CHAPTER 1
INTRODUCTION

Unsupervised learning algorithms have had some impressive successes. Data mining algorithms, for example, are regularly used in the corporate world to extract useful information from large customer databases. Rather than being restricted to the realm of academic research, algorithms from machine learning are now showing up in commercial software, including the PolyAnalyst software offered by Megaputer and Genio Miner from Hummingbird. In addition to being used as commercial tools, learning methods have been used to extend the body of scientific knowledge. Clustering algorithms, which automatically divide a data set into meaningful sub-groups, have been especially successful in both areas. For example, the Autoclass clustering program analyzed a large body of infrared spectral data and discovered a sub-class of stars previously unknown to astronomers (Goebel et al., 1989).

However, the majority of clustering algorithms are limited in what they can achieve. They can detect general trends and patterns in data, but they cannot make use of additional knowledge specific to the problem at hand, as a human expert can. This dissertation develops and evaluates a new kind of clustering algorithm that can take advantage of problem-specific information to become a temporary expert in the domain at hand. In particular, these "intelligent clustering" methods make use of information expressed as constraints between two items in the data set. These algorithms, although general in nature, are able to greatly enhance their performance on individual tasks.

In this chapter, we first discuss unsupervised learning and describe how it is used. We then contrast it with supervised learning methods and identify some important limitations of both approaches. Next we outline our proposed method of intelligent clustering, which falls somewhere in between these two extremes and seeks to address their respective limitations. After describing several scenarios that stand to benefit from our method, we conclude by summarizing the key contributions of our work.

1.1 Unsupervised Learning

We often think of learning as a product of instruction. A student learns by observing a teacher or by carefully examining correct answers. We refer to this kind of situation as *supervised* learning. In contrast, *unsupervised* learners have neither an instructor nor a set of correct answers. Unsupervised learning is exploratory and experimental, seeking knowledge through discovery.

Clustering algorithms are unsupervised in that they do not have access to any specific information about what they should be seeking. A clustering algorithm uses the same principles to perform data analysis whether the objects under scrutiny are stars, or apples, or human beings. Success hinges on how well a given data set conforms to the assumptions encoded in those generic principles. Although currently in widespread use with many different kinds of data, clustering algorithms adopt no special provisions to account for particular characteristics of the data set or problem domain.

Exploratory learning via built-in bias. All unsupervised learning is exploratory in nature; the learner is presented with a set of observations and then encouraged to discover patterns and meaningful structure independently. This is akin to how we learn to interpret visual images or make summaries of long text passages: we do not receive specific guidance that tells us "this is a human face" or "this newspaper article should be summarized with the following three sentences," but we are guided by general notions of pattern matching and simplification, respectively. Similarly, a computer can be instructed to detect patterns in data and to group items by similarity. Despite the lack of specific guidance, unsupervised approaches cannot be said to be completely ignorant. Just as a human makes use of general principles to enable learning, a computer makes use of a *learning bias* of some sort, which encompasses any assumptions it makes about what is to be learned (Mitchell, 1997). For a clustering algorithm, this bias might be a preference for identifying groups of items that are compact and well-separated from other groups. However, we claim that, for many situations, the built-in bias of clustering algorithms is either insufficiently specific or actively misleading.

Clustering as a data mining tool. This limitation is significant because clustering algorithms are currently being used for a variety of purposes. Unsupervised data analysis, or *data mining*, is becoming more and more prevalent as organizations continue to create and maintain vast databases. Corporations stockpile information about their customers; scientists collect gigabytes of observational data. In both cases, "mining" these databases for recurring patterns or other interesting structure is very important. Clustering is even being used to detect patterns in reported computer crimes, so that sites likely to be future targets can be warned (Brown and Gunderson, 2001). Researchers at Stanford University estimate that the worldwide production of data in electronic format is between 500,000 and 1,700,000 *terabytes* per year (Lyman and Varian, 2001). This implies several thousand new terabytes of data every day. The human data analyst is overwhelmed by the sheer volume of information, yet extracting trends and classifications from the data is critical. Currently, clustering algorithms for data mining are successfully culling useful information from these large databases, but as above, standard approaches disregard any domain-specific information.

