A Framework for Pattern-Based Global Models

Arnaud Giacometti, Eynollah Khanjari Miyaneh,
Patrick Marcel, and Arnaud Soulet

Université François Rabelais Tours, L.I.
41000 Blois, France
Eynollah.khanjari@etu.univ-tours.fr,
{arnaud.giacometti,patrick.marcel,arnaud.soulet}@univ-tours.fr

Abstract. Discovering global models on a dataset (e.g., classifiers, clusterings, summaries) has attracted a lot of attention and many approaches can be found in the literature. However no framework has been proposed yet for describing and comparing these approaches in a uniform manner. In this paper we propose such a framework for pattern-based modeling approaches, i.e., approaches that use local patterns to construct a global model. This framework includes a generic algorithm (IGMA) for constructing a global model. We show that the framework allows to describe in an as declarative as possible way various different global model construction methods.

Keywords: global model, model construction, local pattern, clustering, summarization.

1 Introduction

Many methods have been proposed that gather together interesting local patterns for building global models like classifiers, clustering or summaries aiming to provide a global description of the whole dataset. We consider that all such methods can be divided into two major categories. On the one hand, the *separate-and-conquer* methods [6] directly construct the model from the data by employing heuristic-based greedy algorithms to search for the best interesting pattern w.r.t the set of uncovered objects, adding it to the model in progress, and continuing the same process until the resulting model describes the whole data set. On the other hand, the *two-step* methods [10] proceed as follows: (1) The first phase generates an exhaustive collection of local patterns and (2) The second phase, which is often greedy, selects a smaller subset of complementary relevant patterns.

Unfortunately, most of the existing pattern-based modeling methods are defined individually with ad-hoc properties applying problem-specific algorithms instead of favouring a declarative manner. The integration of such methods into a common and flexible environment is an important longterm goal in data mining [9]. Such an integration lies on the proposition of a uniform framework including generic algorithms, allowing the user to specify and compare different approaches in order to construct his desired model without focusing on procedural details of model construction and/or pattern extraction.

E. Corchado and H. Yin (Eds.): IDEAL 2009, LNCS 5788, pp. 433–440, 2009.

The present paper copes with this problem by providing a generic pattern-based framework that uniformly describes major classes of global models, namely classification, clustering and summarization. The framework includes a generic model construction algorithm (IGMA) which bridges the gap between separate-and-conquer and two-step methods. Its main idea is to repeat a more and more constrained exhaustive extraction of local patterns and the addition of the best local patterns to the global model in progress. The model description not only is independent from a specific pattern language (e.g. itemsets, sequences or graphs) but also is declarative in the sense that the user just needs to give a few parameters related to interestingness of patterns and models. Our framework covers a wide range of existing methods including classifiers in [7], clusterings [3,4,8] and summarizations [1,11,13]. The work presented in this paper extends our preliminary work in [7] introduced for rule-based classification.

The outline of this paper is as follows. Section 2 briefly reviews related work. In Section 3, we recall basic definitions and present our formal framework. Section 4 introduces IGMA, the Incremental Global Model construction Algorithm. The generality of the framework and IGMA is illustrated in Section 5. Finally, Section 6 concludes and discusses future works.

2 Related Work

In this section we review the general formalisms proposed for describing global models gathering a set of local patterns.

Separate-and-conquer methods always rely on the same well-known principles because of their greedy nature [2]. Nevertheless, only [6] addresses separate-and-conquer framework for a subset of rule-based classifiers. However, this framework is not suited for two-step methods.

Conversely, [10] proposes a two-step framework (called LeGo) which is a general process for pattern-based model construction. Even if the LeGo framework describes all two-step methods, it does not provide a general algorithm.

