# Aggregate Queries on Probabilistic Record Linkages *

Ming Hua
Facebook Inc.
Menlo Park, CA, USA
arceehua@fb.com

Jian Pei
Simon Fraser University, Canada
Burnaby, BC, Canada
jpei@cs.sfu.ca

## ABSTRACT

Record linkage analysis, which matches records referring to the same real world entities from different data sets, is an important task in data integration. Uncertainty often exists in record linkages due to incompleteness or ambiguity in data. Fortunately, the state-of-the-art probabilistic record linkage methods are capable of computing the probability that two records referring to the same entity.

In this paper, we study the novel aggregate queries on probabilistic record linkages, such as counting the number of matched records. We address several fundamental issues. First, we advocate that the answer to an aggregate query on probabilistic record linkages is a probability distribution of possible answers derived from possible worlds. Second, we identify the category of compatible linkages only on which the answers to aggregate queries can be determined properly when the probabilities of individual linkages are available but the joint distributions of multiple linkages are unavailable. Third, we give a quadratic exact algorithm and two approximation algorithms to answer aggregate queries.

## 1. INTRODUCTION

Record linkages are the linkages among data entries in different data sets referring to the same real-world entities. Building record linkages is an important data integration task in many applications, such as health-care information systems and customer information systems.

In real applications, data is often incomplete or ambiguous. Consequently, record linkages are often uncertain. Probabilistic record linkages are often used to model the uncertainty. For two records, a state-of-the-art probabilistic record linkage method [17] can estimate the probability that the two records refer to the same real-world entity. Often, two thresholds $\delta_M$ and $\delta_U$ ($0 \leq \delta_U < \delta_M \leq 1$) are used: the records are considered not-matched, possibly matched, and matched, respectively, when the linkage probability is less than $\delta_U$, between $\delta_U$ and $\delta_M$, and over $\delta_M$.

While many previous studies focus on building probabilistic record linkages effectively and efficiently, can we answer aggregate queries on probabilistic record linkages?

Example 1 (Aggregate queries). Survival-after-hospitalization is an important measure used in public medical service analysis. To obtain the statistics about the death population after hospitalization, Svartbo *et al.* [33] studied survival-after-hospitalization by linking two real data sets, the hospitalization registers and the national causes-of-death registers in some counties in Sweden.

To elaborate, consider some synthesized records in the two data sets as shown in Figure 1. The column linkage probability $P$ is calculated by a probability record linkage method. In order to obtain the survival-after-hospitalization statistics, we need to count the number of linkages between the hospitalization registers and the causes-of-death registers, which is the death population after hospitalization.

Suppose $\delta_M = 0.75$ and $\delta_U = 0.45$. No records in Table 1 are considered matched, since the linkage probabilities are all lower than $\delta_U$. Is the count of linkages simply 0?

Record $a_1$ in the hospitalization register data set is linked to three records in the causes-of-death register data set, namely $b_1$, $b_2$ and $b_3$, with linkage probability 0.2, 0.3 and 0.4, respectively. Therefore, the probability that "John H. Smith" is linked to some records in the causes-of-death register data set and thus reported dead is $0.2+0.3+0.4 = 0.9$. Similarly, the probability that "Johnson R. Smith" is reported dead is 0.8. Therefore, there is a high probability that the count of linkages is at least 1.

Moreover, the count of linkages is 2 if both "Johnson R. Smith" and "John H. Smith" are reported dead. One may think that the probability is $0.9 \times 0.8 = 0.72$. However, it is incorrect since records $a_1$ and $a_2$ cannot be linked to record $b_3$ at the same time. Due to the linkages involving the same records in one table, the calculation of probability is far from trivial, which will be discussed in Section 3.

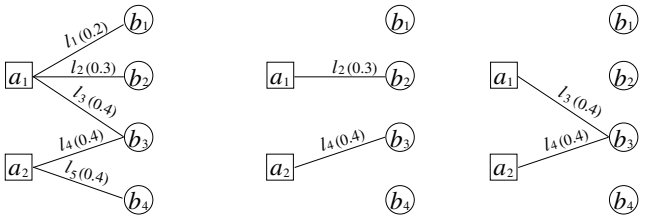
In general, aggregate queries on linkages can use any aggregate functions. For example, if the hospitalization register data set contains the date of discharge for each patient, and the causes-of-death registers data set contains the date of death, then one may ask an aggregate query about the

*The research reported in this paper is part of Ming Hua's Ph.D. thesis (see the monograph version [19]). This research is supported in part by an NSERC Discovery grant, an NSERC Discovery Accelerator Supplement grant, and a BCFRST Foundation NRAS Endowment Research Team Program grant. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.


*EDBT 2012*, March 26–30, 2012, Berlin, Germany.

| LId | hospitalization registers | | causes-of-death registers | | $P$ |
|---|---|---|---|---|---|
| | Id | Name | Id | Name | |
| $l_1$ | $a_1$ | John H. Smith | $b_1$ | Johnny Smith | 0.2 |
| $l_2$ | $a_1$ | John H. Smith | $b_2$ | John Smith | 0.3 |
| $l_3$ | $a_1$ | John H. Smith | $b_3$ | J. Smith | 0.4 |
| $l_4$ | $a_2$ | Johnson R. Smith | $b_3$ | J. Smith | 0.4 |
| $l_5$ | $a_2$ | Johnson R. Smith | $b_4$ | J. R. Smith | 0.4 |

(a) A set of linkages.




(b) Bipartite graph representation. (c) A valid world. (d) An invalid world.

**Figure 1: Record linkages between the hospitalization registers and the causes-of-death registers.**

average survival time of the death population. ∎

Example 1 illustrates two important points. First, aggregate queries on probabilistic record linkages are interesting and useful in record linkage analysis. Second, answering such aggregate queries is far from trivial. The dependencies of linkages can be very complicated.

In this paper, we tackle the problem of aggregate queries on probabilistic record linkages. We make solid contributions by addressing the following interesting and fundamental challenges.

First, *what do aggregate queries mean on probabilistic record linkages?* The answer to an aggregate query on a set of probabilistic record linkages is a probability distribution summarizing the answers in all scenarios that the records are linked. We advocate the histogram summary as a way to summarize the answer distribution due to the simplicity and popularity of histograms in the database and data analysis community.

Second, *as the existing probabilistic linkage methods give only the linkage probabilities of pairs of tuples but not the joint distribution of multiple linkages, on what kind of linkages can possible world probabilities be defined properly?* Here, a possible world is a possible scenario that the records are linked. Thus, the probabilities of possible worlds play a critical role in deriving the answer distribution for an aggregate query on probabilistic linkages. We identify the category of compatible linkages only on which the answers to aggregate queries can be determined properly.

Third, *how should we model the dependencies among probabilistic record linkages?* We develop a notion of probabilistic mutual exclusion graph (PME-graph for short), which is a specific type of Markov networks and can be used to derive the joint distribution of a set of probabilistic linkages.

Fourth, *how can we answer aggregate queries efficiently?* A straightforward approach is to enumerate all possible worlds, calculate the answer to a query in each possible world, and summarize the results. Obviously, the straightforward method can be very costly. In this paper, we give a quadratic exact algorithm by inference on PME-graphs.

We present an extensive empirical study on both real and synthetic data sets demonstrating the effectiveness of the queries and the efficiency of the query answering methods.

The rest of the paper is organized as follows. In Section 2, we formulate aggregate queries on probabilistic record linkages. In Section 3, we develop a notion of probabilistic mutual exclusion graph and discuss the compatibility linkages. Section 4 reviews related work. An exact method for aggregate `count` is developed in Section 6. Section 7 reports an extensive empirical study. Section 8 concludes the paper.

We provide the proofs of mathematical results in the appendixes.

# 2. PROBLEM STATEMENT

In this section, we review the preliminaries and formulate aggregate queries on probabilistic record linkages.

## 2.1 Probabilistic Record Linkages

Let $\mathcal{E}$ be a set of real-world entities. We consider two tables $A$ and $B$ which describe subsets $\mathcal{E}_A, \mathcal{E}_B \subseteq \mathcal{E}$, respectively. In general, $\mathcal{E}_A$ and $\mathcal{E}_B$ may not be identical. Tables $A$ and $B$ may have different schemas as well.

DEFINITION 1 (LINKAGE FUNCTION). *A **probabilistic record linkage method** (or **linkage function** for short) is a function $\mathcal{L} : A \times B \to [0, 1]$ such that, for tuples $t_A \in A$ and $t_B \in B$, $\mathcal{L}(t_A, t_B)$ scores the likelihood that $t_A$ and $t_B$ describe the same entity in $\mathcal{E}$. The larger the score, the more likely the two tuples describe the same entity. A pair of tuples $l = (t_A, t_B)$ is called a **probabilistic record linkage** (or **linkage** for short) if $\mathcal{L}(l) > 0$.* ∎

Any binary classifier can be employed to compute the score $\mathcal{L}(t_A, t_B)$ by classifying the corresponding feature vectors of $t_A$ and $t_B$ and treating the classification confidence as $\mathcal{L}(t_A, t_B)$ [3]. A number of classifiers have been adopted to compute the linkage function, including Naïve Bayes [35], decision trees [34], and Support Vector Machines [2].

A tuple $t_A \in A$ ($t_B \in B$) may participate in zero, one or multiple linkages. The number of linkages that $t_A$ ($t_B$) participates in is called the **degree** of $t_A$ ($t_B$), denoted by $degree(t_A)$ ($degree(t_B)$). $(\mathcal{L}, A, B)$ specifies a bipartite graph, where the tuples in $A$ and those in $B$ are two independent sets of nodes, respectively, and the edges are the linkages between the tuples in the two tables.

In many situations, to perform effective data integration, duplicates are eliminated from the two tables $A$ and $B$ prior to computing the linkage function. Therefore, a one-to-one matching is enforced during the record linkage. The record linkage with a one-to-one matching constraint is called the constrained matching problem [10]. In the constrained matching problem, the probability that a tuple in a table is matched by some tuples in the other table is at most 1. That is, for each tuple $t_A \in A$, $\sum_{t_B \in B} \mathcal{L}(t_A, t_B) \leq 1$ and, symmetrically, for each tuple $t_B \in B$, $\sum_{t_A \in A} \mathcal{L}(t_A, t_B) \leq 1$. The linkage functions for the constrained matching problem are called the **normalized linkage function**, which have been extensively studied [10, 25]. In this paper, we consider normalized probabilistic linkage functions only.

For a tuple $t_A \in A$, let $l_1 = (t_A, t_{B_1}), \ldots, l_{degree(t_A)} = (t_A, t_{B_{degree(t_A)}})$ be the linkages that $t_A$ participates in. For each tuple $t_A \in A$, we can write a **mutual exclusion rule**

$R_{t_A} = l_1 \oplus \cdots \oplus l_{degree(t_A)}$ which indicates that at most one linkage can hold based on the assumption that each entity can be described by at most one tuple in each table. $Pr(t_A) = \sum_{i=1}^{degree(t_A)} Pr(l_i)$ is the probability that $t_A$ is matched by some tuples in $B$. Since the linkage function is normalized, $Pr(t_A) \leq 1$. We denote by $R_A = \{R_{t_A} | t_A \in A\}$ the set of mutual exclusion rules for tuples in $A$. $R_{t_B}$ for $t_B \in B$ and $R_B$ can be defined symmetrically.

