Probabilistic Frequent Itemset Mining with Hierarchical Background Knowledge

André Melo and Johanna Völker

Abstract—**In the recent years, there has been significant development in the field of Probabilistic Frequent Itemset Mining (PFIM). Despite the complexity of calculating the *frequentness probability* of an itemset, approximation techniques allow us to reduce the complexity of the problem with very low approximation error. In this paper we investigate how to incorporate hierarchical taxonomies into the *attribute uncertainty* model, which assumes independence between the existential probability of items in a transaction. We propose scalable methods which can reduce noise, and ensure consistency of the transactions by approximating the dependencies between attributes implied by a background hierarchical taxonomy. We also perform experiments in order to evaluate the scalability, accuracy of the approximation, as well as the denoising performance of the proposed methods.**

Index Terms—**Probabilistic frequent itemset mining, generalized rules, hierarchical background knowledge.**

I. Introduction

Uncertainty is inherent to many types of data and applications. It can originate, for instance, from measurement noise, trustworthiness of the source and confidence values of information extraction, automatic data enrichment, and data cleansing techniques. There are mainly two ways of modeling uncertain data: *attribute uncertainty* and *tuple uncertainty*. In the first, the existence of items in a transaction is uncertain, and in the latter, the items in the transactions are certain, but the existence of a transaction in the dataset is uncertain.

In many applications, the uncertain attribute probability values are generated independently. Besides that, the *attribute uncertainty* is simpler to represent and to work with. However, its main drawback is that it assumes independence between the attributes. This results in the impossibility of modeling dependencies which might be contained in the background knowledge. Moreover, it might happen that, because of noise for example, an uncertain transaction is inconsistent with respect to the background knowledge. In this case, the background knowledge can be used to solve inconsistencies as well as to improve the quality of the data.

Although the *tuple uncertainty* model enables dependencies to be represented, the number of tuples required to represent a single dependent transaction grows exponentially with the number of uncertain items. Even if we consider a hybrid of the *attribute uncertainty* and *tuple uncertainty* models, where only sets of interdependent items are modeled with *tuple uncertainty*, while the independent items are represented with *attribute uncertainty*, this approach would be unfeasible in case of high interdependence between the attributes.

Linked data mining is a concrete example of application where methods for dealing with large amounts of uncertain data *and* explicitly modeled dependencies are required [1]-[3]. Knowledge bases such as DBpedia, which have been generated in a (semi-) automatic way, naturally contain many uncertain statements. We can obtain explicit estimates of these uncertainties by means of data debugging techniques [4], or use the uncertainty values provided by machine learning methods for fact prediction [5]. Additionally, the schemas (or *ontologies*) which are provided by many RDF knowledge bases as background knowledge can help to speed up the mining process and to cope with the noise resulting, e.g., from probabilistic methods for knowledge base enrichment. In any case, the resulting uncertain knowledge base can be huge, and highly scalable mining algorithms are required, in order to deal with the magnitude of the data, uncertainty and available background knowledge.

Hierarchical taxonomies, such as the example from Fig. 1[1], are a common type of background knowledge. These taxonomies imply dependencies which cannot be represented with the *attribute uncertainty* model. These dependencies can be used to speed up the frequent itemset mining process [6], and also to improve the quality of the uncertain transactions. However, exploiting these dependencies to improve the quality of transactions is expensive and has scalability issues, as we will discuss in more details later in Section V.

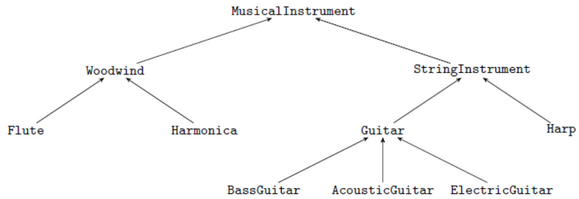

Fig. 1. Example of a hierarchical taxonomy T.

In this paper we propose more scalable techniques that approximate the expensive exact dependencies computation, and can be represented with the attribute uncertainty model. These techniques basically involve recomputing the singletons' existential probabilities in the transaction considering the dependencies implied by taxonomy. We also evaluate the proposed dependency approximations by measuring their performance in terms of runtime and distance to the exact dependency computation. Moreover, we evaluate the impact of these methods on both exact and approximated

Manuscript received March 2, 2015; revised June 3, 2015.

André Melo and Johanna Völker are with University of Mannheim, Germany (e-mail: andre@informatik.uni-mannheim-de, johanna@informatik.uni-mannheim-de).

[1] This is a subset of Google Product taxonomy: http://www.google.com/basepages/producttype/taxonomy.en-US.txt

DOI: 10.7763/IJKE.2015.V1.16

PFIM.

The rest of this paper is organized as follows. In Section II, preliminary concepts used by this paper are presented, Section III defines the problem object of investigation, and in Section IV we discuss the related work. In Section V we describe the exact computation of dependencies, and the proposed approximation approaches are presented in Section VI. Section VII provides the experimental evaluation of these approaches, and finally, in Section VIII we state our conclusions.

