18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Solving conflicts in database fusion with Bayesian networks

Eleonora Laurenza
Università Sapienza, Italy

***Abstract*—Data fusion is a major task in data management. Frequently, different sources store data about the same real-world entities, however with conflicts in the values of their features. Data fusion aims at solving those conflicts in order to obtain a unique global view over those sources. Some solutions to the problem have been proposed in the database literature, yet they have a number of limitations for real cases: for example they leave too many alternatives to users or produce biased results. This paper proposes a novel algorithm for data fusion actually addressing conflict resolution in databases and overcoming some existing limitations.**

I. Introduction

Data fusion is the task of merging multiple representations of the same real-world entities in order to obtain a single and unified view of them. [1]. As the various representations in different data sources are likely to have disagreeing values for corresponding features, data fusion involves detecting and solving such conflicts.

The problem has an increasingly significant industrial relevance, because of the massive proliferation of redundant and often contradictory data. Moreover, the complexity of the most recent data management scenarios (genomic data, linked open data, statistical microdata), together with the always increasing volumes of data, cause quality loss and reduced trustworthiness of the data. [2].

One of the most common settings for data fusion is *database fusion*, where the entities are memorized in a relational database system (RDBMS). Entities are modeled as relations, with their features being the attributes.

As a concrete example, think of the data about companies collected by many different national business registers that need to be integrated at a federal level, as in the example in Figure 1. Local registers collect information about both domestic and foreign companies, therefore different registers store information about the same company, with unavoidable conflicts. This is the case, in the example, for SIEMENS where the number of employees is 360K in the Italian register and 100 in the German one. Thus, a reconciliation of all the features of each company is needed.

Data fusion has been considered a major problem in the database literature, which has however provided only partial results, working for specific cases. Several algorithms simply ignore the conflicts (*conflict-ignoring*), leaving the choice to the final users; other approaches adopt a preference strategy (*conflict-avoiding*), taking the value from the most trustworthy sources. Finally, some others actually try to solve the conflict (*conflict-solving*), but with techniques that are limited to simple algebraic approximations [3].

These approaches have a number of limitations. Ignoring or avoiding conflicts is not practical, especially with the recent explosion of available sources and features for each entity. Users would be exposed to hundreds or even thousands of alternatives for each conflict. Algorithms based on algebraic approximation only lead to local bests, since the specific kind of approximation depends on each user's sensitivity, overall resulting in a biased global view.

This work proposes *BP-fuse* (*Belief Propagation fusion*), a novel algorithm for solving conflicts in database fusion. The technique relies on knowledge about the domain of interest, deriving either from domain experts or from data analysis. It leverages the probabilistic dependencies among the attributes: non-conflicting values are used to discover what the "true" values are.

Such knowledge is compactly represented in a specific data model, *SSM*, (*simple sensor model*), based on Bayesian networks and the technique actively queries them to take decisions about conflicts.

ITALIAN BUSINESS REGISTER

ID	L_NAME	EMP_NO	GEO_AREA	NACE	PROFIT
526	FCA	100k	Ur	AUTO	20M
114	SIEMENS	360k	Co	ICT	700M
834	Ferrari	9k	-	AGRI	200M

GERMAN BUSINESS REGISTER

ID	L_NAME	REV	GEO_AREA	EMP_NO	FORM
38	FCA	-	Ur	200	SPA
73	SIEMENS	6.14G	Ur	100	Gmbh
714	LVMH	3.06G	Co	83k	-

Fig. 1. Sample tables from European business registers

The remainder of the paper is organized as follows. In Section II some related work on the topic is presented. We begin with a motivating case study in Section III. Some background about Bayesian networks is recalled in Section IV. Section V illustrates the adopted data model and Section VI presents BP-fuse. Section VII discusses some properties of the approach, and in Section VIII, some future work on the topic is envisaged.

II. Related work

Database fusion problem aims at achieving a unified view of the same entity represented by a number of sources, by

solving the conflicts among the disagreeing features. In order to fuse databases, some preliminary tasks are needed, which in the literature are typically grouped in the *data integration* problem [4], [5]. It involves *schema integration* [6], [7] and *data matching* [8]. The former aims at fusing the databases at a schema level, hence achieving the same logical representation of entities, that is, the same name for relations and features; the latter concerns the identification of the same real-world entities in the different sources, as it is often the case that common identifiers (such as social security numbers for individuals, VAT code for companies) are not present.

