COMPLETENESS OF INFERENCE AND EFFICIENT COMPUTATION OF ATTRIBUTE-DEPENDENCY FORMULAS FOR TABULAR DATA

R. BELOHLAVEK[1,2] and V. VYCHODIL[2]

[1]*Dept. Systems Science and Industrial Engineering, Binghamton University—SUNY; Binghamton, NY 13902, U. S. A.*
Email: rbelohla@binghamton.edu

[2]*Dept. Computer Science, Palacký University, Olomouc; Tomkova 40, CZ-779 00, Olomouc, Czech Republic*
Email: vilem.vychodil@upol.cz

The present paper contributes to dependencies among attributes in relational data. We present results on attribute-dependency formulas (AD-formulas), namely, on inference of AD-formulas, and their relationship to other data dependencies. In particular, we present Armstrong-like rules and a completeness theorem for reasoning with AD-formulas, and an efficient algorithm for extraction of a non-redundant basis of all AD-formulas from data.

Keywords: attribute dependency, non-redundant basis, Armstrong axioms, completeness, formal concept analysis

1. Introduction

Dependencies in relational data represent one of the main topics in management of data and data mining. In particular, formulas relating two groups of attributes have proven to be both user-friendly and computationally tractable. Our paper deals with dependencies in data tables describing relationships between objects and attributes. A data table like this can be represented by a triplet $\langle X, Y, I\rangle$ where X is a finite set of objects (table rows), Y is a finite set of attributes (columns), and $I \subseteq X \times Y$ is a binary relation between X and Y specifying whether an object x has an attribute y (in which case $\langle x, y\rangle \in I$) or not ($\langle x, y\rangle \notin I$). The dependencies we are interested in are represented by so-called AD-formulas. AD-formulas are expressions of the form $A \sqsubseteq B$ where A and B are sets of attributes. AD-formulas were introduced in [2] as a means of user-defined constraints in formal concept analysis (FCA). In FCA, a collection $\mathcal{B}(X, Y, I)$ of partially ordered clusters, so-called formal concepts, is extracted from data table $\langle X, Y, I\rangle$, see [4,6]. The reason for introducing AD-formulas is the following. It is very often the case that the user has some further information, additional to $\langle X, Y, I\rangle$. An AD-formula, e.g. "green color" $\sqsubseteq$ "cold-blooded" $\sqcup$ "warm-blooded", is an example of such an information expressing a relative importance of attributes (whether an animal is green is less important than whether it is cold-/warm-blooded). Then, for a given set T of AD-formulas, one can consider the collection $\mathcal{B}_T(X, Y, I)$ of concepts from $\mathcal{B}(X, Y, I)$ which respect T, see later. The main benefits are the following: $\mathcal{B}_T(X, Y, I)$ is smaller and thus better comprehensible for a user than $\mathcal{B}(X, Y, I)$; $\mathcal{B}_T(X, Y, I)$ contains only those concepts which respect the constraints represented by AD-formulas from T, i.e. only "the interesting" concepts from $\mathcal{B}(X, Y, I)$.

In [3], we presented results on semantic entailment of AD-formulas. In particular, we presented a theoretical insight leading to an algorithm for testing whether an AD-formula $A \sqsubseteq B$ semantically follows from a set T of AD-formulas, and an algorithm for computing a minimal non-redundant basis T' from a given set T of AD-formulas. In [3] and [1], several connections between AD-formulas and two other types of dependencies in tabular data have been observed. In particular, to so-called attribute implications [6] and dependencies used in Knowledge Spaces [5]. Note that AD-formulas are a particular case of clauses studied in [7].

In this paper, we present further results on entailment of AD-formulas and results on computing AD-formulas from tabular data. The results are based on an observation saying that models of an AD-formula $A \sqsubseteq B$ are just complements of models of the attribute implication $B \Rightarrow A$. First, we present a complete system of deduction rules for reasoning about AD-formulas which results from the well-known Armstrong axioms and the above-mentioned observation. Second, we present a way to compute a non-redundant basis of all AD-formulas which are valid in a given data table by reducing the problem to the corresponding problem regarding attribute implications. Due to space limitation, we omit proofs.

