Utility Driven Clustering

P. Swapna Raj and **Balaraman Ravindran**
Department of Computer Science and Engineering
Indian Institute of Technology Madras
{pswapna,ravi}@cse.iitm.ac.in

Abstract

Data mining has primarily focused on statistical properties of data alone and not necessarily on what could be done with the patterns. While there has been some work on measuring usefulness of patterns in decision making but not on using such measures for driving the mining process. We introduce a framework to mine clusters that support decision making. We use an extrinsic measure that evaluates patterns based on their utility in decision making. We show empirical validation of our approach on several test domains.

Motivation

Traditionally clustering algorithms use inherent statistical properties of the data to evaluate the goodness of the clusters formed. These are known as *intrinsic measures*, examples include cluster purity, average diameter and RAND index.

Consider a scenario from the telecom domain, where a service provider wishes to segment his customer base, so as to promote various calling plans. Typically the promotions are focused on certain demographics and trying to sell a package designed for a business executive to a college student would probably not succeed.

Clustering based on the calling patterns of the customers would yield compact clusters. But in this case trading compactness of the clusters for more homogeneity in the demographic of the users clustered together is a better approach. The degree of the trade-off is usually quantified in the form of a utility function which encapsulates the payoff for taking different decisions in various contexts. The output of a clustering algorithm is evaluated in terms of the over-all payoff earned by promotional decisions made based on the clusters produced.

Our framework is the first of its kind that provides an approach to mine actionable clusters using an extrinsic measure. A cluster is actionable if the user can act upon it to his advantage. The extrinsic measure defines the value of the pattern with respect to an externally defined task. In order to mine these actionable clusters, we must be able to change the clustering process so as to obtain clusters based on which an optimal strategy can be provided. We call this approach *"utility-driven clustering"*.

Framework

In order to drive the mining of actionable patterns based on the utility in decision making we need to come up with a single complex objective function that we must optimize. In most formulations of the evaluation stage, the decision is related to the patterns mined via a complex optimization procedure and no simple relationship exists to the underlying intrinsic measures (Kleinberg, Papadimitriou, and Raghavan 1998). We therefore consider data mining and decision making as two different stages in order to mine clusters that support decision making.

Given a set of values for the hyper-parameters (parameters of the model) the clustering algorithm proceeds as usual. If we consider k-means clustering for example, the hyper-parameter can be the number of clusters, k, as well as the distance function, say weighted euclidean measure. Once the clusters are formed and the decision is taken with respect to the clusters in the decision making stage, we can employ an utility function for evaluating this decision. The evaluation stage then returns a single scalar evaluation of the clusters detected based on the utility of the optimal decisions supported by the patterns.

We impose a neighborhood structure on the hyper-parameter space and use the scalar evaluation to select new settings for the hyper-parameters using local search procedure in order to search for a better pattern with higher utility. We use the evaluation to drive a local search procedure since it is difficult to formulate the single complex objective function in terms of the hyper-parameter. The parameter of k-means is the set of centroids corresponding to the clusters and it is computationally expensive to store the centroids. Hence we chose to tune hyper-parameter space and not parameter space.

Our framework is flexible since we can accommodate any clustering algorithm. There are several potential hyper-parameters such as the diameter of the cluster in hierarchical clustering that can be tuned in order to get small/large clusters accordingly. We can also use a class of supervised clustering algorithm known as constrained clustering by incorporating background knowledge in the form of instance level constraints that are specified apriori. Typically constrained clustering uses a set of must-link and cannot-link constraints meaning that two instances must be in the same cluster or cannot be in the same cluster respectively (Wagstaff et al.

	Utility 1			Utility 2			Utility 3		
	Hyper	Hypo	Normal	Hyper	Hypo	Normal	Hyper	Hypo	Normal
Treatment A	+10	-9	-4	+10	-15	-9	+50	-25	-10
Treatment B	-5	+10	-3	-16	+10	-15	-25	+50	-10
No Treatment	-9	-6	+10	-19	-15	+10	-10	-10	+50

Table 1: Utility function - Thyroid dataset

2001). These constraints can be tuned in order to obtain better actionable clusters.

