Using Prior Knowledge to Improve the Performance of an Estimation of Distribution Algorithm Applied to Feature Selection

Leonardo R. Emmendorfer[1]
Doctoral Programme in Numerical Methods for Engineering – UFPR[2]

Rodrigo Traleski[1]
Department of Informatics UFPR[2]

Aurora Trinidad Ramirez Pozo[1]
Department of Informatics UFPR[2]

Abstract

Feature selection provides a great enhancement in the process of building a classifier model. A recent approach to feature selection is the use of Estimation of Distribution Algorithms (EDAs). Those algorithms's performance is greatly affected by the initial population, so prior knowledge about the problem is very important. The most important prior knowledge about the features is the relative order of importance observed among them, which can be obtained by some statistical measure. Based on the use of that kind of knowledge, some improvements are proposed and theoretically discussed. An experiment is presented, which evaluates potential benefits of those alternatives.

1. Introduction

In a data mining context, classification is one of the most important steps in the whole process of implicit knowledge acquisition from data sets and databases. An instance in a data set is composed of predictive attributes, which may be continuous or discrete, and a discrete class attribute. Classification allows predicting the class of future instances, based on the (somehow learned) relationship between the predictive attributes and the class attribute. In that context, feature selection methods are used in situations where the number of attributes is high. The idea is to apply a feature selection algorithm before obtaining the classification model.

Evolutionary Algorithms (EAs) have proved to be effective in many search and optimization problems. EA's are iterative algorithms based on the idea of natural selection. A population of candidate solutions is evaluated and the best individuals will be chosen in order to create the next population. Genetic Algorithms (GAs) is an important class of EA. Operators like crossover and mutations are used to guarantee variability in the population.

Estimation of Distribution Algorithms (EDAs) comprises another class of EAs and have already been applied to the feature selection task [2]. The difference among EDAs and other EAs is on how iterative evolutionary process is done; instead of applying crossover and mutation operators directly over the best individuals, EDAs create the next generation using a probabilistic model derived from the best individuals selected from the current population.

Probabilistic models used in EDAs are often very simple. Actually, the most known EDAs assume independence among variables, and obtain the probability of a gene being 1 in each generation just by the observed frequency of 1s. In the other hand, the most complex class of EDAs can suitably treat relationships of order *n* among the variables (genes). In that case, when Bayesian Networks are used as the probabilistic model of the EDA, we should generally call these methods BNEDAs (Bayesian network-based Estimation of Distribution Algorithms). Through BNEDAs, the relationship between genes is explicitly learnt and maintained, so the problem of building-block disruption is avoided [3]. Therefore, they are very promising approaches to feature selection, since it is essential to recognize which interactions can actually increase the classification accuracy. A drawback is the high computational cost of creating Bayesian networks. The most representative BNEDA is BOA – Bayesian Optimization Algorithm [4].

Our main purpose is to propose and validate improvements in the use of BNEDAs for feature selection, essentially by incorporating prior knowledge about the problem. The major motivations are (i) reducing the computational cost of the method, by avoiding unproductive search regions, and (ii) increasing the accuracy of the subsets obtained.

[1] {leonardo,rodrigo,aurora}@inf.ufpr.br
[2] Federal University of Paraná (Brasil)

The text is organized as follows. Section 2 presents the problem of feature selection and some discussion is done about the most popular approaches. Section 3 explains theoretical foundation of BNEDAs, focusing on how probabilistic models can be useful in evolutionary search and optimization algorithms. Section 4 discuss about the benefits and weakness of BNEDAs when applied to our specific problem of feature selection, and leads to some insight about improving the basic method by using some prior knowledge. The content of that prior knowledge and the methodology for incorporating it during the evolutionary process is discussed in section 5. Finally, an experiment is proposed in section 6 and the results and potential future improvements are discussed in section 7.

2. Feature Selection

When a high number of characteristics are available, then the creation of a classifier model may be harmed, either by the presence of redundant attributes or irrelevant ones. For that reason, feature selection is very relevant. An appropriate subset must be selected, in order to maximize the accuracy of classification. Particularly, that problem becomes more critical when the number of instances is small when compared to the number of attributes, as often happens in bioinformatics problems, for instance. A smaller number of features may also increase model's comprehensibility.

A very intuitive and few sophisticated approach is the use of filters. The key idea about filtering is to evaluate all attributes, obtaining some measure of their quality as good classifiers. Such measures may be orthogonality, high content of information, among others [5]. So only after the subset selection step is done, then the predictive function is built.