2. Preliminaries

We now present preliminaries from formal concept analysis (FCA), see [4,6]. Let $\langle X, Y, I\rangle$ be a data table as described in Section 1. A (*formal*) *concept* in $\langle X, Y, I\rangle$ is a pair $\langle A, B\rangle$ of a set $A \subseteq X$ of objects (so-called *extent*) and a set $B \subseteq Y$ of attributes (so-called *intent*) such that A is the set of all objects which have all attributes from B, and B is the set of all attributes shared by all objects from A. Formal concepts can be partially ordered by putting $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$ iff $A_1 \subseteq A_2$ (or, equivalently, $B_2 \subseteq B_1$). A set $\mathcal{B}(X, Y, I)$ of all formal concepts in $\langle X, Y, I\rangle$ equipped with $\leq$ is called a *concept lattice* of $\langle X, Y, I\rangle$. $\mathcal{B}(X, Y, I)$ represents a collection of hierarchically ordered clusters extracted from the input data. Many applications of FCA can be found in [4] and in the references therein.

An attribute implication [6] (over Y) is an expression $A \Rightarrow B$, where $A, B \subseteq Y$. *$A \Rightarrow B$ is true in $M \subseteq Y$,* written $M \models A \Rightarrow B$, if we have:

$$\text{if } A \subseteq M \text{ then } B \subseteq M. \tag{1}$$

A set $M \subseteq Y$ is called a *model of T* if, for each $A \Rightarrow B \in T$, $M \models A \Rightarrow B$. Let $\mathrm{Mod}(T)$ denote the set of all models of T. *$A \Rightarrow B$ semantically follows from T* (*T semantically entails $A \Rightarrow B$*) if, for each $M \in \mathrm{Mod}(T)$, $M \models A \Rightarrow B$. Basic results concerning attribute implications can be found in [4, 6]. Attribute implications are also used in database theory (as functional dependencies), see [10].

3. Complete Deduction Rules and Computation of Non-redundant Bases for AD-formulas

3.1. *AD-formulas*

An *attribute-dependency formula* [2] (shortly, an AD-formula) over Y is an expression

$$A \sqsubseteq B,$$

where $A, B \subseteq Y$. *$A \sqsubseteq B$ is true in $M \subseteq Y$,* written $M \models A \sqsubseteq B$, if we have:

$$\text{if } A \cap M \neq \emptyset \text{ then } B \cap M \neq \emptyset. \tag{2}$$

Analogously as in case of attribute implications, we introduce models and semantic entailment. A set $M \subseteq Y$ is called a *model of T* if, for each $A \sqsubseteq B \in T$, $M \models A \sqsubseteq B$. Let $\mathrm{Mod}(T)$ denote the set of all models of T. *$A \sqsubseteq B$ semantically follows from T* (*T semantically entails $A \sqsubseteq B$*) if, for each $M \in \mathrm{Mod}(T)$, $M \models A \sqsubseteq B$. For brevity, we write $y \sqsubseteq B$ ($A \sqsubseteq y$) instead of $\{y\} \sqsubseteq B$ ($A \sqsubseteq \{y\}$).

The following assertion shows a connection between validity truth of attribute implications and validity AD-formulas which is of crucial importance in the sequel, see [3] and also [2].

Lemma 3.1. *For $A, B, M \subseteq Y$, we have $M \models A \sqsubseteq B$ iff $\overline{M} \models B \Rightarrow A$, where $\overline{M} = Y - M$.* $\square$

Remark 3.1. (i) In [2], AD-formulas were introduced as expressions of the form $y \sqsubseteq y_1 \sqcup \cdots \sqcup y_n$, i.e., as expressions of the form $\{y\} \sqsubseteq B$ where $B \subseteq Y$ in the present notation. As shown in [3], considering formulas $A \sqsubseteq B$ instead of $\{y\} \sqsubseteq B$ represents an inessential extension.

