A Methodological View on Knowledge-Intensive Subgroup Discovery

Martin Atzmueller and Frank Puppe

Department of Computer Science
University of Würzburg, 97074 Würzburg, Germany
Phone: +49 931 888-6739; Fax: +49 931 888-6732
{atzmueller, puppe}@informatik.uni-wuerzburg.de

Abstract. Background knowledge is a natural resource for knowledge-intensive methods: Its exploitation can often improve the quality of their results significantly. In this paper we present a methodological view on knowledge-intensive subgroup discovery: We introduce different classes and specific types of useful background knowledge, discuss their benefit and costs, and describe their application in the subgroup discovery setting.

1 Introduction

Knowledge-intensive learning methods (e.g., [1]) use background knowledge for a simple reason: Utilizing background knowledge can often significantly improve both the quality of their results and the efficiency of the search process. In this paper, we describe how to exploit background knowledge for subgroup discovery, a method that has first been formalized by Klösgen [2] and Wrobel [3]: Subgroup discovery is a powerful and broadly applicable technique aiming at discovering interesting subgroups concerning a certain target property of interest, e.g., in the subgroup of smokers with a positive family history the risk of coronary heart disease (target property) is significantly higher than in the general population.

Background knowledge can help to improve subgroup discovery in several ways, e.g., it can increase the representational expressiveness and also focus the subgroup discovery algorithm on the relevant patterns. Then, similar to a constrained query to a web search engine, the user is not flooded with too many (uninteresting) results. Furthermore, for increasing the efficiency of the search method the search space can often be constrained. However, knowledge acquisition is often challenging and costly, known as the 'knowledge acquisition bottleneck': Then, an important idea is to ease knowledge acquisition by reusing existing domain knowledge, i.e., knowledge that is already known to the user, or that is contained in existing ontologies or knowledge bases. Therefore, we propose to apply as much background knowledge as possible, with potentially reduced costs by knowledge reuse.

The rest of the paper is organized as follows: We first briefly introduce subgroup discovery in Section 2. After that, we propose several types of background knowledge in Section 3, discuss their benefit and costs, and describe how they can be applied for subgroup discovery in Section 4. Finally, we conclude the paper with a discussion and summary in Section 5, and point out interesting directions for future work.

S. Staab and V. Svatek (Eds.): EKAW 2006, LNAI 4248, pp. 318–325, 2006.

2 Subgroup Discovery

The main application areas of subgroup discovery [2,3] are exploration and descriptive induction, to obtain an overview of the relations between a target variable and a set of explaining variables. A subgroup discovery setting includes a target variable (concept of interest), a subgroup description language, a specific quality function, and a search strategy for which, e.g., a beam search technique [3] is often applied:

Let Ω_A be the set of all attributes. For each attribute $a \in \Omega_A$ a range $dom(a)$ of values is defined; we assume $\mathcal{V}_A$ to be the (universal) set of attribute values of the form $(a = v)$, $a \in \Omega_A, v \in dom(a)$. A single-relational propositional subgroup description $sd = \{e_1, e_2, \ldots, e_n\}$ is defined by the conjunction of a set of selection expressions (selectors) $e_i = (a_i, V_i)$, i.e., selections on domains of attributes, $a_i \in \Omega_A, V_i \subseteq dom(a_i)$. We define Ω_{sd} as the set of all possible subgroup descriptions. The interestingness of a subgroup can be flexibly formalized by a (user-defined) quality function $q : \Omega_{sd} \rightarrow R$ (e.g., [2]) that is used in order to evaluate a subgroup description $sd \in \Omega_{sd}$. Typical quality criteria include the difference in the distribution of the target variable concerning the subgroup and the general population, and the subgroup size. Usually the (post-processed) k best subgroups and/or the subgroups with a quality above a minimum threshold are presented to the user as the result of the subgroup discovery method.

3 Types and Classes of Background Knowledge

The proposed classes of background knowledge include constraints, ontological knowledge and abstraction knowledge which we describe below: Constraints specify conditions that the mined patterns need to satisfy, e.g., quality and language constraints. Ontological knowledge describes general properties of the objects contained in the domain ontology and can be used to infer additional constraints. Abstraction knowledge is given by 'virtual' rule-based attributes. Figure 1 shows the knowledge hierarchy, from the three knowledge classes to the specific types, and the objects they apply to.

