Scalable Knowledge Discovery in Complex Data with Pattern Structures

Sergei O. Kuznetsov

School of Applied Mathematics and Information Science,
National Research University Higher School of Economics,
Bol. Trekhsvyatitelskii 3, Moscow, Russia
skuznetsov@hse.ru

Abstract. Pattern structures propose a direct way to knowledge discovery in data with structure, such as logical formulas, graphs, strings, tuples of numerical intervals, etc., by defining closed descriptions and discovery tools build upon them: automatic construction of taxonomies, association rules and classifiers. A combination of lazy evaluation with projections of initial data, randomization and parallelization suggest efficient approach which is scalable to big data.

1 Introduction

In many real-world knowledge discovery problems researchers have to deal with complex descriptions different from binary datatables. In the last two decades the use of closed descriptions defined either in terms of Galois connections, semilattical similarity operation (i.e., operation which is idempotent, commutative, and associative) or in equivalent terms of counting inference proved to be very useful in various knowledge discovery applications, such as ontology and taxonomy engineering, mining association rules, machine learning, classification, and clustering. Several attempts were done in defining closed sets of graphs and closed graphs [24,32,26,30,28,1,18], strings [11,6], numerical intervals [23,22], logical formulas [7,10], etc. In [15] a general approach called pattern structures was proposed, which allows one to apply knowledge discovery tools to arbitrary partially ordered data descriptions. Using pattern structures, one can compute taxonomies, ontologies, implicational dependencies and their bases, association rules, and classifiers in the same way as it is done with binary data.

To meet the big data challenge the problem settings of knowledge discovery can be recast to allow for faster procedures. In this paper we show how the classification problems for pattern structures can be reformulated to achieve scalability even for complex descriptions.

The rest of the paper is organized as follows: In Section 2 we recall basic definitions in pattern structures, give examples of graph-based and interval-based pattern structures. In Section 3 we describe our approach to scalable classification with pattern structures, and make a conclusion in Section 4.

P. Maji et al. (Eds.): PReMI 2013, LNCS 8251, pp. 30–39, 2013.

2 Knowledge Discovery with Pattern Structures

2.1 Main Definitions and Results

Let G be a set (of objects), let $(D, \sqcap)$ be a meet-semi-lattice (of all possible object descriptions) and let $\delta : G \to D$ be a mapping. Then $(G, \underline{D}, \delta)$, where $\underline{D} = (D, \sqcap)$, is called a *pattern structure*, provided that the set $\delta(G) := \{\delta(g) | g \in G\}$ generates a complete subsemilattice $(D_\delta, \sqcap)$ of $(D, \sqcap)$, i.e., every subset X of $\delta(G)$ has an infimum $\sqcap X$ in $(D, \sqcap)$. Elements of D are called *patterns* and are naturally ordered by subsumption relation $\sqsubseteq$: given $c, d \in D$ one has $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$. Operation $\sqcap$ is also called a *similarity operation*. A pattern structure $(G, \underline{D}, \delta)$ gives rise to the following derivation operators $(\cdot)^\diamond$:

$$A^\diamond = \sqcap_{g \in A} \delta(g) \qquad \text{for } A \subseteq G,$$
$$d^\diamond = \{g \in G \mid d \sqsubseteq \delta(g)\} \qquad \text{for } d \in (D, \sqcap)$$

These operators form a Galois connection between the powerset of G and $(D, \sqsubseteq)$. The pairs (A, d) satisfying $A \subseteq G$, $d \in D$, $A^\diamond = d$, and $A = d^\diamond$ are called the *pattern concepts* of $(G, \underline{D}, \delta)$, with *pattern extent* A and *pattern intent* d. Pattern concepts are ordered wrt. set inclusion on extents. The ordered set of pattern concepts makes a lattice, called *pattern concept lattice*. For $a, b \in D$ the *pattern implication* $a \to b$ holds if $a^\diamond \subseteq b^\diamond$, and the *pattern association rule* $a \to_{c,s} b$ with *confidence* c and *support* s holds if $s \leq \frac{|a^\diamond \cap b^\diamond|}{|G|}$ and $c \leq \frac{|a^\diamond \cap b^\diamond|}{|a^\diamond|}$. Like in case of association rules [33,34], pattern association rules may be inferred from a concise representation that corresponds to the set of edges of the diagram of the pattern concept lattice. Operator $(\cdot)^{\diamond\diamond}$ is an algebraical closure operator on patterns, since it is idempotent, extensive, and monotone.