II. Foundations

In this section we briefly present some of the works which are fundamental to the understanding of this paper. Firstly we present the attribute uncertainty model, the possible world semantics, and finally exact probabilistic itemset mining and its approximations

A. Attribute Uncertainty Model

In this model, every attribute in each transaction carries some uncertainty information. All the attributes are assumed to be independent, and this assumption is used to compute the probability of composite itemsets. Let $I = \{i_1, \dots, i_n\}$ be a set of n binary attributes called items, $D = \{t_1, \dots t_d\}$ a set of d uncertain transactions, and $t_j = \{y_1, \dots, y_m\}$ an uncertain attributes transaction where for every item $y \in t_j$ has an existential probability $P(y \in t_j)$.

Since the attribute uncertainty model assumes independence between attributes, the probability that an itemset $X \subseteq I$ exists in an uncertain transaction t is simply the multiplication of the probabilities of each item $i \in X$ as shown in Equation (2.1).

$$P(X \subseteq t) = \prod_{i \in X} P(i \in t) \qquad (2.1)$$

Also, the expected support of an itemset X is simply the sum of its existential probability in each transaction of D as shown in Equation (2.2).

$$esup(X) = \sum_{t \in D} P(X \subseteq t) \qquad (2.2)$$

B. Possible World Semantics

For an uncertain dataset D, there is a set of possible worlds $w_i \in W$. For each w_i, D is a certain dataset $D_i = \{t_{i_1,1}, \dots, t_{i_d,d}\}$, and $t_{i,j} \subseteq t_j$ is a possible certain transaction generated from t_j. For the transaction t_1 from Table I, for instance, we have $t_{1,1}, t_{2,1}, t_{3,1}, t_{4,1}$ as possible transactions.

TABLE I: The Possible Worlds of the Transaction T_1

	BassGuitar	Guitar	$P(t_{i,j})$
t_1	0.9	0.8	--
$t_{1,1}$	0	0	0.02
$t_{2,1}$	0	1	0.18
$t_{3,1}$	1	0	0.08
$t_{4,1}$	1	1	0.72

The probability $P(t_{i,j})$ that an uncertain transaction $t = t_{i,j}$ is calculated with Equation (2.3). Hence, the probability of possible world $P(w_i)$ is defined with Equation 2.4, where:

$$P(t_{i,j}) = \prod_{x \in t_{i,j}} P(x \in t_j) \prod_{y \notin t_{i,j}} 1 - P(y \in t_j) \qquad (2.3)$$

$$P(w_i) = \prod_{t_{x,j} \in D_i} P(t_{i,j}) \qquad (2.4)$$

C. Probabilistic Frequent Itemset Mining (PFIM)

In order to determine if an itemset is probabilistic frequent, we calculate the frequentness probability, i.e. the probability that a given itemset is frequent $P(\sup(X) \geq minsup)$ [7], which is the sum of the probabilities of all possible worlds where X is frequent. An itemset X is a Probabilistic Frequent Itemset (PFI) if its frequentness probability satisfies a minimum *probability frequent threshold* (*pft*). Chui *et al.* [8] introduce frequent itemset mining on uncertain data based on expected support. This approach approximates the *spmf* to a unit step function and requires only the computation of $esup$. Wang *et al.* [9] approximate the *spmf* of an itemset X to a Poisson distribution with $\lambda = esup(X)$ and Calders *et al.* [10] approximate the *spfm* with a normal distribution defined by $N(esup(X), var(X))$, where $var(X) = \sum_{t \in D} P(X \subseteq t)(1 - P(X \subseteq t))$. Bernecker *et al.* [11] performed a thorough comparison of the three models presented above, and as a conclusion, the authors propose to generally use the normal distribution approximation because it yields the best trade-off between approximation quality and efficiency.

III. Problem Definition

The *attribute uncertainty* model assumes independence between attributes, however, if there is a hierarchical taxonomy as background knowledge, this independence does not hold (c.f. Section V). Considering the dependencies implied by such background knowledge can help improve the quality of the data, however, the exact computation of the dependencies is an expensive task.

The problem is to exploit hierarchical taxonomies to improve the quality of uncertain data modeled with *attribute uncertainty* in an efficient and scalable way. Our objective is to approximate the dependencies, which are expensive to compute, and use them in order to reduce noise in the transactions. Our setting assumes the attribute uncertainty values to be acquired independently and the noise to be generated independently for each attribute. Moreover, the higher level nodes of the taxonomy to be attributes also present in the data. We also allow partial paths in the hierarchy, i.e., instances do not necessarily have to be leaf nodes. We propose scalable approximations of the dependencies computation, which can applied in the PFIM task improving the quality of the mining outcome.

IV. Related Work

Srikant and Agrawal [6] introduced the problem of mining generalized association rules on certain data with background taxonomies. It enables the mining algorithm to learn rules across different levels of the taxonomy. This is important, because it may happen that at lower levels a rule does not satisfy the minimum support threshold. Considering higher levels allows us to mine rules which would not be learned otherwise, and to learn more concise and generalized rules. The authors propose methods which explore taxonomies to speed up the mining process. These methods make use of *redundant itemset pruning*, which prunes an itemset X containing an item x and its generalization $\hat{x}$, since X has the same support as $X - \hat{x}$, and therefore does not need to have its support computed. This pruning technique plays an important role in this paper and it will be further discussed in Section V. Similar work on learning generalized association rules includes [12], which has a different approach that encodes the taxonomical information as digits into item ids.