Once the schemas have been integrated and the corresponding entities matched with a unique identifier, algorithms for database fusion can apply. In the literature some techniques for fusion have been provided, with the mentioned limitations. Some solutions rely on elementary relational algebra operations available in relational systems and provide the users with all the possible alternatives [9], [10], [11], [12], unaffordable in many real cases. Others actually try to solve the conflicts and propose a combination of the disagreeing values based on simple arithmetics [3]. Their results are not always acceptable, as, for instance, the average of two conflicting values may be out of the acceptable domain or, in any case, tightly coupled to each user's sensitivity.

The problem of conflict resolution has been also studied in the field of Web information extraction. Some approaches try to exploit the different reputation of sources and choose values from the most reliable ones: this is sometimes done within a classical probability theory approach [13], [14], or in a purely Bayesian way [15]. Basically, these studies take into consideration the dependencies among sources to establish their trustworthiness and solve conflicts accordingly. In the Web context, this is meaningful, as the sources are highly interrelated and copy data from one another. Other studies in the Web literature assume that all the sources are equally reliable and delegate the decision to the users on the large scale (crowdsourcing) [16], [17].

All these approaches are not effective for databases, which are typically independent from one another. In addition, the crowdsourcing approach has modest applications in the financial or statistical fields, where precise and quantitative knowledge of the amount of the features are needed. Finally, none of these approaches exploits the dependencies among the features, which are indeed relevant in the database context. In the following sections we will show how BP-fuse overcomes these limitations.

Bayesian networks have been used at length in a variety of scenarios, supporting both causal reasoning and prediction [18]. While they are widespread in many contexts, such as expert systems, to the best of my knowledge Bayesian Networks have never been used in database fusion.

III. A CASE STUDY FOR DATABASE FUSION

This section illustrates the approach to the problem of data fusion by referring to a real application of BP-fuse algorithm. Let us consider two European company registers, which are collections of records about multinational enterprises in EU, held, for example, by two different national statistical institutions of the respective member states, Italy and Germany.

The registers are modeled as two relational tables. Figure 1 shows a fragment of those tables. For one single company some characteristics are known in the Italian register and unknown in the German one and viceversa. Besides, for one company, the two registers have conflicting values for the corresponding attributes.

The goal is obtaining a unified business register, fusing the information coming from the two in such a way that for each company the largest number of features is obtained and the data quality is enriched. For each of the registers, ID is the primary key of the relation and uniquely identifies a record about a company in the system and L_NAME is the legal identifier of the company.[1]

Both the registers store the geographical area (GEO_AREA) of production and the number of employees (EMP_NO). There are also differences in the two database schemas: the Italian register is interested in maintaining the primary economic activity classification (NACE), and the yearly amount of the income, which is the result of enterprise after accounting all costs (PROFIT), whereas the German one ignores it, but contains the yearly sales revenue (REV) of the company, which is the gross amount for PROFIT, and the particular legal form of enterprise differing on the basis of national jurisdiction (FORM).

Here we assume that schema integration and data matching have already been performed with appropriate algorithms. Hence, the corresponding attributes in the two relations have the same names, as a result of the schema integration; the corresponding companies have the same value for the L_NAME attribute, as the assignment of an identifier to a real world entity is the result of the data matching.

The approach presented in this paper relies on knowledge about the domain of interest and models it in Bayesian networks. For the domain in the example, a simple net is shown in Figure 2. It represents some kind of causal dependency relating G and N with E.

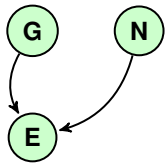

Fig. 2. A simple Bayesian network for business registers

The geographical area where the production site of the company resides, together with the economic classification of its business are reported to influence the number of employees as shown in the probability table in Figure 3. For instance, automotive enterprises (AUTO) situated in the country (Co) tend to have between 10 and 49 employees, while construction

[1]Notice that L_NAME is not part of the primary key in the relation, since different records referring to the same company may exist with different ID.

enterprises (CONST) in urban centers (Ur) have about 70 employees with a probability of 0.33.

		GEO by NACE by EMP_NO							
		AGRI		AUTO		CONST		ICT	
		Ur	Co	Ur	Co	Ur	Co	Ur	Co
	< 10	0.6	0.01	0.2	0.23	0.25	0.19	0.01	0.75
	10-49	0.34	0.1	0.03	0.4	0.3	0.34	0.01	0.21
	50-249	0.03	0.32	0.12	0.07	0.33	0.43	0.2	0.03
	> 249	0.03	0.57	0.65	0.3	0.12	0.04	0.78	0.01

Fig. 3. Relations among GEO, NACE and EMP_NO

Let us consider the fusion of the two records referring to the FCA in Figure 1. FCA is present in both the registers, the attributes NACE and PROFIT are present only in the Italian register: therefore values AUTO and 20M are directly in the result. REV and FORM, which are present only in the German register, appear with their values in the result as well. The two relations agree on the GEO_AREA, but show a conflict for EMP_NO: 100k for the Italian one, 200 for the German one.