## 2.2 Possible Worlds of Probabilistic Linkages

The possible worlds model [29] has been extensively used in modeling multiple probabilistic events. In general, a possible world is a combination of the probabilistic events in question. A possible world carries an **existence probability** which is the likelihood that the possible world happens in reality.

A linkage function can be regarded as the summarization of a set of possible worlds.

Definition 2 (Possible world). *For a linkage function $\mathcal{L}$ and tables $A$ and $B$, let $\mathcal{L}_{A,B}$ be the set of linkages between tuples in $A$ and $B$. A **possible world** of $\mathcal{L}_{A,B}$, denoted by $W \subseteq \mathcal{L}_{A,B}$, is a set of tuple pairs $(t_A, t_B)$ such that (1) for any mutual exclusive rule $R_{t_A}$, if $Pr(t_A) = 1$, then there exists one pair $(t_A, t_B) \in W$. Symmetrically, for any mutual exclusive rule $R_{t_B}$, if $Pr(t_B) = 1$, then there exists one pair $(t_A, t_B) \in W$; and (2) each tuple $t_A \in A$ participates in at most one pair in $W$, so does each tuple $t_B \in B$. $\mathcal{W}_{\mathcal{L},A,B}$ denotes the set of all possible worlds of $\mathcal{L}_{A,B}$.* ∎

Figure 1(c) shows a possible world of the linkages in Figure 1(b). Figure 1(d) illustrates an invalid possible world where $l_3$ and $l_4$ appear together and thus violate the mutual exclusive rule $R_{b_3}$.

We will discuss the existence probability of a possible world in Section 3.

## 2.3 Aggregate Queries

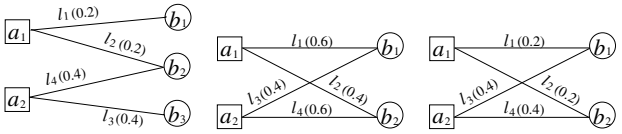
In a possible world, the answer to an aggregate query is certain. Therefore, the answer to an aggregate query on an uncertain data set is in general a multiset of the answers in the possible worlds. Moreover, each possible world is associated with an existence probability. Incorporating the probabilities, the answer to an aggregate query is a probability distribution on possible answers.

Definition 3 (Aggregate query on linkages). *Given a set $\mathcal{L}_{A,B}$ of linkages between tables $A$ and $B$, let $Q_F^P$ be an **aggregate query**, where $P$ and $F$ are a predicate and an aggregate function, respectively, which may involve attributes in $A$, $B$, or both. The **answer to $Q_F^P$ on linkages** is the probability distribution*

$$f(v) = Pr(Q_F^P(\mathcal{L}_{A,B}) = v) = \sum_{W \in \mathcal{W}_{\mathcal{L},A,B}, Q_F^P(W) = v} Pr(W),$$

*where $W$ is a possible world, $\mathcal{W}_{\mathcal{L},A,B}$ is the set of all possible worlds of $\mathcal{L}_{A,B}$, $Q_F^P(W)$ is the answer to $Q_F^P$ on the linkages in $W$, and $Pr(W)$ is the probability of $W$.* ∎

On a large set of linkages, there may be a huge number of possible worlds. Computing a probability distribution exactly is often very costly. Moreover, if there are many possible answers in the possible worlds, enumerating all of them may overwhelm a user. Since histograms are popularly adopted in data analytics and aggregate query answering, here we advocate answering aggregate queries on linkages using histograms. We consider both equi-width histograms and equi-depth histograms.



(a) Compatible. (b) Compatible. (c) Incompatible.

**Figure 2: Linkage compatibility.**

Definition 4 (Histogram answer). *Consider an aggregate query $Q$ on linkages $\mathcal{L}$, let $v_{min}$ and $v_{max}$, respectively, are the minimum and the maximum values of $Q$ on all possible worlds.*

*Given a bucket width parameter $\eta$, and a minimum probability threshold $\tau$, the **equi-width histogram answer** to $Q$ is a set of interval tuples $(\phi_i, p_i)$ $(1 \leq i \leq \lceil \frac{v_{max} - v_{min}}{\eta} \rceil)$ where $\phi_j = [v_{min} + (j-1)\eta, v_{min} + j\eta)$ $(1 \leq j < \lceil \frac{v_{max} - v_{min}}{\eta} \rceil)$ and $\phi_{\lceil \frac{v_{max} - v_{min}}{\eta} \rceil} = [v_{min} + (\lceil \frac{v_{max} - v_{min}}{\eta} \rceil - 1)\eta, v_{max}]$ are $\lceil \frac{v_{max} - v_{min}}{\eta} \rceil$ equi-width intervals between $v_{min}$ and $v_{max}$, $p_i = Pr(Q(\mathcal{L}) \in \phi_i)$. An interval pair $(\phi_i, p_i)$ is output only if $p_i \geq \tau$.*

*Given an integer $k > 0$, the **equi-depth histogram answer** to $Q$ is a set of interval tuples $(\phi_i, p_i)$ $(1 \leq i \leq k)$ where $\phi_j = [v_{j-1}, v_j)$ $(1 \leq j < k$, $v_0 = v_{min}$ and $v_j = \min\{x | Pr(Q(\mathcal{L}) \leq x) \geq \frac{j}{k}\})$ and $\phi_k = [v_{k-1}, v_{max}]$.* ∎

In the rest of this paper, we focus on computing histogram answers to aggregate queries on linkages.

# 3. LINKAGE COMPATIBILITY

The linkage functions defined in Section 2.1 give only the probabilities of individual linkages. In order to obtain the existence probabilities of possible worlds, we need to derive the joint probability distribution of all linkages given by a linkage function. Unfortunately, not every linkage function can lead to a joint distribution which is consistent with the marginal distributions of individual linkages. In this section, we identify the category of compatible linkage functions whose joint distributions can be computed from the marginal distributions of individual linkages.

## 3.1 Dependencies among Linkages

Example 2 (Compatible linkages). Consider the linkages shown in Figure 2(a), where the probabilities of the linkages are labeled. For a linkage $l$, let $l$ and $\neg l$ denote the events that $l$ appears and $l$ is absent, respectively. Since linkages $l_1$ and $l_2$ are mutually exclusive, they cannot both appear in a possible world. The marginal distribution of $(l_1 l_2)$, denoted by $f(l_1 l_2)$, is $Pr(\neg l_1 \neg l_2) = 1 - Pr(l_1) - Pr(l_2) = 0.6$, $Pr(\neg l_1 l_2) = Pr(l_2) = 0.2$, $Pr(l_1 \neg l_2) = Pr(l_1) = 0.2$, and $Pr(l_1 l_2) = 0$. Similarly, the marginal distributions $f(l_2 l_4)$ and $f(l_3 l_4)$ can be calculated from the linkage probabilities and the mutual exclusion rules.

Using Bayes' theorem, we can compute the joint distribution on $l_1$, $l_2$, $l_3$ and $l_4$. For example,

$$Pr(l_1 \neg l_2 l_3 \neg l_4) = Pr(l_1 \neg l_2) Pr(l_3 \neg l_4 | l_1 \neg l_2), \text{ and}$$

$$Pr(l_3 \neg l_4 | l_1 \neg l_2) = Pr(\neg l_4 | l_1 \neg l_2) Pr(l_3 | \neg l_4 l_1 \neg l_2).$$

Since $l_1$ and $l_4$ are conditionally independent given $l_2$, we have $Pr(\neg l_4 | l_1 \neg l_2) = Pr(\neg l_4 | \neg l_2)$. Moreover, $l_3$ and $\{l_1, l_2\}$ are conditionally independent given $l_4$, thus $Pr(l_3 | \neg l_4 l_1 \neg l_2) = Pr(l_3 | \neg l_4)$. Therefore,

$$\begin{aligned} & Pr(l_1 \neg l_2 l_3 \neg l_4) = Pr(l_1 \neg l_2) Pr(\neg l_4 | \neg l_2) Pr(l_3 | \neg l_4) \\ = \; & 0.2 \times \tfrac{1-0.2-0.4}{1-0.2} \times \tfrac{0.4}{1-0.4} = \tfrac{1}{15}. \end{aligned}$$

Figure 2(b) is another example of compatible linkages. There are only two valid assignments in the joint distribution: $l_1 \neg l_2 \neg l_3 l_4$ and $\neg l_1 l_2 l_3 \neg l_4$. The joint distribution probability should be consistent with the marginal probabilities of the linkages. Thus, the joint probabilities are $Pr(l_1 \neg l_2 \neg l_3 l_4) = 0.6$ and $Pr(\neg l_1 l_2 l_3 \neg l_4) = 0.4$.

Figure 2(c) is an example of incompatible linkages. On the one hand, using the marginal probabilities of the linkages and the mutual exclusion rules $l_1 \oplus l_2$, $l_2 \oplus l_4$ and $l_3 \oplus l_4$, similar to the case in Figure 2(b), we have $Pr(l_1 \neg l_2 l_3 \neg l_4) = \frac{1}{15}$. On the other hand, due to the mutual exclusion rule $l_1 \oplus l_3$, $Pr(l_1 l_3) = 0$. Thus, there is inconsistency. ∎

Example 2 indicates that some linkage functions may lead to a situation where the existence probability of a possible world cannot be specified in a consistent way. Formally, we introduce the notion of compatible linkages.

Definition 5 (Compatible linkages). *A set of linkages $\mathcal{L}$ are **compatible** if there is a joint distribution on $\mathcal{L}$ that satisfies the marginal distributions specified by $\mathcal{L}$.* ∎

## 3.2 Probabilistic Mutual Exclusion Graphs

We develop a probabilistic graphic model to capture dependencies among linkages.

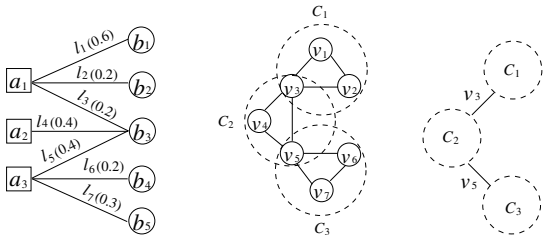
Definition 6 (PME-graph). *For a set of probabilistic linkages $\mathcal{L}_{A,B}$, the **probabilistic mutual exclusion graph** (**PME-graph** for short) $G_{\mathcal{L},A,B} = (V, E)$ is an undirected graph such that (1) a vertex $v_l \in V$ ($l \in \mathcal{L}_{A,B}$) is a binary random variable corresponding to a probabilistic linkage, $Pr(v_l = 1) = Pr(l)$ and $Pr(v_l = 0) = 1 - Pr(l)$; (2) an edge $e = (v_l, v_{l'}) \in E$ ($v_l, v_{l'} \in V$) if linkages $l$ and $l'$ share a common tuple, i.e., they are involved in a mutual exclusion rule $R_t$ ($t \in A$ or $t \in B$).* ∎

Figure 3(a) shows a set of linkages. Figure 3(b) shows the corresponding PME-graph.