Peterson and Tang [13] introduced probabilistic generalized frequent itemset mining on attribute uncertainty databases with taxonomies. On their setting the uncertain database contains exclusively leaf nodes, and the existential probability of a generalization is calculated as the probability of the union of its direct specializations which are assumed to be mutually independent. Their problem setting is different from ours, as we assume the generalized items to be already present in the database, and the uncertainty values to be obtained independently, which can lead to inconsistency, and they do not allow partial paths in the hierarchy Moreover, we exploit the dependencies implied by the background knowledge to improve the quality of this uncertainty values.

V. Exact Computation of Dependencies

We want to take into account the dependencies implied by the background knowledge in order to improve the quality of the transactions and solve inconsistencies. The computation of the dependencies can be done by generating the joint probability table of an uncertain transaction with all its possible worlds. As described in Section II, an uncertain database implies the existence of possible worlds $w_i \in W$, whose probabilities sum up to 1. However, some of the possible worlds can be inconsistent w.r.t. the hierarchical background knowledge T, if for some $\{x \sqsubseteq \hat{x}\} \in T$, a transaction violates some of the constraints imposed by a taxonomy, which are described by Constraints 1, 2 and 3.

TABLE II: Example of Uncertain Transaction

	Bass Guitar	Acoustic Guitar	Electric Guitar	Guitar	String Instrument
t_2	0.9	0.5	0.1	0.8	0.8

For an uncertain transaction *t* containing the items x and $\hat{x}$, where $x \sqsubseteq \hat{x}$ the following constraints apply:

Constraint 1. The existential probability of $\hat{x}$ is greater or equal to that of its specialization x, i.e., $P(\hat{x} \in t) \geq P(x \in t)$

Constraint 2. The existential probability of an itemset X containing $\hat{x}$ and x is equal to that of $X - \hat{x}$, i.e., $P(X \subseteq t) = P(X - \hat{x} \subseteq t)$.

Constraint 3. The existential probability of an itemset X containing x and $\neg\hat{x}$ is zero, i.e., for $\{X \mid \{x, \neg\hat{x}\} \subseteq X\}$, $P(X \subseteq t) = 0$

We define the $W^T \subseteq W$ as the set of consistent, and $W^T \backslash W$ as the set of inconsistent possible worlds w.r.t. a hierarchy T. For the consistent worlds $w_i \in W^T$, we can recompute their probabilities taking into account that $P(w_j) = 0$, $\forall w_j \in W \backslash W^T$ (c.f. Constraint 3), which we define as $P_T(w_i)$:

$$P_T(w_i) = \frac{P(w_i)}{\sum_{w_j \in W^T} P(w_j)} \tag{5.5}$$

Given Constraint 2, we know that the supports of X and $X - \hat{x}$ are exactly the same, therefore we can apply *redundant itemset pruning* (c.f. Lemma 1 from [6]). Also, note that in the attribute uncertainty model, Constraint 2 is always violated if Equation (2.1) is employed to compute $P(X \in t)$, unless $P(\hat{x} \in t) = 1$. This problem can be solved if redundant itemset pruning is applied. Since of X is redundant, its pruned an the PFIM, and its support is not computed with Equation (2.1), but inferred to be equal to that of $X - \hat{x}$.

TABLE III: Dependence Table from T_2 (c.f. Table II) with Possible Worlds $w_i \in W^T$ Probabilities from Fig. 1

	Bass Guitar	Acoustic Guitar	Electric Guitar	Guitar	String Instrument	$P(w_i)$	$PT(w_i)$
w_1	0	0	0	0	0	0.009	0.011
w_2	0	0	0	0	1	0.009	0.011
w_3	0	0	0	1	1	0.036	0.044
w_4	0	0	1	1	1	0.004	0.005
w_5	0	1	0	1	1	0.036	0.044
w_6	0	1	1	1	1	0.004	0.005
w_7	1	0	0	1	1	0.324	0.396
w_8	1	0	1	1	1	0.036	0.044
w_9	1	1	0	1	1	0.324	0.396
w_{10}	1	1	1	1	1	0.036	0.044

A. Dependence Table

The dependencies implied by a background taxonomy can be represented with a joint probability distribution table. Such table can be created by applying 2.4 and 5.5 to compute the probabilities of all possible worlds. Table III shows the resulting dependence table for the transaction t_2 from Table II with T from Fig. 1 as background knowledge. Note that $\forall w_i \in W \backslash W^T, P(w_i) = 0$, since w_i is inconsistent w.r.t. T. Therefore the inconsistent possible worlds are actually impossible and do not need to be represented in the dependence table.

The existential probabilities of the itemsets are calculated by simply summing up the probabilities of consistent possible worlds in each a given itemset occurs, as shown in Equation (5.6). Table IV shows the resulting existential probability values $P_T(x \in t)$ of the singletons calculated with Equation (5.6) on Table III. Note that the resulting transaction t_2^T is now consistent w.r.t. T, and the items probabilities got reinforced or weakened by the probabilities of the other items in the hierarchy.

$$P_T(X \sqsubseteq t) = \sum_{w_i \in W^T \mid X \subseteq t} P_T(w_i) \tag{5.6}$$

Note that in order to precisely represent the dependencies, we need to generate a dependence table for each uncertain transaction in the data, and the size of the table grows exponentially with the number of uncertain items in the transaction.

