Cluster-Grouping: From Subgroup Discovery to Clustering

Albrecht Zimmermann and Luc De Raedt

Institute of Computer Science, Machine Learning Lab, Albert-Ludwigs-University Freiburg, Georges-Köhler-Allee, 79110 Freiburg, Germany
{azimmerm,deraedt}@informatik.uni-freiburg.de

Abstract. The problem of **cluster-grouping** is defined. It integrates subgroup discovery, mining correlated patterns and aspects from clustering. The algorithm *CG* for solving cluster-grouping problems is presented and experimentally evaluated on a number of real-life data sets. The results indicate that the algorithm improves upon the subgroup discovery algorithm *CN2-WRACC* and is competitive with the clustering algorithm *CobWeb*.

Keywords: clustering, subgroup discovery, correlated pattern mining.

1 Problem Specification and Context

The problem of cluster-grouping integrates subgroup discovery, mining correlated patterns and aspects from clustering. Subgroup discovery [1] aims at finding groups in the data that are over- or under-represented w.r.t. a specific target attribute; correlated pattern mining [2] is a form of association rule mining, which aims at finding rules whose condition part correlates strongly with its conclusion part w.r.t. a statistical evaluation criterion (e.g. χ^2 or entropy); and clustering [3] aims at identifying groups that are homogeneous w.r.t. an evaluation criterion such as *category utility*.

Although these three techniques are perceived as being quite different in the literature, it turns out that they are an instance of the more general problem of cluster-grouping that we introduce below. The cluster-grouping problem is concerned with finding rules $b_1 \wedge ... \wedge b_c \rightsquigarrow h_1 \vee ... \vee h_d$ (over boolean variables) that score best w.r.t. an interestingness function σ and set $\mathcal{E}$ of instances. We call d the dimension of the rule.
More formally, the *cluster grouping* problem can be defined as follows:

- **Given**
  - a set of rules $\mathcal{L}$ (the hypothesis space)
  - a set of instances $\mathcal{E}$ (i.e. boolean variable assignments)
  - an convex interestingness measure $\sigma : \mathcal{E} \times \mathcal{L} \mapsto \mathbb{R}$
  - a positive integer k
- **Find** the k rules in $\mathcal{L}$ that have the highest score w.r.t. σ and $\mathcal{E}$.

J.-F. Boulicaut et al. (Eds.): ECML 2004, LNAI 3201, pp. 575–577, 2004.

Subgroup discovery (as studied by [1]) is the special case of cluster-grouping where the conclusion part of the rules in $\mathcal{L}$ is a fixed boolean attribute and σ is *weighted relative accuracy (WRAcc)*; correlated pattern mining (as studied by [2]) allows for rules of dimension 1 and employs convex interestingness measures (such as χ^2 and entropy); and conceptual clustering can be regarded as the problem of finding k rules (whose condition part defines the clusters and whose conclusion part defines the d boolean variables of interest) w.r.t. a measure such as *category utility*.

2 The *CG*-Algorithm

The *CG*-algorithm for cluster grouping is an extension of Morishita and Sese's algorithm [2] for correlated pattern mining. Whereas Morishita and Sese coonsidered only rules of dimension 1, *CG* allows for rules of arbitrary dimension d. *CG* is similar to the correlated pattern mining algorithm of [2] in that it employs a branch-and-bound algorithm to search for the k best patterns w.r.t. the interestingness measure σ. The key idea underlying the algorithm is that for *convex* functions it is possible to compute an upper bound $u(r)$ on the quality of a rule r and all its specializations.

The *CG* algorithm works as follows. It initializes the queue of candidate solutions Q with the most general rule. It then repeatedly deletes the best candidate c from Q and evaluates its refinements w.r.t. u and σ. If a refinement is among the k best patterns already encountered, it is added to the current list of solutions. If a refinement's upper bound u scores worse than that of the worst element on the current list of solutions, it is discarded. All other refinements are added to the current list of candidates. The search continues until the list of candidates becomes empty.

