In particular, we employ feature selection not only to increase the prediction accuracy of an individual classifier but also to promote diversity among component classifiers in an ensemble (Opitz, 1999).

Feature Selection and Ensembles

Recently many researchers have combined the predictions of multiple classifiers to produce a better classifier, an ensemble, and often reported improved performance (Breiman, 1996b, Bauer and Kohavi, 1999). Bagging (Breiman, 1996a) and Boosting (Freund and Schapire, 1996) are the most popular methods for creating accurate ensembles. The effectiveness of Bagging and Boosting comes primarily from the diversity caused by re-sampling training examples while using the complete set of features to train component classifiers.

Recently several attempts have been made to incorporate the diversity in feature dimension into ensemble methods. The Random Subspace Method (RSM) in (Ho, 1998a, Ho, 1998b) was one early algorithm that constructed an ensemble by varying the feature subset. RSM used C4.5 as a base classifier and randomly chose half of the original features to build each classifier. In (Guerra-Salcedo and Whitley, 1999), four different ensemble methods were paired with each of three different feature selection algorithms: complete, random, and genetic search. Using two table-based classification methods, ensembles constructed using features selected by the GA showed the best performance. In (Cunningham and Carney, 2000), a new entropy measure of the outputs of the component classifiers was used to explicitly measure the ensemble diversity and to produce good feature subsets for ensemble using hill-climbing search.

Genetic Ensemble Feature Selection (GEFS) (Opitz, 1999) used a GA to search for possible feature subsets. GEFS starts with an initial population of classifiers built using up to $2 \cdot D$ features, where D is the complete feature dimension. It is possible for some features to be selected more than once in GEFS and crossover and mutation operators are used to search for new feature subsets. Using 100 most-fit members with majority voting scheme, GEFS reported better estimated generalization than Bagging and AdaBoost on about two-thirds of 21 data sets tested. Longer chromosomes, however, make GEFS computationally expensive in terms of memory usage (Guerra-Salcedo and Whitley, 1999). Further, GEFS evaluates each classifier after combining two objectives in a subjective manner using $fitness = accuracy + \lambda\ diversity$, where *diversity* is the average difference between the prediction of component classifiers and the ensemble.

However, all these methods consider only one ensemble. We propose a new algorithm for ensemble feature selection, Meta-Evolutionary Ensembles (MEE), that considers multiple ensembles simultaneously and allows each component classifiers to move into the best-fit ensemble. We evaluate and reward each classifier based on two different criteria, accuracy and diversity. A classifier that correctly predicts data examples that other classifiers in the same ensemble misclassify contributes more to the accuracy of the ensemble to which it belongs.

We imagine that some limited "energy" is evenly distributed among the examples in the data set. Each classifier is rewarded with some portion of the energy if it correctly predicts an example. The more classifiers that correctly classify a specific example, the less energy is rewarded to each, encouraging them to correctly predict the more difficult examples. The predictive accuracy of each ensemble determines the total amount of energy to be replenished at each generation. Finally, we select the ensemble with the highest accuracy as our final model.

Meta-Evolutionary Ensembles

Pseudocode for the Meta-Evolutionary Ensembles (MEE) algorithm is shown in Figure 10, and a graphical depiction of the energy allocation scheme is shown in Figure 11.

```
initialize population of agents, each with energy $\theta/2$
while there are alive agents in $Pop^i$ and $i < T$
    for each ensemble $g$
        for each record $r$ in $Data_{test}$
            $prevCount_{g,r} = count_{g,r}$;  $count_{g,r} = 0$;
        for each agent $a$ in $Pop^i$
            $a' = mutate(crossover(a, randomMate))$;
            $g = group(a)$;
            train($a$);
            for each record $r$ in $Data_{test}$
                if ($class(r) == prediction(r,a)$)
                    $count_{g,r}$++;   $\Delta E = E_{envt}^{g,r} / min(5, prevCount_{g,r})$;
                    $E_{envt}^{g,r} = E_{envt}^{g,r} - \Delta E$;  $E_a = E_a + \Delta E$;
            $E_a = E_a - E_{cost}$;
            if ($E_a > \theta$)
                insert $a, a'$ into $Pop^{i+1}$;  $E_{a'} = E_a / 2$;  $E_a = E_a - E_{a'}$;
            else if ($E_a > 0$)
                insert $a$ into $Pop^{i+1}$;
    for each ensemble $g$
        replenish energy based on predictive accuracy;
    $i = i+1$;
endwhile
```

Figure 10: Pseudo-code of Meta-Evolutionary Ensembles (MEE) algorithm.