Constraint knowledge can be applied, e.g., for filtering patterns by their quality, and for restricting the search space. We distinguish the following types:

- **Language constraints** can, e.g., restrict the maximal number of conjuncts of a subgroup description. The description language itself can range from purely conjunctive languages to languages allowing internal disjunctions and negation.
- **Quality constraints** relate, e.g., to a minimum quality value, a minimum support, or a statistical significance threshold, that the subgroup patterns need to satisfy.
- **Value exclusion constraints** and **attribute exclusion constraints** are applied for filtering the domains of attributes and the attribute space, respectively.
- **Value aggregation constraints** can be specified in order to form abstracted disjunctions of attribute values, e.g., intervals for ordinal values. For example, consider the attribute *age* with the values '< 40', '$40 - 50$', '$50 - 70$', '> 70': Then, we can derive the aggregated values '≤ 50' and '> 50'. In general, aggregated values are not restricted to intervals, but can cover any combination of values.
- **Attribute combination constraints** are applied for filtering/excluding certain combinations of attributes, e.g., if these are already known to the domain specialist.

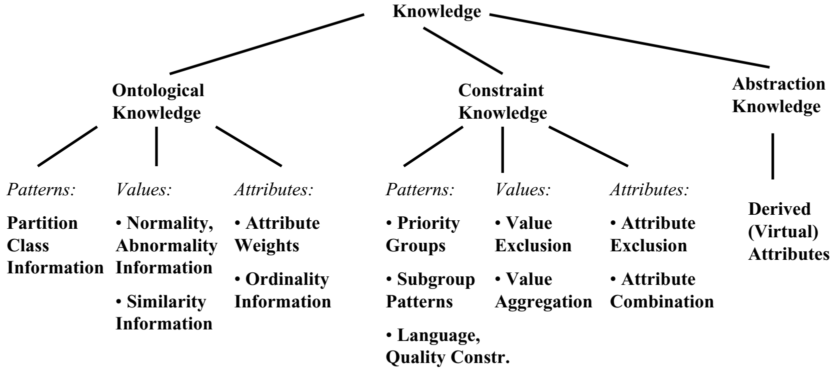

Fig. 1. Hierarchy of (abstract) knowledge classes and specific types

- **Priority groups** are partially disjunctive sets of attributes with an assigned priority. The subgroup discovery method starts with the attribute set with the highest priority. If the currently considered subgroups cannot be improved any further, then it iteratively takes the next (prioritized) set of attributes into account.
- **Subgroup pattern constraints** given by selected subgroup patterns can be used, e.g., to avoid the rediscovery of already known subgroups, for comparison to a (new) set of subgroups, and for deriving new attributes as discussed in Section 4.

Ontological knowledge is commonly used for the development of knowledge systems. The knowledge can either be defined by the user, or can partially be learned semi-automatically (e.g., [4]). It consists of the following types:

- **Attribute weights** denote the relative importance of attributes, and are a common extension for knowledge-based systems, e.g., for case-based reasoning systems [4].
- **Abnormality/Normality information** is usually easy to obtain for diagnostic domains, e.g., in the medical domain the set of 'normal' attribute values contains the expected values, and the set of 'abnormal' values contains the unexpected/pathological ones; often the unexpected values are more interesting for analysis. Each attribute value is attached with a label specifying a normal or an abnormal state. Normality information only requires a binary label. Abnormality information defines several categories, e.g., consider the value range {normal, marginal, high, very high} of the attribute *temperature*. The values *normal* and *marginal* denote normal states of the attribute while the values *high* and *very high* describe abnormal states.
- **Similarity information** between attributes values is often applied in case-based reasoning: It specifies the relative similarity between the individual attribute values. For example, for a nominal attribute *color* with the value range *white*, *gray*, *black* we can state that the value *white* is more similar to *gray* than it is to *black*.
- **Ordinality information** specifies if the value domain of a nominal attribute can be ordered, e.g., the qualitative ones *age* and *liver size* are ordinal while *color* is not.
- **Partition class information** provides semantically distinct groups of attributes. These partially disjoint subsets usually correspond to certain problem areas of the application domain, e.g., in the medical domain of sonography such partitions are representing different organ systems like *liver*, *pancreas*, *stomach*, and *kidney*.