Besides the above frameworks, there exist other general two-step methods dealing with a given set of patterns to construct a global model. The Chosen Few [1] and the Pattern Ordering [13] approaches aim to summarize an initial collection of patterns. Whereas the former gives a summary that characterizes the whole dataset the latter finds k patterns enabling to regenerate approximative information about the collection. Thereby, these methods mainly focus on the second phase of two-step methods by proposing a generic greedy algorithm. On the contrary, [15] proposes a generic constraint-based *pattern set* mining approach, i.e. a non-greedy approach for the second phase. However, this method suffers from several distinct drawbacks: expressivity for separate-and-conquer methods as well as the hardness [11] of pattern set evaluation, scalability, and efficiency in the absence of properties like boundability.

Finally, all the above frameworks and generic methods [10,1,13,15] are unadapted to specify separate-and-conquer model construction.

A proposal that concerns the notion of inductive querying is IQL [14]. Despite strong theoretical backgrounds, IQL suffers from several drawbacks, namely complexity and incapability of describing separate-and-conquer model construction in the form of *virtual relations* in the database.

Thus, to the best of our knowledge, no pattern-based framework subsumes both separate-and-conquer and two-step global model construction methods.

3 Formal Framework

In this section, we give the formal definitions of the notions used throughout the paper to describe the pattern-based global modeling process.

3.1 Local Patterns and Objects

A *local pattern* is assumed as an abstract notion and we do not restrict it into a particular type (e.g. itemsets, sequence or graphs). Let $\mathcal{L}$ be the set of local patterns (pattern and local pattern are used interchangeably in what follows). A dataset, usually denoted by D, is a multi-set of elements of $\mathcal{L}$, and the elements of D are called *object*. Let $\mathcal{D}$ be the set of all such multi-sets, i.e. $D \in \mathcal{D}$. We denote by $|S|$ the cardinality of a set S.

Given a specialization relation $\preceq$ on $\mathcal{L}$, we note $\varphi_1 \preceq \varphi_2$ to mean the pattern φ_1 is more general than φ_2. We say that a pattern φ covers an object d iff $\varphi \preceq d$. Let φ be a pattern and $D = \{\text{ABC,ABC,ABCDE,BCDE,BCDE}\}$ be a dataset (used throughout the paper), the support of φ w.r.t D is defined as usual by: $sup(\varphi, D) = \frac{|\{d \in D | \varphi \preceq d\}|}{|D|}$. For example, let $\mathcal{L}$ be the set of itemsets, and $\preceq$ is $\subseteq$. We have $|D| = 5$ and therefore $sup(\text{AB}, D) = 0.60$ and $sup(\text{BC}, D) = 1$.

3.2 Global Model

We now propose a definition of a pattern-based global model. Informally, a pattern-based global model consists of a set of patterns along with an order relation between them. For example, the order relation may specify the order of patterns in a classifier construction task (which can also be used to predict the class of unlabeled data objects).

Definition 1. (Model) *Given the set $\mathcal{L}$ of patterns, a global model is a tuple $\langle P, <_P \rangle$ where $P \subseteq \mathcal{L}$ and $<_P$ is a strict partial order on P. Let $M = \langle P, <_P \rangle$ be a model, $|M|$ denotes $|P|$ and $\varphi \in M$ denotes $\varphi \in P$. Let $\mathcal{M}$ be the set of all global models.*

For a dataset D, we denote by $covered(D, M)$ the set of objects covered by a model M, i.e. $covered(D, M) = \{d \in D | \exists \varphi \in M, \varphi \preceq d\}$.

An essential aspect in data mining is the capability of combining models using simple operators while retaining the so-called *closure property*. The following operators provide such a property, i.e. the combination of two models is a model.

Concatenation. Let $M_1 = \langle P_1, <_{P_1} \rangle$ and $M_2 = \langle P_2, <_{P_2} \rangle$ be two models such that $P_1 \cap P_2 = \emptyset$. $M_1 \centerdot M_2 = \langle P_1 \cup P_2, <_{\centerdot} \rangle$ where $<_{\centerdot} = <_{P_1} \cup <_{P_2} \cup (P_1 \times P_2)$ is indeed a strict partial order because $P_1 \cap P_2 = \emptyset$. Note that $\centerdot$ does not commute, since × does not commute.