For a PME-graph $G = (V, E)$, any two vertices $v_i$ and $v_j$ in $G$ have the following properties. First, $v_i$ and $v_j$ are independent if $v_i$ and $v_j$ belong to different connected components. Second, $v_i$ and $v_j$ are mutually exclusive if there is an edge $e = (v_i, v_j) \in E$. Third, $v_i$ and $v_j$ are conditionally independent given another vertex $v$ if there is a path between $v_i$ and $v_j$ passing $v$.

Theorem 1. *A PME-graph $G$ is a Markov network.* ∎

For a tuple $t \in A$ or $t \in B$, the edges corresponding to the linkages in the mutual exclusion rule $R_t$ form a maximal clique in $G$. Moreover, any two cliques in $G$ can share at most one common vertex. For example, two maximal cliques $C_1 = \{v_1, v_2, v_3\}$ and $C_2 = \{v_3, v_4, v_5\}$ in Figure 3(b) only share one common vertex $v_3$.




(a) A set of linkages $\mathcal{L}$. (b) PME-graph. (c) Clique graph.

**Figure 3: A PME-graph and a clique graph .**

PME-graphs capture the dependencies among linkages. Moreover, we can create a maximal clique graph to represent the dependencies between maximal cliques. Hereafter, for the sake of simplicity, we refer to maximal cliques as cliques.

Definition 7 (Clique graph). *For a PME-graph $G_{\mathcal{L},A,B}$, the corresponding **clique graph** is a graph $G_{clique}(V, E)$, where a vertex $v_C \in V$ corresponds to a maximal clique $C$ in $G_{\mathcal{L},A,B}$, and an edge $e_{CC'} = (v_C, v_{C'}) \in E$ if cliques $C$ and $C'$ in the PME-graph share a common vertex.*

*Let $C$ be a maximal clique in $G_{\mathcal{L},A,B}$ and $V_C$ be the set of vertices in $C$. The probability of the corresponding vertex $v_C \in V$ in the clique graph is $Pr(v_C) = \sum_{x \in V_C} Pr(x)$.* ∎

Figure 3(c) shows the clique graph corresponding to the PME-graph in Figure 3(b).

Indeed, the clique graph defined in this paper follows the classic definition of clique graph in graphic models. In addition, a clique graph derived from the PME-graph also satisfies the bipartite graph property.

In [11], it is stated that the joint probability distribution of a Markov network exists if the corresponding clique graph is chordal. Since a clique graph derived from a PME-graph satisfies the bipartite graph property, we further strengthen the statement in the following theorem.

Theorem 2 (Compatibility). *Given a set of linkages $\mathcal{L}$ and the corresponding clique graph $G_C$, the linkages in $\mathcal{L}$ are compatible if and only if, for each connected component $G' \in G_C$, either (1) $G'$ is acyclic; or (2) $G'$ is a cycle such that each vertex $v_C$ in the cycle is connected to two edges $e_1$ and $e_2$, whose corresponding vertices $v_1$ and $v_2$ in the PME-graph satisfy $Pr(v_1) + Pr(v_2) = 1$.* ∎

## 3.3 Resolving Incompatibility

Let $\mathcal{L}$ be a set of incompatible linkages, can we find a subset $\mathcal{L}' \subset \mathcal{L}$ such that $\mathcal{L}'$ is compatible and the loss of information is minimized? An intuitive and theoretically interesting measure of the information retained in $\mathcal{L}'$ is the expected number of linkages.

However, finding a subset of linkages maximizing the expected number of linkages is far from trivial. We have not obtained any theoretical results on the complexity of the problem or any approximate algorithms with quality guarantee. One feasible approach in practice is as follows. Let $G_C$ be the clique graph of a connected component of $G$ violating the conditions in Theorem 2. Let the weight of each

edge $e_v$ be the probability of $e_v$. We can find the *maximum spanning tree* [14] of $G_C$ as an approximation of the linkage subset with the maximal expected number of linkages.

## 3.4 Deriving All Possible Worlds

To enumerate all possible worlds of a set of linkages $\mathcal{L}_{A,B}$, a naïve approach is to check each subset of the linkages against Definition 2, which takes $O(2^{|\mathcal{L}_{A,B}|})$ time. However, the actual number of valid possible worlds may be much smaller than $2^{|\mathcal{L}_{A,B}|}$. For example, in Figure 3(a), there are 7 linkages but there are only 11 possible worlds.

We can use PME-graphs to generate all possible worlds in $O(|\mathcal{W}|)$ time, where $|\mathcal{W}|$ is the number of possible worlds.

A possible world $W$ of linkages $\mathcal{L}_{A,B}$ can be regarded as an assignment of values 0 and 1 to the vertices in the PME-graph $G_{\mathcal{L},A,B}$, where a vertex $v_l = 1$ if the corresponding linkage $l \in W$, otherwise $v_l = 0$. For a clique $C$ in $G_{\mathcal{L},A,B}$, if $\sum_{v \in C} Pr(v = 1) < 1$, then at most one vertex in $C$ can be assigned to 1; if $\sum_{v \in C} Pr(v = 1) = 1$, then there is exactly one vertex in $C$ taking value 1. The probability of a possible world $W$ is the joint distribution

$$Pr(W) = Pr((\wedge_{l \in W} v_l = 1) \wedge (\wedge_{l' \notin W} v_{l'} = 0)). \quad (1)$$

Since vertices in different connected components in $G_{\mathcal{L},A,B}$ are independent (Section 3.2), if $G_{\mathcal{L},A,B} = (V, E)$ contains $k$ connected components $V = V_1 \cup V_2 \cup \cdots V_k$, Equation 1 can be rewritten as

$$Pr(W) = \prod_{i=1}^{k} \Pr((\bigwedge_{l \in W \cap V_i} v_l = 1) \bigwedge (\bigwedge_{l' \notin W, l' \in V_i} v_{l'} = 0)) \quad (2)$$

# 4. RELATED WORK

Our study is mainly related to the existing work on record linkages, probabilistic data models, probabilistic graphical models and aggregate queries on uncertain data. There are three significant differences between our work and the existing studies.

First, most existing studies in the area of record linkage focus on how to compute the match probability. To the best of our knowledge, we are the first to study the case of incompatible linkages and how to utilize the results from record linkages.

Second, the PME-graph proposed in this paper is a special form of Markov network, because the clique graph derived from the PME-graph proposed in this paper is a **bipartite graph**. This bipartite property allows us to conduct inference on the graph without triangulating the graph and constructing the junction tree [23] from the clique graph.

Third, in this paper, we focus on how to efficiently compute the approximation answers to aggregate queries. The existing junction tree inference algorithm is expensive and thus cannot be applied to large-scale data sets, while our method can approximate the distribution of aggregate query results efficiently with a quality guarantee.

## 4.1 Record Linkage

Computing record linkages has been studied extensively. Koudas *et al.* [24] presented a nice tutorial. Generally, linkage methods can be partitioned into two categories. The deterministic record linkage methods [28] link two records if their values on certain matching attributes are identical. Those methods are often ineffective in real-life applications due to data incompleteness and inconsistency.

Probabilistic record linkage methods [17] estimate the likelihood of two records being a match based on some similarity measures in the matching attributes. The similarity measures used in probabilistic record linkage methods fall into three classes [24]. First, based on the Fellegi-Sunter theory [15], one can model the values of the records on the matching attributes as comparison vectors, and estimate the probability of two records being matched or unmatched [18]. Second, some "edit-based" measures such as the Levenshtein distance [26] can be used. Third, "term based" measures are proposed, where terms can be defined as words on the matching attributes or Q-grams [16].

In this paper, we focus on how to use probabilistic linkages produced by the existing probabilistic record linkage methods to answer aggregate queries in a meaningful and efficient way. To the best of our knowledge, all existing record linkage methods only return linkage probabilities on a pair of records. There are no previous studies on linkages compatibility and linkage joint distributions.

## 4.2 Probabilistic Data Models

Various models for uncertain and probabilistic data have been developed in literature. One extensively used model is the probabilistic database model [29]. Another popularly used model is the uncertain object model [8].

In [31, 12], probabilistic graphical models are used to represent correlations among probabilistic tuples. Moreover, Sen *et al.* [32] studied the problem of compact representation of correlations in probabilistic databases by exploiting the shared correlation structures.

Uncertainty in data integration is studied in [13, 30], where probabilistic schema mapping is modeled as a distribution over a set of possible mappings between two schemas.

Our probabilistic linkage model can be considered as an extension of the uncertain object model. We can consider each tuple $t_A \in A$ as an uncertain object. A tuple $t_B \in B$ can be considered as an instance of $t_A$ if there is a linkage $l = (t_A, t_B) \in \mathcal{L}$. Object $t_A$ may contain multiple instances. At the same time, an instance $t_B$ may belong to multiple objects. A mutual exclusion rule $R_{t_B} = (t_{A_1}, t_B) \oplus \cdots \oplus (t_{A_d}, t_B)$ specifies that $t_B$ can only belong to one object in a possible world.

Our study is very different from [31, 12]. In [31, 12], the joint distribution of a set of uncertain tuples is known. The objectives there are to compute the marginal probabilities of uncertain tuples based on the joint distribution. In our study, the joint distribution of probabilistic linkages is unavailable. Our goal is to compute the joint distribution and to answer aggregate queries on probabilistic linkages only based on the probabilities of individual linkages.

Last, the uncertainty considered in this paper lies in the matching of records from different data sources. This is the tuple level uncertainty in data integration, which is different from the schema level uncertainty studied in [13, 30].

## 4.3 Probabilistic Graphical Models

Probabilistic graphical models refer to graphs describing dependencies among random variables. There are two types of probabilistic graphical models: directed graphical models [21] and undirected graphical models [23] (also known as Markov networks).

In this paper, we develop PME-graphs as a specific type of undirected graphical models. We exploit the special proper-

ties of PME-graphs beyond the general undirected graphical models, and study the factorization of the joint probabilities in PME-graphs. Moreover, we develop efficient methods to evaluate aggregate queries on linkages using PME-graphs.

## 4.4 Probabilistic Aggregate Queries

Our study is also related to aggregate queries on imprecise data and probabilistic join over uncertain data.

Chen *et al.* [7] studied aggregate queries on data whose attributes may take "partial values", where a "partial value" is a set of possible values with only one being true. Ré *et al.* [27] studied the efficient evaluation of aggregate queries on probabilistic data based on Monte Carlo simulation. Burdick *et al.* [4, 5, 6] extended the OLAP model on imprecise data. The answer to an aggregation query is modeled as an answer random variable with certain probability distribution over a set of possible values. Jayram *et al.* [20] proposed several one pass streaming algorithms to estimate statistical aggregates of a probabilistic data stream.