In the other hand, feature selection itself may be viewed as an optimization problem, where the objective is to maximize the accuracy of the classification, when only a subset of all the predictive attributes is used. The problem of finding the best feature subset can be solved by a wrapper [5]. It involves the use of a black box optimization algorithm, adopting a heuristic (like in BOA) to explore the search space, evaluating each candidate subset of features by the inductor function, which plays the role of the black box.

Although filtering is a much less computationally expensive method, it was shown that some wrappers are more effective in maximizing the accuracy [10] [5], due to better exploitation of the search space. But the wrappers potentially expend much more computer time, as they must apply the inductor algorithm to evaluate each candidate feature subset.

For that reason, it is very important to avoid unproductive subsets, especially during the initial stages of evolutionary process. This emphasizes the importance of using prior knowledge in our context.

3. Estimation of Distribution Algorithms

Evolutionary Algorithms (EAs) have been successfully used to solve a wide range of optimization and search problems. Maybe the Genetic Algorithms (GA) are the most intuitive evolutionary approach. In GA, individuals are explicitly represented in all steps of evolutionary process, and evolution occurs through the use of operators like crossover and mutation. Unfortunately, GAs do not deal properly with the problem of building-block disruption. Instead, crossover and mutation operators actually split related genes in chromosomes [12].

Those reasons motivated the creation of a new kind of algorithm, generally called Estimation of Distribution Algorithm (EDA) [7]. In EDA, new populations are created without the use of crossover and mutation operators. The genetic pool is coded as a distribution of the search space; in each generation the best individuals are selected in order to estimate a distribution of the genes. All individuals of the next generation are created based in that distribution, what completes the cycle.

The following pseudo-code shows a generic scheme of the EDA approach:

1. An initial population D_0, composed by R individuals, is randomly created. Each individual has n genes.
2. In order to create the D_{m+1} population, a number of K $(K < R)$ individuals are selected from D_m, according to a criterion. We call S_m the subset of K selected individuals from generation m.
3. A n-dimensional probabilistic model is inferred from S_m. It is expected that it will better represent the relationships among the n variables. This step is known as the *learning* procedure, and it is crucial to the evolutionary process.
4. Finally, the new population D_{m+1} is obtained after the simulation from the distribution learnt in the previous step. A total of R individuals are generated in this step.

The number of relationships among the variables affects the complexity of learning a model from the population. In the simplest class of models, no relation is found among the variables. In a second class, only paired relations are allowed. The last is the most complex class, and can deal with relationships of higher level. Learning this last kind of model is a NP-

hard problem. This last class comprises Bayesian network-based EDAs (BNEDAs). In this kind of EDA, Bayesian networks are used as probabilistic models representing higher order of relationships among variables. A Bayesian network learning algorithm like K2 [9] must be used in the process. K2 employs a greedy search algorithm to search over possible Bayesian networks, which are represented as graphs. In each iteration, a new relation among two variables is inserted (an arc in the graph). For each new relation, the quality of the network is evaluated, using the Dirichlet metric, until no insertions are seemed to improve the quality of the network. The K2 algorithm is the most representative of the search-and-score approach.

4. Applying BNEDAs to the feature selection task

In this paper, three BNEDAs will be proposed to the feature selection task, in order to take better advantage of prior knowledge about the features and the relations among them. However, a more general description must be done first. Therefore, this section introduces the application of BNEDA's to feature selection.

Population chromosomes can be represented as binary variables. Each gene represents an attribute; if the attribute is selected, then its corresponding gene will be set to "one", otherwise the value will be "zero". The initial population can be randomly obtained from an uniform distribution of the value of each gene.

The fitness can be measured by the mean accuracy. Suppose Naivebayes or another classifier model can be chosen. Since most data sets are relatively small, a 10-fold cross validation is applied to obtain each fitness evaluation. In a k-fold cross validation, the whole data set D is randomly divided in k non-overlapping subsets, $D_1,...,D_k$. On each iteration i (i=1 to k), the network is trained with $D - D_i$ and tested with D_i.

The procedure is repeated at most 5 times, or until a standard deviation of 1% is reached. The final result is the mean accuracy over all iterations. The same empirical methodology for fitness evaluation is also adopted by [1] and [2].