The concept-based learning model for standard object-attribute representation (i.e., formal contexts) [12,25,27] is naturally extended to pattern structures. Suppose we have a set of positive examples G_+ and a set of negative examples G_- w.r.t. a *target attribute*, $G_+ \cap G_- = \emptyset$, objects from $G_\tau = G \setminus (G_+ \cup G_-)$ are called undetermined examples.

A pattern $c \in D$ is a *positive premise (classifier)* iff

$$c^\diamond \cap E_- = \emptyset \text{ and } \exists A \subseteq E_+ : c \sqsubseteq A^\diamond$$

A pattern $h \in D$ is a *positive hypothesis* iff

$$h^\diamond \cap E_- = \emptyset \text{ and } \exists A \subseteq E_+ : h = A^\diamond$$

A positive hypothesis is the *least general generalization* of descriptions ("similarity") of positive examples, which is not contained in (does not cover) any negative example. *Negative* premises (classifiers) and hypotheses are defined similarly. Various classification schemes using premises are possible, as an example consider the following simplest scheme from [12,26,15]: If description $\delta(g)$ of an undetermined example g contains a positive premise (hypothesis) c, i.e., $c \sqsubseteq \delta(g)$,

and no negative premise, then g is *classified positively*. Negative classifications are defined similarly. If $\delta(g)$ contains premises (hypotheses) of both signs, or if $\delta(g)$ contains no premise (hypothesis) at all, then the classification is contradictory or undetermined, respectively, and some probabilistic relaxation of the above definitions of premises and hypotheses should be applied.

For some pattern structures (e.g., for the pattern structures on sets of graphs with labeled vertices) even computing subsumption of patterns may be NP-hard. Hence, for practical situations one needs approximation tools, which would replace the patterns with simpler ones, even if that results in some loss of information. To this end we use a contractive monotone and idempotent mapping $\psi : D \to D$ that replaces each pattern $d \in D$ by $\psi(d)$ such that the pattern structure $(G,\ \underline{D},\ \delta)$ is replaced by $(G,\ \underline{D},\ \psi \circ \delta)$. Under some natural algebraic requirements that hold for all natural projections in particular pattern structures we studied in applications, see [30], the meet operation $\sqcap$ is preserved: $\psi(X \sqcap Y) = \psi(X) \sqcap \psi(Y)$. This property of a projection allows one to relate premises (hypotheses) in the original representation with those approximated by a projection.The representation context of the projected case is obtained from the unprojected one by removing some attributes. If $\psi(a) \to \psi(b)$ and $\psi(b) = b$ then $a \to b$ for arbitrary $a,\ b \in D$. In particular, if $\psi(a)$ is a positive (negative) premise in projected representation, then a is positive (negative) premise in the original representation.

2.2 Pattern Structures in Applications

One may argue that a semi-lattice on descriptions is a too demanding requirement, but we can show easily that this is not the case. Any natural kind of descriptions available for data analysis has an explicitly or implicitly given partial order relation in the form of "is a" or "part of" relation. Having a partially ordered set $(P,\ \leq)$ of descriptions one can define a *similarity operation* $\sqcap$ on sets of descriptions as follows: For two descriptions X and Y from P

$$\{X\} \sqcap \{Y\}\ := \{Z | Z \leq X, Y,\ \forall Z_* \leq X, Y\ Z_* \not\geq Z\},$$

i.e., $\{X\} \sqcap \{Y\}$ is the set of all maximal common subdescriptions of descriptions X and Y. Similarity of non-singleton sets of descriptions $\{X_1,\ \ldots,\ X_k\}$ and $\{Y_1,\ \ldots,\ Y_m\}$ is defined as

$$\{X_1,\ \ldots,\ X_k\} \sqcap \{Y_1,\ \ldots,\ Y_m\} := MAX_{\leq}(\bigcup_{i,j}(\{X_i\} \sqcap \{Y_j\})),$$

where $MAX_{\leq}(\mathcal{X})$ returns maximal elements of $\mathcal{X}$ w.r.t. $\leq$. The similarity operation $\sqcap$ on sets of descriptions is commutative: $\mathcal{X} \sqcap \mathcal{Y} = \mathcal{Y} \sqcap \mathcal{X}$ and associative: $(\mathcal{X} \sqcap \mathcal{Y}) \sqcap \mathcal{Z} = \mathcal{X} \sqcap (\mathcal{Y} \sqcap \mathcal{Z})$. A set $\mathcal{X}$ of descriptions from P for which $\mathcal{X} \sqcap \mathcal{X} = \mathcal{X}$ defines a pattern. Then the triple $(G, (D,\ \sqcap), \delta)$, where D is the set of all patterns, is a pattern structure.