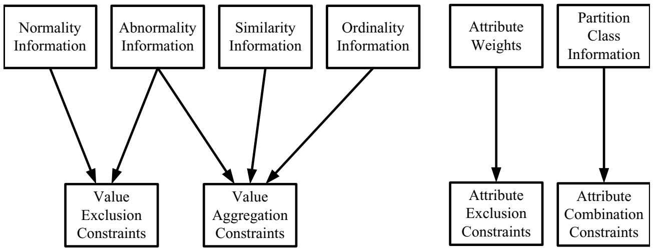
Normality Information
Abnormality Information
Similarity Information
Ordinality Information
Attribute Weights
Partition Class Information
Value Exclusion Constraints
Value Aggregation Constraints
Attribute Exclusion Constraints
Attribute Combination Constraints

Fig. 2. Deriving constraints using ontological knowledge

Figure 2 shows how ontological knowledge can be used in order to derive further 'basic' constraints. Below, we summarize how new constraints can be inferred using ontological knowledge.

- We can construct attribute exclusion constraints using attribute weights to filter the set of relevant attributes by a weight threshold or by subsets of the weight space.
- Using abnormality/normality knowledge we can specify global value exclusion constraints for a set of abnormal values, or for the normal values.
- Using similarity or abnormality/normality information we can filter and model the value ranges of attributes: If the similarity between two attribute values is very high, then they can potentially be analyzed as an *aggregated value*. Similarly, global abnormality groups can be defined by sets of abnormality degrees specifying which values to combine. For example, in the medical domain attribute values such as *probable* and *possible* (with different abnormality degrees) can often be aggregated.
- Ordinality information can be easily used to construct aggregated values, which are often more meaningful for the domain specialist: We can consider all adjacent combinations of attribute values, or all ascending/descending combinations starting with the minimum or maximum value, respectively. Whenever abnormality information is available, we can partition the value range by the given *normal* value and only start with the most extreme value. An example is discussed in Section 4.1.
- Partition class information can be used to infer attribute combination constraints in order to prevent the combination of individual attributes that are contained in separate partition classes. Alternatively, inverse constraints can also be derived, e.g., to specifically investigate inter-organ relations in the medical domain.

Abstraction knowledge is given by derived (rule-based) attributes. These abstractions often correspond to certain known dependencies between attributes, e.g., in the medical domain, we can infer the *body mass index*, given the attributes *height* and *weight*. For deriving a value v_a of a nominal attribute a, a rule of the form $r_{v_a} = cond(r_{v_a}) \rightarrow v_a$ is used, where the rule condition $cond(r_{v_a})$ contains conjunctions and/or disjunctions of (negated) attribute values $v_i \in \mathcal{V}_A$. The derived attributes serve three main purposes:

- They focus the subgroup discovery method on the relevant analysis objects.
- They decrease multi-correlations between attributes that are not interesting.
- Derived attributes can reduce missing values for a given concept, since they can be constructed such that a defined value is more often computed if the respective concept would have a missing value otherwise.

Due to the limited space we refer to [5] for a detailed discussion. Abstraction knowledge is probably the most costly class of background knowledge: If the abstractions are not based on discovery results, then they have to be formalized manually by the expert.

4 Background Knowledge: Applicability, Benefit and Cost

In the table below we summarize the characteristics of the proposed classes and types of background knowledge (CK = constraint knowledge, OK = ontological knowledge, AK = abstraction knowledge) in terms of the 'derivable knowledge' (if applicable), their syntactical and cognitive costs, and their potential contribution to restricting the search space and/or focusing the search process. Considering the costs and the impact of the knowledge types on the search space, the label - indicates no cost/impact; the labels +, ++, and +++ indicate increasing costs and impact. A +(+) signifies, that the respective element has low costs if it can be derived/learned, and moderate costs otherwise. Similarly ++(+) indicates this for moderate and high costs.

