Semi-supervised Clustering: Learning with Limited User Feedback

by

Sugato Basu

Doctoral Dissertation Proposal

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

November 2003

Semi-supervised Clustering: Learning with Limited User Feedback

Publication No. ______________

Sugato Basu
The University of Texas at Austin, 2003

Supervisor: Dr. Raymond J. Mooney

In many machine learning domains (e.g. text processing, bioinformatics), there is a large supply of unlabeled data but limited labeled data, which can be expensive to generate. Consequently, semi-supervised learning, learning from a combination of both labeled and unlabeled data, has become a topic of significant recent interest. In the proposed thesis, our research focus is on semi-supervised clustering, which uses a small amount of supervised data in the form of class labels or pairwise constraints on some examples to aid unsupervised clustering. Semi-supervised clustering can be either search-based, i.e., changes are made to the clustering objective to satisfy user-specified labels/constraints, or similarity-based, i.e., the clustering similarity metric is trained to satisfy the given labels/constraints. Our main goal in the proposed thesis is to study search-based semi-supervised clustering algorithms and apply them to different domains.

In our initial work, we have shown how supervision can be provided to clustering in the form of labeled data points or pairwise constraints. We have also developed an active learning framework for selecting informative constraints in the pairwise constrained semi-supervised clustering model, and proposed a method for unifying search-based and similarity-based techniques in semi-supervised clustering.

In this thesis, we want to study other aspects of semi-supervised clustering. Some of the issues we want to investigate include: (1) effect of noisy, probabilistic or incomplete supervision in clustering; (2) model selection techniques for automatic selection of number of clusters in semi-supervised clustering; (3) ensemble semi-supervised clustering. In our work so far, we have mainly focussed on generative clustering models, e.g. KMeans and EM, and ran experiments on clustering low-dimensional UCI datasets or high-dimensional text datasets. In future, we want to study the effect of semi-supervision on other clustering algorithms, especially in the discriminative clustering and online clustering framework. We also want to study the effectiveness of our semi-supervised clustering algorithms on other domains, e.g., web search engines (clustering of search results), astronomy (clustering of Mars spectral images) and bioinformatics (clustering of gene microarray data).

Contents

Chapter 1

Introduction

Two of the most widely-used methods in machine learning for prediction and data analysis are classification and clustering (Duda, Hart, & Stork, 2001; Mitchell, 1997). Classification is a purely supervised learning model, whereas clustering is completely unsupervised. Recently, there has been a lot of interest in the continuum between completely supervised and unsupervised learning (Muslea, 2002; Nigam, 2001; Ghani, Jones, & Rosenberg, 2003). In this chapter, we will give an overview of traditional supervised classification and unsupervised clustering, and then describe learning in the continuum between these two, where we have partially supervised data. We will then be presenting the main goal of our proposed thesis.

1.1 Classification

Classification is a supervised task, where supervision is provided in the form of a set of labeled training data, each data point having a class label selected from a fixed set of classes (Mitchell, 1997). The goal in classification is to learn a function from the training data that gives the best prediction of the class label of unseen (test) data points. Generative models for classification learn the joint distribution of the data and class variables by assuming a particular parametric form of the underlying distribution that generated the data points in each class, and then apply Bayes Rule to obtain class conditional probabilities that are used to predict the class labels for test points drawn from the same distribution, with unknown class labels (Ng & Jordan, 2002). In the discriminative framework, the focus is on learning the discriminant function for the class boundaries or a posterior probability for the class labels directly without learning the underlying generative densities (Jaakkola & Haussler, 1999). It can be shown that the discriminative model of classification has better generalization error than the generative model under certain assumptions (Vapnik, 1998), which has made discriminative classifiers, e.g., support vector machines (Joachims, 1999) and nearest neighbor classifiers (Devroye, Gyorfi, & Lugosi, 1996), very popular for the classification task.