BP-fuse solves conflicts of this kind, by evaluating the plausibility of the candidate values, given certain ones. Using the simple Bayesian net in Figure 2 with only three variables, the algorithm calculates $P(100k \mid Ur, AUTO)$, which is 0.65; it also calculates $P(200 \mid Ur, AUTO)$, yielding 0.12. The most plausible value is 100k and it is assigned to EMP_NO in the fused record. The case for SIEMENS is quite similar, however particular attention must be paid as both GEO_AREA and EMP_NO disagree. The final two records, Ferrari and LVMH, appear only in one relation and so they are directly part of the result.

In the following sections, with some elaborations on the running example, it will be showed how richer and more expressive networks are effectively exploited by the proposed algorithm.

EUROPEAN BUSINESS REGISTER							
ID	L_NAME	EMP_NO	GEO_AREA	NACE	REV	PROFIT	FORM
1	FCA	100k	Ur	AUTO	-	20M	SPA
2	SIEMENS	360k	Ur	ICT	6.14G	700M	GMBH
3	Ferrari	9k	Co	AGRI	-	200M	-
4	LVMH	83k	Co	-	3.06G	-	-

Fig. 4. The result of BP-fuse algorithm

IV. Preliminaries

Let us first recall the concept of Bayesian Network (BN). They are essentially DAGs (Directly Acyclic Graphs) that specify a multivariate joint probability distribution over a set of random variables used to represent knowledge of variable relationships in an uncertain domain.

The *nodes* represent the random variables that are concerned in the reality of interest. In Figure 2, random variables are the geographical area, the economic classification and the number of employees in a company. Probabilistic dependencies among variables are graphically expressed by *directed edges* in the network.

Each node is labelled with a *condition dependency probability distribution* (CPT) table. It contains the distribution of such variable, as it is conditioned by all the variables corresponding to incoming edges. It encodes the quantitative knowledge about the domain. CPTs of root nodes directly contain the a priori distribution of the corresponding variables as reported, since no conditioning variables are present.

Fitting graphical models is called learning, a term borrowed from artificial intelligence theory, and in general requires a two-step process. The first step consists in finding the graph structure that encodes the conditional independencies present in the data. The second step is called parameter learning and deals with the estimation of the values of the CPTs of the network. Both structure and parameter learning are often performed using a combination of algorithms (a big variety exists indeed[2] and prior expert knowledge of the data [18].

The CPT in Figure 3 shows how the number of employees, which is the variable the table refers to, varies depending on the geographical area and the economic classification of the company.

In this example the involved variables are discrete or discretized according to the knowledge of the domain experts. In general, for continuous variables, any automatic (or knowledge based) discretization technique may as well be adopted. Clearly, techniques yielding a thorough definition of the reality of interest result into finer intervals and thus better represent the levels of the variables. This produces a network that is more reliable and precise.

We call *evidence set*, the collection of all the observed certain variables with their value. Given an evidence set, the Bayesian network allows to calculate the probability distribution of every variable. More formally, the network supports the computation of the probabilities of any subset of variables given evidence about any other subset [22].

The network provides an efficient way to compute the joint probability distribution of all the variables, since every joint or conditional probability can be then derived. With reference to Figure 2, consider the joint probability distribution of all the involved variables, which can be calculated through the chain rule: $P(G, N, E) = P(G)P(N|G)P(E|N, G)$. The structure of a BN implies that the value of a particular node is conditional only on the values of its parents. In the example, the network shows that the economical activity classification (N) and the geographical location (G) are independent. This reduces the chain to $P(G, N, E) = P(G)P(N)P(E|N, G)$. This simplification becomes more useful as the dimension of the network grows.

In general, in a BN containing k nodes, N_1 to N_k, a value in the joint distribution is represented by $P(N_1{=}n_1, N_2{=}n_2, \ldots, N_k{=}n_k)$, or more compactly, $P(n_1, n_2, \ldots, n_k)$. Factorizing with the chain rule, we obtain: $P(n_1, n_2, \ldots, n_k) = \prod_i P(n_i \mid Parents(N_i))$.

[2]For example: maximum likelihood estimation, Bayesian estimation [19], regularized estimation [20] [21].)