TABLE IV: TRANSACTION t_2^T RESULTED FROM TABLE III

	Bass Guitar	Acoustic Guitar	Electric Guitar	Guitar	String Instrument
t_2^T	0.880	0.489	0.098	0.978	0.989

VI. APPROXIMATION APPROACHES

The calculation of the dependencies implied by the background taxonomies can improve the quality of the data, however, creating the whole dependence table for each transaction in order to calculate the exact P_T values is expensive and does not scale. Therefore, in this paper we propose some scalable approximations of the exact dependencies computation.

The proposed approximation approaches consist of computing an approximation of the singletons existential probabilities considering the dependencies implied by the taxonomy. In order to better approximate composite itemsets existential probabilities, the conditional probabilities inherent to the hierarchical structure of the taxonomy can be exploited. However, it requires the approximated singleton probabilities to be consistent with the background taxonomy, as we will discuss in details in Section VI-A.

A. Stratified Computation of Singleton Probabilities

To approximate the singleton marginal probabilities from the dependence table, we propose a stratified approach. In order to reduce the complexity of the dependencies computation, the whole taxonomy is broken into smaller and simpler subsets. This is done by dividing it into non-disjoint strata of two levels. That is, a taxonomy of depth ℓ+1 and levels $\{l_0, \dots, l_\ell\}$, is broken into ℓ strata $\{s_o, \dots, s_{\ell-1}\}$, where each stratum $s_i = \{l_i, l_{i+1}\}$. The dependencies are then computed individually for each stratum, in a top-down or bottom-up manner. Since the all the items in the taxonomy, excluding the root and leaves, belong to two strata, they propagate the updates of the dependencies computation from one stratum to the next.

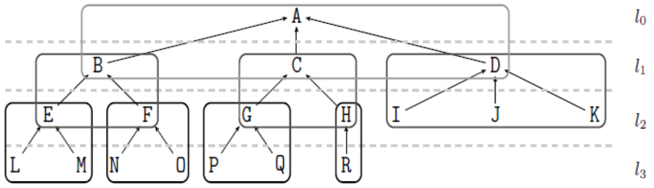

Fig. 2. Example of stratified computation of dependencies.

Each stratum is composed by one or more disjoint subtrees of the taxonomy with depth 2 which are assumed to be independent of each other. Each of these subtrees are composed by one itemˆx and its direct specializations $x_i \sqsubseteq \hat{x}$. For a single subtree the exact dependencies computation is performed, however, we do not need to create the dependence table. Instead, we can directly compute the probability of the parent $P_T(\hat{x} \in t)$ and the children $P_T(x \in t)$ with Equations 6.8, 6.9, 6.10.

$$P(t = \emptyset) = \big(1 - P(\hat{x} \in t)\big) \prod_{\{x \mid x \sqsubseteq \hat{x}\}} 1 - P(x \in t) \tag{6.7}$$

$$P(T \vDash_{cons} t) = P(\hat{x} \in t) + P(t = \emptyset) \tag{6.8}$$

$$P_T(x \in t) = \frac{P(x \in t) P(\hat{x} \in t)}{P(T \vDash_{cons} t)} \tag{6.9}$$

$$P_T(\hat{x} \in t) = \frac{P(\hat{x} \in t)}{P(T \vDash_{cons} t)} \tag{6.10}$$

Fig. 2 shows an example of transaction with three strata $s_0 = \{l_o, l_1\}$, $s_1 = \{l_1, l_2\}$ and $s_2 = \{l_2, l_3\}$. The exact dependencies computation is performed for each subtree (identified by the rectangles in Fig. 2) stratum by stratum in top-down (s_0, s_1, s_2) or bottom-up (s_2, s_1, s_0) fashion. All the subtrees in a given stratum are assuned to be mutually independent, therefore the order in which the subtrees of the stratum have their dependencies computed does not matter.

One problem of the stratified approximation approach is that it does not guarantee consistency. From Equations (6.9) and 6.10, we can infer that for a subtree, the inequations $P_T(x \in t) \leq P(x \in t)$ and $P_T(\hat{x} \in t) \leq P(\hat{x} \in t)$ always hold. That means that applying the stratified dependence computation might result in an inconsistent transaction because, since all the non-root and non-leave nodes belong to two strata, the consistency of in one stratum can be disrupted by the dependence computation in the next stratum.

The transaction t from Table V is an example of transaction where the stratified approximation results in an inconsistent transaction. The result of the bottom-up approach t_{sbu}^1 is inconsistent because P(*StringInstrument*∈ t_{sbu}^1) < P(*BassGuitar*∈ t_{sbu}^1). That happens because although P(*Guitar*∈ t_{sbu}^1)=0.731 after computing the dependencies of the first stratum, its existential probability is reduced to 0.343 after the next stratum because of the low probability of its superclass P(*StringInstrument* ∈ t_{sbu}^1)=0.2.