1.2 Clustering

Clustering is an unsupervised learning problem, which tries to group a set of points into clusters such that points in the same cluster are more similar to each other than points in different clusters, under a particular similarity metric (Jain & Dubes, 1988). Here, the learning algorithm just observes a set of points without observing any corresponding class/category labels. Clustering problems can also be categorized as generative or discriminative. In the generative clustering model, a parametric form of data generation is assumed, and the goal in the maximum likelihood formulation is to find the parameters that maximize the probability (likelihood) of generation of the data given the model. In the most general formulation, the number of clusters K is also considered to be an unknown parameter. Such a clustering formulation is called a "model selection" framework, since it has to choose the best value of K under which the clustering model fits the data. We will be assuming that K is known in the clustering frameworks that we will be considering, unless explicitly mentioned otherwise. In the discriminative clustering setting (e.g., graph-theoretic clustering), the clustering algorithm tries to cluster the data so as to maximize within-cluster similarity and minimize between-cluster similarity based on a particular similarity metric, where it is not necessary to consider an underlying parametric data generation model. In both the generative and discriminative models, clustering algorithms are generally posed as optimization problems and solved by iterative methods like EM (Dempster, Laird, & Rubin, 1977), approximation algorithms like KMedian (Jain & Vazirani, 2001), or heuristic methods like Metis (Karypis & Kumar, 1998).

1.3 Semi-supervised learning

In many practical learning domains (e.g. text processing, bioinformatics), there is a large supply of unlabeled data but limited labeled data, which can be expensive to generate. Consequently, *semi-supervised learning*, learning from a combination of both labeled and unlabeled data, has become a topic of significant recent interest. The framework of semi-supervised learning is applicable to both classification and clustering.

1.3.1 Semi-supervised classification

Supervised classification has a known, fixed set of categories, and category-labeled training data is used to induce a classification function. In this setting, the training can also exploit additional unlabeled data, frequently resulting in a more accurate classification function. Several semi-supervised classification algorithms that use unlabeled data to improve classification accuracy have become popular in the past few years, which include co-training (Blum & Mitchell, 1998), transductive support vector machines (Joachims, 1999), and using Expectation Maximization to incorporate unlabeled data into training (Ghahramani & Jordan, 1994; Nigam, McCallum, Thrun, & Mitchell, 2000). Unlabeled data have also been used to learn good metrics in the classification setting (Hastie & Tibshirani, 1996). A good review of semi-supervised classification methods is given in (Seeger, 2000).

1.3.2 Semi-supervised clustering

Semi-supervised clustering, which uses class labels or pairwise constraints on some examples to aid unsupervised clustering, has been the focus of several recent projects (Basu, Banerjee, & Mooney, 2002; Klein, Kamvar, & Manning, 2002; Wagstaff, Cardie, Rogers, & Schroedl, 2001; Xing, Ng, Jordan, & Russell, 2003). If the initial labeled data represent all the relevant categories, then both semi-supervised clustering and semi-supervised classification algorithms can be used for categorization. However in many domains, knowledge of the relevant categories is incomplete. Unlike semi-supervised classification, semi-supervised clustering (in the model-selection framework) can group data using the categories in the initial labeled data as well as extend and modify the existing set of categories as needed to reflect other regularities in the data.

Existing methods for semi-supervised clustering fall into two general approaches that we call *search-based* and *similarity-based* methods.

Search-based methods

In search-based approaches, the clustering algorithm itself is modified so that user-provided labels or constraints are used to bias the search for an appropriate partitioning. This can be done by several methods, e.g., modifying the clustering objective function so that it includes a term for satisfying specified constraints (Demiriz, Bennett, & Embrechts, 1999), enforcing constraints to be satisfied during the cluster assignment in the clustering process (Wagstaff et al., 2001), doing clustering using side-information from conditional distributions in an auxiliary space (Sinkkonen & Kaski, 2000), and initializing clusters and inferring clustering constraints based on neighborhoods derived from labeled examples (Basu et al., 2002).

Similarity-based methods

In similarity-based approaches, an existing clustering algorithm that uses a similarity metric is employed; however, the similarity metric is first trained to satisfy the labels or constraints in the supervised data. Several similarity metrics have been used for similarity-based semi-supervised clustering, including string-edit distance trained using EM (Bilenko & Mooney, 2003), Jensen-Shannon divergence trained using gradient descent (Cohn, Caruana, & McCallum, 2000), Euclidean distance modified by a shortest-path algorithm (Klein et al., 2002), or Mahalanobis distances trained using convex optimization (Hillel, Hertz, Shental, & Weinshall, 2003; Xing et al., 2003). Several clustering algorithms using trained similarity metrics have been employed for semi-supervised clustering, including single-link (Bilenko & Mooney, 2003) and complete-link (Klein et al., 2002) agglomerative clustering, EM (Cohn et al., 2000; Hillel et al., 2003), and KMeans (Hillel et al., 2003; Xing et al., 2003).