One can think of $\mathcal{X} \sqcap \mathcal{Y}$ in the following way, which also gives a straightforward approach to computing $\sqcap$: One takes the set of all subdesriptions of all

descriptions of $\mathcal{X}$ and takes set-theoretic intersection (i.e., $\cap$) of this set with the set of all subdescriptions of all descriptions of $\mathcal{Y}$. Finally, from the resulting set of subdescriptions one chooses the maximal ones w.r.t. the partial order $\leq$ on descriptions.

Pattern Structures on Sets of Graphs. In [24,26] we proposed a semi-lattice on sets of graphs with labeled vertices and edges. This semilattice is based on a partial order given by subgraph isomorphism or its generalizations. For example, in [26,15] the following natural order relation on graphs with labeled vertices and edges, called *domination relation*, was proposed. Consider connected graphs[1] with vertex and edge labels from set $\mathcal{L}$ partially ordered by $\preceq$. Denote the set of graphs with labeled vertices and edges by P. Each graph Γ from P is a quadruple of the form $((V,\ l),\ (E,\ b))$, where V is a set of vertices, E is a set of edges, $l : V \to \mathcal{L}$ is a function assigning labels to vertices, and $b : E \to \mathcal{L}$ is a function assigning labels to edges. In $(P,\ \leq)$ we do not distinguish isomorphic graphs.

For two graphs $\Gamma_1 := ((V_1,\ l_1),\ (E_1,\ b_1))$ and $\Gamma_2 := ((V_2,\ l_2),\ (E_2,\ b_2))$ from P we say that Γ_1 *dominates* Γ_2 or $\Gamma_2 \leq \Gamma_1$ (or Γ_2 is a *subgraph* of Γ_1) if there exists an injection $\varphi : V_2 \to V_1$ such that it *respects edges*: $(v, w) \in E_2 \Rightarrow (\varphi(v), \varphi(w)) \in E_1$ and *fits under labels*: $l_2(v) \preceq l_1(\varphi(v))$, if $(v, w) \in E_2$, then $b_2(v, w) \preceq b_1(\varphi(v), \varphi(w))$.

Obviously, $(P,\ \leq)$ is a partially ordered set. Having a partial order on graphs, one can use the definitions above to define similarity operation $\sqcap$ and closure operator $(\cdot)^{\diamond\diamond}$. A set of graphs X is called *closed* if $X^{\diamond\diamond} = X$. The closed set of graphs consists of *closed graphs* as defined in [36]. A learning model based on *graph pattern structures* along the lines of the previous subsection was used in series of applications in chemo- and bioinformatics [16,30], in text analysis [14] and conflict analysis [13]. Numerous types of projections were used in these applications, like e.g. k-vertex subgraphs, k-length paths, cylcic subgraphs in chemoinformatics, and noun, verb, and other types of phrases in natural language processing.

Pattern Structures on Intervals. In practice, a typical object-attribute data table is not binary, but has many-valued attributes. Instead of binarizing (scaling) data, one can directly work with many-valued attributes by applying *interval pattern structures*. For two intervals $[a_1,\ b_1]$ and $[a_2,\ b_2]$, with $a_1,\ b_1,\ a_2,\ b_2 \in \mathbb{R}$, we define their meet as

$$[a_1,\ b_1] \sqcap [a_2,\ b_2] = [\min(a_1,\ a_2),\ \max(b_1,\ b_2)].$$

This operator is obviously idempotent, commutative and associative, thus defining a pattern structure on tuples (vectors) of intervals of attribute values. The lattice of interval pattern structures is isomorphic to the concept lattice of the context that arises from the *interordinal scaling* of the initial many-valued numerical context, where for each table value a two binary attributes $\geq a$ and $\leq a$

[1] Omitting the condition of connectedness, one obtains a similar, but computationally much harder model.

are introduced. However, interval tuples give better understanding of results and computation with them is faster than that with the interordinal scaling, as shown in experiments with gene expression data [22].

