Dimensionality Reduction via Genetic Value Clustering

Alexander Topchy and William Punch

Computer Science Department, Michigan State University
East Lansing, MI, 48824, USA
{topchyal,punch}@cse.msu.edu

Abstract. Feature extraction based on evolutionary search offers new possibilities for improving classification accuracy and reducing measurement complexity in many data mining and machine learning applications. We present a family of genetic algorithms for feature synthesis through clustering of discrete attribute values. The approach uses new compact graph-based encoding for cluster representation, where size of GA search space is reduced exponentially with respect to the number of items in partitioning, as compared to original idea of Park and Song. We apply developed algorithms and study their effectiveness for DNA fingerprinting in population genetics and text categorization.

1 Introduction

Abundance of features and their measurement complexity is the central problem in many real-world applications of statistical pattern recognition and machine learning. Typical data models, which are the part of the classification and prediction systems encountered in data mining, may include thousands of variables and databases with millions of entries. Feature selection and extraction algorithms seek to provide a lower dimensional data representation that preserves most of the available information [1]. Many application domains often require that knowledge extracted from the data be comprehensible and compact [2,3]. Finding the best subset of variables relevant to the computational model is the goal of feature *selection*, while feature *extraction* synthesizes new features from the original variables, where features (variables) refer to the components of the pattern measurement vector. Potential benefits of reducing the data dimensions include: better modeling (classification/prediction) accuracy, simplification of the developed model, faster learning with fewer parameters, lower measurement costs, and improved reliability of parameter estimation.

Numerous dimensionality reduction algorithms are known and actively used in image recognition, text processing, computational biology and other domains [2]. Most of them fall into two broad categories. The first category, filter methods, estimates the relevance of a candidate feature transformation by analyzing the data distribution, often heuristically, *without* running the actual model. The second category, wrapper methods, optimizes the actual classification accuracy achieved by the selected (extracted) features [4]. In both cases the evaluation function is usually nonlinear and highly multimodal. Furthermore, the search space tends to be astronomically large resulting in a difficult optimization problem.

Evolutionary algorithms (EAs), in particular genetic algorithms (GAs), are a unique kind of optimization method as applied to feature selection/extraction. Unlike

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 1431–1443, 2003.

conventional sequential search of feature subsets [1], GA is certainly not sensitive to the non-monotonicity of an evaluation function and therefore not as prone to the difficulties involved in finding hidden, nested sets of features [2]. Siedlecki and Sklansky pioneered a GA approach to large-scale feature selection in [5], where classical, direct representation of feature sets was introduced. More sophisticated schemes were proposed in [6,7,8]. Building on this seminal work [5], GAs were also applied to feature extraction coupled with k nearest neighbor classifiers in [9,10] and several other algorithms [11,12]. Typically new features are created through the learning of real-value weights applied to the original features [13] or through a sequence of primitive operators encoded in the chromosome [14]. Recent attribute construction algorithms successfully utilize GA for data mining tasks [15]. A variety of relevant algorithms is reviewed in [3,16,17].

In this study we introduce a different GA approach to dimensionality reduction that does not explicitly operate with whole features, but instead operates on individual values assumed by the feature variables. Features are extracted by clustering several "old" values into a new meta-value which substitutes for the old values in the feature vector. Therefore, new features are created by clustering the values of variables. A GA is used as a search engine for this value clustering. If features assume nominal values then clustering could be viewed as a grouping problem and there are a number of known genetic algorithms that can be used [18]. Grouping as well as clustering GAs is an important research area in itself [18]. Scalability and redundancy are two characteristic problems of grouping algorithms. Here we present a grouping GA with an improved graph-based encoding, that extends work by Park and Song [19]. The proposed cluster representation has lower redundancy while preserving the simplicity of decoding and evaluation. This encoding is not limited to value clustering and therefore applicable to other grouping tasks.

There is an additional motivation behind value clustering. One outcome of dimensionality reduction could be more important than simplified data and knowledge representation, namely such reduction can also help to diminish the effect of the "curse of dimensionality". The curse of dimensionality phenomenon manifests itself in a decrease in classification accuracy when a large number of features are included in the model. Estimation errors will inevitably lead to accuracy degradation for the model induced from a fixed training sample [20]. It is important to note that dimensionality increase occurs not only in the number of features but also in the number of values assumed by the individual variables. In general, appropriate input transformations may eliminate either features or feature values or both. Effectively, such a transformation reduces measurement complexity, which is proportional to the total number of cells in the input value space. Overcoming the curse of dimensionality by lowering the measurement complexity is essential in many real-world tasks.

