Advanced Features of Attribute Annotated Data Sets[*]

Martin Řimnáč

rimnacm@cs.cas.cz
Institute of Computer Science
Pod Vodárenskou věží 2, 182 07 Praha 8

Abstract. The paper compares features of learning and querying process in the situation, when values in the input data set are annotated by attributes or this information is not available. The attribute annotation enables to consider global relationships, which are useful to express the data semantics in a explicit way. It will be shown data can be accessed with no semantic interpretation and then, after the evaluation process, the result can be interpreted.

1 Introduction

Machine learning methods are seen as an significant approach for artificial intelligence, a knowledge is builded according to training examples from a given task domain. The expected result of the learning process is a generalized view on the task, which can be used for prediction (or decision) for data from whole task domain.

The paper deals with advanced features coming together with a attribute annotated data set usage. It assumes that the knowledge base can be described by an incident matrix Φ and compares the situations, when values in the input data set are annotated by attributes or this information is not available.

The incident matrix can be accessed using

$$\boldsymbol{y} = \Phi \cdot \boldsymbol{x} \tag{1}$$

where $\boldsymbol{x}$ is a vector representing a query by fact activations and analogically a vector $\boldsymbol{y}$ a result of the query.

The facts can be defined as

- `values` $v \in \mathscr{D}$ from a discrete domain $\mathscr{D}$
- `elements` $e \in \mathscr{E} \subseteq \mathscr{A} \times \mathscr{D}$, where $\mathscr{A}$ is a set of attributes.

The incident matrix can be trained using a example by example design approach [1].

[*] The work was supported by the project 1ET100300419 of the Program Information Society (of the Thematic Program II of the National Research Program of the Czech Republic) "Intelligent Models, Algorithms, Methods and Tools for the Semantic Web Realization" and partly by the Institutional Research Plan AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Applications".

2 Not Attribute Annotated Data Sets

The incident matrix $\varPhi^{\mathscr{D}}$ can be trained by examples described by tuples $t \in \mathscr{T}$, each tuple consists of several values from the domain $\mathscr{D}$. In this way, the incident matrix can be interpreted as

$$\varPhi^{\mathscr{D}} = \{\phi_{ij} : \forall \phi_{ij} = 1 \; \exists t_k \in T : v_i, v_j \in t_k\} \tag{2}$$

Note, this matrix expresses only the fact there exists a tuple t_k connecting the values v_i and v_j together. From the definition, the $\varPhi^{\mathscr{D}}$ is symmetric, trivially $\phi_{ii} = 1 \; \forall i$ and the transitivity property is not satisfied.

The training process for this matrix can be designed as incremental one (a vector $\boldsymbol{x_L}$ represents a training example):

$$\boldsymbol{\varPhi}^{\mathscr{D}}_{k+1} = \boldsymbol{\varPhi}^{\mathscr{D}}_{k} + \boldsymbol{x_L x_L^T} \tag{3}$$

This matrix can be designed as binary $\phi_{ij} \in \{0, 1\}$ or alternatively as $\phi_{ij} \in \mathbb{N}$, in such a case, to bound the query response (1), the relative values in the matrix or a maximum formula for the $i-$th fact activation has to be used instead:

$$\begin{aligned} \forall y_i : \; & 0 \leq y_i \leq 1 \\ y_i = \; & \sum_{\forall j} \hat{\phi}_{ij} x_j = \sum_{\forall j} \frac{\phi_{ij}}{\sum_{\forall j'} \phi_{ij'}} x_j \qquad (4) \\ y_i = \; & \max_{\forall j'} \{\phi_{ij'} x_{j'}\} \qquad (5) \end{aligned}$$

In this point of view, the values in the matrix can be seen as an evaluation of any special uncertainty fuzzy measure. The result can be used in similar way as one given by information retrieval tools, it expresses only values being connected with values in the query (and enable to order these values), but does not carry any information about.

