

SEMI-SUPERVISED CLUSTERING USING GENETIC ALGORITHMS

AYHAN DEMIRIZ

Dept. of DSES, RPI
Troy, NY 12180

Email: demira@rpi.edu

K. P. BENNETT

Dept. of Math. Sci., RPI
Troy, NY 12180

Email: bennek@rpi.edu

M. J. EMBRECHTS

Dept. of DSES, RPI
Troy, NY 12180

Email: embrem@rpi.edu

ABSTRACT

A semi-supervised clustering algorithm is proposed that combines the benefits of supervised and unsupervised learning methods. The approach allows unlabeled data with no known class to be used to improve classification accuracy. The objective function of an unsupervised technique, e.g. K-means clustering, is modified to minimize both the cluster dispersion of the input attributes and a measure of cluster impurity based on the class labels. Minimizing the cluster dispersion of the examples is a form of capacity control to prevent overfitting. For the the output labels, impurity measures from decision tree algorithms such as the Gini index can be used. A genetic algorithm optimizes the objective function to produce clusters. Experimental results show that using class information improves the generalization ability compared to unsupervised methods based only on the input attributes.

1 INTRODUCTION

In this work, we examine an approach to solve classification problems that combines supervised and unsupervised learning techniques. In supervised learning, we are given a set of labeled training points, and the task is to construct some function that will correctly predict the labels of future points. In unsupervised learning such as clustering, the task is to segment unlabeled training data into clusters that reflect some meaningful structure in the data. For the Semi-Supervised Learning Problem (SSLP) we assume that we are given both labeled and unlabeled points. The task is then to predict the labels of the unlabeled points using all the available labeled data, as well as unlabeled data. One would expect a better generalization ability of the resulting classifier due to the better understanding of the input distribution resulting from using all the available data. The semi-supervised learning problem can be used to perform transductive inference (Vap98). In transduction, we are given a set of training data and a set of testing data, and the learning task is to predict the labels of the specific testing data only. Different testing data will produce different classification functions. The intuition is that transduction is a simpler problem because we are trying to construct a function that is valid only at specific points, versus in induction where the goal is to construct a function valid at all future testing points.

In this paper we take the approach of incorporating label information into an unsupervised learning approach. Our goal is to group both labeled and unlabeled data into the clusters where each cluster is as pure as possible in terms of class distribution provided by labeled data. The advantage of such an approach is that it can be used for both inductive and transductive inference. In addition the clusters help characterize segments of the population likely or unlikely to possess the target characteristic represented by the label.

As a base to our semi-supervised algorithm, we use an unsupervised clustering method optimized with a genetic algorithm incorporating a measure of classification accuracy used in decision tree algorithms, the GINI index (BFOS84). Here we examine clustering algorithms that minimize some objective function applied to k-cluster centers. Each point is assigned to the nearest cluster center by Euclidean distance. The goal is to choose the cluster centers that minimize some measure of cluster quality. Typically a cluster dispersion metric is used. If the mean square error, a measure of within cluster variance, is used than the problem becomes similar to the classic K-means clustering (JD88). We also experimented with an alternative metric, the Davis-Bouldin Index (DB79), that is a function of both the within

cluster variance and between cluster center distances. By minimizing an objective function that minimizes a linear combination of the cluster dispersion measure and the Gini Index, the algorithm becomes semi-supervised. The details of the problem formulation are given in Section 2. Since the objective function is highly nonlinear and discontinuous with many local minima, we optimize it using the C++ based genetic algorithm (GA) library package GALib (Wal96).

GAs are well known for being able to deal with complex search problems by implementing an evolutionary stochastic search. Because of this, GAs have been successfully applied to a variety of challenging optimization problems. The NP-hard nature of the clustering problem makes GA a natural choice for solving it such as in (BHBC96; SYK97; MC96; Cuc98). A common objective function in these implementations is to minimize the square error of the cluster dispersion:

$$E = \sum_{k=1}^K \sum_{x \in C_k} \|x - m_k\|^2 \quad (1)$$

where K is the number of clusters, m_k is the center of cluster C_k . This is indeed the objective function for the K-means clustering algorithms. The algorithm proposed in (SYK97) modifies this objective function by using the inverse of Davies-Bouldin index defined in (DB79; JD88) and minimizing it by using evolutionary programming.