Cheng *et al.* [9] explored the join queries over data sets with attribute level uncertainty, where the values of a tuple in the join attributes are probability distributions in a set of value intervals. Agrawal and Widom [1] studied join queries on data sets with tuple level uncertainty, where each tuple in a table is associated with a membership probability. Kimelfeld and Sagiv [22] studied the maximal join queries on probabilistic data, where only the answers whose probabilities are greater than a threshold and are not contained by any other output answers are returned.

Our study is different from the existing work on aggregate queries over uncertain data in the following two aspects. First, the application scenarios and the data models are different. In our study, the uncertainty lies in the linkages between two data sets, which brings in unique challenges in representing the dependencies among probabilistic data and leads to completely different technical solutions. Second, the summarizations of answers to aggregate queries are different. In this paper, the histogram with minimum probability threshold is used to summarize the answer distribution.

The probabilistic linkages can be considered as the join of two deterministic tables, where the matching relationship between tuples is probabilistic. In this paper, we focus on answering aggregate queries on the joined data, instead of the join methods.

# 5. OTHER AGGREGATE QUERIES

In this section, we discuss how to extend the techniques in Section 6 to answer other aggregate queries.

## 5.1 Sum and Average Queries

Given a set of linkages $\mathcal{L}$ and an attribute $\mathcal{A}$ of interest, a `sum` query and an `average` query return the distribution of the sum and the average values of all linkages in $\mathcal{L}$ (satisfying the query predicate) in $\mathcal{A}$, respectively. In Example 1, "the average survival time of the death population" is an example of `average` queries.

To evaluate `sum` and `average` queries, we use the PME-graph $G$ of $\mathcal{L}$ again. Different from processing `count` queries, for each vertex $v$, if $v$ satisfies predicate $P$, we assign $v.F = v.\mathcal{A}$, where $v.\mathcal{A}$ is the value of $v$ in attribute $\mathcal{A}$. Then, a `sum` query can be evaluated by traversing the clique tree in the similar manner as answering `count` queries. The overall complexity of computing the sum probabilities of $G$ is $O(n^2)$,

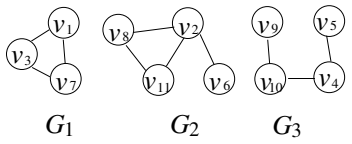



**Figure 4: Reusing the intermediate results.**

where $n$ is the number of values in the distribution of the results to the `sum` query.

An `average` query $Q^P_{\texttt{average}}$ can be evaluated by answering queries $Q^P_{\texttt{sum}}$ and $Q^P_{\texttt{count}}$. During the depth-first traverse of the clique tree, we maintain the distinct pairs of values $(x_{sum}, x_{count})$, where $x_{sum}$ and $x_{count}$ are values in the distributions of the results to $Q^P_{\texttt{sum}}$ and $Q^P_{\texttt{count}}$, respectively. Then, the `average` value distribution can be obtained straightforwardly.

## 5.2 Min and Max Queries

Given a set of linkages $\mathcal{L}$ and an attribute $\mathcal{A}$ of interest, a `min` query and a `max` query return the distributions of the minimum and the maximum values of all linkages in $\mathcal{L}$ (satisfying the query predicate) in $\mathcal{A}$, respectively. Since evaluating `min` and `max` queries are very similar, we only discuss `min` queries in this subsection.

A `min` query $Q^P_{\texttt{min}}$ can be transformed into a set of `count` queries as follows. For each vertex $v$ in the PME-graph satisfying $P$, let $x = v.\mathcal{A}$. We construct a `count` query $Q^{P\wedge P_x}_{\texttt{count}}$, where $P_x$ is a predicate such that $P_x(y)$ is `true` if $y < x$. Then, the probability that $x$ is the minimum value in $G$ is $Pr(min(G), x) = Pr(v)Pr(G, 0|v)$, where $Pr(G, 0|v)$ is the conditional count probability of $G$ with respect to query $Q^{P\wedge P_x}_{\texttt{count}}$ given $v$. We can process the vertices in the value ascending order in $\mathcal{A}$. The algorithms discussed in Section 6 can be applied to answer such a `count` query.

Two techniques can improve the efficiency. First, we can reuse the intermediate results for processing each vertex to reduce the computational cost.

Example 3 (Reusing intermediate results). Consider the PME-graph $G$ in Figure 4. Suppose the list of vertices $v_1, \ldots, v_{11}$ are sorted in the $\mathcal{A}$ value ascending order. Moreover, $x_i = v_i.\mathcal{A}$ $(1 \le i \le 11)$.

To obtain $Pr(min(G), x_8)$, let $Pr(G_i, x)$ be the probability that $x$ vertices appear in $G_i$ whose values in $\mathcal{A}$ are smaller than $x_8$ $(1 \le i \le 3)$. We have $Pr(G_1, 0) = Pr(\neg v_1 \neg v_3 \neg v_7)$, $Pr(G_2, 0|v_8) = \frac{Pr(\neg v_2 \neg v_6 v_8)}{Pr(v_8)}$, and $Pr(G_3, 0) = Pr(\neg v_4 \neg v_5)$.

To obtain $Pr(min(G), x_9)$, let $Pr'(G_i, x)$ be the probability that $x$ vertices appear in $G_i$ whose values in $\mathcal{A}$ are smaller than $x_9$ $(1 \le i \le 3)$. We have $Pr'(G_1, 0) = Pr(\neg v_1 \neg v_3 \neg v_7)$, $Pr'(G_2, 0) = Pr(\neg v_2 \neg v_6 \neg v_8)$, and $Pr'(G_3, 0|v_9) = \frac{Pr(\neg v_4 \neg v_5 v_9)}{Pr(v_9)}$.

Comparing $Pr(min(G), x_8)$ and $Pr(min(G), x_9)$, we find $Pr'(G_1, 0) = Pr(G_1, 0)$ and $Pr'(G_2, 0) = Pr(\neg v_2 \neg v_6) - Pr(G_2, 0|v_8)Pr(v_8)$. Thus, $Pr(G_2, 0|v_8)$ can be reused. Moreover, since $Pr'(G_3, 0|v_9) = \frac{Pr(G_3,0)Pr(v_9|\neg v_4 \neg v_5)}{Pr(v_9)}$, $Pr(G_3, 0)$ can be reused. ∎

Second, effective pruning techniques can avoid checking all vertices in $G$. For two vertices $v_i$ and $v_j$ where $v_i.\mathcal{A} = x_i$, $v_j.\mathcal{A} = x_j$ and $x_i < x_j$, there are a set of components in $G$ that do not contain $v_i$ and $v_j$, as discussed in Example 3.

Let $\pi = \prod_{G_k \subseteq G, v_i, v_j \notin G_k} Pr(G_k, 0)$ be the probability that, among all components not containing $v_i$ and $v_j$, there are no vertices whose values in $\mathcal{A}$ are smaller than $x_i$. Then, $Pr(min(G), x_i) \leq \pi$ and $Pr(min(G), x_j) \leq \pi$. Thus, if $\pi \leq 0$, all vertices that have not been processed can be pruned. Limited by space, we omit the details here.

# 6. ANSWERING COUNT QUERIES

In this section, we discuss evaluating `count` queries in details. Extending the algorithm to answer other aggregate queries is discussed in Section 5. For the sake of simplicity, we write a `count` query $Q^P_{\texttt{count}}$ as $Q$ in this section. To evaluate an aggregate query $Q$ on $\mathcal{L}$, we use the PME-graph $G$ of $\mathcal{L}$.

## 6.1 Preprocessing Predicate

A predicate $P$ selects the vertices in $G$ that satisfy $P$. To obtain high performance, can we remove the vertices not satisfying predicate $P$?

For a vertex $v$ not satisfying $P$, two cases arise. First, if $v$ lies in the path between two vertices satisfying $P$, then $v$ cannot be removed, since removing $v$ leads to loss of some dependency information among the two vertices satisfying $P$. Second, if $v$ does not lie in any path between vertices satisfying $P$, then $v$ can be removed without affecting the answers to the query. We assume the second types of vertices are removed when we process the query.

Therefore, two sets of vertices remain in $G$: $V_1$ contains all vertices satisfying predicate $P$; $V_2$ contains all vertices not satisfying $P$ but connecting two vertices in $V_1$. To distinguish between the vertices in $V_1$ and $V_2$, we associate a flag attribute $F$ with each vertex $v \in G$. If $P(v) = true$, then $v.F = 1$, otherwise $v.F = 0$.

## 6.2 Count Probabilities

$Q(G)$, the answer to a `count` query $Q$ on linkages $G$, is a random variable taking values from 0 to $n$, where $n$ is the number of cliques in $G$, since at most one linkage in a clique can appear in a possible world. The **count probability** $Pr(G, x)$ is the probability that there are $x$ vertices satisfying predicate $P$ appear in $G$.

$G$ may contain multiple connected components $G_1, \ldots, G_m$. The vertices in different components are independent. Therefore, we first focus on computing the count probabilities of each component $G_i$. Then, the overall count probabilities of $G$ can be computed from the convolution of the count probabilities of all components.

As discussed in Section 3, in this section, we only consider the connected component $G_i$ whose corresponding clique graph is acyclic. An acyclic clique graph has the following property.

LEMMA 1 (GRAPH PARTITION). *Given a connected component $G_i$ in a PME-graph $G$ whose clique graph $G_C$ is a tree, let $v$ be a joint vertex, then $v$ partitions $G$ into two disconnected subgraphs.* ∎

From Lemma 1, we know that the two subgraphs $G_1$ and $G_2$ partitioned by $v$ are conditionally independent given $v$. We define conditional count probabilities as follows.

DEFINITION 8 (CONDITIONAL COUNT PROBABILITY). *Given a PME-graph $G$ and a `count` query $Q^P_{\texttt{count}}$, the **conditional count probability** $Pr(G, x|v)$ is the probability that there are $x$ vertices satisfying predicate $P$ appear in $G$ given the condition that $v$ appears, that is,*

$$Pr(G, x|v) = \frac{\sum_{W \in \mathcal{W}, |\{v'|v' \in W \cap G, v'.F=1\}|=x, v \in W} Pr(W)}{\sum_{W \in \mathcal{W}, v \in W} Pr(W)}$$ ∎

Following with Lemma 1, the subgraph probability of a connected component $G_i$ can be computed using the following theorem.

THEOREM 3 (COUNT PROBABILITY). *Given a connected component $G_i$ of a PME-graph $G$, let $v$ be a joint vertex partitioning $G_i$ into two subgraphs $G_i^1$ and $G_i^2$, then*

$$Pr(G_i, x) = Pr(\neg v) \sum_{b=0}^{x} Pr(G_i^1, b|\neg v) Pr(G_i^2, x - b|\neg v)$$
$$+ Pr(v) \sum_{a=0}^{x - v.F} Pr(G_i^1, a|v) Pr(G_i^2, x - v.F - a|v)$$ ∎

Theorem 3 suggests that, in order to compute the count probability of a connected component $G_i$, we can decompose $G_i$ into smaller subgraphs and compute the count probabilities of the subgraphs, which will be pursued in Section 6.3.