In [2], BOA is applied to feature selection. An experiment is made, where BOA is compared to other methods. Results show that selecting features using BOA will be at least equivalent to doing no feature selection at all, and so the authors do not recommend BOA, since other faster methods reached better results.

Later in this paper, BOA is used as a reference method. As a general description, BOA is comprised by (i) a random initial population generator; (ii) K2 as the learning algorithm, which will infer the Bayesian network in each step and finally (iii) a Bayesian network sampler, used to produce the next population from the subsequent networks learned. Probabilistic Logic Sampling algorithm (PLS) [14] is chosen for that last task.

The effectiveness of each attribute selection during BOA execution is evaluated using Naive Bayes, available in VFML library [6]. It employs a simplified version of Bayes formula to decide to which class a test instance belongs. Naive Bayes learns through estimating the probability of attribute values within each class, by straight frequency counts. Since K2 deals only with discrete attributes, then continuous ones were discretized, using Gauss normalization, also available in VFML.

5. Description of the methods proposed

The methods herein proposed are a result of a theoretical prospect for potential improvements in the application of BNEDAs to the problem of feature selection, essentially by incorporating prior knowledge about the problem, which may produce good individuals early in the initial populations, therefore potentially increasing accuracy and reducing the number of generations.

Three methods are proposed, all of them based in the general description of BNEDA of the last section. Each of these improvements uses prior knowledge in a diverse manner, but all need to obtain a list of features, ordered by their importance as classifiers. The chi-square statistic was used to infer the relative importance of each predictive attribute related to the class attribute.

The first method proposed - Mop - incorporates prior knowledge in the evolutionary process in two manners: (i) the distribution of genes in the initial population is not uniformly random, but it takes into account the order of importance of attributes, obtained by the chi-square statistic. (ii) the same order of attributes is also used as the ancestral ordering for K2, the Bayesian network learning algorithm used in BOA.

Through (i) we expected to get a more adjusted initial population, emphasizing the relative presence of more important attributes than less important ones. This can be done simply by setting the expected frequency of each attribute in the initial population as being inversely proportional to the sequential index it receives after ordering. For example, in a population of 1000 individuals with 5 attributes, the attribute number one (the most important) would be present in 100% of individuals, whereas the attribute number 2 would be present in 80% of population, and so on, until the number 5, which would appear in just 20% of the population. On the other hand, through (ii) we hope to reach better subsets, since BOA originally adopts a

random order of the variables, which surely guarantees some degree of diversity, but the best networks may never be exploited.

The second candidate improvement is Mpn. It involves learning the structure of an initial Bayesian network *a priori*, directly from the data set, before the evolutionary process starts. That prior Bayesian network can be captured from the data set by the K2 algorithm.

It is expected that the relation among genes in the EDA will be similar to the relation among the attributes represented by those genes. That approach would potentially reduce the computational effort, since K2 (called from the EDA) would not need to explore previously defined relationships among attributes, since they were already observed and extracted from data set.

Finally, Mppn does exactly the same as Mpn, except for the fact that 50% of the prior network is pruned, so only the best variables are preserved and used as prior knowledge by the evolutionary algorithm. This allows the EDA to search for other potential relationships among attributes.

6. Experimental Methodology

This section describes the experiments performed. The purpose of experiments is to verify if the use of prior knowledge actually impacts on the performance of the proposed BNEDA's when applied to 8 data sets from UCI collection [11]. Mean accuracy is observed, and it is straightforward to verify if significant increase occurs, comparing each of the proposed methods to the original BOA (and to the other reference methods) using paired two-sided *t* tests for each comparison, at a 95% confidence level. Table 1 summarizes the data sets used in experiments.

Table 1. Data sets used in experiments.

Domain	Instances	Numeric features	Discrete features	Classes
Annealing	898	6	32	6
Breast-Cancer	699	0	9	2
Heart-c	303	6	7	5
Horse	368	7	15	2
Ionosphere	351	34	0	2
Segmentation	2310	19	0	7
Sick Euthyroid	3163	7	18	2
Soybean Large	683	0	35	19

The mean accuracy of the three methods proposed is also compared to other two references, in order to allow a better discussion about the results obtained. These reference methods are (i) do no feature selection at all, therefore using all attributes (called the ALL method) and (ii) use a Hill-climbing, non-evolutionary wrapper (called HILL), from the WEKA library [13].