We analyze the performance of the value clustering genetic algorithms in two application domains with discrete variables:

1. Text categorization. Classification of text using naïve Bayes classifier and a "bag of words" data model where words correspond to different feature values. The number of features is determined by the text's length.
2. Population genetics. Genotype based assignment of individuals to their putative populations of origin. The data model is an extension of a multinomial model with multiple features and certain value constraints within features. The allelic values of multilocus individuals are subject to grouping (merging).

In the remainder of the paper we consider existing value clustering methods. A feature extraction genetic algorithm is then developed. We briefly review grouping GAs (GGAs) and describe an enhanced graph-based chromosome encoding of clusters. Experimental problems and implementation details of empirical studies are presented. Finally we discuss the results and interesting future work.

2 Background on Value Clustering

Data mining applications typically deal with instances represented as a set of input attributes (features), where each attribute assumes one of several possible values. We are primarily interested in data represented by nominal values. Throughout the paper, the term "value clustering" refers to a method that replaces several original values of an attribute by a new meta-value. This meta-value is completely artificial and simply a substitute for all the values used in the cluster. Such a clustering affects the data model and consequently the classification rule. This clustering of values represents a kind of generalization or value abstraction which is used to improve classification. We consider important related works below. However, it is instructive to start with a simple example illustrating the idea of value clustering.

2.1 Clustering in Naïve Bayes Model

Consider a naïve Bayes classifier. The classifier learns the conditional probability $P(A_i|C)$ of each attribute A_i given the class label C. Classification is then done by applying Bayes rule to compute the probability of C given the particular instance of $A_1,..,A_n$ and then predicting the class with the highest posterior probability (MAP hypothesis):

$$MAP \equiv \arg\max_i P(c_i \mid a_1,..,a_n), \tag{1}$$

where c_i is an instance of C (a class) and a_j is an instance of A_j. There is one major assumption behind this: all the attributes are conditionally independent given the value of the class C. Therefore, by using Bayes rule:

$$P(c_i \mid a_1,..,a_n) = \frac{P(a_1,..,a_n,c_i)}{P(a_1,..,a_n)} \propto P(c_i)\prod_j P(a_j \mid c_i) \tag{2}$$

Let us assume a multinomial probability distribution for the values of each attribute. Using the maximum likelihood principle one can estimate the class-conditional probabilities $\{P(a_j|c_i)\}$ from training samples as a ratio of number of instances with a specific value of a_j within class c_i to the total number of instances in the same class. For example, if there are only three possible values of the first attribute $a_1 \in \{\alpha,\beta,\gamma\}$, one would estimate the three terms $P(a_1=\alpha|c_i)$, $P(a_1=\beta|c_i)$ and $P(a_1=\gamma|c_i)$. However, if values α and β are merged into one cluster, the meta-value x is constructed $x=(\alpha\ or\ \beta)$, and only two model parameters remain to be estimated: $P(a_1=x|c_i)$ and $P(a_1=\gamma|c_i)$. In the multinomial model, the probability of the meta-value x obeys:

$$P(a_1=x|c_i) = P(a_1=\alpha|c_i) + P(a_1=\beta|c_i). \quad (3)$$

As a result of this clustering all the occurrences of α or β are replaced by x. Of course, the estimation of parameters is classifier specific.

2.2 Related Work

In traditional database research there are several studies for finding an approximate answer for a database query with no exact answer [21]. Automatic query expansion systems offer approximations from a cluster of similar values, obtained by term clustering [22]. Term clustering creates groups of related words by analyzing their co-occurrence in a document collection. A hierarchy of values can either be garnered from experts or obtained automatically. In [23] the attribute values are clustered automatically in an attribute abstraction hierarchy. This hierarchy is discovered using rules derived from database instances. For rules with the same consequence, values found in the premise are clustered. This process is called pattern based knowledge induction. However, information retrieval approaches cannot be easily combined with arbitrary classifiers.