Although the simplification BNs introduce is very effective, it works well only with respect to small networks, since the evaluation of the simplified chain requires a non-polynomial algorithm. This causes the computation time to grow exponentially as the network complexity grows [3].

In practice, more efficient algorithms, based on a message-passing strategy, have been proposed. They all are variations on Pearl's belief propagation algorithm, which has been proven to have polynomial complexity when applied to particular network topologies [23]. Belief propagation allows to calculate and update a *belief* status vector (BEL) for every node in a Bayesian network as the algorithm converges. BEL vector is the posterior probability distribution of the corresponding random variable, given the a priori evidence. Nodes have a *causal support vector* π and an *evidential support vector* λ, representing the probability distribution of the corresponding variable, given all the evidential information coming from their ancestors or descendants.

Every node V receives all messages $\pi_V(U_i)$ from its parents and $\lambda_{Y_j}(V)$ from its children and then calculates its belief vector as follows: $BEL(V)$=$\pi(V)\lambda(V)$, where $\pi(v) = \sum_u P(v \mid u)\ \pi_V(u)$ and $\lambda(v) = \prod_j \lambda_{Y_j}(v)$. The node, using the received λ messages, computes a new message $\lambda_X(U)$, which is sent to its parents U and computes new π messages to be sent to each of its children. $\lambda_V(u) = \sum_v \lambda(v)\ P(v \mid u)$ and $\pi_{Y_j}(v) = \alpha\ \pi(v)\ \sum_{k \neq j} \lambda_{Y_k}(v)$, where α is a normalization factor.

V. Simple sensor model

Simple sensor model (SSM) is the data model envisaged to support data fusion in this paper. This model uses a terminology that is typical of the multi-sensor fusion context and data to be fused are modeled as the measures in a physical sensor. We introduce a generic data model to allow for a solution that is independent of the specific data representation. Indeed, the correspondence with the relational model is quite straightforward: relations correspond to sensors, with the attributes being their variables; real-world keys in the relations correspond to the sensor identifier. For the other models, for example XML, the approach works as well with a mapping of the respective constructs into SSM. For instance, XML nodes would be also mapped into sensors and their attributes into the variables.

Specifically, a *sensor* $S(I, \mathbf{V})$ is characterized by an identifier I and a set of variables $V = V_1, \ldots, V_n$. The identifier and the variables represent the attributes of the entity measured by the sensor, in particular the identifier is the real-world name. The instances of each sensor are the measures $m(i, v)$, where each one is an assignment i for I and $v = (v_1, \ldots, v_n)$ for $\mathbf{V}$. SSM comprises the information about the causal dependencies among the variables of the sensors, which is the perceived logical implications behind the real-world entities, and adopts constructs from Bayesian networks to model them.

[3]Network complexity is not measured by the number of nodes, but by a quantity that is related to the connectivity of the network and to the numbers of possible values for the attributes.

The identifiers are also the link between different sensors, because they allow to tell what measures refer to the same entity. Measures can be incomplete, either because they miss some values for certain variables or because in a particular point in time all the variables are missing. A useful summary of the SSM model is depicted in Figure 5 in the form of a UML domain model.

A. The data fusion problem

Let us give some definitions to build techniques for database fusion. Given three sensors $S_1(I, \mathbf{V}_1)$, $S_2(I, \mathbf{V}_2)$ and $S_3(I, \mathbf{V}_3 = \mathbf{V}_1 \cup \mathbf{V}_2)$, with the same identifier I, where V_1 and V_2 are two sets of variables with a possibly non empty intersection, $\mathbf{V}_3$ is the union set of $\mathbf{V}_1$ and $\mathbf{V}_2$, S_3 is a fusion for S_1 and S_2 and we write $S_3 = \text{fuse}(S_1, S_2)$ if for each pair of measures $m_1(i, v_1) \in S_1$, $m_2(i, v_2) \in S_2$, there exists a measure $m_3(i, v_3)$ in S_3, where:
$v_3 = (v_{11}, v_{12}, \ldots, v_{1n}, v_{c1}, v_2, \ldots, v_{cr}, v_{21}, v_{22}, \ldots, v_{2m})$, $v_{11}, \ldots, v_{1n}$ is an assignment for $\mathbf{V}_1 - \mathbf{V}_2$ variables with values from S_1; $v_{21}, \ldots, v_{2m}$ is an assignment for $\mathbf{V}_2 - \mathbf{V}_1$ variables with values from S_2; $v_{c1}, \ldots, v_{cr}$ is an assignment for $\mathbf{V}_1 \cap \mathbf{V}_2$ variables where each value v_i is derived with a conflict-solving strategy.