3 Scalable Classification with Pattern Structures

The goal of computing implications, association rules, premises, hypotheses, and there concise representations is to "understand" data by creating "knowledge" in the form of implicational dependencies, and to use these dependencies for making predictions for new data. Intractability results on the sizes of concepts [27], implication bases [29,9,3], (minimal) hypotheses [27,2] say that the amount of "knowledge" generated from data without due filtering can be larger than data themselves. If one uses knowledge for making predictions, i.e., defining missing information, say classes of objects described by new data, one does not need having all "knowledge" given explicitly, one just needs having predictions equivalent to those made when all knowledge is there. To attain this goal, one can note that classification by means of premises can be done in a "lazy" way without explicit computation of all premises. Classification can be described in the extended pattern structure

$$(G, (D*, \sqcap*), \delta*) = (G, ((D, \sqcap) \times (\{0, 1\}, \wedge)), \delta \cup val),$$

where $\wedge$ is logical conjunction and the mapping $val : G \to \{0, 1\}$ says whether an object has the target attribute or not. Below we show how it works for premises in the classification scheme defined above.

Many algorithms for computing concept lattices, like NextClosure [17] and CbO [25], may be adapted to computing pattern lattices in bottom-up way. The worst-case time complexity of computing all pattern concepts of a pattern structure $(G,\ \underline{D},\ \delta)$ in the bottom-up way is $O((p(\sqcap) + p(\sqsubseteq)|G|) \cdot |G| \cdot |L|)$, where $p(\sqcap)$ is time needed to perform $\sqcap$ operation, $p(\sqsubseteq)$ is time needed to test $\sqsubseteq$ relation, and L is the set of all pattern concepts. In case of graphs, even $p(\sqsubseteq)$ may be exponential w.r.t. the number of graph vertices, that is why approximations (like those given by projections) are often needed. For a fixed projection size $p(\sqsubseteq)$ and $p(\sqcap)$ can be considered constant. To compute graph patterns in the top-bottom way, e.g., for computing *frequent patterns*, one can update CbO algorithm by getting access to the "fine" structure of descriptions, like it was done for graphs in [28]. The worst-case time complexity of computing the set of interval pattern structures is $O(|G|^2 \cdot |M| \cdot |L|)$, where M is the set of attributes, which in practice can be much lower than the worst-case complexity of computing the set L of all concepts of the interordinally scaled numerical context, which is $O(|G|^2 \cdot |W| \cdot |L|)$, where W is the set of all attribute values.

3.1 Lazy Classification

Suppose we have a training set (negative and positive examples wrt. some target attribute) and unclassified examples, all object descriptions given by a pattern

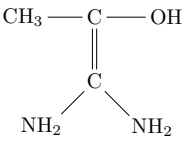
structure. Then the target attribute of the description of an object g_n to be classified with respect to the premises can be computed as the closure (w.r.t. $(G, (D*, \sqcap*), \delta*)$) of the intersection of the description of g_n with description of every object $g \in G$. If for some object g the closure contains the target attribute, g_n is classified positively by premises of $(G, (D*, \sqcap*), \delta*)$, otherwise it is classified negatively. Computationally, this can be described as the following simple two-stage procedure:

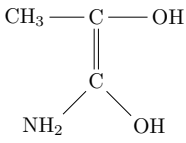
1. For every $g \in G$ compute $(\delta(g_n) \sqcap \delta(g))^\diamond$, i.e. select all objects from G whose descriptions contain $\delta(g_n) \sqcap \delta(g)$. This takes $O(|G| \cdot (p(\sqcap) + |G| \cdot p(\sqsubseteq)))$ time.

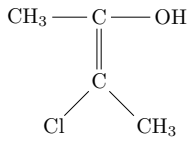
2. If for some $g \in G$ all objects from $(\delta(g_n) \sqcap \delta(g))^\diamond$ have the target attribute, classify g_n positively, otherwise negatively. This takes $O(|G|^2)$ time for looking for the target attribute in object descriptions in at most $|G|$ families of object subsets, each subset consisting of at most $|G|$ objects.