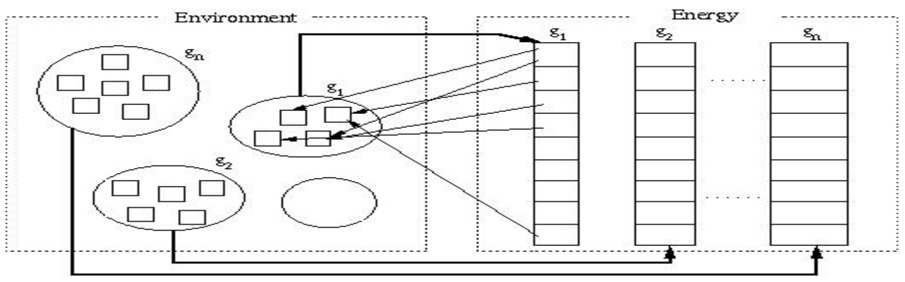

Figure 11: Graphical depiction of energy allocation in the MEE. Individual classifiers (small boxes in the environment) receive energy by correctly classifying test points. Energy for each ensemble is replenished between generations based on the accuracy of the ensemble. Ensembles with higher accuracy have their energy bins replenished with more energy per classifier, as indicated by the varying widths of the bins.

Each agent (candidate solution) in the population is first initialized with randomly selected features, a random ensemble assignment, and an initial reservoir of energy. The representation of an agent consists of $D + log_2(G)$ bits. D bits correspond to the selected features (1 if a feature is selected, 0 otherwise). The remaining bits are a binary representation of the ensemble index, where G is the maximum number of ensembles. Mutation and crossover operators are used to explore the search space and are defined in the same way as in previous section.

In each iteration of the algorithm, an agent explores a candidate solution (classifier) similar to itself, obtained via crossover and mutation. The agent's bit string is parsed to get a feature subset J. An ANN is then trained on the projection of the data set onto J, and returns the predicted class labels for the test examples. The agent collects ΔE from each example it correctly classifies, and is taxed once with E_{cost}. The net energy intake of an agent is determined by its classification accuracy. But the energy also depends on the state of the environment. We have an energy source for each ensemble, divided into bins corresponding to each data point. For ensemble g and record index r in the test data, the environment keeps track of energy $E_{envt}^{g,r}$ and the number of agents in ensemble g, $count_{g,r}$ that correctly predict record r. The energy received by an agent for each correctly classified record r is given by

$$\Delta E = E_{envt}^{g,r} \,/\, min(5, prevCount_{g,r}). \qquad (10)$$

An agent receives greater reward for correctly predicting an example that most in its ensemble get wrong. The *min* function ensures that for a given point there is enough energy to reward at least 5 agents in the new generation. Candidate solutions 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, where more energy is available. The result is a natural bias toward diverse solutions in the population. E_{cost} for any action is a constant ($E_{cost} < \theta$).

In the selection part of the algorithm, an agent compares its current energy level with a constant reproduction threshold θ. If its energy is higher than θ, the agent reproduces: the agent and its mutated clone become part of the new population, with the offspring receiving half of its parent's energy. If the energy level of an agent is positive but lower than θ, only that agent joins the new population.

The environment for each ensemble is replenished with energy based on its predictive accuracy, as determined by majority voting with equal weight among base classifiers. We sort the ensembles in ascending order of estimated accuracy and apportion energy in linear proportion to that accuracy, so that the most accurate ensemble is replenished with the greatest amount of energy per base classifier. Since the total amount of energy replenished also depends on the number of agents in each ensemble, it is possible that an ensemble with lower accuracy can be replenished with more energy in total than an ensemble with higher accuracy.