In Section 2, the problem definition and the proposed algorithm are given. Details about the GA implementation are given in Section 3. Experimental results are given in Section 4. A comparison with 3 nearest neighbor and linear and quadratic discriminant analyses is also reported in Section 4. Finally, we summarize our findings in the Section 5.

Related approaches for combining supervised and unsupervised learning exist. For example LVQ (Koh88) and CTM (CKM97) also use the approach of adapting a primarily unsupervised method to perform classification. This paper helps address the interesting, but still open question, of how well such methods can exploit the information in unlabeled data to support transductive inference.

2 PROBLEM DEFINITION

Clustering, in general, is defined as grouping similar objects together by optimizing some similarity measure for each cluster such as the within group variance. Since clustering generally works in an unsupervised fashion, it is not necessarily guaranteed to group the same type (class) of objects together. In this case, we need to introduce supervision to the our learning scheme through some measure of cluster impurity. The basic idea is to find a set of clusters then minimize a linear combination of the cluster dispersion and cluster impurity measures. More specifically, select $K > 2$ cluster centers, m_k ($k = 1, \dots, K$), that minimize the following objective function:

$$\min_{m_k, k=1, \dots, K} \beta * Cluster_Dispersion + \alpha * Cluster_Impurity \quad (2)$$

where $\alpha > 0$ and $\beta > 0$ are positive regularization parameters.

If $\alpha = 0$, then the result is a purely unsupervised clustering algorithm. If $\beta = 0$ the result is a purely supervised algorithm that tries to minimize the cluster impurity. As in the K-means algorithm, each point is assumed to belong to the nearest cluster center as measured by Euclidean distance. Each non-empty cluster is assigned a class label corresponding to the majority class of points belonging to that cluster. Two cluster dispersion measures from the clustering literature will be examined: mean square error (see (DBE99)) and Davis-Bouldin Index (see (DBE99)). It is important to note that for the induction case, cluster dispersion is based on the labeled training data. For the transduction case, the cluster dispersion is based on all available data, both labeled and unlabeled. For the cluster impurity measure, a measure of partition quality common in decision tree algorithms, the Gini Index, is used (see (DBE99)). In this case, Gini Index of a certain cluster is computed as: $GiniP_j = 1.0 - \sum_{i=1}^k (\frac{P_{ji}}{N_j})^2$ j in $1, \dots, K$ where P_{ji} is the number of points belong to i^{th} class in cluster j . N_j is the total number of points in cluster j . The impurity measure of a particular partitioning

into K clusters is: $impurity = \frac{\sum_{j=1}^K T_{P_j} * GiniP_j}{N}$ where N is the number of points in the dataset. Since the objective function (Eq.2) is highly discontinuous with many local minima, we optimize it using the genetic algorithm library (see Section 3) GA-Lib. When a solution is found, it might contain clusters with little or no points assigned to them. These clusters are deleted and relevant points are reassigned to their nearest cluster centers. Practical details of this algorithm are discussed in the computational results section (see Section 4). The resulting algorithm can be summarized as follows:

Algorithm 1. *Semi-supervised clustering algorithm*

Within genetic algorithm:

1. *Determine cluster centers*
2. *Partition the labeled data by distance to closest cluster center.*
3. *Find non-empty clusters, assign a label to non-empty clusters by majority class vote within them.*
4. *Compute dispersion and impurity measures:*

Induction: Use labeled data.

Transduction: Use labeled + unlabeled data.

Prune clusters with few members.

Reassign the points to final non-empty clusters.

We found the Gini index to be generally preferable over other impurity measures such as the number of misclassified points. If the simple number of misclassified points is used, then the misclassified points may be distributed evenly throughout all the clusters. The Gini Index favors solutions with pure clusters even at the expense of total classification error. Other decision tree splitting criterion such as Information Gain could also be used for cluster impurity measures (Qui84).