However, similarity-based and search-based approaches to semi-supervised clustering have not been adequately compared in previous work, and so their relative strengths and weaknesses are largely unknown. In Section 3.4, we will be presenting a new semi-supervised clustering algorithm that unifies these two approaches.

1.4 Goal of proposed thesis

In the proposed thesis, the main goal is to study semi-supervised clustering algorithms, characterize some of their properties and apply them to different domains. In our completed work, we have already shown how supervision can be provided to clustering in the form of labeled data points or pairwise constraints. We have also developed an active learning framework for selecting informative constraints in the pairwise constrained semi-supervised clustering model, and proposed a method for unifying search-based and similarity-based techniques in semi-supervised clustering. Details of the completed work are given in Chapter 3.

In future, we want to look at the following issues, details of which are given in Chapter 4:

- Investigate the effects of noisy supervision, probabilistic supervision (e.g., soft constraints) or incomplete supervision (e.g., labels not specified for all clusters) in clustering;
- Study model selection issues in semi-supervised clustering, which will help to characterize the difference between semi-supervised clustering and classification;
- Study the feasibility of semi-supervising other clustering algorithms, especially in the discriminative clustering or online clustering framework;
- Create a framework for ensemble semi-supervised clustering;
- Apply the semi-supervised clustering model on other domains apart from text, especially web search engines, astronomy and bioinformatics;
- Study the relation between different evaluation metrics used to evaluate semi-supervised clustering;
- Investigate other forms of semi-supervision, e.g., attribute-level constraints;
- Do more theoretical analysis of certain aspects of semi-supervision, especially semi-supervised clustering with labeled data and the unified semi-supervised clustering model.

Chapter 2

Background

This chapter gives a brief review of clustering algorithms on which our proposed semi-supervised clustering techniques will be applied. It also gives an overview of different popular clustering evaluation measures, and describes the measures we will be using in our experiments.

2.1 Overview of clustering algorithms

As explained in Chapter 1, clustering algorithms can be classified into two models — generative or discriminative. There are other categorizations of clustering, e.g., hierarchical or partitional (Jain, Myrthy, & Flynn, 1999), depending on whether the algorithm clusters the data into a hierarchical structure or gives a flat partitioning of the data.

2.1.1 Hierarchical clustering

In hierarchical clustering, the data is not partitioned into clusters in a single step. Instead, a series of partitions take place, which may run from a single cluster containing all objects to N clusters each containing a single object. This gives rise to a hierarchy of clusterings, also known as the cluster dendrogram. Hierarchical clustering can be further categorized as:

- Divisive methods: Create the cluster dendrogram in a top-down divisive fashion, starting with every data point in one cluster and splitting clusters successively according to some measure till a convergence criterion is reached, e.g., Cobweb (Fisher, 1987), recursive cluster-splitting using a statistical transformation (Dubnov, El-Yaniv, Gdalyahu, Schneidman, Tishby, & Yona, 2002), etc.;
- Agglomerative methods: Create the cluster dendrogram in a bottom-up agglomerative fashion, starting with each data point in its own cluster and merging clusters successively according to a similarity measure till a convergence criterion is reached, e.g., hierarchical agglomerative clustering (Kaufman & Rousseeuw, 1990), Birch (Zhang, Ramakrishnan, & Livny, 1996), etc.

2.1.2 Partitional clustering

Let $\mathcal{X} = \{x_i\}_{i=1}^N$ be the set of N data-points we want to cluster with each $x_i \in \mathbb{R}^d$. A partitional clustering algorithm divides the data into K partitions (K given as input to the algorithm) by grouping the associated feature vectors into K clusters. Partitional algorithms can be classified as:

- Graph-theoretic: These are discriminative clustering approaches, where an undirected graph $G = (V, E)$ is constructed from the dataset, each vertex in $v_i \in V$ corresponding to a data point $\mathbf{x}_i$ and the weight of each edge $e_{ij} \in E$ corresponding to the similarity between the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ according to a domain-specific similarity measure. The K clustering problem becomes equivalent to finding the K-mincut in this graph, which is known to be a NP-complete problem for $K \geq 3$ (Garey & Johnson, 1979). So, most graph-based clustering algorithms try to use good heuristic methods to group nodes so as to find low-cost cuts in G. Several different graph-theoretic algorithms have been proposed: methods like Rock (Guha, Rastogi, & Shim, 1999) and Chameleon (Karypis, Han, & Kumar, 1999) group nodes based on the idea of defining neighborhoods using inter-connectivity of nodes in G, Metis (Karypis & Kumar, 1998) performs fast multi-level heuristics on G at multiple resolutions to give good partitions, while Opossum (Strehl & Ghosh, 2000) uses a modified cut criterion to ensure that the resulting clusters are well-balanced according to a specified balancing criterion.

- Density-based: These methods model clusters as dense regions and use different heuristics to find arbitrary-shaped high-density regions in the data space and group points accordingly. Well-known methods include Denclue, which tries to analytically model the overall density around a point (Hinneburg & Keim, 1998), and WaveCluster, which uses wavelet-transform to find high-density regions (Sheikholesami, Chatterjee, & Zhang, 1998). Density-based methods typically have difficulty scaling up to very high dimensional data ($>$ 10000 dimensions), which are common in domains like text.

- Mixture-model based: In mixture-model based clustering, the underlying assumption is that each of the N data points $\{\mathbf{x}_i\}_{i=1}^{N}$ to be clustered are generated by one of K probability distributions $\{p_h\}_{h=1}^{K}$, where each distribution p_h represents a cluster C_h. The probability of observing any point $\mathbf{x}_i$ is given by:

$$\mathbf{Pr}(\mathbf{x}_i|\Theta) = \sum_{i=1}^{K} \alpha_h p_h(\mathbf{x}_i|\theta_h)$$

where $\Theta = (\alpha_1, \cdots, \alpha_K, \theta_1, \cdots, \theta_K)$ is the parameter vector, α_h are the prior probabilities of the clusters ($\sum_{h=1}^{K} \alpha_h = 1$), and p_h is the probability distribution of cluster C_h parameterized by the set of parameters θ_h. The data-generation process is assumed to be as follows – first, one of the K components is chosen following their prior probability distribution $\{\alpha_h\}_{i=1}^{K}$; then, a data-point is sampled following the distribution p_h of the chosen component.

Since the cluster assignment of the points are not known, we assume the existence of a random variable $\mathcal{Z}$ that encodes the cluster assignment z_i for each data point $\mathbf{x}_i$. It takes values in $\{h\}_{h=1}^{K}$ and is always conditioned on the data-point $\mathbf{x}_i$ under consideration. The goal of clustering in this model is to find the estimates of the parameter vector Θ and the cluster assignment variable $\mathcal{Z}$ such that the log-likelihood of the data:

$$\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) = \sum_{i=i}^{N} \log \mathbf{Pr}(\mathbf{x}_i, \mathcal{Z}|\Theta)$$

is maximized. Since $\mathcal{Z}$ is unknown, the log-likelihood cannot be maximized directly. So, traditional approaches iteratively maximize the *expected* log-likelihood in the Expectation Maximization (EM) framework (Dempster et al., 1977). Starting from an initial estimate of Θ, the EM algorithm iteratively improves the estimates of Θ and $p(\mathcal{Z}|\mathcal{X}, \Theta)$ such that the expected value of the complete-data log-likelihood computed over the class conditional distribution $p(\mathcal{Z}|\mathcal{X}, \Theta)$ is maximized. It can be shown that the EM algorithm converges to a local maximum of the expected log-likelihood distribution (Dempster et al., 1977), and the final estimates of the conditional distribution $p(\mathcal{Z}|\mathcal{X}, \Theta)$ are used to find the cluster assignments of the points in $\mathcal{X}$.

Most of the work in this area has assumed that the individual mixture density components p_h are Gaussian, and in this case the parameters of the individual Gaussians are estimated by the EM procedure. The popular KMeans clustering algorithm (MacQueen, 1967) can be shown to be an EM algorithm on a mixture of K Gaussians under certain assumptions. Details of this derivation are shown in Section 2.2.1.

2.2 Our representative clustering algorithms

In our work, we have chosen KMeans and Hierarchical Agglomerative Clustering as two representative clustering algorithms, from the partitional and hierarchical clustering categories respectively, on which our proposed semi-supervised