Experimental Results

Experimental Results of MEE/ANN

We tested the performance of MEE combined with neural networks on several data sets that were used in (Opitz, 1999). In our experiments, the weights and biases of the neural networks are initialized randomly between 0.5 and -0.5, and the number of hidden node is determined heuristically as the square root of *inputs*. The other parameters for the neural networks include a learning rate of 0.1 and a momentum rate of 0.9. The number of training epochs was kept small for computational reasons. The values for the various parameters are: Pr(*mutation*) = 1.0, Pr(*crossover*) = 0.8, $E_{cost} = 0.2$, $\theta = 0.3$, and $T = 30$. The value of $E_{envt}^{tot} = 30$ is chosen to maintain a population size around 100 classifier agents.

Experimental results are shown in Table 4. All computational results for MEE are based on the performance of the best ensemble and are averaged over five standard 10-fold cross-validation experiments. Within the training algorithm, each ANN is trained on two-thirds of the training set and tested on the remaining third for energy allocation purposes. We present the performance of a single neural network using the complete set of features as a baseline algorithm. In the win-loss-tie results shown at the bottom of Table 4, a comparison is considered a tie if the intervals defined by one standard error[8] of the mean overlap. On the data sets tested, MEE shows consistent improvement over a single neural network.

Data sets	Single net		Baggin g	AdaBoost	GEFS	MEE		
	Avg.	S.D.				Avg.	S.D.	Epochs
Credita	84.3	0.30	86.2	84.3	86.8	86.4	0.52	40
Creditg	71.7	0.43	75.8	74.7	75.2	75.6	0.78	50
Diabetes	76.4	0.93	77.2	76.7	77.0	76.8	0.42	50
Glass	57.1	2.69	66.9	68.9	69.6	61.1	1.73	100
Cleveland	80.7	1.83	83.0	78.9	83.9	83.3	1.54	50
Hepatitis	81.5	0.21	82.2	80.3	83.3	84.9	0.65	40
Votes-84	95.9	0.41	95.9	94.7	95.6	96.1	0.44	40
Hypo	93.8	0.09	93.8	93.8	94.1	93.9	0.06	50
Ionosphere	89.3	0.85	90.8	91.7	94.6	93.5	0.81	100
Iris	95.9	1.10	96.0	96.1	96.7	96.5	0.73	100
Krvskp	98.8	0.63	99.2	99.7	99.3	99.3	0.10	50
Labor	91.6	2.29	95.8	96.8	96.5	94.4	0.78	50
Segment	92.3	0.97	94.6	96.7	96.4	93.2	0.28	50
Sick	95.2	0.47	94.3	95.5	96.5	99.3	0.03	50
Sonar	80.5	2.03	83.2	87.0	82.2	85.2	1.57	100
Soybean	92.0	0.92	93.1	93.7	94.1	93.8	0.19	50
Vehicle	74.7	0.48	79.3	80.3	81.0	76.4	1.12	50
Win-loss-tie	15-0-2		7-4-6	9-6-2	4-7-6			

Table 4: Experimental results of MEE/ANN.

We also include the results of Bagging, AdaBoost, and GEFS from (Opitz, 1999) for indirect comparison. In these comparisons, we did not have access to the accuracy results of the individual runs. Therefore, a tie is conservatively defined as a test in which the one-standard-deviation interval of our test contained the point estimate of accuracy from (Opitz, 1999). In terms of predictive accuracy, our algorithm demonstrates better or equal performance compared to single neural networks, Bagging and Boosting. However, MEE shows slightly worse performance compared to GEFS, possibly due to the methodological differences. For example, it is possible that the more complex structure of neural networks used in GEFS can learn more difficult patterns in data sets such as Glass and Labor data.

From the perspective of computational complexity, our algorithm can be very slow compared to Bagging and Boosting. However, MEE can be very fast compared to GEFS because GEFS uses twice as many as input features as used in MEE. Further, the larger number of hidden nodes and longer training epochs can make GEFS extremely slow.