3 GENETIC REPRESENTATION AND ALGORITHM

The objective function (Eq. 2) defined in the previous section is discontinuous and non-convex. Finding an optimal solution to this problem is extremely difficult, so heuristic search is desirable. We utilized a genetic algorithm approach because the objective function defined can be readily used as a fitness function in the GA. As opposed to developing a genetic algorithm from scratch, we customized the general purpose GA library, GALib (Wal96), utilizing the floating-point representation but otherwise utilizes Goldberg’s simple GA approach (Gol89).

In a genetic algorithm application major concerns are genome representation, initialization, selection, crossover and mutation operators, stopping criteria and most importantly the fitness function. The objective function (Eq.2), defined above, is directly used as the fitness function without any scaling. The genome representation consists of an array of Kd real numbers, where d is the number of dimensions in the data and K is the number of clusters. Each set of d numbers represents one cluster center. This type of representation brings several advantages over prior discrete representation of cluster membership. First, cluster memberships are assigned based on Euclidean distance metric in this case instead of assigning them based on the values of genome. Second, each genome requires less search space than previous applications for the large datasets, since the length of the genomes depends only on the number of clusters (K) and the dimensionality (d) of the dataset, not the number of data points. It is therefore possible to handle the large datasets with this representation.

Default genetic operators defined for GAREalGenome class in GALib were applied. A mutation with Gaussian noise is the default in this case. Uniform crossover was applied as the default crossover operation. Although the uniform initializer is used by default, the population was initialized by sampling from the data. A uniform selection rule was used for selecting mating individuals (parents).

Two stopping criteria were applied . The algorithm stops when either of them is satisfied. One of these criteria is the maximum number of generations. The other one is the convergence after a certain number of consecutive generations.

The genetic algorithm yields reasonable results for both induction and transduction problems. The experimental findings are summarized in the next section.

Table 1. Results Using MSE in Fitness Function

Data Set	Induction			Transduction
	MSE-Only	GINI-Only	MSE+GINI	MSE+GINI
Bright	0.06585	0.01084	0.02507	0.02263
Sonar	0.43279	0.2541	0.22951	0.26066
Heart	0.23636	0.21477	0.2	0.19659
Iono.	0.25673	0.14423	0.12788	0.12981
Housing	0.25828	0.15629	0.18874	0.16887
House	0.09846	0.06692	0.06	0.06308
Prognos.	0.1	0.05059	0.06235	0.05235
Pima	0.32402	0.27118	0.30131	0.30393

4 EXPERIMENTAL RESULTS

The goals in this computational approach are to determine if combining supervised and unsupervised learning approaches techniques could lead to improved generalization, and to investigate if performing transductive inference using unlabeled data for training could lead to improvements over inductive inference. We experimented with eight datasets from the UC-Irvine Machine Learning Repository (MA92)¹. The datasets have all originally two-class output variable except Housing. The output variable for this dataset was categorized at the level of 21.5. Each dataset was divided into three subsets after a standard normalization. We call these subsets the learning, testing and working sets. Currently 40% of data is for learning, 30% is for testing and remaining 30% is for working sets. For each dataset, two scenarios have been tested to understand the difference between inductive and transductive inferences. For inductive inference, the algorithm is applied to labeled training data and then tested on the testing data. For transductive inference, the algorithm is applied to labeled training data, unlabeled working data, and unlabeled testing data, and then tested on the testing data.

We report experiments using seven different fitness functions. The two different cluster dispersion measures, MSE and Davies-Bouldin Index, are applied to induction in a completely unsupervised mode ($\beta = 1, \alpha = 0$) and semi-supervised mode ($\alpha > 0, \beta > 0$), and transduction in a semi-supervised mode ($\alpha > 0, \beta > 0$). We also tried the completely supervised case based on only the Gini index ($\beta = 0, \alpha = 1$). For transduction, both the cluster dispersion measure and the Gini index are based on the labeled and unlabeled data. In transduction, the Gini index becomes: $GiniP_j = 1.0 - \sum_{i=1}^k (\frac{P_{ij}}{\hat{N}_j})^2$ j in $1, \dots, K$ where \hat{N}_j is equal to number of labeled and unlabeled points in cluster j .