3 Attribute Annotated Data Set

The incident matrix for the attribute annotated case can be defined analogically as

$$\varPhi^{\mathscr{A} \times \mathscr{D}} = \{\phi_{ij} : \begin{matrix} \forall \phi_{ij} = 1 \; \exists t_k \in T : v_i, v_j \in t_k \\ \forall \phi_{ij}^{\neg} = 1 : \phi_{ij} = 0 \end{matrix} \} \tag{6}$$

While the previous definition for set without attribute annotation does not contain any restriction, the attribute annotated set enables to distinguish between the instances (as in the previous case) and global relationships between attributes. When any pair of tuples does not satisfy some relationship, this relationship can not be considered in the future (it can be seen as incorrect). Only instances of valid relationships can be stored in the incident matrix.

There are many ways how these global relationships can be expressed. One can be an object hierarchy or, further detailed described, a extensional functional dependency system [2–5]. In such a case, the restriction is given by

$$\forall t_1, t_2 \in \mathscr{T} : \begin{array}{l} e_i, e_j \in t_1, e_i, e'_j \in t_2, \\ e_j \neq e_{j'}, \mathscr{A}(e_j) = \mathscr{A}(e_{j'}) \end{array} \forall e_x, e_y \in E \begin{array}{l} \mathscr{A}(e_j) = \mathscr{A}(e_y), \\ \mathscr{A}(e_i) = \mathscr{A}(e_y) \end{array} : \phi^{\neg}_{xy} = 1 \tag{7}$$

This restriction plays an important role; It defines, which positions in the matrix can be activated. The corrupted functional dependencies can be determined using

$$\mho^{\varDelta}_k = \varDelta((\varPhi^T_{k+1} \cdot \varDelta^T) \gtrdot 1) \tag{8}$$

where $\varDelta$ is a binary matrix corresponding to element - attribute active domain projection.

The restriction (7) also makes a requirement to the input data; each tuple has to be consistent, i.e. having one value per one attribute.

$$\forall t \in \mathscr{T}\ \forall e_1, e_2 \in t : \mathscr{A}(e_1) \neq \mathscr{A}(e_2) \tag{9}$$

The consequence of the restriction (7) is only instances not corresponding to the invalid global relationships are considered [5], when new tuple is gathered:

$$\begin{aligned} \varPhi^{\mathscr{A}\times\mathscr{D}}_{k+1} = {} & (\varPhi^{\mathscr{A}\times\mathscr{D}}_k + \boldsymbol{x_L x_L^T}) \odot (1 - \varPhi^{\neg}_{k+1}) = \\ & (\varPhi^{\mathscr{A}\times\mathscr{D}}_k + \boldsymbol{x_L x_L^T}) \odot (1 - \varDelta \mho_k \varDelta^T) \end{aligned} \tag{10}$$

$$\text{where } \mho_k = \sum_{l=1}^{k} \mho^{\varDelta}_l \tag{11}$$

In this point of view, the matrix $\mho_k$ consisting of invalid global relationships known at $k-$th step can be seen as very important data characteristic and has to be stored to satisfy the restriction during next steps.

Note, the matrix $\varPhi^{\mathscr{A}\times\mathscr{D}}$ satisfies transitivity property and is not at general symmetric. Thanks to this fact, two different operators are defined - the generalization in the same way as in the previous case (1) and the specialization as:

$$\boldsymbol{y} = \varPhi^T \cdot \boldsymbol{x} \tag{12}$$

The used formalism allows consideration of situations, when several element activations are needed for any element activation (several conditions has to be satisfied). These situations can be modeled by functional dependencies with a complex attribute $A_L \subset \mathscr{A}$ on the left side. It can be shown [5] that only instances of functional dependencies with single attributes $A \in \mathscr{A}$ can be considered (stored in $\varPhi^{\mathscr{A}\times\mathscr{D}}$) with no information lost under the condition that exists at least one key element implying each element corresponding the attribute on any side of the related functional dependency - attributes on the left side are given by a vector γ_L and alternatively attributes on the right side by a vector

γ_R. In such a case, the elements on the right side can be activated only under the condition all elements on the left side to be activated.

$$\boldsymbol{y}_{(\gamma_L,\gamma_R)} = (\mathit{\Delta} \cdot \boldsymbol{\gamma}_R) \cdot \mathit{\Phi} \cdot (((\mathit{\Phi}^T \cdot \boldsymbol{x}) \odot (\mathit{\Delta} \cdot \boldsymbol{\gamma}_L)) == \sum \boldsymbol{\gamma}_L) \tag{13}$$