Guidelines Toward Optimized Ensemble Construction

In this section, we use MEE to examine ensemble characteristics and provide some guidelines for building optimal ensembles. We expect that by optimizing the ensemble construction process, MEE will in general achieve comparable accuracy to other methods using fewer individuals. We use data collected from the first fold of the first cross-validation routine for the following analyses.

We first investigate whether the ensemble size is positively related with the predictive accuracy. It has been well established that to a certain degree, the predictive accuracy of an ensemble improves as the number of classifiers in the ensemble increases. For example, our result in Figure 12 indicates that accuracy improvements flatten out at an ensemble size of approximately 15-25. We also investigate whether the diversity among classifiers is positively related with the ensemble's classification performance. In our experiments, we measured the diversity based on the difference of predicted class between each classifier and the ensemble. We first define a new operator $\oplus$ as follows: $\alpha \oplus \beta = 0$ if $\alpha = \beta$, 1 otherwise. When an ensemble e consists of g classifiers, the diversity of ensemble e, $diversity^e$, is defined as follows:

$$diversity^e = \frac{\sum_{i=1}^{g}\sum_{j=1}^{N}(pred_j^i \oplus pred_j^e)}{g \cdot N} \qquad (11)$$

where N is the number of records in the test data and $pred_j^i$ and $pred_j^e$ represent the predicted class label for record j by classifier i and ensemble e respectively. The larger the value of $diversity^e$, the more diverse ensemble is.

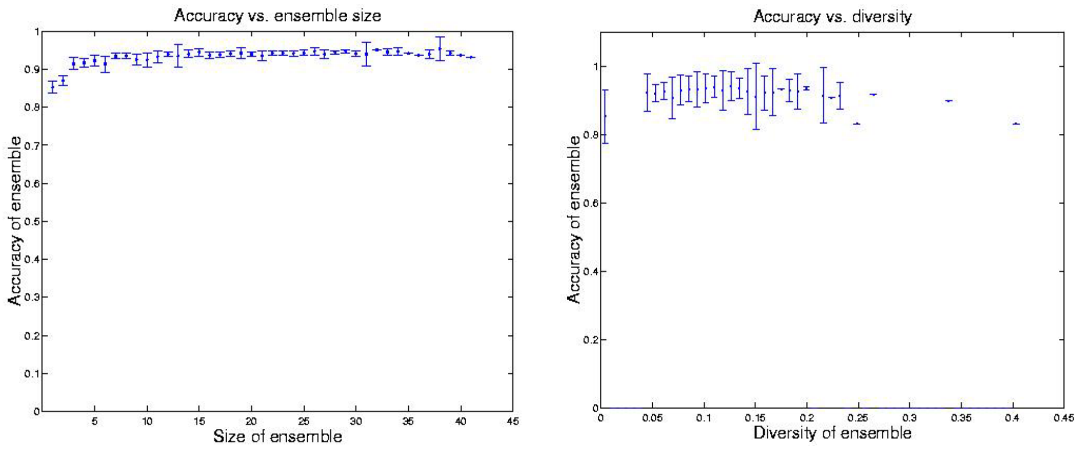

Figure 12: The relationship between the predictive accuracy and ensemble size (left), and between the predictive accuracy and ensemble diversity (right) with 95% confidence interval on the Soybean data. We observed similar patterns on other data sets.

We also show the relationship between the predictive accuracy and ensemble diversity in Figure 12. This shows the expected the positive relationship between accuracy and diversity. However, our results show that too much diversity among classifiers can deteriorate ensemble performance, as the final decision made by ensemble becomes a random guess.

Conclusions

In this section, we propose a new two-level ensemble construction algorithm, Meta-Evolutionary Ensembles (MEE) that uses feature selection as the diversity mechanism. At the first level, individual classifiers compete against each other to correctly predict held-out examples. Classifiers are rewarded for predicting difficult points, relative to the other members of their respective ensembles. At the top level, the ensembles compete directly based on classification accuracy.