The best parameter set for the problem was picked by trial and error. We use same set of GA parameters for each dataset. The maximum number of generations is 500, mutation probability is 0.01, probability of crossover is 0.95, and number of generations to converge is 50. Experiments are conducted based on 10 bootstrap samples. For brevity only the average testing set error results are reported here. A paired t-test was used to assess the significance of difference of the testing set errors within a dataset. Errors with a p-value less than 0.2 were considered significant. To insure that the weaker performance of MSE was not based on poor choice of parameters, (K, β, α) for each dataset were chosen based on trials with the inductive MSE with Gini index.² For the DBI based results, the same values of K were used for each dataset, and the fixed values of $\beta = 0.01$ and $\alpha = 0.1$ were used for all datasets.

The results using the first dispersion measure, MSE, are reported in Table 1. The first column, MSE-only, indicates how the totally unsupervised approach of clustering based on only the unlabeled training data would perform. The second column, GINI-only shows how the completely supervised approach of clustering using the GINI index on the labeled training data performs. The third column is the proposed approach using both the MSE and GINI

¹The datasets and their corresponding sizes are: Bright(14 variables, 2462 points), Sonar(60,208), Cleveland Heart(13,297), Ionosphere(34,351), Boston Housing(13,506), House Votes (16,435), Breast Cancer Prognosis(30,569), and Pima Diabetes (8,769)

²The (k, β, α) values applied for each dataset were bright (15, 0.01,0.99), sonar (7,0.1,1), heart (7,0.25,0.75), ionosphere (7, 0.01,0.99), house (7,0.1,0.9), housing (11,0.01, 0.99), prognosis (11,0.4,0.6), and pima (11,0.01,0.99).

Table 2. Results Using DBI in Fitness Function

Data Set	Induction			Transduction
	DBI-Only	GINI-Only	DBI+GINI	DBI+GINI
Bright	<i>0.26897</i>	0.01084	<i>0.01992</i>	0.01165
Sonar	<i>0.50656</i>	0.2541	<i>0.27049</i>	0.23771
Heart	<i>0.3841</i>	<i>0.21477</i>	<i>0.21136</i>	0.19155
Iono.	<i>0.34327</i>	0.14423	0.12885	0.13558
Housing	<i>0.4563</i>	0.15629	<i>0.17086</i>	0.15497
House	<i>0.11769</i>	0.06692	0.07462	0.06923
Prognos.	<i>0.38059</i>	<i>0.05059</i>	<i>0.04941</i>	0.04353
Pima	<i>0.34585</i>	0.27118	0.28428	0.28122

Table 3. Comparison between Transductive DBI+GINI and 3-NN, LD, and QD

Data Set	3-NN	LinDisc	QuadDisc	DBI+GINI
Bright	<i>0.01247</i>	<i>0.02387</i>	<i>0.02112</i>	0.01165
Sonar	0.2098	<i>0.38025</i>	<i>0.35256</i>	0.23771
Heart	0.19773	0.1745	<i>0.22334</i>	0.19155
Iono.	<i>0.18846</i>	0.14624	0.1294	0.13558
Housing	<i>0.16291</i>	0.16013	<i>0.19946</i>	0.15497
House	0.06154	0.0414	0.06995	0.06923
Prognos.	0.04235	0.04797	<i>0.05348</i>	0.04353
Pima	0.28777	0.2313	<i>0.26401</i>	0.28122

based on the labeled training data. The fourth column indicates how MSE+GINI performs transductive inference when all the available data is used. A **bold** number is the minimum error for a given dataset, an *italic* number indicates that the result is significantly different from the transduction result. The totally unsupervised MSE-only approach always performs significantly worse than any of the supervised methods. Surprisingly, the GINI-only complete supervised approach was the best on four of the eight datasets. The transductive MSE+GINI method based on all available data showed no consistent improvements over the induction approach. This is consistent with other researchers who have reported that doing transductive inference using a regression estimate where the variance estimate was based on all the available data (both labeled and unlabeled) actually degraded results (Bot99).

