Astronomical Data Analysis Software and Systems XII
ASP Conference Series, Vol. 295, 2003
H. E. Payne, R. I. Jedrzejewski, and R. N. Hook, eds.

Novel Approaches to Semi-supervised and Unsupervised Learning

David Bazell

Eureka Scientific, Inc., 6509 Evensong Mews, Columbia, MD 21044

David J. Miller

Department of Electrical Engineering, Pennsylvania State University, University Park, PA 16802

Kirk Bourne

Institute for Science and Technology at Raytheon, NASA/GSFC, Code 630, Greenbelt, MD 20771

Abstract. We discuss a novel approach to the exploration, understanding, and classification of astronomical data. We are exploring the use of unlabeled data for supervised classification and for semi-supervised clustering. Current automated classification methods rely heavily on supervised learning algorithms that require training data sets containing large amounts of previously classified, or labeled, data. While unlabeled data are often cheap and plentiful, using a human to classify the data is tedious, time consuming, and expensive. We are examining methods whereby supervised classification techniques can use cheaply available, large volumes of unlabeled data to substantially improve their ability to classify objects.

1. Introduction

Machine learning falls into two broad categories. One is called supervised learning, the other unsupervised learning. An area of research that has had only mild activity over the past decade attempts to combine supervised and unsupervised learning (Miller & Uyar 1997; Nigam et al. 2000). The hope is to develop powerful new classification algorithms that combine the strengths of the two methods while minimizing their shortcomings. We are investigating one approach to this problem and applying it to astronomical data.

Examples of supervised learning algorithms include neural networks, decision trees, and mixture models. Neural networks consist of a set of individual cross-linked processing units connected by weights. Training samples are fed into the network which produces an output representing one of several possible classes. Each training sample has an associated target class, i.e., the correct class for that sample. The algorithm compares the network output and the target values and changes the connection weights in order to make the network and target outputs match. Iterating this procedure trains the network.

Decision trees consist of a large number of nodes at which decisions are made regarding which path to follow down the tree. The decisions are typically whether a certain feature value is greater than or less than a threshold value at a node. Training samples are passed down the tree. The number of nodes and the threshold values at the nodes are changed during training in order to make the class value at a terminal leaf node match the desired target value. This results in a tree structure wherein new examples trickle down through the tree and end up on a terminal leaf node corresponding to the class predicted by the decision tree.

The technique we are examining is based on using mixture models to describe the probability densities from which features are drawn. A mixture model is a representation of the target function as a linear combination of probability densities. In our case the target function is an unknown function describing the features of our data. The parameters for each of the component probability densities (e.g., mean, standard deviation, and component amplitude) are fit by maximum likelihood using the Expectation Maximization algorithm described briefly below.

While supervised learning algorithms can do very well given an adequate training data set, they have a number of drawbacks. Typically they require a large training data set. Generation of a training data set can be very costly. Furthermore, it is generally not possible to add incrementally to the training data while training the classifier. If the training data are changed in any way, the entire training data set must be used to retrain the classifier.

By contrast, unsupervised learning algorithms often attempt to find groups or clusters in data. There are several approaches to clustering algorithms. For example, one might attempt to find clusters or groups of data points with all data within a cluster being similar to each other. Another approach would be to find groups where the emphasis is on the groups being distinct, rather than the points within the group being similar. These two extremes can be accomplished by minimizing different forms of the objective function.

Some of the drawbacks of unsupervised learning include not being able to guide the cluster generation, and finding too many or too few clusters. From a scientific standpoint we often have an idea of what we would like to see in terms of clustering. Thus, the ability to guide clustering algorithms or to incorporate limits on the number of clusters allowed would be very useful.

Looked at the problem another way, the traditional distinction between supervised and unsupervised learning is the use of labeled vs. unlabeled data. Our project relies on statistical learning to combine labeled and unlabeled data in new ways. We are developing new methods that allow us to train classifiers with a small amount of labeled data, and boost performance by adding unlabeled data. This allows us to benefit from the vast quantities of unlabeled astronomical data, and to modify our training by adding in new labeled data when it becomes available.