B. Approximation of Composite Itemsets Probabilities

We assume that all the sibling nodes are conditionally mutually independent given their parent. This assumption can be exploited when computing the existential probability of composite itemsets. The probability of the union of an itemset X with an item $i \notin X$, is calculated with Equation (6.11), where i_{cp} is the least general generalization of items in X and i, including their respective generalizations.

$$P(X \cup \{i\} \subseteq t) = \frac{P(X \subseteq t) P(i \in t)}{P(i_{cp} \in t)} \tag{6.11}$$

If we want to calculate the probability of { Flute, Harmonica}, for instance, i_{cp} is Woodwind. Now if we want to compute the probability of {*Flute*, *Harmonica*, *BassGuitar*}, where X={*Flute*, *Harmonica*} and i= *BassGuitar*, then i_{cp} =*MusicalInstrument*.

The use of conditional probabilities for computing the support of composite itemsets can improve the accuracy of the approximation for composite itemsets. However, it has the additional cost of checking the taxonomy structure in order to compute the i_{cp} for every composite itemset. This involves obtaining all the generalizations of the items, computing their intersection and finding its least general item. Moreover, it requires the transaction to be consistent, which is not guaranteed to happen if the stratified computation approach from Section VI-A is used. If Equation (6.11) is applied on an inconsistent transaction, the anti monotonicity of support is violated when $P(i_{cp} \in t) < P(i \in t)$, and it might also happen that the resulting existential probability is greater than 1.

C. Iterative Stratified Computation

As seen it Section VI-A, the stratified approach does not guarantee convergence. However, if applied iteratively on a transaction, the stratified approximation will gradually approach a convergence point which is always consistent.

This convergence is illustrated in Table V, where t_{sbu}^{i} shows the resulting transaction after iteratively applying the bottom-up stratified approximation i time, and t_{sbu}^{∞} shows the consistent transaction to which the iterative bottom-up approach converges. We define as converged iterative stratified bottom-up, and top-down approaches (*csbu* and *cstd* respectively) as the iterative application of the stratified approximations (*sbu* and *std*) until the transaction converges. The convergence stop criterion is defined by as $d(t_{sbu}^{i+1}, t_{sbu}^{i}) < d_{conv}$ for *csbu*, and $d(t_{std}^{i+1}, t_{std}^{i}) <$ for *std*, where $d(x, y)$ is the euclidean distance between two uncertain transactions x and y, and d_{conv} is the convergence threshold. We define $t_{csbu} = t_{sbu}^{i}$ and $t_{cstd} = t_{std}^{i}$, where the stopping criterion is satisfied for *i*. Both iterative approaches also feature the approximation of composite itemsets probabilities presented in the last subsection.

TABLE V: STEPS OF A BOTTOM-UP STRATIFIED DEPENDENCE COMPUTATION

	Musical Instrument	String Instrument	Guitar	Bass Guitar
t	0.95	0.2	0.55	0.55
t_{exa}	0.973	0.469	0.343	0.188
t_{sbu}^{1}	0.973	0.469	0.352	0.402
t_{sbu}^{2}	0.99	0.621	0.299	0.191
t_{sbu}^{3}	0.997	0.712	0.246	0.066
t_{sbu}^{4}	0.999	0.769	0.2	0.017
t_{sbu}^{5}	1	0.965	0	0

TABLE VI: COMPARISON OF A TRANSACTION T, WITH ITS EXACT REASONED (T_{EXA}) AND DIFFERENT APPROXIMATIONS

	Musical Instr.	Wood- wind	String Instr.	Harmo- nica	Flute	Guitar	Harp	Bass Guitar	Acoustic Guitar	Electric Guitar
t	0.6	0.2	0.7	0.5	0.7	0.1	0.2	0.2	0.2	0.1
t_{exa}	0.947	0.592	0.736	0.296	0.414	0.119	0.147	0.024	0.024	0.012
t_{std}	0.862	0.581	0.679	0.291	0.407	0.112	0.136	0.022	0.022	0.011
t_{sbu}	0.947	0.592	0.736	0.312	0.437	0.126	0.155	0.032	0.032	0.016
t_{csbu}	1	1	1	0.184	0.257	0.008	0.009	0	0	0
t_{cstd}	1	1	0.876	0.141	0.197	0	0	0	0	0
$t_{pet\,13}$	0.931	0.85	0.539	0.5	0.7	0.424	0.2	0.2	0.2	0.1

VII. EXPERIMENTS

We divide the evaluation of the proposed approximations of dependencies computation into two parts. In the first part we measure the distance between the resulting transactions of the exact and the approximated dependencies computations as well as their runtimes. With that we can evaluate the accuracy and the cost of each of the approximations. In the second part we evaluate how the approximations can improve the quality of a noisy probabilistic dataset. We apply generalized FIM on a certain dataset with a taxonomy as background knowledge, and use the resulting frequent itemsets as gold standard. We then add noise to the dataset to generate the uncertain data and apply PFIM with the various proposed approaches. Finally we compare the resulting probabilistic frequent itemsets with the gold standard by calculating the precision, recall and F-measure.

The methods compared in the experiments and their abbreviations are listed in Table VII. Table VI shows a comparison which illustrates the results of the different dependence computation approaches presented in this paper on an uncertain transaction *t*. The *pet*13[13] approach requires the existential probabilities of non-leaf nodes in the taxonomy to be ignored, as explained in Section 4. We also compare with a baseline method which ignores the taxonomical background knowledge (*nbk*) and another which employs only redundant itemset pruning (*rip*).