(ii) Given a concept lattice $\mathcal{B}(X, Y, I)$ associated to data $\langle X, Y, I\rangle$, a formal concept $\langle C, D\rangle \in \mathcal{B}(X, Y, I)$ is compatible with an AD-formula $A \sqsubseteq B$ iff D is a model of $A \sqsubseteq B$, see [2]. For the purpose of illustration, let $A \sqsubseteq B$ be "green color" $\sqsubseteq$ "cold-blooded" $\sqcup$ "warm-blooded". In order for a formal concept $\langle C, D\rangle$ to be compatible with "green color" $\sqsubseteq$ "cold-blooded" $\sqcup$ "warm-blooded", its intent D needs to contain either "cold-blooded" or "warm-blooded" as an attribute if it contains "green color". This way, attributes describing color can be declared less important that attributes "cold-blooded" and "warm-blooded".

(iii) Note that (2) is just the condition of validity of dependencies considered in Knowledge Spaces [5]. In Knowledge Spaces, the condition of validity of $A \sqsubseteq B$ in M is interpreted as follows. M is interpreted as a set of questions an individual can answer. $A \sqsubseteq B$ being true in M means that if that individual fails in answering all questions from A then he/she fails in answering all questions from B.

(iv) Therefore, although having different aims, AD-formulas coincide with the dependencies studied in Knowledge Spaces from the technical point of view.

3.2. *Complete systems of deduction rules for AD-formulas*

Deduction rules for AD-formulas which we use are of the form

$$\frac{A_1 \sqsubseteq B_1, \ldots, A_n \sqsubseteq B_n}{A \sqsubseteq B}. \tag{3}$$

These rules will be used in proofs in the usual way. That is, having AD-formulas which match the "input part" of the rule, i.e. $A_1 \sqsubseteq B_1, \ldots, A_n \sqsubseteq B_n$, we can infer, in a single step, the AD-formula corresponding to the "output part" of the rule. In particular, we will use the following system of deduction rules:

$$(\overline{\text{Ref}})\ \frac{}{A \sqsubseteq A},\quad (\overline{\text{Wea}})\ \frac{A \sqsubseteq B}{A \sqsubseteq B \cup C},\quad (\overline{\text{Cut}})\ \frac{A \sqsubseteq B,\ C \sqsubseteq A \cup D}{C \sqsubseteq B \cup D},$$

for each $A, B, C, D \subseteq Y$. The notions of a proof and provability are defined the usual way. That is, A *proof* of an AD-formula $A \sqsubseteq B$ from a set T of AD-formulas is a sequence $\varphi_1, \ldots, \varphi_n$ of AD-formulas such that $\varphi_n = A \sqsubseteq B$ and each φ_i either is from T or can be inferred from some preceding formulas φ_j, $j < i$, using some of the deduction rules $(\overline{\text{Ref}})$–$(\overline{\text{Cut}})$. An AD-formula

$A \sqsubseteq B$ is *provable* from a set T of AD-formulas if there is a proof of $A \sqsubseteq B$ from T; in this case, we write $T \vdash A \sqsubseteq B$. The following assertion is a completeness theorem for reasoning with AD-formulas.