To compute the upper bound, Morisha and Sese introduce the concept of a *stamp point* $\langle x, y \rangle$ with x denoting the coverage of a rule and y denoting the number of true positives. Correlation measures are then treated as functions defined on stamp points. While the *actual* future stamp points for specializations of the rule cannot be known in advance, the current stamp point constrains the set of *possible* future stamp points S_{poss}. The upper bound mentioned above is calculated by evaluating the correlation measure on the points lying on the convex hull of S_{poss}. We have extended this technique to arbitrary dimension d, allowing it to be used in clustering (in which the behavior of a rule with regard to all attributes is used as guidance in the search). For rules of dimension 1, the convex hull is a parallelogram, for dimension d one has to consider a hyperbody. In determining the vertices of that body, additional restrictions have to be observed preventing a simple recursion of the two-dimensional technique.

3 Experiments

We performed experiments on a variety of UCI data sets. We compared our approach to *CobWeb* [3] for clustering and to *CN2-WRAcc* [1] for subgroup discovery.

For clustering we applied *CG* to the initial data set, mining the rule with the highest *category utility* and used the condition part of the resulting rule as a splitting criterion. *CG* was then applied on the resulting subsets. In this way we construct a hierachical clustering. For comparison we computed the *category utility* of *Cobweb*'s solutions (averaged over 10 randomized orderings) and *CG*'s solutions and also the agreement between the respective solutions using the *Rand Index*. The resulting *category utilities* are shown in the left-hand table below.

Dataset	*CU CG*	*CU CobWeb*
Breast-w	0.62	0.6496 ± 0.0001
Breast-w-equal	1.088	$1.147 \pm 1.95 * 10^{-5}$
Credit-a	0.379	0.374 ± 0.0178
Credit-a-equal	0.6241	0.6243 ± 0.00067
Glass	0.301	0.291 ± 0.0125
Hepatitis	0.446	0.459 ± 0.0142
Iris	0.5369	0.5321 ± 0.0083
Sick	0.2132	$.2077 \pm 0.0171$
Voting	1.362	1.468 ± 0.0001
Zoo (6 clusters)	0.6398	0.6349 ± 0.005
Zoo (5 clusters)	0.7187	0.7196 ± 0.004

Data set	*CG*	*CN2-WRAcc*
Car	44.5 ± 38.8	84.75 ± 9.2
Zoo	1531 ± 1980.1	2133.6 ± 27.7
Nursery	82.6 ± 108.1	141.4 ± 13.1
Breast-W	95.5 ± 6.4	529
Voting	36 ± 4.2	301
Mushroom	196.5 ± 34.7	1806 ± 4.2

Some of *CobWeb*'s solutions had lower *category utility* than the *CG* solution. While *CobWeb* also found solutions having higher *category utility*, those could not easily be described by conjunctive rules. In general the agreement between the *CobWeb* and *CG* solutions is very high ($93.2\% \pm 5.3\%$).

For subgroup discovery we used *CG* to compute all rules achieving optimal value. We compared those rules to *CN2-WRAcc*'s solutions w.r.t whether the rules with highest *WRAcc* value were found and whether all such rules had been found. In the right-hand table above the average number of candidate rules considered during the search process are shown for *CN2-WRAcc* with beam size 1 and for *CG*.

CN2-WRAcc fails to always find the highest-scoring rules as *CG* does. This is the case even for beam sizes in excess of 10 and up to 50. Additionally *CN2-WRAcc* considered more candidate rules during the search even for small beam sizes.

The results presented above show that *CG* is a valid alternative to *CN2-WRAcc* and *CobWeb*.

References

1. Todorovski, L., Flach, P., Lavrac N. Predictive Performance of Weighted Relative Accuracy. *Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases.* Lyon, France. September 2000
2. Morishita, S., Sese, J. Traversing Itemset Lattice with Statistical Metric Pruning. *Proceedings of the 19th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems.* Dallas, Texas, USA. May 2000
3. Fisher, D.H. Knowledge acquisition via incremental conceptual clustering. *Machine Learning, Volume 2, Number 2*, 139-172, Kluwer Academic. 1987