We use a couple of evaluation settings in our framework. In the first setting each cluster is associated with a single decision which is unique across all clusters. In the second setting each decision might be applied to multiple clusters as well. We can use other approaches to mine better patterns other than local search such as meta-learning.

Experiments

For the purposes of validating our framework, we use k-means clustering algorithm and the hyper-parameter that we use is the weighted euclidean measure. The neighborhood structure is a Manhattan structure, where only one hyper-parameter is allowed to change at a time, by a fixed step size (constant 1).

We conduct two different experiments : The experiment 1 tunes the hyper-parameters (weights of the distance metric). In order to speed up the local search process, we use a "momentum" term, wherein we keep increasing the weight of a particular feature which provides a high utility till no further improvement is possible. The experiment 2 performs feature selection. We perform feature selection by allowing hyper-parameters to take only binary values either 0 or 1 (attribute ignored or considered). We adopt a first-improvement strategy in order to speed up the process when the neighborhood is large.

We consider a simple florist task to illustrate the first evaluation setting using Iris dataset from UCI repository that contains 3 classes of flowers (iris setosa, iris versicolour and iris virginica). The dataset consists of 150 data points and 4 attributes. The task is to create bouquets according to users preference (not known apriori). We assume that each user can pick only one bouquet according to his preference from the available bouquets and leave with some amount of utility with respect to the bouquet. The decision making stage is modeled using the matching problem - Hungarian assignment, which solves the utility maximization problem and provides the utility which indicates the best bouquet each user picks according to the availability.

We create three users whose preferences over certain attributes is modeled using different α, β (parameters of the preferred features - say petal and sepal length) and γ (clusters cardinality) variables such as User 1 : 10, 3, 0.1, User 2 : 5, 5, 0.2; User 3 : 11, 4, 0.3. We conduct experiment 1 with different pairs of preferred features ordered in such a way that the first feature is given more preference than the second [(4,3), (2,4), (1,2)]. We observe that the features 4, 3 has the least average percentage improvement (10%), since the variance of these features are very high in our data. It is evident that features 2, 4 has higher improvement (21%) since the variance of these features are low and the user also expects more uniformity on those features. The experiment 2 is not conducted in this setup since the dataset already has minimal number of features.

We illustrate the second evaluation setting using thyroid dataset adopted from UCI repository. The dataset contains 3 classes (Hypo-thyroid, Hyper-thyroid, and Normal) with 3772 instances and 21 attributes. The decisions that we consider are providing either Treatment A, B or No treatment. The task is to cluster patients with similar symptoms and assign a single decision to the cluster in order to maximize the utility. The utility in this case is the sum of costs of applying a particular treatment to patients within the cluster.

We conduct experiments 1 and 2 using weighted and non-weighted approaches on small balanced dataset (93 hyper, 191 hypo, and 372 normal) and skewed dataset (93 hyper, 191 hypo, and 3488 normal). We use the weighted approach to handle data skewness. The weights used in skewed data for hyper, hypo and normal are 0.97, 0.95, 0.07. While the weights for balanced data are 0.95, 0.7, 0.43. We generally give more weightage to the rarer labels, in this case the hyper-patients.

	Experiment 1			Experiment 2		
	U1	U2	U3	U1	U2	U3
Wtg, Balanced	28	17	12	73	32	49
Wtg, Skewed	26	10	4	48	22	35
Non-wtg, Balanced	11	27	3	36	32	29

Table 2: Average percentage improvement

We used three different utility functions corresponding to different scenarios as shown in Table 1. We have reported results for k = 5 in Table 2. Our approach takes advantage of the trade-offs specified in the utility function to improve the results. For example we obtain average improvement of 73 % with balanced data, weighted approach, utility 1 since most of the hypo and normal patients are given the right treatment and most of the hyper-patients are classified as hypo-patients and are given treatment B.

References

Kleinberg, J.; Papadimitriou, C.; and Raghavan, P. 1998. A microeconomic view of data mining. *Data Mining Knowledge Discovery* 311–324.

Wagstaff, K.; Cardie, C.; Rogers, S.; and Schrodl, S. 2001. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML, 577–584.