## 6.3 Computing Count Probabilities

For a connected component $G_i$ of a PME-graph $G$, let $G_C$ be the corresponding acyclic clique graph of $G_i$. In order to compute the count probabilities of $G_i$, we scan $G_C$ in the depth first manner. During the scan, for two adjacent vertices $v_i$ and $v_j$, if $v_i$ is scanned before $v_j$, then we say $v_i$ is the parent of $v_j$ and $v_j$ is a child of $v_i$. A leaf vertex does not have any child.

We can apply the following **vertex compression** technique to reduce the number of vertices without affecting any count probabilities. If a clique $C$ in $G_i$ contains $m$ ($m > 1$) private vertices $v_{c_1}, \ldots, v_{c_m}$, then we can replace those vertices with a single vertex $v_p$ where $Pr(v_p) = \sum_{1 \leq i \leq m} Pr(v_{c_i})$. Moreover, for all other vertices $v \in V_C - \{v_{c_1}, \ldots, v_{c_m}\}$, an edge $(v, v_p)$ is added to $E$.

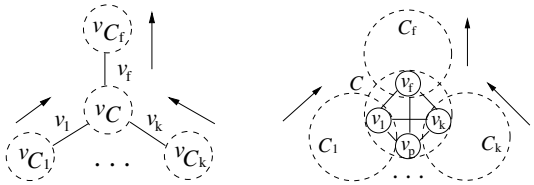
### 6.3.1 Count Probabilities of Leaf Vertices

Recall that a leaf vertex in the clique tree is corresponding to a clique that shares a common vertex with at most one clique. When we scan a leaf node $v_1$ in a clique graph, its count probability is calculated and sent to its parent vertex $v_2$. After vertex compression, the corresponding clique of $v_1$ in the PME-graph only contains two vertices: the private vertex $v_p$ and the joint vertex $v$.

THEOREM 4 (COUNT PROBABILITY OF LEAF VERTEX). *Given an acyclic clique graph $G_C$ and its leaf node $v_1$, let $v_2$ be the parent of $v_1$ and $e_v$ be the edge connecting $v_1$ and $v_2$. Denote by $C_1$ and $C_2$ the corresponding cliques of $v_1$ and $v_2$ in the PME-graph, respectively, and by $v$ the corresponding vertex of $e_v$ in the PME-graph. Then the conditional count probability of $C_1$ given $v$ is $Pr(C_1, x|v) = \begin{cases} 1, & x = v.F; \\ 0, & otherwise. \end{cases}$*

*The conditional count probability of $C_1$ given $\neg v$ is*

$$Pr(C_1, x|\neg v) = \begin{cases} \frac{1 - Pr(v_p) - Pr(v)}{1 - Pr(v)}, & x = 0; \\ \frac{Pr(v_p)}{1 - Pr(v)}, & x = 1; \\ 0, & otherwise. \end{cases}$$ ∎




(a) A clique tree. (b) A PME-graph.

**Figure 5: An intermediate node $v_C$ in the clique tree.**

### 6.3.2 Count Probabilities of Subtrees

Consider an intermediate vertex $v_C$ in the clique graph $G_C$. Denote by $v_{C_f}$ the parent of $v_C$, and by $v_{C_1}, \ldots, v_{C_k}$ the children of $v_C$. Let $T_{v_C}$ be the subtree whose root is $v_C$. We compute the count probabilities of $T_{v_C}$ by integrating the count probabilities of $T_{v_{C_1}}, \ldots, T_{v_{C_k}}$, the subtrees whose roots are children of $v_C$.

In the PME-graph, the corresponding clique $C$ of $v_C$ contains $k+1$ joint vertices: vertex $v_f$ also belonging to $C_f$ (the clique corresponding to $v_{C_f}$ in the clique graph), and vertices $v_1, \ldots, v_k$ also belonging to $C_1, \ldots, C_k$ (the cliques corresponding to $v_{C_1}, \ldots, v_{C_k}$). The clique tree and the PME-graph are illustrated in Figure 5. We want to compute the conditional count probabilities of $T_{v_C}$ given the conditions that $v_f$ appears and $v_f$ does not appear, respectively.

If $v_f$ appears, then no other vertices in $C$ appear. Then, the probability that $x$ vertices appear in $T_{v_C}$ is the probability that $x - v_f.F$ vertices appear in $T_{v_{C_1}} \cup \cdots \cup T_{v_{C_k}}$, given the condition that no vertices in $\{v_{C_1}, \ldots, v_{C_k}\}$ appears. That is

$$Pr(T_{v_C}, x|v_f) = Pr(T_{v_{C_1}} \cup \cdots \cup T_{v_{C_k}}, x - v_f.F|v_f) \quad (3)$$

In Equation 3, when $v_f$ appears, no vertices in $\{v_{C_1}, \ldots, v_{C_k}\}$ can appear. The count probabilities of subtrees $T_{v_{C_i}}$ $(1 \leq i \leq k)$ are conditionally independent. $Pr(T_{v_{C_1}} \cup \cdots \cup T_{v_{C_k}}, x - v_f.F|v_f)$ can be computed by the convolution of $Pr(T_{v_{C_i}}, x|\neg v_{C_i})$ $(1 \leq i \leq k)$. That is

$$\begin{array}{rl} & Pr(T_{v_{C_1}} \cup \cdots \cup T_{v_{C_k}}, x|v_f) \\ = & \sum_{x_1+\cdots+x_k=x} \prod_{i=1}^{k} Pr(T_{v_{C_i}}, x_i|\neg v_{C_i}) \end{array} \quad (4)$$

If $v_f$ does not appear, then the probability that another vertex $v' \in C$ appears is $Pr(v'|\neg v_f) = \frac{Pr(v')}{1-Pr(v_f)}$. The probability that $x$ vertices appear in $T_{v_C}$ is the sum of the probabilities in the following two cases.

**Case 1: the private vertex $v_p$ of $C$ appears.** Then, no vertex in $\{v_{C_1}, \ldots, v_{C_k}\}$ appears. The probability that $x$ vertices appear in $T_{v_C}$ is the probability that $x - v_p.F$ vertices appear in $T_{v_{C_1}} \cup \cdots \cup T_{v_{C_k}}$, given the condition that no vertex in $\{v_{C_1}, \ldots, v_{C_k}\}$ appears.

**Case 2: the private vertex $v_p$ of $C$ does not appear.** Then $x$ vertices in $T_{v_C}$ appears if and only if $x$ vertices appear in $T_{v_{C_1}} \cup \cdots \cup T_{v_{C_k}}$.

Summarizing the above two cases, the conditional count probability of $T_{v_C}$ given $\neg v_f$ is

$$\begin{array}{l} Pr(T_{v_C}, x|\neg v_f) \\ = Pr(v_p|\neg v_f) Pr(T_{v_{C_1}} \cup \cdots \cup T_{v_{C_k}}, x - v_p.F|v_p) \\ +Pr(\neg v_p|\neg v_f) Pr(T_{v_{C_1}} \cup \cdots \cup T_{v_{C_k}}, x|\neg v_p \neg v_f) \end{array} \quad (5)$$

In Equation 5, $Pr(T_{v_{C_1}} \cup \cdots \cup T_{v_{C_k}}, x|v_p) = Pr(T_{v_{C_1}} \cup \cdots \cup T_{v_{C_k}}, x|v_f)$, which can be computed using Equation 4. To compute $Pr(T_{v_{C_1}} \cup \cdots \cup T_{v_{C_k}}, x|\neg v_p \neg v_f)$, there are $k+1$ cases, namely, $v_{C_i}$ appears $(1 \leq i \leq k)$ and no vertices in $\{v_{C_1}, \ldots, v_{C_k}\}$ appears. In any of the $k+1$ cases, the count probabilities of $T_{v_{C_i}}$ are conditionally independent and thus can be computed using the similar convolution as in Equation 4.

To analyze the complexity of the complete procedure, we first analyze the cost of a convolution operation. Given $k$ subtrees $T_{v_{C_1}}, \ldots, T_{v_{C_k}}$, let $n$ be the total number of vertices in $T_{v_{C_1}} \cup \cdots \cup T_{v_{C_k}}$. Computing the convolution of $T_{v_{C_1}}, \ldots, T_{v_{C_k}}$ using Equation 4 requires $O(n^2)$ time. The number of count probabilities computed as the intermediate results is $O(n)$.

Then, we analyze the number of convolution operations required for each intermediate node. Equation 3 requires one convolution. Equation 5 requires $k+2$ convolutions. Therefore, for an intermediate node $v_C$ with $k$ children, let $T_{v_C}$ be the subtree with root $v_C$ and containing $n$ vertices. The overall complexity of computing the count probability of $T_{v_C}$ is $O((k+3)n^2) = O(kn^2)$. $(k+3)n$ count probabilities are computed as the intermediate results.

### 6.3.3 Optimal Tree Scan Order

Different tree depth first scan order of a tree may lead to different cost in computing count probabilities.

Given an acyclic clique graph $G_C$, let $v_1, \ldots, v_d$ be the vertices in $G_C$ whose degrees are 1, then we can scan $G_C$ in the depth first order from $v_i$ $(1 \leq i \leq d)$. Thus, there are $d$ different tree scan orders.

Once the root $v_i$ of the clique graph $G_C$ is selected, the scan order of $G_C$ is uniquely determined. We can compute, for each intermediate vertex $v$, the size of the subtree with root $v$. Therefore, the number of count probabilities computed as the intermediate results in $G_C$ with root $v$ is

$$Cost(v) = \sum_{|children(v)|>0} (|children(v)| + 3) \times |T_v|$$

where $|children(v)|$ is the number of children of $v$ and $|T_v|$ is the size of the subtree with root $v$.

Therefore, by scanning $G_C$ once from root $v_i$, we can calculate the overall cost of computing the count probability of $G_C$ from $v$. This takes $O(n)$ time, where $n$ is the number of vertices in $G_C$. Moreover, in time $O(dn)$ we can decide which vertex in $\{v_1, \ldots, v_d\}$ will lead to the minimal computational cost.

## 6.4 Histogram Answer Approximation

Once the count probabilities of $G$ is calculated, we can compute the answer to the aggregate histogram query by partitioning the count probabilities into $\eta$ equi-width intervals or $k$ equi-depth intervals.

The bottleneck of answering count queries is the m-fold convolution on the answers in the $m$ components. We introduce two techniques that accelerate the computation for equi-width histogram answers and equi-depth histogram answers, respectively, by calculating the approximate answers to convolutions.

### 6.4.1 Equi-width Histogram Answer Approximation

When computing the count probability of $G$ using m-fold convolution, intuitively, we can ignore the values whose probabilities are very small.

Let $x_1, \ldots, x_m$ be the list of values in the $i$-fold convolution $\zeta_{i-1}(x)$ in the probability ascending order ($2 \leq i \leq m$). Let $x_\mu = \max_{1 \leq j \leq m}\{x_j | \sum_{1 \leq h \leq j} \zeta_{i-1}(x_h) < \epsilon\}$. We approximate the probabilities as

$$\zeta'_{i-1}(x_i) = \begin{cases} 0, & 1 \leq i \leq \mu; \\ \frac{\zeta_{i-1}(x_i)}{\sum_{\mu \leq h \leq m} Pr(x_h)}, & \mu < x \leq m. \end{cases} \quad (6)$$

Then, the convolution of $\zeta'_{i-1}(x)$ and $Pr(G_i, x)$ are used to estimate $\zeta_i(x)$. The quality of the approximation answer is guaranteed by the following theorem.