Clustering as a data compression tool. An additional application of clustering algorithms is *data compression*, where massive data sets are reduced to more manageable sizes by replacing a group of similar items with a single representative data point. This alleviates the intense demand for data storage and transmission resources. For example, probes and rovers operating remotely on other planets or in deep space are capable of collecting large amounts of data, but are limited by size and bandwidth in terms of how much they can store or transmit back. If we can only obtain a fraction of the full data set, it is preferable to receive information about a carefully selected subset of the most interesting items, rather than simply the first ones that are lined up in the queue (Mjolsness and DeCoste, 2001). Clustering algorithms can perform this kind of filtering to make better use of limited resources such as bandwidth or storage space. *Intelligent* clustering algorithms can use domain knowledge to make better decisions about which items are redundant (i.e., should be grouped together) and which are significantly different.

Disadvantages of unsupervised learning. Despite the many applications of clustering algorithms, their very generality also limits their performance on any specific task. Because they make use of very general notions to identify patterns and interesting structure in data, they cannot specialize very well for a specific problem. Further, because unsupervised methods are unguided, they may report patterns or trends that, while present in the data, are completely uninteresting (e.g., a pattern caused by a systematic error in the data gathering process). The traditional alternative to unsupervised learning has been supervised learning, which adapts to the problem at hand by not only exploiting but *requiring* problem-specific knowledge.

1.2 Supervised Learning

Supervised algorithms differ from unsupervised algorithms in that they depend on having a set of labeled examples that demonstrate what kind of distinctions should be made. The labels indicate the correct answer for each of a set of sample problems. Testing what has been learned then involves presenting the learner with additional, unlabeled, problems and comparing the learner's responses to the correct answers. This is not so different from how teaching and testing of humans often occur in our classrooms; we seem to be very good at learning by example.

Disadvantages of supervised learning. Despite the proven success of supervised approaches to learning, they too have certain limitations and drawbacks. A supervised algorithm performs best when a large body of labeled examples is available for the algorithm to learn from. The algorithm is able to train itself to generate the correct responses, according to the labeled examples, and (hopefully) to also generalize to unseen data. But as the adage "garbage in, garbage out" reminds us, supervised algorithms are limited by the quality of the labeled examples they are given. In addition, they may be susceptible to over-fitting the training data: an algorithm that focuses too closely on getting all of the given examples right may then be incapable of performing well on new, different examples.

The most important drawback to supervised learning is that an expert must manually create a sufficiently large set of labeled examples. Supervised learning is an all-or-nothing paradigm; it cannot make use of any unlabeled data. Unfortunately, individual labeling is a time-consuming and tedious process. Further, in some cases it may not be possible to provide labels at all. For some problems, the true answers are not known, and we would like the computer to automatically discover interesting patterns and classifications on its own. Supervised methods cannot provide solutions to these problems.

Finally, it is not possible to select one learning paradigm as the universal best approach. Supervised and unsupervised methods serve different learning goals. Supervised methods are appropriate when the goal is to build a predictive model (e.g., neural network, decision tree, support vector machine) for the problem domain. Unsupervised approaches are used when the goal is to analyze the data to discover interesting structure (e.g., clusters or association rules). Our focus is on the latter type of problem.

Table 1.1: Census data represented in feature-vector format

Name	SSN	Age	Gender	Occupation	Income
Diana	111-11-1111	30	F	Professional Dancer	\$50,000
Gary	444-44-4444	30	M	Professional Dancer	\$50,000
Lyra	222-22-2222	29	F	Rocket Scientist	\$90,000
Tim	777-77-7777	28	M	Assistant Professor	\$70,000
...	...	...	...	...	...

1.3 Intelligent Clustering

We are faced with two extremes. Supervised learning requires that every item be assigned a label. Unsupervised learning ignores any such guidance. What if we have a data set where some, but not all, items are labeled? What if we have additional information about the data set that cannot be expressed as individual labels? In the absence of other alternatives, we are forced to use an unsupervised approach and to discard all of this extra information.