Difference. The difference $M_1 \setminus M_2$ is the model $\langle P_1 \setminus P_2, <_{P_1} \setminus \{\langle \varphi, \varphi' \rangle \in <_{P_1} | \varphi \in P_2 \vee \varphi' \in P_2\} \rangle$. The definition of union or intersection is straightforward.

Example 1. Given the dataset D and $\mathcal{L}$, some example models are as follows: $M_1 = \langle \{\text{BCDE}\}, \emptyset \rangle$, $M_2 = \langle \{\text{ABC,BCDE}\}, \{(\text{BCDE,ABC})\} \rangle$, and $M_3 = \langle \{\text{ABC}\}, \emptyset \rangle$. We have $M_2 = M_1 \centerdot M_3$, $covered(D, M_1) = \{\text{ABCDE,BCDE,BCDE}\}$ and $covered(D, M_2) = D$.

3.3 Generic Operators

Now we introduce the two basic but generic operators used in the description of the modeling process. Note that we give here an abstract and generic definition of these operators, in the sense that they are given for any set $\mathcal{S}$, e.g. $\mathcal{L}$ or $\mathcal{M}$. The first one is the theory computation operator *Th* that extracts a subset of a given set $\mathcal{S}$ satisfying a given selection predicate [12]. Given a set $\mathcal{S}$, a dataset $D \in \mathcal{D}$ and a selection predicate q which is a boolean function on $\mathcal{S} \times \mathcal{D}$, the theory of elements of $\mathcal{S}$ is the set $Th(\mathcal{S}, D, q) = \{s \in \mathcal{S} | q(s, D) = true\}$. For example, $Th(\mathcal{L}, D, sup(\varphi, D) \geq 0.50) = \{\varphi \in \mathcal{L} \mid \varphi \subseteq \text{ABC or } \varphi \subseteq \text{BCDE}\}$.

The second operator, Top_k, extracts the k best elements [5] of a set $\mathcal{S}$ w.r.t. a given order on $\mathcal{S}$. This order may depend on a given dataset. To this end, we first define the notion of data-dependent order.

Data-dependent order ($\mathcal{D}$-order). A data-dependent order (or $\mathcal{D}$-order) on a given set $\mathcal{S}$ is a relation $\alpha \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{D}$ such that, for a given $D \in \mathcal{D}$ and every $s_1, s_2 \in \mathcal{S}$, the relation $\alpha[D] = \{\langle s_1, s_2 \rangle | \langle s_1, s_2, D \rangle \in \alpha\}$ is an order on $\mathcal{S}$. We note $\alpha(s_1, s_2, D) = true$ if $\langle s_1, s_2, D \rangle \in \alpha$.

Operator (Top_k). Given a set $\mathcal{S}$, a dataset D and a $\mathcal{D}$-order α on $\mathcal{S}$, the top k elements of $\mathcal{S}$ w.r.t. α is the set $Top_k(\mathcal{S}, D, \alpha) = \{s_1 \in \mathcal{S} | (|\{s_2 \in \mathcal{S} / \alpha(s_2, s_1, D) = true\}| < k)\}$.

Example 2. Let $O_{largest}$ be the $\mathcal{D}$-order defined for every model $M_1 \in \mathcal{M}$, $M_2 \in \mathcal{M}$ and $D \in \mathcal{D}$ by $O_{largest}(M_1, M_2, D) = true$ iff $covered(D, M_2) \subset covered(D, M_1)$, i.e. the more the number of objects covered the more interesting the model. For instance, $O_{largest}(M_2, M_1, D)$ holds for the models in Example 1 and $Top_1(\{M_1, M_2, M_3\}, D, O_{largest}) = \{M_2\}$.

The next section introduces our global model construction algorithm.

4 IGMA: An Incremental Global Model Construction Algorithm

In this section we present a generic algorithm, named IGMA (Incremental Global Model construction Algorithm), that incrementally constructs a global model from the underlying dataset.