THEOREM 5 (APPROXIMATION QUALITY). *Given a* `count` *query $Q$ on linkages with $z$ components, a bucket width parameter $\eta$, and a minimum probability threshold $\tau$, let $(\phi_i, p_i)$ be the equi-width histogram answer to $Q$, and $(\phi_i, \hat{p}_i)$ be the approximation of $(\phi, p_i)$ computed using Equation 6, then $|p_i - \hat{p}_i| < z\epsilon$ for $1 \leq i \leq \lceil \frac{v_{max} - v_{min}}{\eta} \rceil$.* ∎

Under the probability approximation, $\zeta'_{i-1}(x) > \epsilon$ holds for every value $x$ with a non-zero probability. Thus, the number of values with a non-zero probability is at most $\frac{1}{\epsilon}$. The overall complexity of computing the $\epsilon$-approximation of $\zeta_i(x)$ is $O(\frac{1}{\epsilon^2})$. The overall complexity is $O(\frac{m}{\epsilon^2})$, where $m$ is the number of components in $G$.

### 6.4.2 Equi-depth Histogram Answer Approximation

To accelerate probability calculation for equi-depth histogram answers, we introduce an approximation method that keeps a constant number of values in the intermediate results.

If the probability $\zeta_{i-1}(x)$ after the $i$-fold convolution contains values $x_1, \ldots, x_{n_{i-1}}$ ($n_{i-1} > k$) in the value ascending order, then we compute the $\rho$-quantiles $x'_i = \arg\min_x\{\sum_{a \leq x} \zeta_{i-1}(a) \geq \frac{i}{\rho}\}$ ($0 \leq i \leq \rho$). From the $\rho + 1$ values, we construct an approximation of $\zeta_{i-1}(x)$ as:

$$\zeta'_{i-1}(x) = \begin{cases} \frac{1}{\rho}, & x = \frac{x'_{i-1} + x'_i}{2} \ (1 \leq i \leq \rho); \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

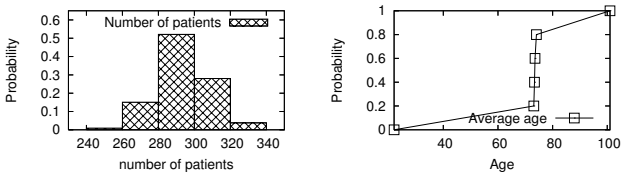
Then, the convolution of $\zeta'_{i-1}(x)$ and $Pr(G_i, x)$ are used to estimate $\zeta_i(x)$. The approximation quality is guaranteed by the following theorem.

THEOREM 6 (APPROXIMATION QUALITY). *Given a* `count` *query $Q$ on a PME-graph $G$ with $m$ components, an integer $k > 0$, let $(\phi_i, p_i)$, where $\phi_i = [v_{i-1}, v_i)$ $(1 \leq i \leq k)$, be the equi-depth histogram answer computed using the $\rho$-quantile approximation, then $|Pr(Q(G) \leq v_i) - Pr'(Q(G) \leq v_i)| < \frac{m}{\rho}$, where $Pr'(Q(G) \leq v_i) = \sum_{a \leq v_i} \zeta'_m(a)$ is the probability computed using Equation 7.* ∎

Using the above approximation method, the overall complexity of computing the approximate $k$ equi-depth histogram answer is $O(m\rho^2)$, where $m$ is the number of connected components.

# 7. EMPIRICAL EVALUATION

In this section, we report a systematic empirical study.




(a) Equi-width answer to $Q_1$. (b) Equi-depth answer to $Q_2$.

**Figure 6: Answers to queries on real data sets.**

## 7.1 Experimental Settings

All experiments were conducted on a PC computer with a 3.0 GHz Pentium 4 CPU, 1.0 GB main memory, and a 160 GB hard disk, running the Microsoft Windows XP Professional Edition operating system, Our algorithms were implemented in C++ compiled by Microsoft Visual Studio 2005.

The real data used in our experiments is the Cancer Registry data set and the Social Security Death Index provided in Link Plus 2.0 (`http://www.cdc.gov/cancer/npcr/tools/registryplus/lp.htm`).

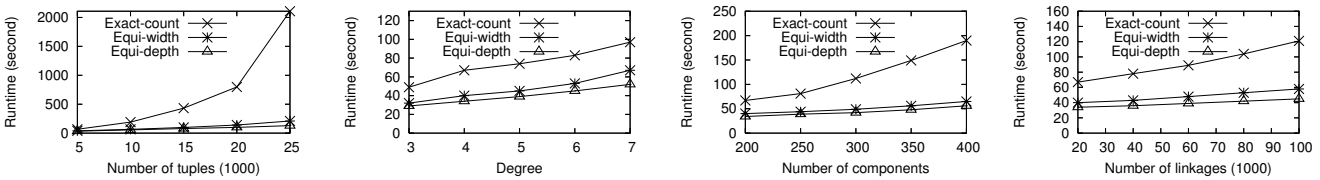
The Cancer Registry data set contains 50,000 records and each record decribes the personal information of a patient, such as `name` and `SSN`. The Social Security Death Index data set contains 10,000 records and each record contains the personal information of an individual, such as `name`, `SSN` and `Death Date`. Since the information of some records are incomplete or ambiguous, we cannot find the exact match for records in the two data sets.

Link Plus is a popularly used tool that computes the probability that two records referring to the same individual. It matches the records on the two data sets based on `name`, `SSN` and `Date of Birth` and returns 4,658 pairs of records whose linkage probabilities are greater than 0. The system suggests that a user should set a matching linkage probability threshold. The pairs of records passing the threshold are considered matching. If we set the threshold as the default value 0.25 suggested by the system, only 99 pairs of records are returned.

The synthetic data sets are generated using the following settings. A data set contains $N_l$ linkages between tables $A$ and $B$ containing $N_t$ tuples each. The degree of each tuple follows the Normal distribution $N(\mu_t, \sigma_t)$. A data set contains $N_c$ connected components. The corresponding PME-graph contains $N_l$ vertices. The number of vertices in the clique graph is the number of tuples whose degrees are greater than 1. We generate the linkages as follows. First, for each tuple $t_A \in A$, a set of linkages are generated associating with $t_A$. Then, for each tuple $t_B \in B$, we randomly pair the tuples in $A$ to $t_B$. In order to avoid loops, once a linkage $(t_A, t_B)$ is created, all tuple $t'_A \in A$ that are in the same connected component with $t_A$ cannot be assigned to $t_B$. The membership probability of each linkage is randomly assigned and normalized so that the probability of each tuple is between $(0, 1]$.

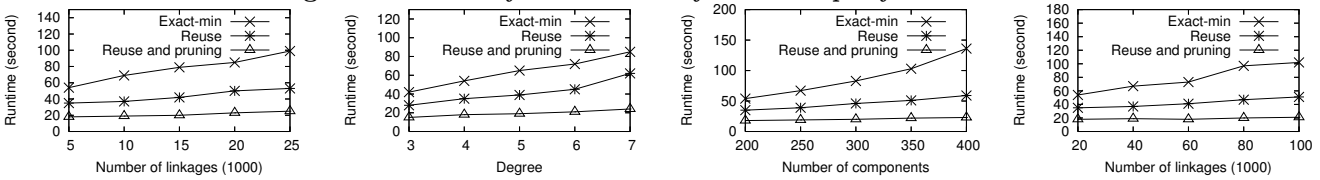
## 7.2 Effectiveness on Real Data Sets

First, we apply the aggregate queries on the Cancer Registry data set and the Social Security Death Index provided in Link Plus 2.0. Both data sets contain the personal information of an individual. Since the information of some records are incomplete or ambiguous, we cannot find the exact match for records in the two data sets. Link Plus




(a) Runtime vs. tuple size. (b) Runtime vs. degree. (d) Runtime vs. component. (d) Runtime vs. linkage size.

**Figure 7: Efficiency and scalability of `count` query evaluation.**




(a) Runtime vs. tuple size. (b) Runtime vs. degree. (c) Runtime vs. component. (d) Runtime vs. linkage size.

**Figure 8: Efficiency and scalability of `min` query evaluation.**

computes the probability that two records referring to the same individual. It returns 4,658 pairs of records whose linkage probabilities are greater than 0. If we set the matching threshold as the default value 0.25 suggested by the system, only 99 pairs of records are returned.

To elaborate the effectiveness of aggregate queries on probabilistic linkages, we ask the following `count` query on the data sets. $Q_1$: *the number of patients appearing in both data sets?* The answer histogram is shown in Figure 6(a). As shown, likely the count is much larger than 99, the number of linked pairs passing the matching threshold 0.25.

To demonstrate aggregate queries other than `count`, we try an `average` query $Q_2$: *the average age of the patients appearing in both data sets.* The answer histogram is shown in Figure 6(b). If only the 99 records of matching probabilities over 0.25 are considered, the `average age` is 71.7.

Clearly, comparing to the minimum matching probability threshold methods, our approach provides more informative answers to aggregate queries over probabilistic linkages using histograms.

## 7.3 Performance on Synthetic Data Sets

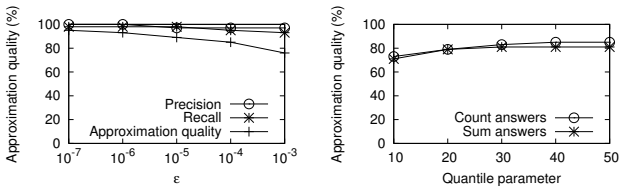
We further evaluate the efficiency and the approximation quality of our approaches on synthetic data sets with different parameter settings.

By default, a synthetic data set contains 20,000 linkages between tables $A$ and $B$ with 5,000 tuples each. The degree of a tuple follows the Normal distribution $N(4,1)$. The bucket width $\eta = 1,000$ and the minimum probability threshold $\tau = 0.1$. The parameter $k$ for equi-depth histogram answer is set to 10.

First, we evaluate the efficiency and scalability of the query answering methods. Figure 7 shows the runtime of the query answering methods for `count` queries in various parameter settings. *Exact-count* is the exact algorithm described in Section 6.3. *Equi-width* and *Equi-depth* denote the approximation algorithms discussed in Section 6.4. By default, $\epsilon = 10^{-4}$ and $\rho = 30$. Clearly, the two approximation algorithms are more efficient than the exact algorithm.

Since answering `sum` queries is very similar to answering `count` queries, we omit the results on `sum` queries for the interest of space.

Figure 8 shows the efficiency of the `min` query evaluation. *Exact-min* is the algorithm that transforms the `min` query to a set of `count` queries. *Reuse* is the algorithm that explores the sharing of computation among different linkages. *Reuse+Pruning* is the algorithm that applies the pruning technique discussed in Section 5 in addition to the reuse method. Clearly, the two techniques improve the efficiency significantly.