However, deliberately ignoring information is contrary to what we see as the basic philosophy of learning. Successful learning is opportunistic, and it attempts to exploit information from as many sources as possible. Some algorithms, referred to as *weakly supervised* or *semi-supervised* methods, seek to limit the amount of manual labeling that must be done yet still make use of domain-specific information when it is available. We extend this body of hybrid algorithms by proposing the use of "intelligent clustering" algorithms. In this section, we identify four situations where these algorithms, which can take advantage of partially labeled data and information that transcends the label format, are particularly useful.

1.3.1 Knowledge Not Expressed as Labels

The usual division between supervised and unsupervised learning recognizes only one source of information: individual data labels. However, as hinted at above, there are many other forms that knowledge can take. It may be presented as structural or spatial information about the problem, heuristic guides, relational information between items, or some combination of these elements. In recognition of this variety, the intelligent clustering algorithms we propose are able to exploit any such information that can be expressed as a set of constraints between items in the data set.

Information that describes a relationship between items in a data set has proven particularly challenging for machine learning applications because it cannot be represented in the usual data format. The most common way to represent items in a data set is with a *feature vector*. Each item possesses a value for each of a given set of features, and those values can be strung together to create a feature vector. For example, consider a database of census records, as shown in Table 1.1. Each person is represented by a variety of features, including social security number, age, gender, occupation, and annual income.

However, although it is possible to describe an item with a series of features, it is not possible to use those features to express a relationship between two items. For example, we cannot use a feature to capture the fact that Gary and Diana are married to each other, because this requires a way to express relationship information *between* items (people). This representational issue is a problem because a learning algorithm cannot make use of information in formats it does not understand.

We propose a method for representing instance-level relationships and for naturally integrating those relationships into clustering algorithms. In particular, we focus on relationships that are consistent with the desired clustering output, and describe how these can be expressed as a set of instance-level constraints. To continue the example, we may want clusters of people to represent households or neighborhoods, and therefore desire that clusters preserve the marriage relationship. We will also show how this formulation is a convenient way to express useful background knowledge about a variety of real-world problems.

1.3.2 Partially Labeled Data Sets

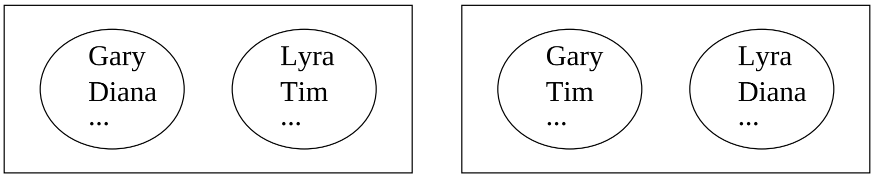
Section 1.2 described supervised learning and hinted at some reasons why it might not be possible (or feasible) to provide individual labels for every item in a data set. This becomes more and more common as researchers encounter data sets with thousands or millions of elements; in such cases, manual labeling is simply unrealistic. While supervised methods can restrict themselves to the subset of data that does possess labels, the unlabeled items also contain information that can contribute to better system performance. Significant work has been done in modifying supervised algorithms to enable them to operate on data that is not fully labeled (e.g., Blum and Mitchell, 1998; Nigam et al., 1998) However, less work has been done in enabling unsupervised algorithms to exploit partially labeled data sets.

We will show how to take existing labels, convert them into a set of instance-level constraints, and then ensure that they are preserved in the final output of a clustering algorithm. Our approach can therefore integrate labels, traditionally only useful for supervised learning, into unsupervised methods.

1.3.3 Intelligent Exploratory Data Analysis

In contrast to situations where informative labels are lacking for practical reasons, there are also domains where supervisory information simply is not known. This is the case for many data mining applications where the goal is to detect novel patterns in data. Examples of such situations include market basket analysis, spectral analysis of astronomy data, and automated image segmentation. Although specific labels for individual items are not known, the user may have specific requirements for the output partitions. Ideally, we would like user to be able to indicate those requirements to the clustering algorithm.