Our model shows consistently improved classification performance compared to a single classifier at the cost of computational complexity. Compared to the traditional ensembles (Bagging and Boosting) and GEFS, our resulting ensemble shows comparable performance while maintaining a smaller ensemble. Further, our two-level evolutionary framework confirms that more diversity among classifiers can improve predictive accuracy. Up to a certain level, the ensemble size also has a positive effect on the ensemble performance.

The next step is to compare this algorithm more rigorously to others on a larger collection of data sets, and perform any necessary performance tweaks on the EA energy allocation scheme. This new experiment is to test the claim that there is relatively little room for other ensembles algorithm to obtain further improvement over decision forest method (Breiman, 1999). Along the way, we will examine the role of various characteristics of ensembles (size, diversity, etc.) and classifiers (type, number of dimensions / data points, etc.). By giving the system as many degrees of freedom as possible and observing the characteristics that lead to successful ensembles, we can directly optimize these characteristics and translate the results to a more scalable architecture (Street and Kim, 2001) for large-scale predictive tasks.

CONCLUSIONS

In this chapter, we proposed a new framework for feature selection in supervised and unsupervised learning. In particular, we note that each feature subset should be evaluated in terms of multiple objectives. In supervised learning, ELSA with neural networks model (ELSA/ANN) was used to search for possible combinations of features and to score customers based on the probability of buying new insurance product respectively. The ELSA/ANN model showed promising results in two different experiments, when market managers have clear decision scenario or not. ELSA was also used for unsupervised feature selection. Our algorithm, ELSA/EM, outperforms a greedy algorithm in terms of classification accuracy. Most importantly, in the proposed framework we can reliably select an appropriate clustering model, including significant features and the number of clusters.

We also proposed a new ensemble construction algorithm, Meta-Evolutionary Ensembles (MEE), where feature selection is used as the diversity mechanism among classifiers in the ensemble. In MEE, classifiers are rewarded for predicting difficult points, relative to the other members of their respective ensembles. Our experimental results indicate that this method shows consistently improved performance compared to a single classifier and the traditional ensembles.

One major direction of future research on the feature selection with ELSA is to find a way that can boost weak selection pressure of ELSA while keeping its local selection mechanism. For problems requiring effective selection pressure, local selection may be too weak because the only selection pressure that ELSA can apply comes from the sharing of resources. Dynamically adjusting the local environmental structure based on the certain ranges of the observed fitness values over a fixed number of generations could be a promising solution. In this way, we could avoid the case in which the solution with the worst performance can survive into the next generation because there are no other solutions in its local environment.

Another major direction of future research is related with the scalability issue. By minimizing the communication among agents, our local selection mechanism makes ELSA efficient and scalable. However, our models suffer the inherent weakness of the wrapper model, the computational complexity. Further by combining EAs with ANN to take the advantages of both algorithms, it is possible that the combined model can be so slow that it cannot provide solutions in a timely manner. With the rapid growth of records and variables in database, this failure can be critical. Combining ELSA with faster learning algorithms such as decision tree algorithms and Support Vector Machine (SVM) will be worth to pursue.

---

1 Continuous objective functions are discretized.

2 This is one of main tasks in the 2000 CoIL challenge (Kim *et al.*, 2000). For more information about CoIL challenges and the data sets, please refer to *http://www.dcs.napier.ac.uk/coil/challenge/*.

3 If other objective values are equal, we prefer to choose a solution with small variance.

4 This is reasonable because as we select more prospects, the expected accuracy gain will go down. If the marginal revenue from an additional prospect is much greater than the marginal cost, however, we could sacrifice the expected accuracy gain. Information on mailing cost and customer value was not available in this study.

5 The other four features selected by the ELSA/logit model are: contribution to bicycle and fire policy, and number of trailer and lorry policies.

6 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 $K_{min} = 2 \leq K \leq K_{max}$.

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

8 In our experiments, standard error is computed as standard deviation / $iter^{0.5}$ where $iter = 5$.