4.1 Principles and Parameters

Intuitively, IGMA iteratively constructs a model based on a given dataset. The basic principle is that at each iteration of IGMA, a set of k best interesting patterns w.r.t. a local selection predicate is extracted. However, different methods apply their own method-specific criteria to answer the following questions:

- **Which patterns are selected**? Each method incorporates its own selection predicate in order to evaluate whether a pattern is interesting. Therefore, a selection predicate has to be generated according to the current model in order to only retain the relevant patterns. This is exactly what is described by the *predicate generator* function parameter *PredGen* in IGMA.
- **Which patterns are the best ones**? All the potentially interesting patterns are compared thanks to an order linked to the method. Of course, this order depends on the dataset and the patterns included in the current model. For providing such a linkage, IGMA incorporates the *order generator* function parameter *OrdGen* to establish a $\mathcal{D}$-order on $\mathcal{L}$ w.r.t the set of uncovered objects.

Though adding the set of best k patterns to the current model potentially augments its descriptive accuracy, however, the user may possibly establish some interestingness criterion on models and finally select the best one. The $\mathcal{D}$-order described by parameter O in IGMA incorporates such an interestingness on the models constructed during the whole process. The next section presents and discusses the algorithm.

4.2 The IGMA Algorithm

In this section we introduce the IGMA algorithm and briefly describe its incremental model construction process.

```
Input: A pattern language $\mathcal{L}$, a dataset $D$, an integer $k$, a predicate generator function $PredGen$,
    an order generator function $OrdGen$, and a $\mathcal{D}$-order $O$ on $\mathcal{M}$
Output: A global model $M$
 1: $i = 0$ and $M_0 = \langle \emptyset, \emptyset \rangle$
 2: repeat
 3:     $i = i + 1$
 4:     $q_i = PredGen(M_{i-1})$ and $\alpha_i = OrdGen(M_{i-1})$
 5:     $L_i = Th(\mathcal{L} \setminus M_{i-1}, D, q_i)$
 6:     $T_i = Top_k(L_i, D, \alpha_i)$
 7:     $M_i = M_{i-1} \cdot \langle T_i, \alpha_i[D] \rangle$
 8: until $M_i = M_{i-1}$
 9: $M = Top_1(\{M_j \mid 1 \leq j < i\}, D, O)$
10: return $M$
```

The algorithm starts from an empty model (Line 1) and constructs its output model by iterating until the model remains unchanged, meaning that there are no more patterns to be added to the model (Line 8). At each step of the loop, a model is constructed with a set of k best patterns along with an ordering relation over them. The patterns are extracted with the *Th* operation (Line 5), taking into account the current model for generating the selection predicate (Line 4) and ignoring the patterns already obtained (Line 5). Then a new model is obtained by concatenating the current model with the newly extracted ordered set of k best patterns (Line 7), i.e. a locally optimal model. Finally, out of all constructed models the best one w.r.t O is returned as the result (Line 9 and 10).

Example 3. Given the dataset D, the language $\mathcal{L}$ of itemsets, $k = 1$, $O = O_{largest}$ (see Example 2), and the two function parameters, defined for every $M \in \mathcal{M}$ and every $\varphi, \varphi' \in \mathcal{L}$ and every $D \in \mathcal{D}$ as follows (let $D' = D \setminus covered(D, M)$):

- $PredGen(M) = q$ s.t $q(\varphi, D) = true$ if $sup(\varphi, D') \geq 0.40$.
- $OrdGen(M) = \alpha$ s.t $\alpha(\varphi, \varphi', D) = true$ if $sup(\varphi, D') \cdot |\varphi| > sup(\varphi', D') \cdot |\varphi'|$.