Theorem 3.1 (completeness). $(\overline{\mathrm{Ref}})$–$(\overline{\mathrm{Cut}})$ *is a sound and complete system of deduction rules, i.e., for any set T of AD-formulas and an AD-formula $A \sqsubseteq B$, we have $T \vdash A \sqsubseteq B$ iff $T \models A \sqsubseteq B$.* □

The proof of Theorem 3.1 is based on a couple of observations we are going to make. For an AD-formula $A \sqsubseteq B$ and a set T of AD-formulas, put

$$(A \sqsubseteq B)^{\mathrm{AI}} = B \Rightarrow A,\ T^{\mathrm{AI}} = \{B \Rightarrow A \,|\, A \sqsubseteq B \in T\}.$$

Moreover, for a deduction rule (R) over AD-formulas, such as (3), denote by $(\mathrm{R})^{\mathrm{AI}}$ the deduction rule

$$\frac{B_1 \Rightarrow A_1, \ldots, B_n \Rightarrow A_n}{B \Rightarrow A}.$$

That is, $(\cdots)^{\mathrm{AI}}$ transforms AD-formulas, sets of AD-formulas, and deduction rules over AD-formulas, the corresponding attribute implications, sets of attribute implications, and deduction rules over attribute implications. Lemma 3.1 immediately gives the following observation (see also [1]):

Lemma 3.2. *For a set T of AD-formulas,*

$$\mathrm{Mod}(T) = \{\overline{M} \,|\, M \in \mathrm{Mod}(T^{\mathrm{AI}})\}.$$ □

Then we have the following assertion.

Lemma 3.3. *For a set T of AD-formulas and an AD-formula $A \sqsubseteq B$ we have $T \models A \sqsubseteq B$ iff $T^{\mathrm{AI}} \models (A \sqsubseteq B)^{\mathrm{AI}}$.*

Note that a rule (R) such as (3) is called sound if $\{A_1 \sqsubseteq B_1, \ldots, A_n \sqsubseteq B_n\} \models A_n \sqsubseteq B_n$. Therefore, we have

Corollary 3.1. *A rule* (3) *over AD-formulas is sound iff the corresponding rule* (4) *over attribute implications is sound.* □

Note that due to Corollary 3.1, we can automatically retrieve further sound deduction rules over AD-formulas from the well-known rules for attribute implications (or, equivalently, functional dependencies). For instance, the following rules are sound (for $A, B, C \subseteq Y$):

$$(\overline{\mathrm{Ax}})\ \frac{}{A \sqsubseteq A \cup B}, \quad (\overline{\mathrm{Add}})\ \frac{B \sqsubseteq A, C \sqsubseteq A}{B \cup C \sqsubseteq A}, \quad (\overline{\mathrm{Pro}})\ \frac{A \cup B \sqsubseteq C}{A \sqsubseteq C}, \quad (\overline{\mathrm{Tra}})\ \frac{A \sqsubseteq B, B \sqsubseteq C}{A \sqsubseteq C}.$$

Remark 3.2. Due to the above remarks, Theorem 3.1 provides us with a complete set of deduction rules for the dependencies studied in knowledge spaces. This "fills a gap" because syntactical reasoning and completeness was not investigate in knowledge spaces [5].

The above results enable us to apply results on attribute implications (functional dependencies) regarding entailment of AD-formulas. Due to space limitations, we restrict ourselves to the problem of computing non-redundant bases.

3.3. *Computing minimal non-redundant basis of AD-formulas from data*

As mentioned above, AD-formulas were introduced as a means of constraining a concept lattice by an information regarding relative importance of attributes. This information is provided by a user who is supposed to specify it using AD-formulas. On the other hand, one might want to extract the information about relative importance of attributes from data automatically. For instance, one might ask an expert to look at a concept lattice $\mathcal{B}(X, Y, I)$ extracted from data with no additional information available and tell which formal concepts are "relevant", i.e., compatible with some background information regarding relative importance of attributes, which is, nevertheless, not explicitly formulated. Then, denoting by $\mathcal{B}_C(X, Y, I)$ the collection of "relevant" formal concepts, the question is whether there is a set T of AD-formulas such that $\mathcal{B}_C(X, Y, I)$ equals (or is close) to the set of all formal concepts of $\mathcal{B}(X, Y, I)$ which are compatible with T. A way to find whether such T exists is the following. Denote by $\mathrm{Int}_C(X, Y, I)$ the collection of all intents of formal concepts from $\mathcal{B}_C(X, Y, I)$, i.e.,

$$\mathrm{Int}_C(X, Y, I) = \{D \mid \langle C, D\rangle \in \mathcal{B}_C(X, Y, I)\}.$$