VI. BP-fuse algorithm

Let us introduce BP-fuse algorithm. It is formulated with reference to the simple sensor model introduced in Section V, and deals with the data fusion problem defined in Section V-A.

BP-fuse takes as input a number of sensors $S_1(I, \mathbf{V_1}), \ldots, S_s(I, \mathbf{V_s})$ with the same identifier I, where variable sets $\mathbf{V}_1 \ldots \mathbf{V}_s$ can be overlapping and a Bayesian network defined on such variables. It returns a sensor $S_r(I, \mathbf{V_r})$ such that $S_r = \text{fuse}(S_1, \ldots, S_s)$.

BP-fuse has two phases: the former, *emission*, is devoted to the extraction of the measures from the input sensors; the latter, *unification*, has the responsibility to actually solve conflicts among the values of the variables in all the sensors.

For every sensor and for every measure $m(i, v_1, \ldots, v_k)$, the emission phase produces a set of triples (i, V_1, v_1), $\ldots$, (i, V_k, v_k). The triples are then grouped by identifier i into *candidate entities* (CE) which are collections of triples referring to the same real world entity. In a candidate entity the triples are in turn grouped by V_i into *candidate sets* (CS). A candidate set collects for each variable and entity, all the possible values coming from different measures and sensors [4].

The unification phase has the responsibility to produce from every candidate entity a measure for S_r. To achieve this, BP-fuse needs to reduce every candidate set to a unique value. Four cases are possible with respect to candidate set reduction: i) *there is only one non null value in the candidate set*: BP-fuse chooses the non null candidate value; ii) *null set*: the candidate set only contains the null value, BP-fuse chooses the null value; iii) *no conflict*: the candidate set has exactly one value, BP-fuse chooses this value; iv) *conflict*: there are

[4]Notice that for a given i, different candidates for a variable can also derive from the same sensor, in case of duplicate measures.

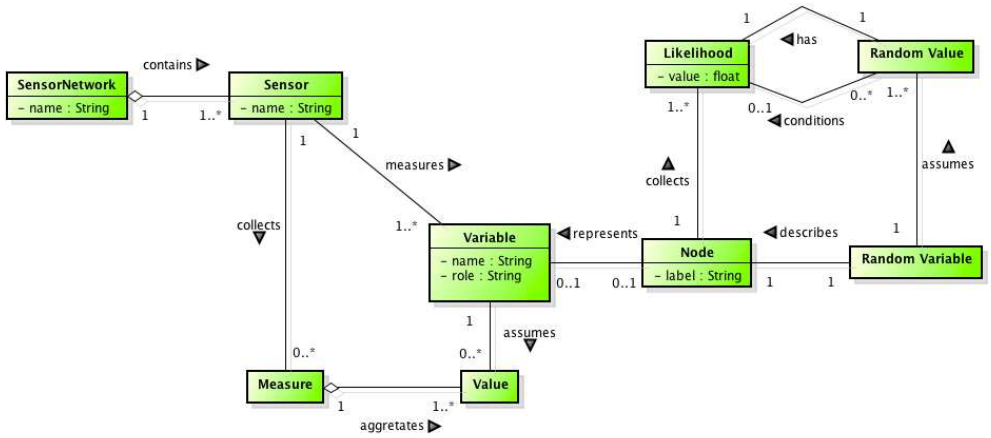

Fig. 5. A graphical representation of SSM: Domain Model

different values in the candidate set. Case iv) is indeed very common and, moreover, several variables are likely to be contemporaneously conflicting in a measure.

For every candidate entity, BP-fuse considers all the conflicting variables at the same time. Let $V_1, \ldots, V_t$, [5] be such variables. BP-fuse generates all the possible assignments $a = (v_1, \ldots, v_t)$, where v_i is chosen from candidate set V_i. Then the algorithm investigates the plausibility of each assignment a as follows. Let $V_{t+1}, \ldots, V_q$ be the other variables of the measure, the ones for which the respective candidate sets have already been reduced by applying cases i-iii. For each assignment a, BP-fuse estimates the plausibility with the support of the associated Bayesian net. It generates and evaluates queries such as:

$$P(v_1, \ldots, v_t \mid v_{t+1}, \ldots, v_q). \tag{1}$$

In order to efficiently compute 1 for a, BP-fuse applies some basic manipulations. Query 1 is turned into the ratio between two conjunctive forms:

$$\frac{P(v_1, \ldots, v_t, v_{t+1}, \ldots, v_q)}{P(v_{t+1}, \ldots, v_q)} \tag{2}$$

Each conjunctive form is factorized into $P(v_1)P(v_2 \mid v_1) \ldots P(v_n \mid v_{n-1}, \ldots, v_1)$ by applying the chain rule. Now, BP-fuse orderly calculates each factor $P(v_i \mid v_1, \ldots, v_j)$ by applying belief propagation. It starts from initial factors $P(v_i)$ of the chain and then uses each v_i in the evidence set for the following factors. It eventually extracts the belief BEL$(V_i = v_i)$ for the conditioned variable v_i. BP-fuse calculates the plausibility of a by replacing previously calculated factors in ratio 2.