In machine learning studies we find a number of flexible value clustering methods. Perhaps the most interesting method is the information bottleneck by Tishby et al [24]. The information bottleneck method replaces the original random variable X by a compact representation $\tilde{X}$, which tries to keep as much information as possible about the random variable Y. In particular, variable X could stand for feature values and variable Y is a class label. The information bottleneck method maximizes mutual information $I(\tilde{X};Y)$ between $\tilde{X}$ and Y, conditioned on the information content $I(\tilde{X};X)$ resulting from the clustering $\tilde{X}$ with respect to X. Most significantly, the optimal solution of this information theoretic problem can be found in terms of probability distributions $p(\tilde{x} \mid x)$, $p(y \mid \tilde{x})$, and $p(\tilde{x})$ with an arbitrary joint distribution $p(x, y)$. The exact solution is given by a set of nonlinear equations [24]. Iterative solving requires the user-defined cardinality of $\tilde{X}$, and $\tilde{X}$ is usually initialized to random values that potentially may lead to sub-optimal local solutions. The resulting clustering is not "hard", where each value of X belongs to a single cluster in $\tilde{X}$, but rather "soft" with membership probabilities $p(\tilde{x} \mid x)$. However, hard clustering can be achieved in the agglomerative information bottleneck method [25]. The agglomerative information bottleneck method is a hierarchical process that uses distance measures between distributions and greedy search by merging clusters pair-wise. The multivariate information bottleneck method [26, 27] further extends the approach for multivariate distributions, maximizing mutual information between Y and several "bottlenecked" variables simultaneously.

Recently, the value abstraction approach (clustering in our terms) led to considerable progress in linkage analysis for genetic mapping [28], and was expanded for more general likelihood computations and faster Bayesian network inference in [29]. All these works can be characterized as filter-type feature extraction, since classifier accuracy or induction step feedback is not used for value clustering.

3 Feature Extraction by Value Clustering

Genetic search for better clusters is the core of the proposed dimensionality reduction methodology. As such it is the part of a complete feature extraction system, which also involves classifier induction and performance evaluation. The overall view of the system is shown on Fig. 1.

The fidelity of the computational model can be judged by its classification accuracy. Cross-validation or holdout estimators of true accuracy are typically used during the search [4]. A candidate solution receives a fitness value proportional to the estimated classification accuracy of a corresponding classifier. However, our ultimate goal is to improve the predictive accuracy on a previously unseen sample(s). That is why final performance estimation is necessary for the best solution once the search is completed. Our induction step is classifier specific. For example, induction of a naïve Bayes classifier consists of estimating the probabilities of the feature values and is relatively efficient. For other models, training may be quite computationally consuming, e.g. for neural networks, or especially in GA since it operates with an entire population of solutions. We provide further details on the classifiers used in the experimental results section.

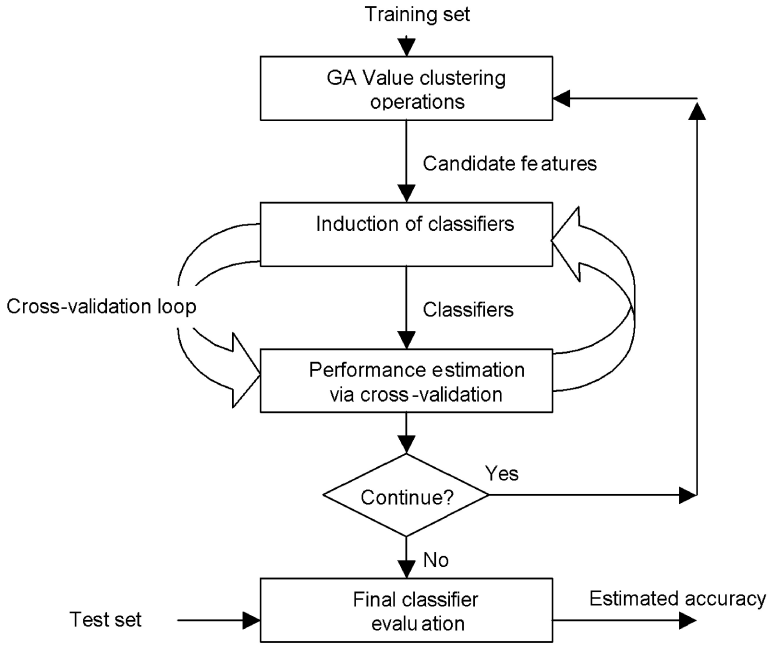

Fig. 1. Feature extraction wrapper-type approach with GA search engine

3.1 Cluster Representation and Genetic Operators

Value clustering can be regarded as a grouping problem, where a number of objects (items) must be placed in several groups. Examples of grouping problems include bin

packing, graph coloring and line balancing, all of which are NP-hard. The search space size for partitioning N objects in M groups grows exponentially with N:

$$\frac{1}{M!}\sum_{i=1}^{M}(-1)^{M-i}\binom{M}{i}i^{N} \tag{4}$$

The major difficulty of typical GA approaches is in designing a representation of a grouping as a chromosome. A good encoding must ensure the complete coverage of the space of possible partitions with minimal redundancy. This encoding should also allow meaningful inheritance of information from parents to children. Finally, it is important that validity checks and chromosomes repair have insignificant computational overhead, if in fact such a repair is necessary.

