Mining Group Differences

Shane M. Butler
Monash University, Australia

Geoffrey I. Webb
Monash University, Australia

INTRODUCTION

Finding differences among two or more groups is an important data-mining task. For example, a retailer might want to know what the different is in customer purchasing behaviors during a sale compared to a normal trading day. With this information, the retailer may gain insight into the effects of holding a sale and may factor that into future campaigns. Another possibility would be to investigate what is different about customers who have a loyalty card compared to those who don't. This could allow the retailer to better understand loyalty cardholders, to increase loyalty revenue, or to attempt to make the loyalty program more appealing to non-cardholders.

This article gives an overview of such group mining techniques. First, we discuss two data-mining methods designed specifically for this purpose—Emerging Patterns and Contrast Sets. We will discuss how these two methods relate and how other methods, such as exploratory rule discovery, can also be applied to this task.

Exploratory data-mining techniques, such as the techniques used to find group differences, potentially can result in a large number of models being presented to the user. As a result, filter mechanisms can be a useful way to automatically remove models that are unlikely to be of interest to the user. In this article, we will examine a number of such filter mechanisms that can be used to reduce the number of models with which the user is confronted.

BACKGROUND

There have been two main approaches to the group discovery problem from two different schools of thought. The first, Emerging Patterns, evolved as a classification method, while the second, Contrast Sets, grew as an exploratory method. The algorithms of both approaches are based on the Max-Miner rule discovery system (Bayardo Jr., 1998). Therefore, we will briefly describe rule discovery.

Rule discovery is the process of finding rules that best describe a dataset. A dataset is a collection of records in which each record contains one or more discrete attribute-value pairs (or items). A rule is simply a combination of conditions that, if true, can be used to predict an outcome. A hypothetical rule about consumer purchasing behaviors, for example, might be *IF buys_milk AND buys_cookies THEN buys_cream*.

Association rule discovery (Agrawal, Imielinski & Swami, 1993; Agrawal & Srikant, 1994) is a popular rule-discovery approach. In association rule mining, rules are sought specifically in the form of where the antecedent group of items (or *itemset*), A, implies the consequent itemset, *C*. An association rule is written as $A \rightarrow C$. Of particular interest are the rules where the probability of *C* is increased when the items in *A* also occur. Often association rule-mining systems restrict the consequent itemset to hold only one item as it reduces the complexity of finding the rules.

In association rule mining, we often are searching for rules that fulfill the requirement of a minimum support criteria, *minsup*, and a minimum confidence criteria, *minconf*. Where support is defined as the frequency with which *A* and *C* co-occur:

$$\text{support}(A \rightarrow C) = \text{frequency}(\text{A} \cup \text{C})$$

and confidence is defined as the frequency with which *A* and *C* co-occur, divided by the frequency with which *A* occurs throughout all the data:

$$\text{confidence}(A \rightarrow C) = \frac{\text{support}(A \rightarrow C)}{\text{frequency}(A)}$$

The association rules discovered through this process then are sorted according to some user-specified interestingness measure before they are displayed to the user.

Another type of rule discovery is *k*-most interesting rule discovery (Webb, 2000). In contrast to the support-confidence framework, there is no minimum support or

confidence requirement. Instead, *k*-most interesting rule discovery focuses on the discovery of up to *k* rules that maximize some user-specified interestingness measure.

MAIN THRUST

Emerging Patterns

Emerging Pattern analysis is applied to two or more datasets, where each dataset contains data relating to a different group. An Emerging Pattern is defined as an itemset whose support increases significantly from one group to another (Dong & Li, 1999). This support increase is represented by the growth rate—the ratio of support of an itemset in group 1 over that of group 2. The support of a group G is given by:

$$\text{supp}_G(X) = \frac{\text{count}_G(X)}{|G|}$$

The $GrowthRate(X)$ is defined as 0 if $\text{supp}_1(X) = 0$ and $\text{supp}_2(X) = 0$; ∞ if $\text{supp}_1(X) = 0$ and $\text{supp}_2(X) \neq 0$; or else $\text{supp}_2(X)/\text{supp}_1(X)$. The special case where $GrowthRate(X) = \infty$ is called a Jumping Emerging Pattern, as it is said to have jumped from not occurring in one group to occurring in another group. This also can be thought of as an association rule having a confidence equaling 1.0.

Emerging Patterns are not presented to the user, as models are in the exploratory discovery framework. Rather, the Emerging Pattern discovery research has focused on using the mined Emerging Patterns for classification, similar to the goals of Liu et al. (1998, 2001). Emerging Pattern mining-based classification systems include CAEP (Dong, Zhang, Wong & Li, 1999), JEP-C (Li, Dong & Ramamohanarao, 2001), BCEP (Fan & Ramamohanarao, 2003), and DeEP (Li, Dong, Ramamohanarao & Wong, 2004). Since the Emerging Patterns are classification based, the focus is on classification accuracy. This means no filtering method is used, other than the infinite growth rate constraint used during discovery by some the classifiers (e.g., JEP-C and DeEP). This constraint discards any Emerging Pattern X for which $GrowthRate(X) \neq \infty$.

Contrast Sets

Contrast Sets (Bay & Pazzani, 1999, 2001) are similar to Emerging Patterns, in that they are also itemsets whose support differs significantly across datasets. However, the focus of Contrast Set research has been to develop an exploratory method for finding differences between one group and another that the user can utilize, rather than as a classification system focusing on prediction accuracy. To this end, they present filtering and pruning methods to ensure only the most interesting and optimal number rules are shown to the user, from what is potentially a large space of possible rules.

Contrast Sets are discovered using STUCCO, an algorithm that is based on the Max-Miner search algorithm (Bayardo Jr., 1998). Initially, only Contrast Sets are sought that have supports that are both significant and the difference large (i.e., the difference is greater than a user-defined parameter, *mindev*). Significant Contrast Sets (*cset*), therefore, are defined as those that meet the criteria:

$$P(cset \mid G_i) \neq P(cset \mid G_j)$$

Large Contrast Sets are those for which:

$$\text{support}(cset, G_i) - \text{support}(cset, G_j) \geq mindev$$

As Bay and Pazzani have noted, the user is likely to be overwhelmed by the number of results. Therefore, a filter method is applied to reduce the number of Contrast Sets presented to the user and to control the risk of type-1 error (i.e., the risk of reporting a Contrast Set when no difference exists). The filter method employed involves a chi-square test of statistical significance between the itemset on one group to that Contrast Set on the other group(s). A correction for multiple comparisons is applied that lowers the value of α as the size of the Contrast Set (number of attribute value pairs) increases.

Further pruning mechanisms also are used to filter Contrast Sets that are purely specializations of other more general Contrast Sets. This is done using another chi-square test of significance to test the difference between the parent Contrast Set and its specialization Contrast Set.

Mining Group Differences Using Rule Discovery

Webb, Butler, and Newlands (2003) studied how Contrast Sets relate to generic rule discovery approaches. They used the OPUS_AR algorithm-based Magnum Opus software to discover rules and to compare them to those discovered by the STUCCO algorithm.

OPUS_AR (Webb, 2000) is a rule-discovery algorithm based on the OPUS (Webb, 1995) efficient search technique, to which the Max-Miner algorithm is closely related. By limiting the consequent to a group variable, this rule discovery framework is able to be adapted for group discovery.