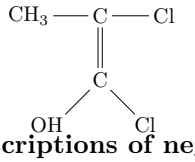
Example. Consider a training sample with four positive examples having descriptions $\{G_1\}$, $\{G_2\}$,$\{G_3\}$,$\{G_4\}$, three negative examples having descriptions $\{G_5\}$,$\{G_6\}$,$\{G_7\}$, and unclassified examples having descriptions $\{G_8\}$,$\{G_9\}$.

Descriptions of positive examples:

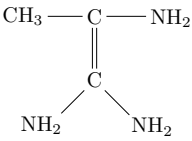
G_1 : 

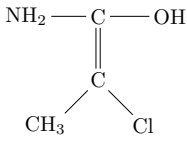
G_2 : 

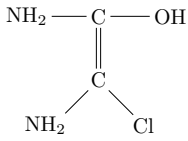
G_3 : 

G_4 : 

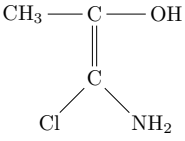
Descriptions of negative examples:

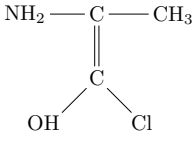
G_5 : 

G_6 : 

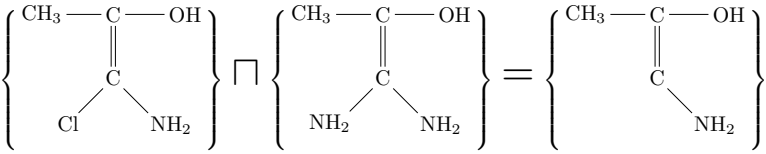
G_7 : 

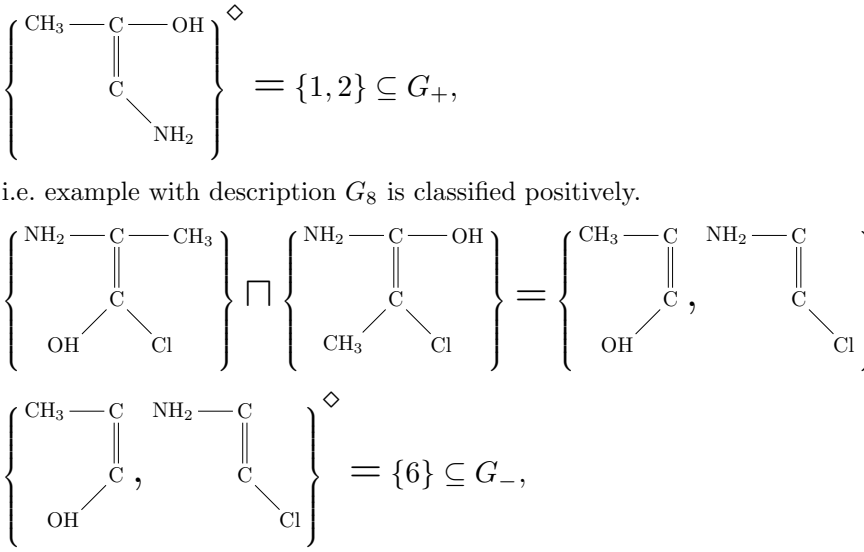
Descriptions of unclassified examples:

G_8 : 

G_9 : 

Classification results:





i.e. example with description G_8 is classified positively.

i.e. example with description G_9 is classified negatively.

Proposition 1. *Classification of an object with premises can be done in* $O(|G| \cdot (|G| \cdot p(\sqsubseteq) + p(\sqcap)))\cdot$ *time and in* $O(|G|^2)$ *time in projections of fixed size.*

The lazy approach to classification described above is close to some standard approaches like Nearest Neighbors [37] (finding nearest classes in metric spaces), Case-Based Reasoning [20] (classifying similar to classification of similar cases), abduction in Horn theories [21] (lazy evaluation from models instead of generating implications on Boolean variables), however differs from them in being based on partially ordered structures, not metric or Boolean.