Several genetic algorithms are specifically known to solve grouping problems. The most straightforward approach uses a standard GA with group-numbers, one gene per object, where each gene contains a group-number for the object. Unfortunately, standard n-point crossover fails to transmit grouping information between chromosomes, since group-numbers do not carry meaning by themselves and could be arbitrarily permuted for any given grouping. This representation also requires knowing the correct number of clusters in advance. Similar difficulties are encountered with permutation encodings. In contrast, GGA algorithm [18] guarantees meaningful inheritance using permutation with separators and a sophisticated crossover operator with repairs.

Graph-based encoding is a different method for solving clustering problems [19]. If there are N items to cluster then a solution is represented by a string with N genes, one gene per item. Given that original items are numbered from 1 to N, then each gene assumes a value between 1 and N. Thus if i-th gene contains value j, then items with numbers i and j belong to the same cluster and can be connected by a link in a graph of all items. In this representation an arbitrary partitioning can be reached without pre-defining a correct number of clusters. No special crossover is required for this representation, since even standard crossover both transfers the item's link and preserves information about the partitioning. However, the redundancy of the encoding is extremely large, since there are N^N possible individuals. Many points in the GA search space correspond to the same cluster partitioning, because a cluster can be formed by any connected graph containing its items. Park and Song [19] proposed a remedy to this redundancy problem, by limiting the number of possible links, based on domain knowledge. For example, if the distance between objects can be defined, then one may restrict the possible links for this object to a set of nearest neighbors. Unfortunately, feature value clustering as well as combinatorial grouping problem in general do not allow introduction of meaningful metrics or definitions of neighborhood for nominal values.

Here we present an improved compact graph-based encoding, where redundancy is reduced exponentially with respect to number of objects N, as compared to the original idea of [19]. The proposed representation keeps all the good qualities of the raw graph-base encoding, without sacrificing its representation power, as we strictly prove below.

First of all, we note that the regular graph-based encoding picks N links out of a total N^2 available, because the chromosome has N genes with N possible values per gene. However, even a complete graph on N vertices has only $N(N\text{-}1)/2$ edges. This results in a major combinatorial explosion in the number of ways to encode the same cluster. Furthermore, to represent any partition of items on a graph we need at most

N-1 edges. Thus an arbitrary tree connecting all the objects within a cluster would be sufficient. Indeed, the largest possible cluster contains *N* items that could be represented by a tree with *N*-1 edges.

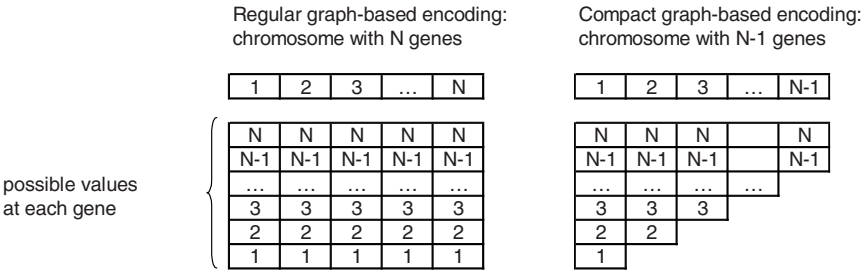

Fig. 2. Regular graph-based clustering GA selects *N* genes from the total of N^2 edges, while compact encoding needs only *N*-1 from *N* (*N*-1)/2 edges. GA search space size is reduced to *N*! from N^N

We propose an enhanced encoding that draws gene values only from *N*(*N*-1)/2 edges and needs *N*-1 genes. The main idea is to remove duplicated edges from a pool of possible gene values. Retaining only values from *N* to *i*, for the *i*-th gene, does this exactly. The first gene assumes values in {1,…,*N*}, the second gene in {2,…,*N*}, etc. The *N*-th gene is unnecessary, since it always is set to *N*, and can be omitted. If the gene value happens to be equal to the gene's number, it means that no edge is placed on the graph. We cannot have less than *N*(*N*-1)/2 edges of a complete graph, because clusters with only two objects must be obtained by a unique link between them. Fig. 2 compares the regular and compact graph-based encoding. Fig.3 shows graph representations of the sample chromosomes.