In our census data example, we may wish to analyze the data to detect interesting subgroup structure, without having labeled examples of the kinds of distinctions we wish to discover. To this end, we could apply any traditional clustering algorithm, which would produce one possible partition of the data set. Figure 1.1a shows a default partition of the data from Table 1.1. Gary and Diana have been placed in the same cluster, and Lyra and Tim appear in a different cluster. However, if we have a more sophisticated clustering goal, such as finding clusters that are consistent with the gender feature, the partition shown in Figure 1.1b is a better solution. We would like to require that individuals of different genders end up in different clusters, but regular clustering algorithms are inadequate for such tasks.



(a) Default partition (b) Consistent with gender

Figure 1.1: Different partitions of the census data shown in Table 1.1

We propose extending current methods for exploratory data analysis by enabling them to make use of feature-based clustering heuristics, such as the above restriction with respect to gender, thereby producing more "intelligent" results.

1.3.4 Structured Data Sets

Our final example involves data sets that possess an inherent structure. For example, clustering is often applied to images, where each pixel represents an item in the data set. One application is in *image segmentation*, where the goal is to automatically divide an image into homogeneous sub-regions or clusters. Although it is possible to apply a regular clustering algorithm to this problem, and cluster the pixels based on their color or intensity values, this kind of approach abandons the information inherent in the two-dimensional structure of the image. The fact that certain pixels are adjacent is important when attempting to segment an image, and that structure should be preserved as much as possible in the output.

We propose methods that will integrate information about the structure of a data set and preserve that structure in the final output, to the degree desired by the user.

As the previous examples demonstrate, there are many cases where the experimenter possesses useful information about what assumptions can be made about potential solutions, even if individual labels for every item in the data set cannot be made (or would be prohibitively expensive to make). These situations include, among others, knowledge expressed in non-label forms, partially labeled data sets, intelligent exploratory data analysis, and structured data sets.

1.4 Contributions

In this dissertation, we investigate methods for enabling clustering algorithms to take advantage of such problem-specific information in an automatic, efficient fashion. This research creates a bridge between powerful data analysis techniques and expert knowledge. In summary, this dissertation presents the following contributions:

A flexible instance-level constraint formulation. We propose the use of instance-level constraints for encoding domain knowledge. We will use both hard and soft constraints. *Hard* constraints encode restrictions that must be satisfied in the final output of the algorithm.

Soft constraints express preferences about the final partition generated by the algorithm and therefore offer additional flexibility and expressiveness. By accommodating both kinds of constraints, we provide a rich language for encoding information that is important for clustering algorithms.

A method for intelligent clustering. We describe a general method for augmenting clustering algorithms to enable them to make use of background knowledge about the problem domain. In particular, we demonstrate how to incorporate knowledge in the form of instance-level constraints. This method is not specific to any single clustering algorithm.

Three intelligent clustering algorithms. We first apply this general technique by modifying two widely-used clustering algorithms, k-means (MacQueen, 1967) and COBWEB (Fisher, 1987), so that they can accommodate hard instance-level constraints. These algorithms, though fundamentally unsupervised, are able to take advantage of problem-specific knowledge to improve their clustering accuracy. Later, we present a second modified version of k-means that can also accommodate soft constraints.

Empirical demonstration of scalability and improvements in clustering accuracy on real-world problems. We evaluate our new clustering algorithms using artificially-generated constraints and on real problems, where constraints are generated from heuristic knowledge about the problem domain. Specifically, we focus on the problems of a) automatically identifying road lanes from GPS data, b) determining noun phrase coreference relationships, and c) analyzing infra-red spectral observations of Mars. In applying our methods to such diverse problems, we are able to show that general constrained clustering algorithms can become adept experts on particular problems when making use of domain information.

1.5 Roadmap

In the next chapter, we will examine a large body of relevant work and discuss the advantages of our approach. Chapter 3 describes the details of our method of integrating background knowledge with clustering algorithms and presents two modified clustering algorithms, based on k-means and COBWEB. We then examine the application of constrained clustering to several real-world problems in Chapters 4, 5, and 6. In addition, Chapter 6 presents a third intelligent clustering algorithm that can make use of soft constraints.