i	Remaining objects: $D \setminus covered(D, M_{i-1})$	$Th(\mathcal{L} \setminus M_{i-1}, D, q_i)$: L_i	$Top_1(L_i, D, \alpha_i)$: T_i	New model: M_i
1	D	$\{\varphi \mid \varphi \subseteq ABC\} \cup \{\varphi \mid \varphi \subseteq BCDE\}$	{BCDE}	$\langle\{BCDE\}, \emptyset\rangle$
2	{ABC,ABC}	$\{\varphi \mid \varphi \subseteq ABC\}$	{ABC}	$\langle\{ABC,BCDE\}, \{(BCDE,ABC)\}\rangle$
3	{}	{}	{}	$\langle\{ABC,BCDE\}, \{(BCDE,ABC)\}\rangle$

Starting with $M_0 = \langle\emptyset, \emptyset\rangle$, the table illustrates the construction of a model $M = IGMA(\mathcal{L}, D, 1, PredGen, OrdGen, O_{largest})$. At each iteration i the Th and the Top_1 operators exploit (respectively) the selection predicate q_i and the $\mathcal{D}$-order α_i which are generated in turn w.r.t the current model (M_{i-1}) and the set of objects not yet covered by it (D'). Finally, M_2 is returned as the resulting model which provides a global view, i.e. a summary, on the dataset D.

The next section shows how different global model constructions are expressed.

5 Generality of the Framework and IGMA

Now we specify the construction of various global models based on our framework by applying IGMA. More than 30 methods including separate-and-conquer and two-step methods encompassing classification, clustering and summarization have been represented in terms of IGMA using a wide range of parameters values. Due to space limitation we only specify single instances of clustering and summarization and we skip the rule-based classifiers we described in [7].[1] In what follows the desired model is constructed just by calling IGMA($\mathcal{L}, D, 1, PredGen, OrdGen, O_{largest}$) in which the functions $PredGen$ and $OrdGen$ are given for every $\varphi, \varphi' \in \mathcal{L}$, every $D \in \mathcal{D}$ and every $M \in \mathcal{M}$ as will be described for each method in the next two sections.

5.1 Clustering

Among many clustering methods (of both separate-and-conquer and two-step), e.g. [3,4,8], in this section we only represent ECCLAT [3]. The ECCLAT method

[1] The interested reader will find in [7] examples of different values for parameters of IGMA.

constructs clusters by extracting a subset of the frequent closed itemsets. A cluster is described by a closed itemset which is considered as the representative of the set of objects it covers. To build a clustering, ECCLAT follows an iterative approach to select clusters among the set of frequent closed patterns having the highest interestingness when evaluated on the dataset.
The functions *PredGen* and *OrdGen* are defined as follows:

- **Predicate generator:** The generated predicate, denoted by q_{ECC}, selects only the frequent closed patterns that cover at least a minimum number of unclustered objects. This is formulated as $PredGen(M) = q_{ECC}$ where $q_{ECC}(\varphi, D) = true$ iff $sup(\varphi, D).|D| \geq minfr \wedge closed(\varphi, D) \wedge |\{d \in D \setminus covered(D, M) | \varphi \preceq d\}| \geq m$, where $minfr$ is a frequency threshold and $closed(\varphi, D)$ is used to denote that φ is a closed pattern, and m is the minimal number of objects not yet covered by M and covered by φ.
- **Order generator:** The order on patterns is generated according to their interestingness w.r.t D. Formally, $OrdGen(M)= \alpha_{ECC}$ where $\alpha_{ECC}(\varphi, \varphi', D) = true$ iff $interestingness(\varphi, D) \geq interestingness(\varphi', D)$, where the *interestingness* is the average of an intra-cluster similarity (*homogeneity*) and an inter-cluster dissimilarity (*concentration*) measure.