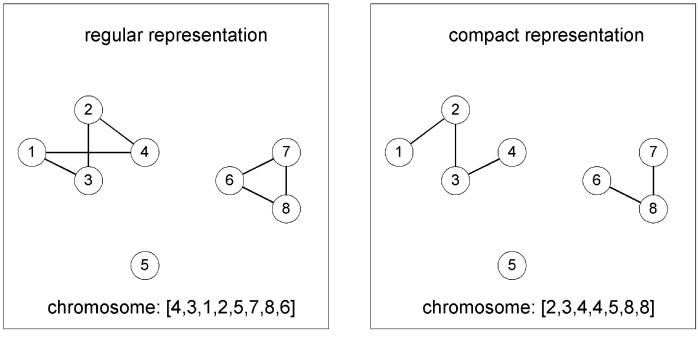

Fig. 3. Sample chromosomes and corresponding clusterings are shown by connected subgraphs for regular and compact encodings

In the compact encoding, each gene obtains its values from an alphabet of a different cardinality. This raises a question as to how required clusters are formed. It is easy to prove that any possible clustering is still attainable in new encoding. For this purpose consider a cluster containing objects with arbitrary numbers. One can sort objects in a cluster by their number. Any such cluster can be formed, by connecting an

object with next higher numbered object from the sorted list, because the required edges are always available: any object has access to edges incident to any object with greater number. Therefore, the proposed encoding has the same representational power at the original encoding but has an exponentially lower space, as we approximately estimate from:

$$\delta = \frac{N^N}{N!} \approx \frac{e^N}{\sqrt{2\pi N}}, \tag{5}$$

where δ is the ratio of sizes of GA search space in old and new encodings. For example, if N=20 we find that δ>43000000, a dramatic reduction.

The compact encoding has one problem, namely different genes carry different information because of the different cardinalities. This could be an issue for a standard crossover (e.g. 1- or 2-point) since lower numbered positions on the chromosome are more important than the others. As an alternative, we offer a modification of the compact encoding that equalizes the cardinality of alphabets for different genes. The genes with the most available edges can transfer some of their edges to other genes, so we still have the very same pool of $N(N\text{-}1)/2$ edges at our disposal. For example, we can delete the value $(N\text{-}1)$ in the alphabet of the first gene, and insert value 1 in the alphabet of $(N\text{-}1)$th gene, leaving us with the same edges. Equalization creates cardinality $1+N/2$ for every gene, and can be done exactly with even N, with minor deviation at some positions when N is an odd number. In this equalized compact representation, the reduction in size of search space is still greater than before, namely:

$$\delta = \frac{N^N}{(1+N/2)^{N-1}} = \left(1+\frac{N}{2}\right)\left(\frac{2}{1+2/N}\right)^N \tag{6}$$

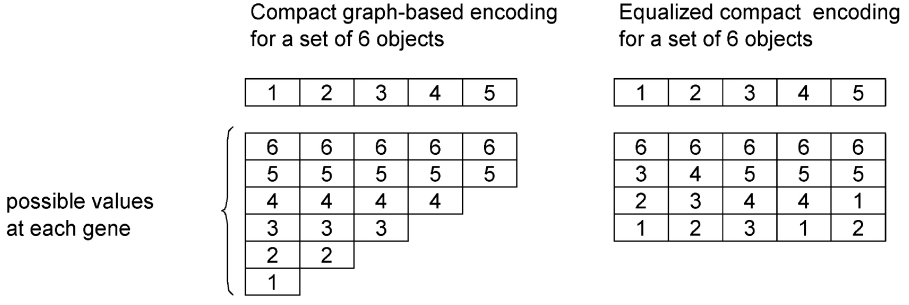

Fig. 4. An example of encoding alphabets at each chromosome position in compact and equalized compact graph-based representations of a set of 6 items

Let us prove that it is possible to realize any clustering in the equalized compact representation. We will show that a cluster with arbitrary objects can be created. The alphabet of the genes (corresponding to any chosen objects) contains all possible links between these objects, because equalization preserves the edges by simple transfer. Now an auxiliary construction is necessary: imagine a complete graph on chosen objects. In this complete graph we can assign direction to the edge from object I to J if the alphabet of gene I contains value J, otherwise the edge goes in the opposite

direction. This directed graph is called a tournament since it is a complete and directed graph. Graph theory provides us with the important fact: every tournament has a Hamiltonian path [30]. Hamiltonian path connects all the objects following the arcs and visits each and every object only once. Therefore, arcs from such a directed path are equivalent to valid assignment of edges to genes. Since arbitrary chosen objects are connected, we can obtain any partitioning.

For our experiments we used equalized compact encoding and 2-point crossover. The next section presents the details of our empirical study of compact encodings for value clustering in some applications.

4 Empirical Study

The main goal of the empirical study is to compare the performance of select classifiers with and without GA value clustering. Even though an overall accuracy improvement is very valuable, we are also interested in reduction of measurement complexity.