A. Datasets

We report results for experiments in two datasets. One is transactional data extracted from DBpedia 3.8[2] for the statistical schema induction [3], with the original ontology's class subsumption hierarchy as background knowledge. Every transaction in the dataset corresponds to a DBpedia instance, and the items correspond to classes and properties assigned to the instances. We also use the dataset T10I4D100K from the Frequent Itemset Mining Dataset Repository[3]. This dataset has 1000 items and no background knowledge, therefore we synthesize a hierarchical taxonomy where the original 1000 items are leaf nodes, and the taxonomy structure is generated based on the fanout parameter, which determines the number of specializations an item in the taxonomy should have. The number of levels ℓ in a taxonomy with n leaf nodes and fanout ϕ is $\ell = \lceil \log_{\phi} n + 1 \rceil$, thus the greater the fanout the shallower, and

[2] http://wiki.dbpedia.org/Downloads38
[3] http://fimi.ua.ac.be/data/

the lower the fanout the deeper the synthesized taxonomy is. The uncertain datasets used in the experiments are generated from certain data as follows: items contained in a given certain transaction are assigned an existential probability drawn from a normal distribution $N(\mu_1, \sigma_1)$, and the items not contained are chosen with probability p_0 to be assigned an existential probability $N(\mu_0, \sigma_0)$. Uncertainty values are set to 1 or 0 when the value drawn from the normal distributions are greater than 1 or less than 0 respectively. For our experiments, we define the level of noise in the data with a variable x, which defines $\mu_1 = (1 - x)$, $\mu_0 = x$ and$\sigma_1 = \sigma_0 = x$. That means the greater the x the noisier the the generated uncertain dataset is.

TABLE VII: Abbreviations of the Compared Methods

	Method
exa	Exact Dependency Computation
nbk	No Dependency and No Redundant Itemsets Pruning
rip	Redundant Itemsets Pruning Only
std	Stratified Top-down
sbu	Stratified Bottom-up
cstd	Converged Iterative Stratified Top-down
csbu	Converged Iterative Stratified Bottom-up
*pet*13	Peterson et at. 2013 [13]

B. Approximation Quality

In this experiment we use the dataset T10I4D100K with $x = 0.15$ and $p_0 = 0.1$. Since in this experiment we need to do the exact dependencies computation, whose runtime grows exponentially with the number of uncertain attributes, we limit the number of items in the transaction to k. In case the size of a transaction exceeds k, we keep the top-k most general items and remove the rest. For each transactions of the dataset, we measure the euclidean distance between each approximation and the exact computation. We also measure the average runtime of the exact and approximated dependencies computation.

Fig. 3 shows how the runtime and distance of the proposed approximations is affected by the taxonomy fanout, number of items per transaction, and level of noise. The results reveal that the bottom-up approaches are overall better than the top-down, being less sensible to noise and more accurate on deeper taxonomies. Also, the iterative approaches significantly increase the runtime, and also increases the distance to *exa*. This is because repeatedly applying the stratified approach ends up exaggerating the effects of the dependence computation and therefore increasing the distance to *exa*. Overall *sbu* is the best performer with very low distance to the exact computation, low runtime, and high robustness.

C. PFIM Performance

In this experiment we evaluate how the dependence approximations can improve the PFIM task. In order to do so, we use the certain dataset T10I4D100K with a synthesized taxonomy, and apply a generalized FIM algorithm to obtain the set of frequent itemsets, which will be used as gold standard. It is important to mention that redundant frequent itemsets are not contained in the gold standard because of the redundant itemset pruning. Afterwards, we add noise to the data, as described earlier in this section, and perform PFIM using all the approaches listed in Table VII. For each approach we measure the runtime, and we compare the resulting set of probabilist frequent itemsets with the gold standard in order to calculate the precision, recall and F-measure.

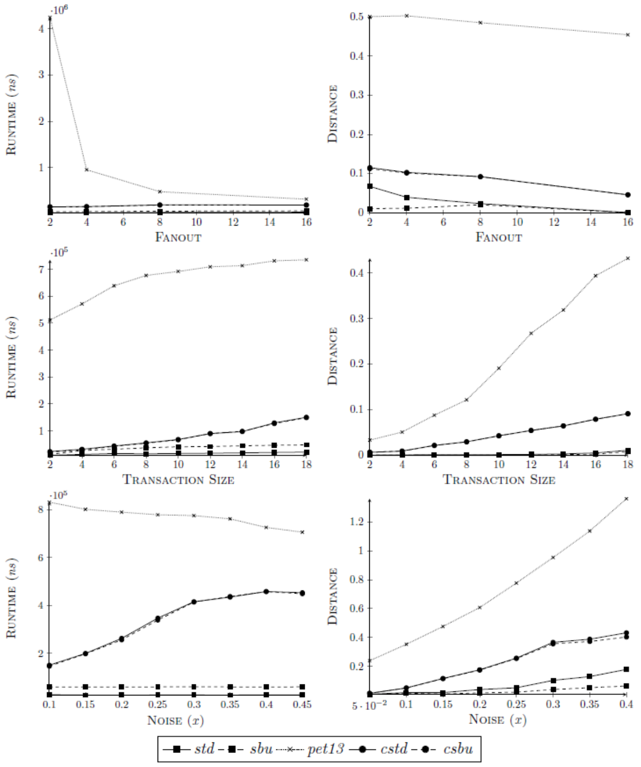

Fig. 3. Approximations runtime and euclidean distance from the approximations to the exact computation.