(a) Equi-width histogram. (d) Equi-depth histogram.

**Figure 9: Approximation quality.**

Last, we evaluate the quality of the two approximation algorithms discussed in Section 6.4. The quality of $\epsilon$-approximate euqi-width histogram answers is computed as $\frac{1}{|\{\phi|Pr(\phi)>\tau\}|}\sum_{Pr(\phi)>\tau} 1 - \frac{|\widehat{Pr}(\phi)-Pr(\phi)|}{Pr(\phi)}$, where $Pr(\phi)$ is the probability of interval $\phi$ and $\widehat{Pr}(\phi)$ is the estimated probability of $x \in \phi$. The quality of the equi-depth histogram answer approximation is measured by $\frac{1}{k}\sum_{i=1}^{k} 1 - \frac{|\hat{Pr}(v_i)-Pr(v_i)|}{Pr(v_i)}$, where $v_i$ is the value output as the approximation of the $i$-th $k$-quantile, $Pr(v_i)$ is the real probability of $v_i$, and $\hat{Pr}(v_i)$ is probability computed using the approximation method. The experimental results show that our approximation methods have good quality.

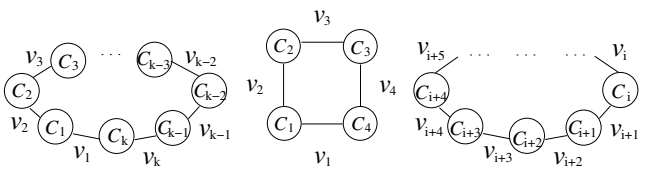
# 8. CONCLUSIONS

In this paper, we investigate aggregate query evaluation on probabilistic linkages. In contrast to the traditional methods that use simple probability thresholds to obtain a set of deterministic linkages, we fully utilize the probabilities produced by the record linkage methods and consider aggregates on linked records as distributions over possible worlds. By preserving the distribution information, we can provide more meaningful answers to aggregate queries. Moreover, we propose efficient exact and approximate query answering methods.

# 9. REFERENCES

[1] P. Agrawal and J. Widom. Confidence-aware joins in large uncertain databases. *Technical report, Stanford University CA, USA*, March 2007.

[2] M. Bilenko and R. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *SIGKDD'03*.
[3] B. Bilenko *et al.* Adaptive Product Normalization: Using Online Learning for Record Linkage in Comparison Shopping. In *ICDM'05*.
[4] D. Burdick *et al.* OLAP over uncertain and imprecise data. In *VLDB '05*.
[5] D. Burdick *et al.* Efficient allocation algorithms for olap over imprecise data. In *VLDB'06*.
[6] D. Burdick *et al.* Olap over imprecise data with domain constraints. In *VLDB '07*.
[7] A. L. P. Chen *et al* Evaluating aggregate operations over imprecise data. *TKDE*, 8(2):273–284, 1996.
[8] R. Cheng *et al.* Evaluating probabilistic queries over imprecise data. In *SIGMOD'03*.
[9] R. Cheng *et al.* Efficient join processing over uncertain data. In *CIKM '06*.
[10] W. Cohen and J. Richman. Learning to Match and Cluster Large HighDimensional Data Sets For Data Integration. In *SIGKDD'02*.
[11] R. Cowell *et al.* Probabilistic Networks and Expert System. *Springer*, 1999.
[12] A. Deshpande *et al.* Graphical models for uncertain data. In *Managing and Mining Uncertain Data*, pages 77–105. Springer, 2008.
[13] X. Dong *et al.* Data integration with uncertainty. In *VLDB '07*.
[14] J. R. Evans and E. Minieka. Optimization Algorithms for Networks and Graphs. Marcel Dekker Inc., 1992.
[15] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *JASA*, 64:1183–1210, 1969.
[16] L. Gravano *et al.* Approximate string joins in a database (almost) for free. In *VLDB '01*.
[17] L. Gu *et al.* Record linkage: Current practice and future directions. Technical Report 03/83, CSIRO, 2003.
[18] T. N. Herzog *et al.* Estimating the parameters of the Fellegi-Sunter record linkage model. In *Data Quality and Record Linkage Techniques*, pages 93–106. Springer New York, 2007.
[19] M. Hua and J. Pei. Ranking Queries on Uncertain Data Springer, 2011.
[20] T. S. Jayram *et al.* Estimating statistical aggregates on probabilistic data streams. In *PODS '07*.
[21] F. V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, 1st edition, 1996.
[22] B. Kimelfeld and Y. Sagiv. Maximally joining probabilistic data. In *PODS '07*.
[23] R. Kindermann and J. Snell. *Markov random fields and their applications*. American Mathematical Society, Providence, Rhode Island, 1980.
[24] N. Koudas *et al.* Record linkage: similarity measures and algorithms. In *SIGMOD '06*.
[25] M. Larsen. Advances in Record Linkage Theory: Hierarchical Bayesian Record Linkage Theory. In *ASA'05*.
[26] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8, 1966.
[27] C. Ré *et al.* Efficient top-$k$ query evaluation on probabilistic data. In *ICDE'07*.
[28] L. J. Roos *et al.* The art and science of record linkage: methods that work with few identifiers. *Comput Biol Med.*, 16:45–57, 1986.
[29] A. D. Sarma *et al.* Working models for uncertain data. In *ICDE'06*.
[30] A. D. Sarma *et al.* Uncertainty in data integration. In *Managing and Mining Uncertain Data*, pages 185–217. Springer, 2008.
[31] P. Sen and A. Deshpande. Representing and querying correlated tuples in probabilistic databases. In *ICDE'07*.
[32] P. Sen *et al.* Exploiting shared correlations in probabilistic databases. *VLDB'08*.
[33] B. Svartbo *et al.* Survival during and after hospitalization: a medical record linkage. *International Journal of Health Care Quality Assurance*, 12:13–17(5), 1 January 1999.
[34] V. Verykios *et al.* A Bayesian decision model for cost optimal record matching. In *The VLDB Journal*, 12(1):28–40, 2003.
[35] W. Winkler. Using the EM algorithm for weight computation in the Fellegi-Sunter model of record linkage. In *American Statistical Association*, 667–671, 1988.




(a) A set of linkages $\mathcal{L}$. (b) Case 1. (c) Case 2.

**Figure 10: A cycle of $k$ cliques.**

# APPENDIX

## Proof of Theorem 1

In a PME-graph $G_{\mathcal{L},A,B}$, a vertex $v$ is mutually exclusive with its adjacent vertices $N_v$. For any other vertex $v' \in V - \{v\} - N_v$, $v$ and $v'$ are independent conditional on $N_v$. The Markov property [23] is satisfied. ∎

## Proof of Theorem 2

(Sufficiency) We consider the two conditions one by one.

*Condition 1.* If the clique graph $G_C$ is acyclic, then the joint distribution of the linkages can be derived using the methods discussed in Section 3.4.

*Condition 2.* If the second condition holds, then the joint distribution of the linkages involved in $G'$ can be uniquely determined. Suppose $G'$ contains vertices $\{v_{C_1}, \ldots, v_{C_k}\}$ as shown in Figure 10(a), whose corresponding cliques are $\{C_1, \ldots, C_k\}$ in the PME-graph ($k$ must be an even number since the linkages between tuple sets $A$ and $B$ form a bipartite graph). In the clique graph, since each vertex $v_{C_i}$ has degree 2, which means, the corresponding clique in the PME-graph shares 2 vertices with 2 other cliques. There are $k$ edges $e_{v_1}, \ldots, e_{v_k}$ involved in the cycle, whose corresponding vertices in the PME-graph are $v_1, \ldots, v_k$. Each vertex $v_i$ belongs to two cliques $C_{i-1}$ and $C_i$ $(2 \le i \le k)$. $v_1$ belongs to cliques $C_1$ and $C_k$. Since the probability sum of each two connected edges is 1, we have $Pr(v_1) = Pr(v_{2i+1})$ $(1 \le i \le \frac{k}{2} - 1)$ and $1 - Pr(v_1) = Pr(v_{2j})$ $(1 \le j \le \frac{k}{2})$. Thus, the joint distribution of all vertices in the PME-graph is given by

$$Pr((\bigwedge_{0 \le i \le \frac{k}{2}-1} v_{2i+1}) \bigwedge (\bigwedge_{1 \le j \le \frac{k}{2}} \neg v_{2j})) = Pr(v_1)$$
$$Pr((\bigwedge_{0 \le i \le \frac{k}{2}-1} \neg v_{2i+1}) \bigwedge (\bigwedge_{1 \le j \le \frac{k}{2}} v_{2j})) = 1 - Pr(v_1).$$

The joint distribution is consistent with the marginal distribution specified by each linkage. Thus, the linkages are compatible.

(Necessity). Consider a set of compatible linkages whose clique graph is $G_C$.

Suppose $G'$ contains a cycle, then we need to show that $G'$ can only form a cycle satisfying condition 2 in the theorem. We prove this in two cases.

*Case 1:* The cycle in $G'$ contains 4 vertices $v_{C_1}, v_{C_2}, v_{C_3}, v_{C_4}$, whose corresponding cliques in the PME-graph are $\{C_1, C_2, C_3, C_4\}$, as illustrated in Figure 10(b). Let $v_i$ be the vertex contained by $C_{i-1}$ and $C_i$ $(2 \le i \le 4)$ and $v_1$ be contained by $C_1$ and $C_4$. The joint probability of $v_1$ and $v_4$ can be expressed as $Pr(v_1 v_4) = Pr(\neg v_2 \neg v_3) Pr(v_1 | \neg v_2) Pr(v_4 | \neg v_3)$. Since $v_1$ and $v_4$ are contained in clique $C_4$, $Pr(v_1 v_4) = 0$ holds. Moreover, $Pr(v_1 | \neg v_2) > 0$ and $Pr(v_4 | \neg v_3) > 0$. Thus, $Pr(\neg v_2 \neg v_3) = 0$. Since $v_2$ and $v_3$ are contained in the same clique $C_2$, we have $Pr(\neg v_2 \neg v_3) = 1 - Pr(v_2) - Pr(v_3)$. Therefore, $Pr(v_2) + Pr(v_3) = 1$, which means that $C_2$ only contains

two vertices $\{v_2, v_3\}$ and $Pr(C_2) = 1$. Similarly, we can show that other clique $C_i$ $(1 \le i \le 4)$ only contains 2 vertices and the probability sum of the two vertices is 1.

*Case 2:* The cycle in $G'$ contains $k$ vertices $v_{C_1}, \ldots, v_{C_k}$ $(k > 4)$, as illustrated in Figure 10(c). The corresponding cliques in the PME-graph are $C_1, \ldots, C_k$, respectively. Let $v_i$ be the vertex contained by $C_{i-1}$ and $C_i$ $(2 \le i \le k)$ and $v_1$ be contained by $C_1$ and $C_k$. We show that, for any clique $C_i$, $Pr(v_i) + Pr(v_{i+1}) = 1$.