The DBI dispersion measure was much more effective than the MSE with regards to transduction. The results for the DBI dispersion measure on the UC-Irvine Data are reported in Table 2. The same experimental setup and parameters as mentioned above were used except that the maximum number of generations is set to 300. The same K was used as in MSE approach, but the regularization parameters were modified due to the different magnitude of the DBI. As a purely unsupervised approach DBI-only was even worse than MSE-only at classification. The inductive DBI+GINI approach was not significantly different from the GINI-Only approach. The transductive DBI+GINI was either better or not significantly worse than GINI-only approach. Transductive DBI+GINI consistently produced the best classification results of all the seven approaches tested primarily due to the more compact and better separated clusters found by DBI over MSE. The evidence indicates that capacity control based on both labeled and unlabeled data is much more effective using the DBI criterion than MSE.

Another finding from the results is that DBI dispersion measure favors a fewer number of non-empty clusters compared to the MSE dispersion measure. On average, using DBI dispersion measure resulted in 53.33%, 33.33%, 28.36%, 17.91%, 30.84%, 33.33%, 51.11% and 50.91% fewer non-empty clusters than MSE dispersion measure for transductive inference on the UCI datasets respectively.

In Table 3, results from some other classification techniques are compared with transductive DBI+GINI – specifically 3 nearest neighbor classifier, linear and quadratic discriminant classifiers. The discriminant analysis was done using the SAS procedure DISCRIM (SAS89). All results are reported on the test datasets. The DBI-GINI is consistently one of the better methods.

5 CONCLUSION

A novel method for semi-supervised learning that combines supervised and unsupervised learning techniques has been introduced in this paper. The basic idea is to take an unsupervised clustering method, label each cluster with class membership, and simultaneously optimize the misclassification error of the resulting clusters. The intuition behind the approach is that the unsupervised component of objective acts like a form of regularization or capacity control during supervised learning to avoid overfitting. The objective function is a linear combination of a measure of cluster dispersion and a measure of cluster impurity. The method can exploit any available unlabeled data during training since the cluster dispersion measure does not require class labels. This allows the approach to be used for transductive inference, the process of constructing a classifier using both the labeled training data and the unlabeled testing data. Experimental results also show that using DBI for cluster dispersion instead of MSE helps transductive inference. This is due to the compact and well separated clusters found by minimizing DBI. DBI finds solution using much fewer clusters than MSE with much greater accuracy. The basic ideas in this paper: incorporating classification information into an unsupervised algorithm and using the resulting algorithm for transductive inference are applicable to many types of unsupervised learning and are promising areas of future research.

REFERENCES

- L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International, California, 1984.
- A.M. Bensaid, L.O. Hall, J.C. Bezdek, and L.P. Clarke. Partially supervised clustering for image segmentation. *Pattern Recognition*, 29(5):859–871, 1996.
- L. Bottou. About direct regression and density estimation. Presentation at Learning Workshop, Snowbird, Utah, 1999.
- V. Cherkassy, Y. Kim, and F. Mulier. Constrained topological maps for regression and classification. In *Proceedings of Int. Conf. on Neural Information Processing (ICONIP-97)*, New Zealand, November 1997.
- R. Cucchiara. Genetic algorithms for clustering in machine vision. *Machine Vision and Applications*, 11:1–6, 1998.
- D.L. Davies and D.W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.
- A. Demiriz, K.P. Bennett, and M.J. Embrechts. Semi-supervised clustering using genetic algorithms. R.P.I. Math Report No. 9901, Rensselaer Polytechnic Institute, Troy, New York, 1999.
- D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.
- A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, N.J., 1988.
- T. Kohonen. Learning vector quantization. *Neural Networks*, 1988.
- P.M. Murphy and D.W. Aha. *UCI repository of machine learning databases*. Department of Information and Computer Science, University of California, Irvine, California, 1992.
- C.A. Murthy and N. Chowdhury. In search of optimal clusters using genetic algorithms. *Pattern Recognition Letters*, 17:825–832, 1996.
- J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1984.
- SAS. *SAS Language and Procedures: Usage, Version 6*. SAS Institute Inc., Cary, North Carolina, 1989.
- M. Sarkar, B. Yegnanarayana, and D. Khemani. A clustering algorithm using an evolutionary programming-based approach. *Pattern Recognition Letters*, 18:975–986, 1997.
- V.N. Vapnik. *Statistical Learning Theory*. Wiley Inter-Science, 1998.
- M. Wall. *GAlib: A C++ Library of Genetic Algorithm Components*. MIT, <http://lancet.mit.edu/ga/>, 1996.