We use the p-Apriori [14] and U-Eclat [10] as PFIM algorithms, and we run them with all the approximation approaches listed in Table VII. Experiments were done also with the exact *spmf* calculation [7], the normal distribution [10] and expected support [8] approximations. Neither the PFIM algorithm nor the *spmf* approximation choice affected the results. Therefore, for simplicity, we choose to report in the plots the results for U-Eclat with normal distribution model only. The compared approaches include all the methods listed in Table VI with the exception of *exa*, which could not be run due to time restrictions. The plots in Fig. 4 report how the runtime and F-measure are affected by the taxonomy fanout, transaction size, noise level,and minimum support threshold.

The results shown in Fig. 4 indicate that *csbu* has best F-measure overall. The *cstd* approach has an almost similar performance, however *csbu* is better on more noisy data. Both iterative approaches have a consistently higher F-measure than the single iteration stratified approaches *sbu* and *std*. It is also noticeable that the bottom-up approaches are more robust to noise than the top-down approaches. The improved F-measure of the proposed approximations in comparison to *rip* show the impact of the noise reduction resulted from the dependencies computations. All the proposed approaches also had better F-measure than *pet*13, whose low F-measure values are due to low precision. Since *pet*13 only considers the uncertain values at leaf nodes of the taxonomy, it tends to incorporate noise to the higher levels and increase the support of the itemsets. This results in a high recall, but also high number of false positives and therefore low precision.

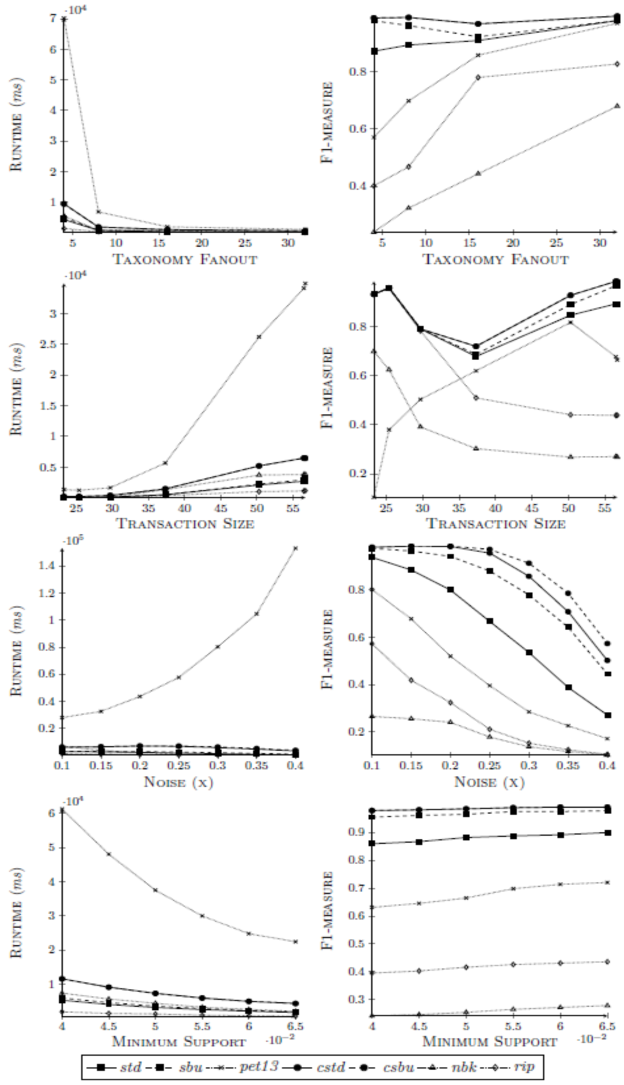

Fig. 4. Comparison of PFIM runtime and F-measure.

TABLE VIII: EXPERIMENTS ON DBPEDIA 3.8 SCHEMA INDUCTION

	Runtime (s)	Precision	Recall	F_1-measure
nbk	66753	0.1699	0.2222	0.1925
rip	**17433**	**1.0**	0.2222	0.3636
std	27258	**1.0**	0.3463	0.5145
sbu	26735	0.9977	0.3554	0.5242
cstd	67285	0.9952	0.3321	0.4980
csbu	74013	0.9956	**0.3665**	**0.5357**
*pet*13	104399	0.8944	0.2641	0.4078

When analyzing the runtime, *rip* has the shortest runtime overall, as it makes use of the background knowledge in order to speed up the mining by pruning redundancies, and it does not perform any dependency computations. The stratified approaches *sbu* and *std* have a significantly longer runtime. This happens especially because of the increased support of itemsets after the noise reduction and consequent increased number of candidate itemsets. The time spent with the dependencies computation accounts on average for less than 10% of the total runtime. For the iterative approaches, the cost of the additional iterations and the approximation of the existential probabilities of composite itemsets significantly increase the runtime.