4.1 Implementation Details

The driver GA program included the following major steps:

1. Initialization. The population is initialized with random individuals using the equalized compact graph-based encoding as described before. Each individual is created by selecting its genes from the alphabets of respective chromosome positions. For the i-th position, the probability to select the value i is set to $(N\text{-}1)/N$, to prevent formation of a single big cluster covering all the objects. If the value i, is not generated, we initialize this gene to one of the remaining available values (edges) for this position using a uniform probability. Prevention of excessive clustering is motivated by a result from graph theory on critical probability of connectedness of random graphs [31]. Finally, we always seeded a chromosome [1,2,…,N], corresponding to a N non-merged objects (feature values).
2. Fitness evaluation of each individual in the population. Chromosomes are easily decoded to the actual cluster partitioning of feature values as described above. Model parameters are estimated according to the extracted clustering. Classifier accuracy is evaluated using hold-out samples or 10-fold cross-validation (in one experiment).
3. Termination criteria check. The number of generations was the measure of computational effort. The search was stopped after about 200 generations in each run
4. Tournament selection (tournament size = 2) of parents. Pairs are selected at random with replacement and their number is equal to the population size. The better of the two individuals becomes a parent at the next step.
5. Crossover and reproduction. Standard 2-point crossover is used. Each pair of parents produces two offspring. Mutation with small probability p_m=0.01 is applied to each position. In addition, elitism was always used and the best individual of the population survives unchanged.
6. Continue to step 2.

4.2 Test Problems and Results

Two different applications were chosen as benchmarks: text categorization and assignment of individuals to their putative population based on their DNA markers. Feature extraction is very important here, because computations in all these domains suffer from excessive number of features. Certainly, our goal is not necessarily to outperform existing classification algorithms in the area, but rather to demonstrate that GA value clustering effectively reduces feature dimensions as well as improving accuracy in comparison to direct classification.

Text Categorization. Text categorization attempts to automatically place documents into one of a set of predefined classes. Training samples for the experiments were taken from the Reuters-21578 documents collection. We use a classical "bag of words" data model using a naïve Bayes classifier. The number of features is equal to the number of words in a document and each feature can assume any value from the observed dictionary of size N. The dictionary is created from all the training data. The classifier is induced by estimating the probabilities of words in the entire training documents collection. If previously unseen word appears during the inference phase, its probability is set to a small value $0.1/N$. Note that the number of features changes from document to document, while the number of feature values is constant for a fixed training set. It is our goal to cluster the values within features. The probability distribution for each feature is the same due to the independence assumption used in naïve Bayes. Therefore, the clustering of feature values is the same for all features. Thus each GA chromosome represents a clustering of words in the dictionary. The full dictionary of the Reuters data set contains tens of thousands of words (values to be clustered). It is commonly acknowledged that preprocessing is necessary to keep reduce computational costs to a reasonable level. We performed preprocessing by retaining features having the highest mutual information within the category label. Mutual information was assessed independently for each feature. In two different setups we selected 10 and 50 words respectively, corresponding to chromosome lengths with 9 and 49 genes in the equalized compact encoding.

The experiments were designed to distinguish between Reuters articles on "acquisitions" or "earnings" with 2308 and 3785 non-overlapping documents in each category respectively. The known expected accuracy is extremely high, so we made the task more challenging by considering only the first two lines in the body of every document. 1500 documents from each class were split between training and holdout sets (for classifier fitness evaluation), the rest of documents were used to estimate the true accuracy of the best found solution. After the search phase, the best classifier was induced from a combination of training and holdout sets before the final testing. The comparison between the three methods has been made, namely, a naïve non-clustered approach, our proposed clustering GA and a pure best-first hill climbing with the same number of fitness evaluations. We performed 100 runs of each algorithm with different sizes of splits between training and holdout sets. Our GA approach ran for 200 generations with a population of size 100. The results for the estimated true accuracy and percentage of runs with improvement are reported in Table 1.

Table 1. Comparison of test accuracy for the best solutions found by GA value clustering and best-first hillclimbing with "naive" non-clustered solution. First two columns, are training and holdout set sizes. Last column shows percentage of runs when GA found better than naive solution. Each row is the average over 100 runs, with training and holdout sets sampled at random in each run from original data

	Tr. set	Holdout	"Naïve" model	Hill-climbing	Clustering GA	Best runs, %
10 words	400	1100	79.9	79.9	82.3	87
	600	900	79.8	80	81.6	81
The	800	700	79.7	79.9	81.5	78
results	1000	500	79.9	79.9	81.1	74
50 words	600	900	88.2	88.7	90.8	73
	800	700	88.4	88.8	90.5	60

The results show that GA typically improves the predictive accuracy by 1-2% and most runs are successful. Perhaps the main advantage of GA was in dimensionality reduction. The average number of clusters for the 10-words problem was 7.9 and 38.6 for 50-words problem.