At this step, BP-fuse chooses for the candidate entity under consideration the assignment a with the highest plausibility. It reduces every candidate set to a unique value and, as a consequence, produces a measure for S_r. The application of the explained steps to all the candidate entities results in the generation of all the fused measures for S_r.

Let us now come again over the running example introduced in Section III in order to see a fully detailed application of BP-fuse algorithm. To this end, let us consider the two business registers in Figure 1 and the related support network depicted in Figure 6.

It contains the supplementary nodes FORM, REV and PROFIT, modeling the legal form, the revenue and the net profit of the company. These three nodes correspond to attributes of the tables. The network relates the involved attributes also with other concepts of the domain of interest that are not present in the tables. They are J_LABOUR_COST and EXPORT_VOL, modeling the yearly workforce cost and percentage of export for every company. Like GEO and NACE, FORM is deemed to have some relation with the number of employees EMP_NO. The geographical location of the production plant influences the workforce cost, which, in turn, affects the PROFIT together with REV. The economic activity classification NACE has some implication on the volume of export EXPORT_VOL.

Let us consider the two records referring to SIEMENS, which are in conflict both on GEO_AREA and on EMP_NO. In BP-fuse terms, we have two sensors, S_I and S_G, with their measures:
m_I = (L= SIEMENS, E=360k, G=Co, N=ICT, P=700M)
m_G = (L= SIEMENS, E=100, G=Ur, R=6.14G, F=Gmbh)

A. Emission phase

The emission phase produces the following triples:

CE(SIEMENS) = {(SIEMENS, E, 360k)(SIEMENS, G, Co),(SIEMENS, N, ICT), (SIEMENS, P, 700M),(SIEMENS, E, 100), (SIEMENS, G, Ur), (SIEMENS, R, 6.14G), (SIEMENS, F, Gmbh)}

These triples refer to the same entity (SIEMENS), so BP-fuse maps them into the same candidate entity. Then,

[5] We should distinguish between the variables and the respective sets to be reduced and adopt a different symbol for the two; however, here, for shortness the same letter will be used.

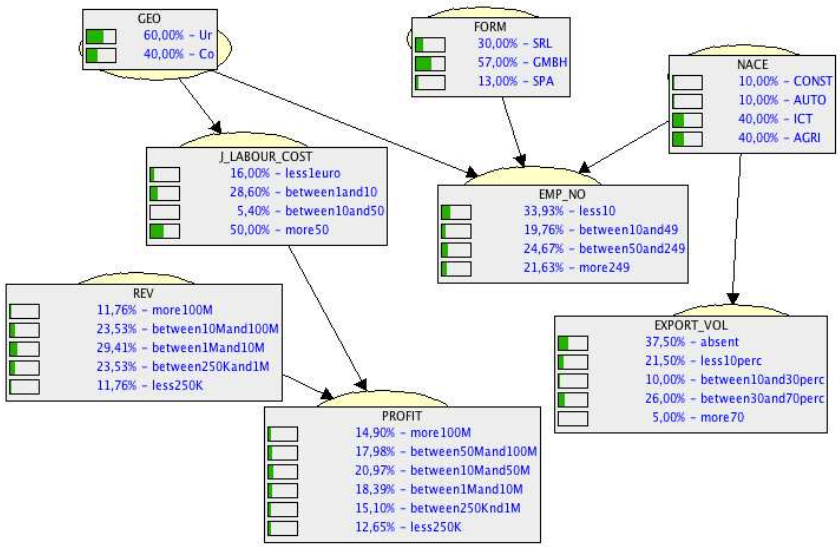

Fig. 6. Example of Bayesian network describing the involved variables and the respective belief vectors

within the entity, the algorithm constructs the candidate sets referring to the triples, so for each variable, a set is built:

CS(E) = {360k, 100}, CS(G) = {Co, Ur},
CS(N) = {ICT}, CS(R) = {6.14G},
CS(P) = {700M}, CS(F) = {Gmbh}

Candidate sets CS(N), CS(R), CS(P), CS(F) contain a single value; hence, they are reduced to the value itself by applying case iii. Now, CS(E) and CS(G) show conflicts.