In our experience, the most important types of knowledge with an especially good cost/benefit ratio are *quality constraints*, *attribute exclusion constraints*, *normality information*, *ordinality information*, and especially *derived attributs* (indicated in bold type). In the next section, we provide examples for applying most of these knowledge types. After that, we summarize how we can exploit background knowledge for subgroup discovery.

Knowledge		*Derivable*	*Costs*		Search	
Class	*Type*	*Knowledge*	*Syn.*	*Cog.*	*Restr.*	*Foc.*
CK	Language C.	–	–	+	++	+
CK	**Quality Constr.**	–	–	++	++	++
CK	Value Exclusion Constr.	–	(+)	+	+	+
CK	Val. Aggregation Constr.	–	(+)	+(+)	++	+
CK	**Attr. Exclusion Constr.**	–	(+)	+(+)	++	++
CK	Attr. Combination Constr.	–	(+)	+(+)	++	++
CK	Priority Group Constr.	–	+	++	–	+
CK	Subgroup Pattern Constr.	Deriv. Attr.	+(+)	+(+)	–	++
OK	**Normality Information**	Val. Excl.	+	+	++	++
OK	Abnormality Information	Val. Excl.	++	++	++	++
		Val. Aggr.			+	++
OK	Similarity Information	Val. Aggr.	+(+)	+(+)	++	++
OK	**Ordinality Information**	Val. Aggr.	+	+	+++	++
OK	Attribute Weights	Attr. Excl.	(+)	+(+)	++	++
OK	Partition Class Inform.	Attr. Comb.	+	+	++	++
AK	***Derived Attributes***	Deriv. Attr.	+++	+++	–	+++

4.1 Background Knowledge – Examples

Let A be a nominal attribute with the range $dom(A) = \{a_1, a_2, a_3, a_n, a_5, a_6, a_7\}$ of attribute values, e.g., A could correspond to the (discretized) attribute *body weight* with values like *massive underweight*, *strong underweight*, *underweight*, *normal weight*, *overweight*, *strong overweight*, and *massive overweight*. Ordinality information can be easily applied in order to derive a restricted set of aggregated values denoting different weight groups. If we want to exclude all combinations not being neighbors (excluding irrelevant combinations like (a_1, a_3)), we obtain only 77 combinations of all adjacent attribute values, in contrast to considering all possible 127 attribute value combinations:

$$(a_1, a_2), (a_1, a_2, a_3), \ldots, (a_1, \ldots, a_7), (a_2, a_3), (a_2, \ldots, a_7), \ldots (a_6, a_7) .$$

In the medical domain we often know that a certain attribute value denotes the *normal* value (in our example 'normal weight' = a_4). This value is often not interesting for the analyst who might focus on the 'abnormal' value combinations. Combining normality and ordinality information, we then only need to consider 10 combinations:

$$(a_1), (a_1, a_2), (a_1, a_2, a_3), (a_2, a_3), (a_3), (a_7), (a_7, a_6), (a_7, a_6, a_5), (a_6, a_5), (a_5) .$$

If we are interested only in combinations including the most extreme value (typical in medicine), we can further reduce the number of 'meaningful' combinations to 6:

$$(a_1), (a_1, a_2), (a_1, a_2, a_3), (a_7), (a_7, a_6), (a_7, a_6, a_5) .$$

The savings of such a reduction of value combinations, which can be derived using ordinality, normality information and interestingness assumptions, are huge: If there are 10 attributes like A with seven values each, then the size of the search space considering all possible selector combinations is reduced from $128^{10} = 10^{21}$ to $7^{10} = 3 \cdot 10^{8}$.

Concerning abstraction knowledge, let us consider an additional attribute B denoting the *body height* with the (ordinal) value range $dom(B) = \{b_1, b_2, b_3, b_n, b_5, b_6, b_7\}$.