DNA Fingerprinting. Assignment of individuals to their putative populations of origin is a practical problem in population genetics. In a classical framework [32], to assign an individual it is necessary to compute likelihood functions for each source population. The population having the highest likelihood value is deemed to be the most probable population of origin. Genetic markers, taken from DNA samples, serve as features and their values (allelic configurations) are assumed to be multinomially distributed [33]. Classification is then performed using a naïve Bayes approach. The large number of possible alleles (e.g.10-20) for each feature seriously complicates the assignment in many practical situations, because the reliability of the estimated parameters is low, since a typical baseline sample consists of only 50-100 individuals. Value clustering is quite promising for binning the alleles together to improve the assignments.

The experiments were done with data set of two lake trout hatchery strains. The data has been collected from trout populations in Lake Seneca and Lake Superior, USA and kindly made available to us by Dr. Kim Scribner. We used 100 individuals per sampled population. Each individual is given as a set of two alleles per locus (feature) with 8 loci in total. While most of the features can have only 2-4 different alleles, there are 10 possible alleles for one of the loci. We ran GA value clustering only for the allelic values at this particular locus. The classification step and GA search are very similar to what was done for text categorization. However, the probabilities of individuals are computed differently, because for trout genotypes there are two alleles specified at each locus (feature). Namely, the probability of allelic configuration (i,j) at each locus is estimated as $2p_ip_j$, if alleles i and j are different, or as p_i^2, if $i=j$, where p_i is an estimate of i-th allele probability at a locus. This modification of the regular naïve Bayes classifier is trivial and appears only in computations of fitness for candidate solutions. Also the fitness was estimated by 10-fold cross-validation due to the limited number of individuals in the original data set. As a main result, the value clustering GA was able to improve classification accuracy (cross-validated) from 77.5% to 81.0%. In most runs, the best solution for a locus was consistently represented by only 6 meta-values instead of the 10 original values.

5 Future Work and Conclusion

The novelty of the approach is in adopting a genetic algorithm for clustering the values of variables. In contrast, traditional genetic algorithm approaches for data mining operate on the features as a whole by including/excluding them from a subset or adjusting the appropriate weights. We make a step further and organize the search in the entire input space. The enhanced graph-based encoding has a much less redundant chromosome while maintaining complete coverage of possible partitions, as we strictly prove. Performed experiments demonstrate reduction in measurement complexity and improvement in classification accuracy due to the better reliability of meta-values.

In our future work we want to generalize and apply genetic clustering to a difficult problem of parametric learning in the context of Bayesian networks. Bayesian network classifiers, even augmented to the trees over the feature nodes, requires estimation of a much greater number of parameters than a typical naïve Bayes classifier. We optimistically expect that the conditional probability distribution in a Bayesian network can be considerably simplified and learned from data with the help of GA value clustering.

Acknowledgements. The authors are grateful to Dr. Kim Scribner for providing DNA data samples. The work of Alexander Topchy has been supported by graduate research award from the Research Excellence Fund Center for Biological Modeling at Michigan State University.

References

1. Devijver, P.A. and Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice-Hall International, (1982)
2. Jain A.K., Duin R.P. and Mao J, Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, (2000) 4–37
3. Freitas A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer-Verlag, (2002)
4. Kohavi R. and John G.: Wrappers for Feature Subset Selection. Artificial Intelligence Journal 97 (1–2), (1997) 273–324
5. Siedlecki W. and Sklansky J.: On automatic feature selection. International Journal of Pattern Recognition and Artificial Intelligence, 2, (1988) 197–220
6. Vafaie H., and De Jong K.: Robust feature selection algorithms. In Proc. of the 5th IEEE International Conference on Tools for Artificial Intelligence, Boston, MA, (1993) 356–363
7. Whitley D., Beveridge R., Guerra C. and Graves C.: Messy Genetic Algorithms for Subset Feature Selection. International Conference on Genetic Algorithms. T. Baeck, ed. Morgan Kaufmann, (1997)
8. Yang J. and Honavar V.: Feature Subset Selection Using a Genetic Algorithm. In: Feature Extraction, Construction, and Subset Selection: A Data Mining Perspective. Motoda, H. and Liu, H. (Eds.) New York, Kluwer, (1998)
9. Punch W.F., Goodman E.D., Pei M., Chia-Shun L., Hovland P. and Enbody R.: Further Research on Feature Selection and Classification Using Genetic Algorithms. In Proc. 5th International Conference on Genetic Algorithms, Urbana-Champaign IL, (1993) 557–562
10. Raymer M., Punch W., Goodman E., Sanschagrin P., and Kuhn L., Simultaneous Feature Extraction and Selection using a Masking Genetic Algorithm. In Proc. of 7th International Conference on Genetic Algorithms (ICGA), San Francisco CA, (1997) 561–567