B. Unification phase

BP-fuse proceeds to generate all the possible assignments in order to estimate the most plausible.

a_1 = (E=360k, G=Co, N=ICT, R=6.14G, P=700M, F=Gmbh)
a_2 = (E=360k, G=Ur, N=ICT, R=6.14G, P=700M, F=Gmbh)
a_3 = (E=100, G=Co, N=ICT, R=6.14G, P=700M, F=Gmbh)
a_4 = (E=100, G=Ur, N=ICT, R=6.14G, P=700M, F=Gmbh)

For each assignment, the calculation reduces, with some algebraic simplifications on formula 2, to the product of the the factors: $P(E|N,R,P,F)\ \ P(G|N,R,P,F,E)$. The factors are calculated as the belief of the conditioned variable, extracted from the associated Bayesian network after belief propagation convergence. Thus, with the appropriate evidence sets, we obtain:

a_1: BEL(E=360k) BEL(G=Co) = 0,
a_2: BEL(E=360k) BEL(G=Ur) = 0.24,
a_3: BEL(E=100) BEL(G=Co) = 0.18,
a_4: BEL(E=100) BEL(G=Ur) = 0.

BP-fuse returns the measure corresponding to a_2, the assignment with the highest plausibility, solving both the conflicts together.

VII. Algorithm discussion

The major novelty of BP-fuse is the fact it exploit the dependencies among the attributes to solve the conflicts. Also, these dependencies may include features that are neither modeled in the database nor measured, yet are however part of the domain of interest. It is the case, for instance, of J_LABOUR_COST and EXPORT_VOL in Figure 6.[6] Once it has been captured by the Bayesian network, the knowledge can be used independently of the data. In this sense BP-fuse is context independent but domain aware.

Let us make some informal considerations about the correctness of BP-fuse. The unification generates all the possible assignments, taking into account, all the candidate solutions for the conflicts. The plausibility of each of them is estimated by extracting the belief value after the belief propagation convergence. BP-fuse converges under the constraint that the Bayesian network is a *singly connected graph*, also known as *polytree* [23]. This is a direct consequence of the termination conditions of belief propagation [24].

Also the complexity of BP-fuse can be easily derived from the one of belief propagation. For each conflict, we need the equilibrium of the belief propagation algorithm over the used Bayesian network. Such convergence is guaranteed to be reached in time proportional to the network diameter [24][7].

[6]Indeed, it can be referred to as a *latent variable*, since it is not measured but helps relate geographical location to profit.

[7]The diameter of a network is the length of the longest path between a pair of nodes.

A. Quality of data fusion answer in BP-fuse

Let us evaluate the quality of BP-fuse result by means of *completeness* and *conciseness*, two indicators introduced in [3] to study fusion techniques with an approach recalling the more usual terms of *precision* and *recall*.

The two indicators can be easily defined on sets: completeness of a set is the ratio between the number of unique elements in the set and the number of unique elements in the universe $\frac{|S|}{|U|}$; conciseness is the ratio between the number of unique elements in a set and the number of all the elements in the set $\frac{|S|}{|SET|}$. The two indicators are used here to estimate the quality of data fusion algorithms, by comparing the representativity and the redundancy of the input sensors with the ones of the data fusion answer. A good fusion algorithm would be expected to increase the completeness and, at least, not to decrease the conciseness with duplicates. In our context, the *extensional completeness* of a sensor is the ratio between the number of unique entities referred to by the measures and the total number of entities in the universe (that is all the available sensors together). This indicator expresses how widely the sensor covers the reality: the higher the value, the larger the coverage. In our running example in Figure 1, both S_I and S_G have extensional completeness $3/4$. The *intensional completeness* of a sensor is the ratio between the number of variables and the total number of unique variables measured by all the available sensors. In the example, both the sensors have an intensional completeness of $5/7$. Conciseness can be extensional and intensional as well, amounting the number of unique entities or variables in a sensor. This indicator conveys the idea of how compact the sensor is: the higher the value the more compact the sensor. For both the sensors in the example, conciseness values are 1, since there are no duplicates either in the measures or in the variables. The fused sensor, returned by BP-fuse, has the best value for intensional completeness (the ratio is 1) as it contains the union of the variables from all the operands by definition (Section V-A); BP-fuse maximizes extensional completeness as well (the ratio is 1), since the key-value pairs are generated for all the involved sensors and no measures are discarded during the unification phase.