In the following, we assume that both A and B are quantitative nominal attributes. Then, we can derive the attribute *body mass index (BMI)* given the body weight (attribute *A*) and the body height (attribute *B*). The matrix shows the combinations of the respective attribute values: The derived attribute values corresponding to a high body mass index are given by the entries $1, 2, 3, 4$ in ascending order, while a '0' denotes the 'normal' case.

	a_1	a_2	a_3	a_4	a_5	a_6	a_7
b_1	0	0	1	2	3	4	4
b_2		0	0	1	2	3	4
b_3			0	0	1	2	3
b_4				0	0	1	2
b_5					0	0	1
b_6						0	0
b_7							0

It is easy to see that in this example the 'meaningful' combinations of the respective attribute values are always on the diagonal, or form triangular matrices, e.g., considering the entries '3' and '4' of the matrix. In our example, these combinations correspond to relatively small people with a large body weight: In principle, the distribution of the individual values can be arbitrary. Then, the distributions of the combined attribute values can also be of arbitrary shape. By constructing selection expressions containing internal disjunctions we can only select quadrangular sub-matrices and would thus include larger groups that can 'confound' the 'new values', i.e., the original value combinations, since the quadrangular sub-matrices might contain at least one potentially misleading value combination. In contrast, using derived attributes we can carve out arbitrary parts of the matrix, e.g., the triangular sub-matrices shown in the example. Then, a derived attribute capturing the specific value combinations is more expressive and meaningful for the user, and can focus the analysis significantly.

4.2 Applying Background Knowledge for Subgroup Discovery

In the following, we describe knowledge elements considering their effect(s) for the subgroup discovery task, i.e., restricting the search space, focusing the search process, post-processing the results, and increasing the representational expressiveness.

Restricting the Search Space and Focusing Search. Most of the knowledge classes described in Section 3 can be directly integrated in the subgroup discovery step:

- Language constraints and quality constraints are applied as filters in order to restrict the search space and to focus the search process, e.g., by providing concise/simple description languages and by pruning uninteresting hypotheses below minimal quality and interestingness thresholds.
- Constraint knowledge (and ontological knowledge that is used to derive constraint knowledge) such as value exclusion constraints, value aggregation constraints and attribute exclusion constraints helps to focus the search process. While attribute exclusion and value exclusion constraints restrict the search space just by construction, value aggregation constraints do not necessarily restrict the search process since new values are introduced. However, value aggregation constraints can

provide significant quality improvements with low costs, if the aggregated values are more meaningful for the user. Additionally, if only the generated new values are taken into account, e.g., for ordinal value groups, then the search space remains the same or is even restricted. Furthermore, attribute combination constraints that inhibit the examination of specified sets of attributes can prune large (uninteresting) areas of the search space. Priority groups are utilized to focus the search process by construction: The attributes of the different priority groups are taken into the search space subsequently according to the requirements of the user.

- Subgroup pattern constraints contained in the background knowledge can be included into the process by considering them as starting points for the search process. Furthermore, derived attributes can be incrementally defined using (discovered) subgroup patterns during the discovery step. Additionally, by comparison to already known subgroup patterns we can inhibit the rediscovery of subgroups.
- Abstraction knowledge can be applied for increasing the representational expressiveness as discussed below, and for focusing the search process on the relevant objects. If only these are considered, then the search space can also be restricted.

Post-processing the Discovered Subgroups. The most important type of background knowledge for post-processing is given by specific known subgroup patterns itself: For example, in the medical domain often a lot of the existing relations are already known and can be formalized as subgroup patterns. By comparison with the discovered knowledge, (unexpected) patterns that conform to, deviate, or contradict the given domain knowledge can be identified. In addition to specific subgroup patterns we can also apply partition class information in order to mark subgroups that conform to the partition classes, or to identify subgroups that contain attributes included in different partition classes. This depends on the requirements of the user, e.g., in the medical domain different organ systems can be considered.

Increasing the Expressiveness of Subgroup Patterns. For increasing the representational expressiveness, (derived) attributes and subsets of the value range of an attribute can be utilized to infer new attributes and values, respectively, that are more meaningful for the user: The power of derived attributes lies in their ability of abstracting (known) associations of attributes into new attributes. These correspond to new concepts that are usually more meaningful, reasonable, and ultimately more important for the user. Thus, the search process can be focused significantly. Furthermore, the power of the statistical evaluations is increased significantly if missing values are minimized: Since abstraction knowledge can be used to infer missing values in their respective context, derived attributes can help to improve the missing value problem significantly.