The vectors $\boldsymbol{\gamma}_L$ can be stored in the matrix of left sides $\mathit{\Gamma}_L$ and due to the same idea as in (11), the matrix of right sides related to corrupted functional dependencies $\mho_R = [1 - \boldsymbol{\gamma}_R]$. These matrices can be extended for covering also functional dependencies between single attributes:

$$\mathit{\Gamma}_L^\star = [\mathbb{E}|\mathit{\Gamma}_L] \ , \mho_R^\star = [\mho|\mho_R] \tag{14}$$

$$\begin{aligned} \boldsymbol{y} = & \sum_{\forall l} \boldsymbol{y}_{(\gamma_{L_l},\gamma_{R_l})} = \\ & \sum_{\forall l} (\mathit{\Delta} \cdot (1 - \boldsymbol{\gamma}^{\neg}_{R_l})) \cdot \mathit{\Phi} \cdot (((\mathit{\Phi}^T \cdot \boldsymbol{x}) \odot (\mathit{\Delta} \cdot \boldsymbol{\gamma}_{L_l})) == \sum \boldsymbol{\gamma}_{L_l}) \end{aligned} \tag{15}$$

4 Transforming Task

The formula (15) for querying attribute annotated sets corresponds to the formula (1) for not annotated sets, which is much more simpler. In this section, the proposal of value-attribute assignment suppression leading to the complexity reduction is given.

Note, the annotated case satisfies the transitivity property (instead of not annotated one). This is a reason, why a query result can be reached not in one step ($K = 1$), but generally in a dynamic process ($K \to \infty$).

$$\boldsymbol{y}_{k+1} = \pi(\boldsymbol{y}_k), \ \boldsymbol{y}_0 = \boldsymbol{x}, \ k+1 < K \tag{16}$$

Instances of functional dependencies between single attributes given by the matrix $\mathit{\Phi}^{\mathscr{A} \times \mathscr{D}}$ can be easily transformed using maximum degree finding formula (5) as in the not annotated case.

The second part of the formula supports relationship with complex attributes and needs to distinguish the direction (instead of the symmetric matrix $\mathit{\Phi}^{\mathscr{D}}$). This fact leads to the definition of the matrix $\mathit{\Psi} = [\psi_{ij}]$:

$$\psi_{ij} = \frac{\phi_{ji}}{\sum_{\forall i'} \phi_{ji'}} \tag{17}$$

Interpretation of this matrix $\mathit{\Psi}$ is that each element activates its key element in a degree δ given by a key element arity a (a number of related elements to the key one). The degree δ is determined under the assumption of saturation of the state the key element is fully activated by all a connected elements. The saturation is set to 1, so $\delta = 1/a$.

These key element activations may lead to an inconsistent result (basically taking only the fact the elements to be connected in any input tuple), but it can be shown that one of element is activated in a higher degree for all situations corresponding to the instances of functional dependencies. A value assigned to the attribute A by a query vector $\boldsymbol{x}$ can be determined using

$$v_A^{\boldsymbol{x}} = \arg \max_{\mathscr{A}(e_i)=A} \{y_i\} = \arg \max_{\mathscr{A}(e_i)=A} \{\max_{\forall j}\{\phi_{ij} x_j\}\} \tag{18}$$

Situations, when a maximum is not exists, are caused by not satisfying all conditions given by the left side attributes in the corresponding relationship. Further, the maximum exists also in the situation, when the key element is not fully activated, but in the given subdomain, there is no connection to the element with another value (under the condition to values of several attributes on the left side of functional dependency, if these values are fixed, a functional dependency with the same right side and a subset of attributes on left side may exist in the domain given by the value condition). Note, generally these relationships can not be interpreted as functional dependencies.