BP-fuse also maximizes the conciseness of the result. The fusion answer is intensionally concise by construction, since we assume that the schema matching has already been performed, associating semantically equal variables to the same name, and that the fused sensor contains the union of the variables, where duplicates are not allowed. Also, extensional conciseness is 1. In facts, BP-fuse emission produces a key-value pair for each measure and variable, and the unification phase collects all the pairs with the same real-world key into a single fused measure.

VIII. Conclusions

This paper presented BP-fuse as a novel algorithm to solve conflicts in database fusion. The major result is the possibility to exploit the dependencies among the features to solve the conflicts. These dependencies are modeled in Bayesian networks that represent domain knowledge deriving from experts or data analysis. Dependencies among the variables and non-conflicting values are used in conjunction in a global perspective, to tell which values are more plausible in the result.

Furthermore, the technique that has been described is off-line, in the sense that it considers the present data and applies transformations to obtain the result. Successive extensions to the system would also include the possibility to perform the algorithm in streaming and at runtime.

A critical goal, especially in finance and statistics, is keeping the lineage of data. This aspect is even more important with respect to data fusion, where conflict resolution may hide the relationship between the original data sources and the fused ones. BP-fuse partially addresses this issue though requiring some specific extensions. The original values are not lost, once they are merged into their fused version.

References

[1] M. E. Liggins, J. Llinas, and D. L. Hall, *Multisensor data fusion*. CRC Press, Inc., 2008.

[2] E. Rahm and H. H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Eng. Bull.*, vol. 23, no. 4, pp. 3–13, 2000.

[3] J. Bleiholder and F. Naumann, "Data fusion," *ACM Computing Surveys (CSUR)*, vol. 41, no. 1, p. 1, 2008.

[4] J. D. Ullman, "Information integration using logical views," in *Database Theory - ICDT '97, 6th International Conference, Delphi, Greece, January 8-10, 1997, Proceedings*, 1997, pp. 19–40.

[5] M. Lenzerini, "Data integration: A theoretical perspective." in *PODS*, 2002, pp. 233–246.

[6] A. Y. Halevy, "Answering queries using views: A survey," *The VLDB Journal*, vol. 10, no. 4, pp. 270–294, 2001.

[7] P. A. Bernstein, "Applying model management to classical meta data problems." in *CIDR*, vol. 2003, 2003, pp. 209–220.

[8] P. Christen, *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer, 2012.

[9] S. Raghavan and H. Garcia-Molina, "Integrating diverse information management systems: A brief survey," *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 44–52, 2001.

[10] C. Galindo-Legaria, "Outerjoins as disjunctions," in *SIGMOD Conference*, 1994, pp. 348–358.

[11] S. Greco, L. Pontieri, and E. Zumpano, "Integrating and managing conflicting data," in *Ershov Memorial Conference*, 2001, pp. 349–362.

[12] L. Yan and M. Tamer, "Conflict tolerant queries in aurora." in *CoopIS*. IEEE Computer Society, 1999, pp. 279–290.

[13] X. L. Dong, L. Berti-Equille, and D. Srivastava, "Integrating conflicting data: the role of source dependence," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 550–561, 2009.

[14] M. Bronzi, V. Crescenzi, P. Merialdo, and P. Papotti, "Extraction and integration of partially overlapping web sources," *PVLDB*, vol. 6, no. 10, pp. 805–816, 2013.

[15] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han, "A bayesian approach to discovering truth from conflicting sources for data integration," *Proceedings of the VLDB Endowment*, vol. 5, no. 6, pp. 550–561, 2012.

[16] E. Kamar, S. Hacker, and E. Horvitz, "Combining human and machine intelligence in large-scale crowdsourcing," in *Proceedings of the 11th ICAAMAS - Volume 1*, ser. AAMAS '12. IFAAMS, 2012, pp. 467–474.

[17] V. Crescenzi, P. Merialdo, and D. Qiu, "Crowdsourcing large scale wrapper inference," *Distributed and Parallel Databases*, vol. 33, no. 1, pp. 95–122, 2015.

[18] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[19] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine learning*, vol. 20, no. 3, pp. 197–243, 1995.

[20] J. Hausser and K. Strimmer, "Entropy inference and the james-stein estimator, with application to nonlinear gene association networks," *The Journal of Machine Learning Research*, vol. 10, pp. 1469–1484, 2009.

[21] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.
[22] S. K. Das, *High-level data fusion*. Artech House, 2008.
[23] J. Pearl and S. Russel, *Bayesian networks*, 2011.
[24] J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artificial intelligence*, vol. 29, no. 3, pp. 241–288, 1986.