Furthermore, aggregated values forming a disjunctive selection expression can be more meaningful and reasonable for the user, e.g., considering different aggregated age groups in the medical domain. We can apply abnormality or similarity information in order to derive value aggregation constraints. Then, these new values can be directly utilized in the search process. Additionally, the description language itself plays an important role, since it is used to define the subgroups. As a simple and concise description language often conjunctive languages without internal disjunctions are applied.

5 Conclusion

In this paper we presented a methodological view on exploiting background knowledge for subgroup discovery. We described several classes of background knowledge, and discussed the benefit, cost, and application of the particular types of knowledge.

In contrast to existing approaches utilizing background knowledge, including Inductive Logic Programming (ILP) (e.g., [6]), constraint-based data mining (e.g., [7]), and association rule learning techniques (e.g., [8]), we propose to integrate several new types of additional background knowledge: It can be used to easily infer new background knowledge on the fly, e.g., constraints, and can be refined incrementally according to the requirements of the discovery task. Furthermore, we propose special abstraction knowledge that can be applied dynamically. Compared to common preprocessing methods, the background knowledge concerning aggregations of attributes or attribute values is applied dynamically on the data. The original data set is not changed by the knowledge-intensive approach; instead, either the discovery method is 'configured' applying the knowledge, or 'virtual' attributes/attribute values are introduced.

We already successfully applied parts of the presented approach in different case studies in the medical domain [5,9]: For these, the application of background knowledge was essential, since a naive approach resulted in (too) many subgroups that were not regarded as interesting or were already known to the domain specialists.

In the future, we want to examine methods that enable the automatic construction of abstraction knowledge. An 'intelligent' adaptation and fine-tuning of aggregations of attribute values is another interesting issue to consider.

References

1. Richardson, M., Domingos, P.: Learning with Knowledge from Multiple Experts. In: Proc. 20th Intl. Conference on Machine Learning (ICML-2003), AAAI Press (2003) 624–631
2. Klösgen, W.: Explora: A Multipattern and Multistrategy Discovery Assistant. In: Advances in Knowledge Discovery and Data Mining. AAAI Press (1996) 249–271
3. Wrobel, S.: An Algorithm for Multi-Relational Discovery of Subgroups. In Komorowski, J., Zytkow, J., eds.: Proc. 1st European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD-97), Berlin, Springer (1997) 78–87
4. Baumeister, J., Atzmueller, M., Puppe, F.: Inductive Learning for Case-Based Diagnosis with Multiple Faults. In: Advances in Case-Based Reasoning. Volume 2416 of LNAI., Berlin, Springer (2002) 28–42 Proc. 6th European Conference on Case-Based Reasoning.
5. Atzmueller, M., Puppe, F., Buscher, H.P.: Exploiting Background Knowledge for Knowledge-Intensive Subgroup Discovery. In: Proc. 19th Intl. Joint Conference on Artificial Intelligence (IJCAI-05), Edinburgh, Scotland (2005) 647–652
6. Zelezny, F., Lavrac, N., Dzeroski, S.: Using Constraints in Relational Subgroup Discovery. In: Intl. Conference on Methodology and Statistics, University of Ljubljana (2003) 78–81
7. Boulicaut, J.F., Jeudy, B.: Constraint-based data mining. In: The Data Mining and Knowledge Discovery Handbook. Springer (2005)
8. Liu, B., Hsu, W.: Post-Analysis of Learned Rules. In: Proc. 13th National Conference on Artificial Intelligence (AAAI-96), Menlo Park, CA, AAAI Press (1996) 828–834
9. Atzmueller, M., Baumeister, J., Hemsing, A., Richter, E.J., Puppe, F.: Subgroup Mining for Interactive Knowledge Refinement. In: Proc. 10th Conference on Artificial Intelligence in Medicine (AIME 05). LNAI 3581, Berlin, Springer (2005) 453–462