The data sets differ in some aspects. The number of instances ranges widely, from 303 to 3163, and the characteristics of data vary from the numerical majority to a categorical dominance. The choice of so diverse data sets was motivated by the explicit intention of representing situations which can be found in practical applications, and therefore be able to discuss robustness of the proposed methods. Also, these data sets are often used in benchmark evaluations related to feature selection [15].

Proposed and reference methods are evaluated through a 5-fold cross validation, followed by a 2-fold. In each of five iterations the data set was divided in half - namely halves 1 and 2. Initially, half 1 is used for training and half 2 for validation; afterward the opposite occurs. Therefore, each wrapper is executed 10 times for each data set. For each set of attributes obtained by a wrapper, the performance must be calculated, by recording the accuracy reported by Naive Bayes, for each of the resulting subsets. Accuracy is averaged for each method, when applied to each data set. In order to allow performing paired *t*-tests, the same 10 partitions are preserved and used in all methods.

7. Results and Conclusions

Initially, mean accuracy obtained with the reference methods – ALL, HILL and BOA – and with the proposed methods – Mop, Mpn and Mppn - are presented in tables 2 and 3 respectively. Next, the p-values resulting from all *t* tests are shown in tables 4 thru 6. P-values are in bold when statistical significant difference occurs, at a 95% confidence level. It is important to highlight that the proposed methods always reached at least the equivalent results when compared with each of the reference methods.

Moreover, after analyzing confidence intervals (not shown) we can conclude that a proposed method is never significantly worse that any reference method, in any data set, when considering accuracy. Otherwise, all proposed methods seem to overperform the reference methods in most data sets. Mop presented slightly better results than the other methods.

We can also verify that more significant results were obtained for data sets with a higher number of attributes. This suggests that the proposed methods are actually performing better exactly when feature selection is more relevant. Further work may confirm that trend.

Table 7 compares the mean number of generations of BOA, Mop, Mpn and Mppn. Although Mop

presented a smaller number of generations for all data sets when compared to BOA, this reduction is not very statistically significant for most data sets.

Since additional computational effort is required to learn an initial Bayesian network, then Mpn and Mppn may not be as advantageous as expected, considering also that they did not reach better results than Mop. Thus, our experiment brings some evidence that Mop is the best choice among the methods compared here, but further work should be done.

Table 2. Mean accuracy over data sets, using reference methods

Data set	ALL	HILL	BOA
Annealing	94.89	91.25	93.86
Breast-Cancer	96.27	96.39	96.79
Heart-c	81.66	79.07	80.66
Horse	79.93	82.73	82.95
Ionosphere	90.29	89.54	89.73
Segmentation	89.93	90.19	86.52
Sick Euthyroid	91.70	94.31	91.48
Soybean Large	88.57	88.36	79.38

Table 3. Mean accuracy over data sets, using all methods tested

Data set	Mop	Mpn	Mppn
Annealing	95.50	95.59	95.34
Breast-Cancer	97.08	97.02	96.87
Heart-c	82.02	82.06	82.26
Horse	83.81	83.27	83.03
Ionosphere	91.38	91.27	91.15
Segmentation	91.04	90.85	91.05
Sick Euthyroid	94.97	94.95	94.97
Soybean Large	91.02	91.03	90.82

Table 4. p-values resulting from paired two-sided t tests comparing each method to ALL*

Data set	Mop	Mpn	Mppn
Annealing	0.091	**0.026**	0.184
Breast-Cancer	**0.032**	0.052	0.109
Heart-c	0.451	0.451	0.202
Horse	**<0.001**	**0.001**	**0.001**
Ionosphere	0.106	0.082	0.101
Segmentation	**0.001**	**0.001**	**0.001**
Sick Euthyroid	**<0.001**	**<0.001**	**<0.001**
Soybean Large	**<0.001**	**<0.001**	**0.001**

* statistically significant difference (p-value<0.05) in bold

Table 5. p-values resulting from paired two-sided t tests comparing each method to HILL*

Data set	Mop	Mpn	Mppn
Annealing	**0.004**	**0.001**	**0.004**
Breast-Cancer	**0.001**	**0.001**	**0.033**
Heart-c	**0.039**	**0.039**	**0.022**
Horse	0.173	0.567	0.729
Ionosphere	**0.002**	**0.017**	**0.011**
Segmentation	0.115	0.319	0.144
Sick Euthyroid	**0.022**	0.056	**0.027**
Soybean Large	**<0.001**	**<0.001**	**0.001**