The joint distribution of $v_{i+2}$ and $v_{i+3}$ can be expressed as $Pr(v_{i+2}v_{i+3}) = Pr(\neg v_{i+1}\neg v_{i+4})Pr(v_{i+2}|\neg v_{i+1})Pr(v_{i+3}|\neg v_{i+4})$. Since $v_{i+2}$ and $v_{i+3}$ belong to the same clique $C_{i+2}$, we have $Pr(v_{i+2}v_{i+3}) = 0$. Moreover, $Pr(v_{i+2}|\neg v_{i+1}) > 0$ and $Pr(v_{i+3}|\neg v_{i+4}) > 0$. Therefore, $Pr(\neg v_{i+1}\neg v_{i+4}) = 0$. We can express $Pr(\neg v_{i+1}\neg v_{i+4})$ as

$$\begin{aligned} &Pr(\neg v_{i+1}\neg v_{i+4}) \\ &= Pr(\neg v_{i+1}\neg v_{i+4}v_i v_{i+5}) + Pr(\neg v_{i+1}\neg v_{i+4}v_i\neg v_{i+5}) \\ &+Pr(\neg v_{i+1}\neg v_{i+4}\neg v_i v_{i+5}) + Pr(\neg v_{i+1}\neg v_{i+4}\neg v_i\neg v_{i+5}) = 0 \end{aligned} \tag{8}$$

Since all probability values are non-negative, each component in Equation 8 has to be 0. Therefore, we have

$$\begin{aligned} &Pr(\neg v_{i+1}\neg v_{i+4}v_i v_{i+5}) \\ &= Pr(v_i v_{i+5})Pr(\neg v_{i+1}|v_i)Pr(\neg v_{i+4}|v_{i+5}) = 0 \end{aligned} \tag{9}$$

and

$$\begin{aligned} &Pr(\neg v_{i+1}\neg v_{i+4}\neg v_i v_{i+5}) \\ &= Pr(\neg v_i v_{i+5})Pr(\neg v_{i+1}|\neg v_i)Pr(\neg v_{i+4}|v_{i+5}) = 0 \end{aligned} \tag{10}$$

In Equation 9, since $Pr(\neg v_{i+1}|v_i) = Pr(\neg v_{i+4}|v_{i+5}) = 1$, we have $Pr(v_i v_{i+5}) = 0$. Therefore, in Equation 10, $Pr(\neg v_i v_{i+5}) = Pr(v_{i+5}) - Pr(v_i v_{i+5}) > 0$. Thus, $Pr(\neg v_{i+1}|\neg v_i) = 0$. Since $Pr(\neg v_{i+1}|\neg v_i) = \frac{1-Pr(v_{i+1})-Pr(v_i)}{Pr(\neg v_i)}$, we have $Pr(v_{i+1}) + Pr(v_i) = 1$. Therefore, $C_i$ only has 2 vertices $\{v_i, v_{i+1}\}$ and $Pr(v_{i+1}) + Pr(v_i) = 1$. ∎

## Proof of Lemma 1

Let $v$ belong to two cliques $C_1$ and $C_2$. In the clique graph $G_C$, let $v_{C_1}$ and $v_{C_2}$ be the two vertices corresponding to $C_1$ and $C_2$, respectively, and $e_v$ be the edge in $G_C$ corresponding to $v$. Since $G_C$ is a tree, there is only one path between $v_{C_1}$ and $v_{C_2}$ and the path must contain $e_v$. Therefore, removing $e_v$ will lead to two disconnected subgraphs in $G_C$. Correspondingly, removing $v$ will produce two disconnected components in $G_i$. ∎

## Proof of Theorem 3

The theorem immediately follows with the conditional independency of $G_i^1$ and $G_i^2$ given $v$. ∎

## Proof of Theorem 4

Since $v_1$ is a leaf node, the corresponding clique $C_1$ only contains one joint vertex $v$ in the PME-graph. After the vertex compression, there is only one private vertex $v_p$ of $C_1$ satisfying the query predicate $P$. The private vertices of $C_1$ not satisfying $P$ are removed in the predicate processing step.

When $v$ appears, if $v.F = 1$, then there is one vertex in $C_1$ satisfying $P$, and thus $Pr(C_1, 1|v) = 1$. If $v.F = 0$, then no vertex in $C_1$ satisfying $P$, so $Pr(C_1, 0|v) = 1$.

When $v$ does not appear, then $Pr(C_1, 0|\neg v)$ is the probability that $v_p$ does not appear, which is $\frac{1-Pr(v)-Pr(v_p)}{1-Pr(v)}$. Moreover, $Pr(C_1, 1|\neg v) = \frac{Pr(v_1)}{1-Pr(v)}$ is the probability that $v_p$ appears. ∎

## Proof of Theorem 5

We will show that an approximation error of $\epsilon$ is introduced each time when integrating one connected component $G_t$ $(2 \le t \le m)$. Let $x_1, \ldots, x_m$ be the list of values in $\zeta_{t-1}(x)$ in the probability ascending order, $v_1 = v_{min} + (i-1)\eta$ and $v_2 = v_{min} + i\eta$. Let $p_i^t = \sum_{v_1 \le x \le v_2} \zeta_t(x)$ be the probability of bucket $[v_1, v_2)$. Since

$$p_i^t = \sum_{b=1}^{m} \zeta_{t-1}(x_b) \sum_{v_1 - x_b \le x \le v_2 - x_b} Pr(G_t, x)$$

where $\zeta_{t-1}(x_b)$ is the exact count distribution in components $\{G_1, \ldots, G_{t-1}\}$. Let $\hat{p}_i^t = \sum_{v_1 \le x \le v_2} \zeta'_t(x)$ be the approximate probability computed based on the $\epsilon$-approximation $\zeta'_{t-1}(x)$, then

$$\hat{p}_i^t = \sum_{b=1}^{m} \zeta'_{t-1}(x_b) \sum_{v_1 - x_b \le x \le v_2 - x_b} Pr(G_t, x)$$

Let $g(v_1 - x_b, v_2 - x_b) = \sum_{v_1 - x_b \le x \le v_2 - x_b} Pr(G_t, x)$, we have

$$\begin{aligned} p_i^t - \hat{p}_i^t &= \sum_{d=1}^{\mu} \zeta_{t-1}(x_d) g(v_1 - x_d, v_2 - x_d) \\ &+ \sum_{b=\mu+1}^{m} \big(\zeta_{t-1}(x_b) - \zeta'_{t-1}(x_b)\big) \cdot g(v_1 - x_b, v_2 - x_b) \end{aligned} \tag{11}$$

Let $A = \sum_{d=1}^{\mu} \zeta_{t-1}(x_d) g(v_1 - x_d, v_2 - x_d)$, then

$$\sum_{b=\mu+1}^{m} \zeta_{t-1}(x_b) g(v_1 - x_b, v_2 - x_b) = p_i^t - A.$$

According to Equation 6, Equation 11 can be rewritten as

$$p_i^t - \hat{p}_i^t = A + \left(1 - \frac{1}{\sum_{\mu \le h \le m} \zeta_{t-1}(x_h)}\right)(p_i^t - A)$$

On the one hand, since $\sum_{\mu \le h \le m} \zeta_{t-1}(x_h) \le 1$ and $p_i^t > A$, we have $p_i^t - \hat{p}_i^t \le A$. Moreover, $A = \sum_{d=1}^{\mu} \zeta_{t-1}(x_d) g(v_1 - x_d, v_2 - x_d) \le \sum_{d=1}^{\mu} \zeta_{t-1}(x_d) \le \epsilon$. Thus, $p_i^t - \hat{p}_i^t \le \epsilon$.

On the other hand, $\sum_{\mu \le h \le m} \zeta_{t-1}(x_h) \ge 1 - \epsilon$, and thus $1 - \frac{1}{\sum_{\mu \le h \le m} \zeta_{t-1}(x_h)} \ge \frac{-\epsilon}{1-\epsilon}$. Moreover, $p_i^t - A \ge p_i^t - \epsilon$. Therefore, $p_i^t - \hat{p}_i^t \ge -\epsilon \times \frac{p_i^t - \epsilon}{1-\epsilon} \ge -\epsilon$.

By integrating the $m$ components, we have $|p_i - \hat{p}_i| \le \epsilon$. ∎

## Proof of Theorem 6

We only need to show that each time when we integrate one connected component $G_t$ $(2 \le t \le m)$, we introduce an approximation error of $\frac{1}{\rho}$.

Let $g_t(v_i) = Pr(Q(G) \le v_i) = \sum_{x \le v_i} \zeta_t(x)$ and $g_{t-1}(x_b) = \sum_{x \le x_b} \zeta_{t-1}(x)$. Then,

$$g_t(v_i) = \sum_{x_d=0}^{x_n} Pr(G_t, x_d) \cdot g_{t-1}(v_i - x_d)$$

where $x_n$ is the number of cliques in $G_t$.

Let $g'_{t-1}(x_b) = Pr'(Q(G) \le v_i) = \sum_{x \le x_b} \zeta'_{t-1}(x)$ where $\zeta'_{t-1}(x)$ is the approximation of $\zeta_{t-1}(x)$ using Equation 7, then

$$g'_t(v_i) = \sum_{x_d=0}^{x_n} Pr(G_t, x_d) \cdot g'_{t-1}(v_i - x_d)$$

Therefore, $|g_t(v_i) - g'_t(v_i)| =$
$|\sum_{x_d=0}^{x_n} Pr(G_t, x_d) \cdot \big(g_{t-1}(v_i - x_d) - g'_{t-1}(v_i - x_d)\big)|$.

Let $x'_c$ $(1 \le c \le \rho)$ be the $\rho$-quantiles of $\zeta_{t-1}(x)$. Suppose $x'_{c-1} \le v_i - x_d \le x'_c$ $(1 \le c \le \rho)$, there are two cases:

First, if $v_i - x_d \le \frac{x'_{c-1} + x'_c}{2}$, then $g'_{t-1}(v_i - x_d) = \frac{c-1}{\rho}$ and $\frac{c-1}{\rho} \le g_{t-1}(v_i - x_d) \le \frac{c}{\rho}$. Thus, $0 \le g_{t-1}(v_i - x_d) - g'_{t-1}(v_i - x_d) \le \frac{1}{\rho}$.

Second, if $v_i - x_d > \frac{x'_{c-1} + x'_c}{2}$, then $g'_{t-1}(v_i - x_d) = \frac{c}{\rho}$ and $\frac{c-1}{\rho} \le g_{t-1}(v_i - x_d) \le \frac{c}{\rho}$. Thus, $-\frac{1}{\rho} \le g_{t-1}(v_i - x_d) - g'_{t-1}(v_i - x_d) \le 0$.

In both cases, $|g_{t-1}(v_i - x_d) - g'_{t-1}(v_i - x_d)| \le \frac{1}{\rho}$. Therefore,

$$|g_t(v_i) - g'_t(v_i)| \le \sum_{x_d=0}^{x_n} Pr(G_t, x_d) \cdot \frac{1}{\rho} = \frac{1}{\rho}.$$