Then, compute a small but fully representative set T of all AD-formulas valid in $\mathrm{Int}_C(X, Y, I)$, such as a non-redundant basis. A *non-redundant basis* of AD-formulas of a collection $\mathcal{M} \subseteq 2^Y$ is a set T of AD-formulas such that

1. $\mathcal{M} \subseteq \mathrm{Mod}(T)$, i.e., each AD-formula from T is true in each M from $\mathcal{M}$,
2. every AD-formula which is true in each M from $\mathcal{M}$ semantically follows from T,
3. no $T' \subset T$ satisfies 1. and 2.

Then, one tests whether $\mathcal{B}_C(X, Y, I)$ are just the formal concepts from $\mathcal{B}(X, Y, I)$ which are compatible with T. If the result of such a test is positive, T is a non-redundant set of AD-formulas which constraints $\mathcal{B}(X, Y, I)$

to $\mathcal{B}_C(X, Y, I)$. If the result is negative, no set of AD-formulas exists which constraints $\mathcal{B}(X, Y, I)$ to $\mathcal{B}_C(X, Y, I)$.

Computation of non-redundant bases of AD-formulas from data can be performed based on the following theorem. A direct application of Lemma 3.2 and Lemma 3.3 yields

Theorem 3.2. *Let $\mathcal{M} \subseteq 2^Y$ be a collection of sets of attributes. Let T be a non-redundant basis of attribute implications which are true in $\overline{\mathcal{M}} = \{\overline{M} \mid M \in \mathcal{M}\}$. Then $\{A \sqsubseteq B \mid B \Rightarrow A \in T\}$ is a non-redundant basis of AD-formulas of $\mathcal{M}$.* $\square$

Since there exists an efficient algorithm for computing a non-redundant basis (which is, moreover, minimal, i.e. smaller than any other basis) of attribute implications from data, this solves our problem of computing a non-redundant basis of AD-formulas from data.

Acknowledgement Supported by grant No. 1ET101370417 of GA AV ČR, by grant No. 201/05/0079 of the Czech Science Foundation, and by institutional support, research plan MSM 6198959214.

References

1. E. Bartl, R. Belohlavek: Knowledge spaces, attribute dependencies, and graded knowledge states. FUZZ-IEEE 2007, London, UK (to appear).
2. R. Belohlavek, V. Sklenář: Formal concept analysis constrained by attribute-dependency formulas. In: Ganter B., Godin R. (Eds.): ICFCA 2005, *LNCS* **3403**, pp. 176–191, Springer-Verlag, Berlin/Heidelberg, 2005.
3. R. Belohlavek, V. Vychodil: Semantic entailment of attribute-dependency formulas and their non-redundant bases. Proc. JCIS 2006, pp. 747–750.
4. C. Carpineto, G. Romano: *Concept Data Analysis. Theory and Applications.* J. Wiley, 2004.
5. J. P. Diognon, J. C. Falmagne: *Knowledge Spaces.* Springer, Berlin, 1999.
6. B. Ganter, R. Wille: *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin, 1999.
7. B. Ganter, R. Wille: Contextual attribute logic. In: Tepfenhart W., Cyre W. (Eds.): *Proceedings of ICCS 2001*, Springer, 2001.
8. J. L. Guigues, V. Duquenne: Familles minimales d'implications informatives resultant d'un tableau de données binaires. *Math. Sci. Humaines* **95**(1986), 5–18.
9. J. W. Lloyd: *Foundations of Logic Programming* (2nd ed.). Springer-Verlag, New York, 1987.
10. D. Maier: *The Theory of Relational Databases.* Computer Science Press, Rockville, 1983.