* statistically significant difference (p-value<0.05) in bold

Table 6. p-values resulting from paired two-sided t tests comparing each method to BOA*

Data set	Mop	Mpn	Mppn
Annealing	0.157	0.084	0.167
Breast-Cancer	0.060	0.149	0.704
Heart-c	0.151	0.151	0.102
Horse	**0.037**	0.668	0.896
Ionosphere	**0.009**	**0.014**	**0.034**
Segmentation	**<0.001**	**<0.001**	**<0.001**
Sick Euthyroid	**<0.001**	**<0.001**	**<0.001**
Soybean Large	**<0.001**	**<0.001**	**<0.001**

* statistically significant difference (p-value<0.05) in bold

Table 7. Mean number of generations over data sets, using all methods tested

Data set	BOA	Mop	Mpn	Mppn
Annealing	5.90	3.8*	5.30	3.8*
Breast-Cancer	9.20	6.4*	8.00	9.00
Heart-c	5.80	3.30	7.30	8.60
Horse	4.10	3.40	3.50	4.10
Ionosphere	6.10	4.60	4.80	5.60
Segmentation	4.50	3.40	5.90	5.20
Sick Euthyroid	2.60	2.10	4.00	6.70
Soybean Large	2.00	1.90	4.70	3.40

*statistically significant reduction, when compared to BOA

Some future improvements can be considered, even analyzing other data sets, or proposing the use of an entropy-based variable ordering algorithm (for instance) or adopting other classifier algorithms, which could deal with continuous variables directly. Additionally, other algorithms can be used for inferring individuals from the Bayesian network, instead of PLS. Unsupervised Bayesian learning algorithms should also be tested.

8. References

[1] Inza, I., Larrañaga, P., Etxeberria, R., and Sierra, B. "Feature subset selection bye Bayesian networks based on optimization", *Artificial Inteligence*, 1999, pp. 157-184.

[2] Cantú-Paz, E., "Feature subset selection by estimation of distribution algorithms", *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, Morgan Kaufmann Publishers, San Francisco, 2002, pp. 303-310.

[3] Thierens, D., "Scalability problems of simple genetic algorithms", *Evolutionary Computation*, 1999, pp. 331-352.

[4] Pelikan, M., Goldberg, D. E., and Cantú-Paz, E., "The bayesian optimization algorithm", *Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann Publishers, San Francisco, 199, pp. 525-532.

[5] John, G., Kohavi, R., and Phleger, K., "Irrelevant features and the feature subset problem", *Proceedings of the 11th International Conference on Machine Learning*, Morgan Kaufmann Publishers, San Francisco, 1994, pp. 121-129.

[6] Hulten, G. and Domingos, P. "VFML - A toolkit for mining high-speed time-changing data streams", http://www.cs.washington.edu/dm/vfml, 2003.

[7] Larrañaga, P., and Lozano, J. A., *Estimation of distribution algorithm. A new tool for evolutionary computation,* Kluwer Academic Publishers, 2001.

[8] Heckermann, D., Geiger, D. and Chickering, D. M., "Learning bayesian networks: The combination of knowledge and statistical data", *KDD Workshop*, 1994, pp. 85-96.

[9] Cooper, G. F. and Herskovits, E., "A bayesian method for the induction of probabilistic networks from data", *Machine learning*, vol. 9, 1992, pp. 309-347.

[10] Rohn Kohavi and George H. John, "Wrappers for feature subset selection". *Artificial Intelligence*, vol. 97, 1997, pp. 273-324.

[11] C. Blake, E. Keogh, and C.J. Merz, "UCI Repository of Machine Learning Data Bases", University of California, Department of Information and Computer Science, Irvine, CA, 1998.

[12] Holland, J. H., *Adaptation in natural and artificial systems*, The University of Michigan Press, 1975.

[13] Witten, I,.H. and Frank, E., *Data mining: Practical machine learning tools and techniques with Java implementations.* Morgan Kaufmann, San Francisco, CA, 2000.

[14] M. Henrion. "Propagating uncertainty in Bayesian networks by probabilistic logic sampling" *Uncertainty in Artificial Intelligence*, vol. 2, 1998, pp. 149-163.

[15] M.A.Hall and G. Holmes. "Benchmarking Attribute Selection Techniques for Discrete Class Data Mining", IEEE Transactions on Knowledge and Data Engineering, vol. 15, n. 3, 2003, pp. 1437-1447.