With respecting this fact, the function π_i for evaluating degree y_i of the element e_i can be now defined as

$$\pi_i : y_i = \max\{\max_{\forall j}\{\phi_{ij} x_j\}, \sum_{\forall j} \psi_{ij} x_j\} \tag{19}$$

This formula is describing a step of the dynamic process (based on the transitivity), but the transitivity (due to not corresponding to the functional dependencies) has no meaning, the result can be seen equivalent to one given in not annotated case. This fact causes

$$\lim_{k\to\infty} y_{ik} = \sigma,\ 0 \leq \sigma \leq 1,\ \forall e_i \text{ connected with any element in the query} \tag{20}$$

To avoid the σ convergence effect, the consistency requirement can be improved by consideration of the matrix $\Theta = [\theta_{ij}]$ defined as

$$\theta_{ij} = \begin{cases} 1 & i = j \\ \frac{-1}{\left\|\mathscr{D}_\alpha(\mathscr{A}(e_i))\right\| - 1} & \mathscr{A}(e_i) = \mathscr{A}(e_j) \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

where $\left\|\mathscr{D}_\alpha(\mathscr{A}(e_i))\right\|$ is a size of a related attribute active domain. The querying algorithm can be now expressed as

$$\boldsymbol{y}_{k+1} = \Theta \cdot \pi(\boldsymbol{y}_k) \tag{22}$$

The query result $\boldsymbol{y}$ can be interpreted as an activation of element e_i, when $y_i = \lim_{k\to\infty} y_{ik} = 1$ and as a disactivation, when $y_i = \lim_{k\to\infty} y_{ik} = -1$. Not connected elements are returned as $y_i = \lim_{k\to\infty} y_{ik} = 0$.

Note, to reach a consistent result from $\boldsymbol{y}_{k+1}$ for $k \to \infty$, a usage of the value preference criterion (18) is required. The result can be given as

$$y_i = \begin{cases} 1\ e_i = (A, v) : v = v_A^{\boldsymbol{x}} \\ 0 \text{ otherwise} \end{cases} \tag{23}$$

Evaluation of the formula (22) can be realized, for example, by a neural network, and principally has no interpretation (no effect of global relationships included). But activations returned by this process can be finally interpreted by (23). The result returned in this way can not be proofed.

5 Conclusion

The paper dealt with the main aspects and features of data set, which values are annotated by attributes.

Sets with no attribute annotation can be used in similar way as information retrieval tools, the repository trained by this kind of set can express only the fact (in a proposed special case extended also by some fuzzy measure), which symbols (values) are connected together (are related in some input tuple).

On the other hand, a formalism used for attribute annotated sets enables to distinguish two views on the data, the first local concerning instances (data) as in the previous case, and the second estimating the global relationships valid on the universum subdomain covered by an input data set. These relationships plays an important role in the following training process and also effects the querying process, because they are useful to explicitly express estimated semantics.

Finally, the paper showed, how the repository trained by an attribute annotated set can be handled as one with no annotation and tried to extract the part strongly connected with the global relationships. It eliminates all global relationships in the steps (19 to 22), the result given by (19) is quantitatively on the same level as a result from a case with no annotation and then by applying steps (23) extend the result by all features coming with global relationships.

The consequence of this paper leads to separation of a learning process into two parts. The first one processes an input data set as a symbols with no extra meaning and this part result is independent on the interpretation - it only assigns one symbol to another one. When the formalism covering also the attribute-value assignment is used, the second part expressing global relationships enables to estimate (explicitly defined) semantics of the used symbols - elements and interpret the result of the first part in any semantic point of view.

References

1. H. Mannila, K.J. Räithä. ”Design by Example: An Applications of Armstrong Relations.”
   Journal of computer and system sciences 33, pp. 129-141. Academic Press. 1986.
2. P.A.Flach, I.Savnik. “Database Dependency Discovery: A Machine Learnig Approach”. In *AI Communications*, Volume 12/3, pp. 139–160. 1999.
3. H. Mannila, K.J. Räihä “Dependency Inference”.
   In *Proc. of VLDB*. pp. 155–158. ISBN: 0-934613-46-X. 1987.
4. H. Mannila, K.J. Räithä. ”Algorithms for Inferring Functional Dependencies from Relations.” In *Data & Knowledge Engineering* 12, pp 83-99. Elsevier. 1994.
5. M. Řimnáč Data Structure Estimation For RDF Oriented Repository Building In *Procedings of CISIS 2007*, IEEE, 2007 (in print).