We have reduced classification to computing $(\delta(g) \sqcap \delta(g_n))^\diamond$ and testing the target attribute in all objects of this set. This computation is easily parallelizable: one partitions the dataset G in $G = G_1 \cup \ldots \cup G_k$, where k is the number of processors, computes in each G_i the set of objects $(\delta(g) \sqcap \delta(g_n))^{\diamond_i}$, tests the target attribute for all objects in the union of these sets over i. Thus, we have the following

Proposition 2. *Classification of* m *objects with premises using* k *processors can be done in* $O(|G| \cdot (|G| \cdot p(\sqsubseteq) + p(\sqcap)) \cdot m/k)$ *time and in* $O(|G|^2 \cdot n/k)$ *time in projections of fixed size.*

The computation can also be easily randomized by taking random objects from each of G_i for computing the closures. To this end, one needs to use a probabilistic relaxation of the definitions of premises and classifications.

4 Conclusion

Pattern structures propose a useful means for discovering dependencies in data given by complex ordered descriptions, such as numerical data, data given by

graphs as in chemoinformatics and natural language processing, or strings in the analysis of processes. Using projections, parallel computations and randomization, one can propose scalable approach to knowledge discovery with pattern structures by reducing algorithmic complexity from double exponential to low degree polynomial.

Acknowledgments. This work was done within the project "Mathematical models, algorithms, and software tools for intelligent analysis of structural and textual data" supported by the Basic Research Program of the National Research University Higher School of Economics (Moscow).

References

1. Arimura, H., Uno, T.: Polynomial-Delay and Polynomial-Space Algorithms for Mining Closed Sequences, Graphs, and Pictures in Accessible Set Systems. In: Proc. SDM, pp. 1087–1098 (2009)
2. Babin, M.A., Kuznetsov, S.O.: Enumeration Minimal Hypotheses and Dualizing Monotone Boolean Functions on Lattices. In: Jäschke, R. (ed.) ICFCA 2011. LNCS (LNAI), vol. 6628, pp. 42–48. Springer, Heidelberg (2011)
3. Babin, M.A., Kuznetsov, S.O.: Computing Premises of a Minimal Cover of Functional Depedencies is Intractable. Discr. Appl. Math. 161(6), 742–749 (2013)
4. Baixeries, J., Kaytoue, M., Napoli, A.: Computing Functional Dependencies with Pattern Structures. In: Proc. 9th International Conference on Concept Lattices and their Applications (CLA 2012), Malaga (2012)
5. Birkhoff, B.: Lattice Theory. ACM (1991)
6. Buzmakov, A.V., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, A.: On Projections of Sequential Pattern Structures with an Application on Care Trajectories. In: Proc. 10th International Conference on Concept Lattices and their Applications (CLA 2013), La Rochelle (2013)
7. Chaudron, L., Maille, N.: Generalized Formal Concept Analysis. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS (LNAI), vol. 1867, pp. 357–370. Springer, Heidelberg (2000)
8. Coulet, A., Domenach, F., Kaytoue, M., Napoli, A.: Using pattern structures for analyzing ontology-based annotations of biomedical data. In: Cellier, P., Distel, F., Ganter, B. (eds.) ICFCA 2013. LNCS (LNAI), vol. 7880, pp. 76–91. Springer, Heidelberg (2013)
9. Distel, F., Sertkaya, B.: On the Complexity of Enumerating Pseudo-intents. Discrete Applied Mathematics 159(6), 450–466 (2011)
10. Férré, S., Ridoux, O.: A Logical Generalization of Formal Concept Analysis. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS (LNAI), vol. 1867, pp. 371–384. Springer, Heidelberg (2000)
11. Férré, S., King, R.D.: Finding Motifs in Protein Secondary Structure for Use in Function Prediction. Journal of Computational Biology 13(3), 719–731 (2006)
12. Finn, V.K.: Plausible Reasoning in Systems of JSM Type. Itogi Nauki i Tekhniki, Seriya Informatika 15, 54–101 (1991) (in Russian)
13. Galitsky, B.A., Kuznetsov, S.O., Samokhin, M.V.: Analyzing Conflicts with Concept-Based Learning. In: Dau, F., Mugnier, M.-L., Stumme, G. (eds.) ICCS 2005. LNCS (LNAI), vol. 3596, pp. 307–322. Springer, Heidelberg (2005)