5.2 Summarization

Among many different methods for summarization, e.g. [1,11,13], we describe the Chosen Few [1] due to its generality when compared with the others. The Chosen Few approach [1] provides a summary M which is an ordered subset of a given collection $\mathcal{L}$ of patterns and that characterizes the underlying dataset. This method applies a general algorithm which incrementally adds the best pattern φ, w.r.t a given order $\prec$, from $\mathcal{L}$ to the summary M. A measure Φ evaluates the interestingness of φ regarding M. Additionally, φ has to increase the size of the partition induced by the summary M. Supposing $M = \langle P, <_P \rangle$, the *partition set* of D over P, denoted by $D/\sim_P$, is the set of groups whose objects are equivalent w.r.t $\sim_P$. Two objects d_1 and d_2 are equivalent (under a pattern set P), denoted by $d_1 \sim_P d_2$, iff they are exactly covered by the same patterns in M.
For the summarization built by the Chosen Few approach the functions *PredGen* and *OrdGen* are defined as follows:

- **Predicate generator:** The generated predicate selects all the non-examined patterns satisfying the minimal interest w.r.t Φ and inducing a larger partition set. More formally, $PredGen(M) = q_{CHF}$ where $q_{CHF}(\varphi, D) = true$ iff $\Phi(D, P, \varphi) \geq t \wedge |D/\sim_{P \cup \{\varphi\}}| > |D/\sim_P|$, where t is a threshold.
- **Order generator:** This is a constant total order on patterns given as input.

6 Conclusion

This paper proposes a uniform and flexible framework for describing pattern-based global models. We introduce the IGMA algorithm that achieves various kind of global models covering classification, clustering and summarization techniques. From a technical point of view, this generic algorithm covers both existing separate-and-conquer and two-step methods. Finally, we illustrate the expressiveness of IGMA by specifying different global models.

Further work addresses an optimized implementation of IGMA by leveraging the properties on the predicate and the order generators. We would like also to test other pattern-based models by defining new IGMA's parameters. Thanks to the properties of the framework, developing a declarative querying environment linked to IGMA allows the user to interactively compare different approaches and construct the desired models.

References

1. Bringmann, B., Zimmermann, A.: The chosen few: On identifying valuable patterns. In: ICDM, pp. 63–72. IEEE Computer Society, Los Alamitos (2007)
2. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. The MIT Press, Cambridge (2001)
3. Durand, N., Crémilleux, B.: ECCLAT: A new approach of clusters discovery in categorical data. In: SGAI-ES 2002, pp. 177–190. Springer, Heidelberg (2002)
4. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, pp. 226–231. AAAI Press, Menlo Park (1996)
5. Fu, A.W.-c., Kwong, R.W.-w., Tang, J.: Mining N-most interesting itemsets. In: Ohsuga, S., Raś, Z.W. (eds.) ISMIS 2000. LNCS (LNAI), vol. 1932, pp. 59–67. Springer, Heidelberg (2000)
6. Fürnkranz, J.: Separate-and-conquer rule learning. AIR 13(1), 3–54 (1999)
7. Giacometti, A., Khanjari Miyaneh, E., Marcel, P., Soulet, A.: A generic framework for rule-based classification. In: Proceedings of LeGo 2008, an ECML/PKDD 2008 Workshop, pp. 37–54 (2008)
8. Han, E.-H., Karypis, G., Kumar, V., Mobasher, B.: Clustering based on association rule hypergraphs. In: DMKD, pp. 9–13. ACM, New York (1997)
9. Imielinski, T., Mannila, H.: A database perspective on knowledge discovery. Commun. ACM 39(11), 58–64 (1996)
10. Knobbe, A., Crémilleux, B., Fürnkranz, J., Scholz, M.: From local patterns to global models: The lego approach to data mining. In: Proceedings of LeGo 2008, an ECML/PKDD 2008 Workshop, pp. 1–16 (2008)
11. Knobbe, A.J., Ho, E.K.Y.: Pattern teams. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 577–584. Springer, Heidelberg (2006)
12. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Min. Knowl. Discov. 1(3), 241–258 (1997)
13. Mielikäinen, T., Mannila, H.: The pattern ordering problem. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 327–338. Springer, Heidelberg (2003)
14. Nijssen, S., Raedt, L.D.: IQL: A proposal for an inductive query language. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 189–207. Springer, Heidelberg (2007)
15. Raedt, L.D., Zimmermann, A.: Constraint-based pattern set mining. In: SDM, pp. 237–248. SIAM, Philadelphia (2007)