11. Vafaie H. and DeJong K.: Feature Space Transformation Using Genetic Algorithms. IEEE Intelligent Systems 13(2), (1998) 57–65
12. Lin C. and Wu J.: Automatic facial feature extraction by genetic algorithms. IEEE Trans. on Image Processing, vol. 8(6), (1999) 834–845
13. Raymer M.L., Punch W.F., Goodman E.D., Kuhn L.A. and Jain A.K.: Dimensionality Reduction Using Genetic Algorithms. IEEE Trans. on Evolutionary Computations 4(2), (2000) 164–171
14. Brumby S.P., Theiler J., Perkins S.J., Harvey N.R., Szymanski J.J., Bloch J.J., and Mitchell M.: Investigation of Feature Extraction by a Genetic Algorithm. Proc. SPIE 3812, (1999) 24–31
15. Larsen O., Freitas A.A. and Nievola J.C.: Constructing X-of-N attributes with a genetic algorithm. In Proc. 4th Int. Conf. on Recent Advances in Soft Computing, (2002) 326–331
16. Pudil P. and Novovicová J.: Feature Subset Selection Using a Genetic Algorithm in Feature Extraction. In: Huan Liu, Hiroshi Motoda (eds.): Construction and Selection: A Data Mining Perspective, Kluwer (1998)
17. Martin-Bautista M. and Vila M.-A.: A survey of genetic feature selection in mining issues. In Proceedings of the Congress on Evolutionary Computation (CEC 99), (1999) 13–23
18. Falkenauer E., Genetic Algorithms and Grouping Problems. John Wiley & Son Ltd., (1998)
19. Park Y-J. and Song M-S.: A genetic algorithm for clustering problems. In Proc. 3rd Annual Conf. on Genetic Programming, (1998) 568–575.
20. Trunk, G.V.: A problem of dimensionality: a simple example. IEEE Trans. Patt. Anal. Mach. Intell. 1, (1979) 306–307
21. Minker, J., Wilson, G.A., Zimmerman, B.H., An evaluation of query expansion by the addition of clustered terms for a document retrieval system. Information Storage and Retrieval 8(6), (1972) 329–348
22. Spark-Jones K. and Jackson D.M.: The use of automatically-obtained keyword classifications for information retrieval. Information Processing and Management 5, (1970) 175–201
23. Merzbacher M. and Chu W. W.: Pattern-based clustering for database attribute values. In Proc. of AAAI Workshop on Knowledge Discovery in Databases, Wash., D.C., (1993)
24. Tishby N., Pereira F.C., and Bialek W.: The information bottleneck method. In Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing, (1999) 368–377
25. Slonim N. and Tishby N.: Agglomerative Information Bottleneck. In Advances in Neural Information Processing Systems (NIPS–12), MIT Press, (1999) 617–623
26. Friedman N., Mosenzon O., Slonim N., and Tishby N.: Multivariate Information Bottleneck. In Proc. of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI), (2001)
27. Slonim N., Friedman N., and Tishby N.: Agglomerative Multivariate Information Bottleneck. In Advances in Neural Information Processing Systems (NIPS–14), (2001)
28. O'Connell J.R. and Weeks D.E.: The VITESSE algorithm for rapid exact multilocus linkage analysis via genotype set-recoding and fuzzy inheritance. Nature Genetics 11, (1995) 402–408
29. Friedman N., Geiger D., and Lotner N.: Likelihood Computation with Value Abstraction. In Proc. Sixteenth Conf. on Uncertainty in Artificial Intelligence (UAI), (2000)
30. Chartrand, G. and Oellermann O.R.: Applied and Algorithmic Graph Theory. McGraw-Hill, Inc., New York (1993)
31. Bollob'as B.: Random Graphs. Academic Press, London, (1985)
32. Waser P.M. and Strobeck C.: Genetic signatures of interpopulation dispersal. Trends Ecol Evol 13, (1998) 43–44
33. Guinand, B., Topchy A., Page K.S., Burnham-Curtis M.K., Punch W.F., and Scribner K. T.: Comparisons of likelihood and machine learning methods of individual classification. Journal of Heredity 93(4), (2002) 260–269