14. Galitsky, B.A., Kuznetsov, S.O., Usikov, D.: Parse Thicket Representation for Multi-sentence Search. In: Pfeiffer, H.D., Ignatov, D.I., Poelmans, J., Gadiraju, N. (eds.) ICCS 2013. LNCS, vol. 7735, pp. 153–172. Springer, Heidelberg (2013)
15. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In: Delugach, H.S., Stumme, G. (eds.) ICCS 2001. LNCS (LNAI), vol. 2120, pp. 129–142. Springer, Heidelberg (2001)
16. Ganter, B., Grigoriev, P.A., Kuznetsov, S.O., Samokhin, M.V.: Concept-based Data Mining with Scaled Labeled Graphs. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) ICCS 2004. LNCS (LNAI), vol. 3127, pp. 94–108. Springer, Heidelberg (2004)
17. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999)
18. Garriga, G., Khardon, R., De Raedt, L.: Mining Closed Patterns in Relational, Graph and Network Data. Annals of Mathematics and Artificial Intelligence (2012)
19. Guigues, J.-L., Duquenne, V.: Familles minimales d'implications informatives resultant d'un tableau de donnees binaires. Math. Sci. Humaines 95, 5–8 (1986)
20. Hullermeier, E.: Case-Based Approximate Reasoning. Springer (2007)
21. Kautz, H.A., Kearns, M.J., Selman, B.: Reasoning with characteristic models. In: Proc. AAAI 1993, pp. 1–14 (1993)
22. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in formal concept analysis. Inf. Sci. 181(10), 1989–2001 (2011)
23. Kuznetsov, S.O.: Stability as an Estimate of the Degree of Substantiation of Hypotheses on the Basis of Operational Similarity. Nauchno-Tekhnicheskaya Informatsiya, Ser. 2 24(12), 21–29 (1990)
24. Kuznetsov, S.O.: JSM-method as a machine learning method. Itogi Nauki i Tekhniki, Ser. Informatika 15, 17–50 (1991) (in Russian)
25. Kuznetsov, S.O.: Mathematical aspects of concept analysis. J. Math. Sci. 80(2), 1654–1698 (1996)
26. Kuznetsov, S.O.: Learning of Simple Conceptual Graphs from Positive and Negative Examples. In: Żytkow, J.M., Rauch, J. (eds.) PKDD 1999. LNCS (LNAI), vol. 1704, pp. 384–391. Springer, Heidelberg (1999)
27. Kuznetsov, S.O.: Complexity of Learning in Concept Lattices from Positive and Negative Examples. Discr. Appl. Math. 142, 111–125 (2004)
28. Kuznetsov, S.O.: Computing Graph-Based Lattices from Smallest Projections. In: Wolff, K.E., Palchunov, D.E., Zagoruiko, N.G., Andelfinger, U. (eds.) KONT 2007 and KPP 2007. LNCS (LNAI), vol. 6581, pp. 35–47. Springer, Heidelberg (2011)
29. Kuznetsov, S.O., Obiedkov, S.A.: Some Decision and Counting Problems of the Duquenne-Guigues Basis of Implications. Discrete Applied Mathematics 156(11), 1994–2003 (2008)
30. Kuznetsov, S.O., Samokhin, M.V.: Learning Closed Sets of Labeled Graphs for Chemical Applications. In: Kramer, S., Pfahringer, B. (eds.) ILP 2005. LNCS (LNAI), vol. 3625, pp. 190–208. Springer, Heidelberg (2005)
31. Kuznetsov, S.O., Revenko, A.: Finding Errors in Data Tables: An FCA-based Approach. Annals of Mathematics and Artificial Intelligence (2013)
32. Liquiere, M., Sallantin, J.: Structural Machine Learning with Galois Lattice and Graphs. In: Proc. ICML 1998 (1998)
33. Luxenburger, M.: Implications partielle dans un contexte. Math. Sci. Hum (1991)

34. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient Minining of Association Rules Based on Using Closed Itemset Lattices. J. Inf. Systems 24, 25–46 (1999)
35. Ryssel, U., Distel, F., Borchmann, D.: Fast computation of proper premises. In: Proc. CLA 2011 (2011)
36. Yan, X., Han, J.: CloseGraph: Mining closed frequent graph patterns. In: Proc. KDD 2003, pp. 286–295. ACM Press, New York (2003)
37. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search - The Metric Space Approach. Springer (2006)