2. How do we do it?

The key point of this approach—and I hope you remember this if you remember nothing else from this paper—is that unlabeled data provide information about

the joint probability distribution of features. For example, if we perform a search on-line for documents discussing "open source" we will find that both "Linux" and "Gnu" also commonly show up (try it using Google). This suggests that Linux, Gnu, and open source are all related in some way. In this example open source, Linux and Gnu are all features. Given enough documents, we can find a reasonable estimate of the joint probability distribution of the features. That is, we can find an expression that tells us the probability of finding two or more features with specific values in a given document. We don't know what these documents are (white papers, theses, manuals), because they're unlabeled. Joint probability distributions help us use unlabeled data effectively.

If features are generated by a two component Gaussian mixture model, we can recover the model parameters using unlabeled data alone. However, we need class labels to determine classes. Some labeled data are needed to actually classify rather than cluster the data.

We assume that our training data form two disjoint sets. One set, χ_l, consists of the labeled data (x_i, c_i) where x_i is a feature vector and c_i is a class label. The other set, χ_u consists of the unlabeled data (x_j). We assume the features are generated from a conditional probability density $f(x|\theta)$, where the density parameters are contained in the parameter vector θ. The class labels are assumed to be generated from the conditional probability density $P(c|m, x)$, where m denotes the mixture model component.

The best classifier is then found by using the maximum a posteriori (MAP) rule, which maximizes the total posterior probability of the model.

The map rule depends upon both $f(x|\theta)$ and $P(c|m, x)$, which gives us the critical point in this procedure: *Even though $f(x|\theta)$ is independent of class labels, improving $f(x|\theta)$ can improve classification.*

3. Expectation Maximization

The expectation maximization (EM) algorithm is used to iteratively refine estimates of model parameters when using incomplete data. EM can be used to estimate the values of missing parameters such as class labels. The basic algorithm can be described as follows. Assume an initial set of parameter values: $\theta = \{\mu_i, \sigma_i, \alpha_i\}$.

E-Step Use current parameter estimates, θ, to find the "best" values of class membership, i.e., best probabilistic labels.

M-Step Refine the parameters θ by using the map rule to maximize the total likelihood.

The steps are iterated until the change in parameter values falls below some predefined threshold.

4. Status and Applications

In testing our approach we are attempting classification of large data sets using limited labeled data. We are currently using a data set called ESOLV that

has been used previously when testing neural networks (Storrie-Lombardi et al. 1992) and decision trees (Owens, Griffiths, & Ratnatunga 1996). This data set contains galaxy morphology parameters for over 5000 galaxies of a range of morphological types. We are trying to classify these data into five morphologial classes. Using about 100 labeled samples we achieve a test set error of around 50%. Using standard backpropagation neural networks or decision trees and training using about 1700 samples we get a test set error of about 35% to 38%.

Another interesting feature of this model is the ability to identify "interesting" objects based on class conditional probabilities. After classification of the data set has been completed we can examine the class membership probabilities associated with each of the objects.

While not specifically designed for this task, our method does have a limited ability to do class discovery. Class discovery means identifying objects that are not well categorized by existing classes. This can obviously be very useful when examining large data sets. A method that can put known objects into existing classes and create new classes when needed will help researchers direct future observations and analysis to some interesting areas.

Acknowledgments. This work is being funded by a contract from the NASA/Applied Information Systems Research Program.

References

Miller, D.J. & Uyar, H.S. 1997, Advances in Neural Information Processing Systems, 9, 571

Nigam, K., McCallum, A.K., Thrun, S. & Mitchel, T. 2000, Machine Learning, 39, 103

Owens, E.A., Griffiths, R.E., & Ratnatunga, K.U. 1996, MNRAS, 281, 1530

Storrie-Lombardi, M.C., Lahav, O., Sodré, Jr., L., & Storrie-Lombardi, L.J 1992, MNRAS, 259, 8p