Finally, we perform an experiment on a schema induction table [3] from DBpedia 3.8 with $x = 0.5$, $p_0 = 0.05$, where we use the its ontology class subsumption hierarchy as background knowledge. The results are shown in Table VIII, and for every measure the best performer is shown in bold. Note that all the proposed approximations (*std*,*sbu*,*csbu*,*cstd*) significantly improved the recall in comparison to *rip* without compromising the precision. The *csbu* approximation is able to improve *sbu*, however, it takes more than twice the runtime and the F-measure gain is small.

VIII. CONCLUSION

In this paper we proposed scalable techniques which approximate dependencies from a taxonomical background knowledge in uncertain data modeled with attribute uncertainty. The proposed approximations, and in particular the bottom-up approaches can accurately approximate the exact computation. Applying these approximations on PFIM can improve the results quality without significantly affecting its scalability. The experiments indicate that the iterative bottom-up stratified approach *csbu* is the best overall performer, however, the extra iterations and the use of conditional probabilities from the background knowledge to compute the support of composite itemsets increase the runtime. In the future we plan to investigate approximation methods for other types of dependencies such as disjointness, and apply these methods on a large scale for Linked Data.

ACKNOWLEDGMENT

The work reported in this paper was partially funded by the DFG project GOLD[4]. Johanna Völker is supported by a Margarete-von-Wrangell scholarship of the European Social Fund (ESF) and the Ministry of Science, Research and the Arts Baden-Württemberg.

REFERENCES

[1] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek, "Amie: Association rule mining under incomplete evidence in ontological knowledge bases," in *Proc. the 22Nd International Conference on World Wide Web*, pp. 413-422, May 2013.

[2] Z. Abedjan and F. Naumann, "Improving rdf data through association rule mining," *Datenbank-Spektrum*, 2013, pp. 111–120.

[3] J. Völker and M. Niepert, "Statistical schema induction," in *Proc. the 8th Extended Semantic Web Conference on The Semantic Web*, Springer-Verlag, pp. 124-138, June 2011.

[4] J. Lehmann, D. Gerber, M. Morsey, and A. Ngomo, "Defacto deep fact validation," in *Proc. International Semantic Web Conference*, September 2012, vol. 7649, Springer, pp. 312-327.

[5] A. Rettinger, U. Lösch, V. Tresp, C. D'Amato, and N. Fanizzi, "Mining the semantic web," *Data Min. Knowl. Discov*., vol. 24, no. 3, pp. 613-662, May 2012.

[6] R. Srikant and R. Agrawal, "Mining generalized association rules," in *VLDB*, U. Dayal, P. M. D. Gray, and S. Nishio, Eds., Morgan Kaufmann, September 1995, pp. 407–419.

[7] T. Bernecker, H.-P. Kriegel, M. Renz, F. Verhein, and A. Züfle, "Probabilistic frequent itemset mining in uncertain databases," in *Proc. the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, July 2009, pp. 119-128.

[8] C.-K. Chui, B. Kao, and E. Hung, "Mining frequent itemsets from uncertain data," in *Proc. the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD, Springer-Verlag, pp. 47-58, June 2007.

[9] L. Wang, R. Cheng, S. D. Lee, and D. W.-L. Cheung, "Accelerating probabilistic frequent itemset mining: A model-based approach," in *CIKM*, J. Huang, N. Koudas, G. J. F. Jones, X. Wu, K.

[4] http://gold.linkeddata.org

Collins-Thompson, and A. An, Eds., ACM, pp. 429-438, November 2010.

[10] T. Calders, C. Garboni, and B. Goethals, "Efficient pattern mining of uncertain data with sampling," *PAKDD*, vol. 6118, Springer, pp. 480-487, June 2010.

[11] T. Bernecker, R. Cheng, D. W. Cheung, H. P. Kriegel, S. D. Lee, M. Renz, F. Verhein, L. Wang, and A. Züfle, "Model-based probabilistic frequent itemset mining," *Knowl. Inf. Syst.*, vol. 37, London: Springer-Verlang, 2013, pp. 181-217.

[12] J. Han and Y. Fu, "Discovery of multiple-level association rules from large databases," *VLDB*, pp. 420-431, September 1995.

[13] E. A. Peterson and P. Tang, "Mining probabilistic generalized frequent itemsets in uncertain databases," in *Proc. ACM Southeast Regional Conference*, ACM, April 2013, p. 1.

[14] L. Sun, R. Cheng, D. W. Cheung, and J. Cheng, "Mining uncertain data with probabilistic guarantees," in *KDD*, ACM, pp. 273–282, July 2010.

André Melo is currently a PhD student at the Data and Web Science Group of the University of Mannheim in Germany. He received his B.Sc. in computer science at the Fluminense Federal University in Brazil and his M.Sc. in computer science from the Saarland University with the master thesis "Learning rules with numerical attributes from linked data". He is interested in data mining, natural language processing, and semantic web.

Johanna Völker currently works as a junior professor in the data and web science group at the University of Mannheim. Since 2003, she holds a diploma in computer science with a special focus on computational linguistics from the Saarland University. In 2008, she received a PhD in applied computer science from the Karlsruhe Institute of Technology.