A Bayesian Approach for on-line max auditing of Dynamic Statistical Databases

Gerardo Canfora
Bice Cavallo
University of Sannio, Benevento, Italy,
{gerardo.canfora,bice.cavallo}@unisannio.it

ABSTRACT

In this paper we propose a method for on-line max auditing of dynamic statistical databases. The method extends the Bayesian approach presented in [2], [3] and [4] for static databases. A Bayesian network addresses disclosures based on probabilistic inferences that can be drawn from released data; we have developed algorithms to update the network whenever the database changes. In particular, we consider the case in which records are added or deleted, or some sensitive values change their value. The paper introduces the algorithms and discusses results of a preliminary set of of experimental trials.

Keywords

Dynamic Statistical Databases, Bayesian network, uncertainty

1. INTRODUCTION

A Statistical Database (SDB) is a database system that enables its users to retrieve only aggregate statistics (e.g., mean, max, min, and count) for a subset of the entities represented in the database. Consider, for example, a company database containing salaries of employees. A user may want to determine the max or a min salary of the employees in a subset of records in the database. He/she cannot, however, be allowed to glean the salary of any one employee in particular.

Several methods for protecting privacy in SDBs have been suggested in the literature; see reference [1] for a survey. These methods can be classified under four general approaches: conceptual, data perturbation, output perturbation, and query restriction. We focus on the query restriction approach, which prevents malicious inferences by denying some unsafe queries. In particular, we deal with the on-line auditing problem [6], [8], [9], [10], [12]. With on-line auditing, queries are answered one by one, in sequence, and the auditor has to determine whether the SDB is compromised when answering a new query.

In references [2], [3] and [4], we have proposed, for the on-line max and min auditing, a Bayesian network (BN) as a disclosure control tool, based on probabilistic inferences that can be drawn from released data. The model is able to:

- deal with on-line max and min auditing without maintaining query logs;
- deal with a probabilistic definition of privacy, independently of the probability distribution of the sensitive field;
- manage efficiently duplicated values of the sensitive field;
- provide a graphical representation of user knowledge;
- capture user prior knowledge;
- consider the case in which a denial leaks information.

In references [2], [3] and [4], the database is static.

The original contribution of this paper is to extend the approach proposed in [2], [3] and [4] to dynamic databases for on-line max auditing. A static database is one that never changes after it has been created. Most census are static: whenever a new version of the database is created, the new version is considered to be another static database. A dynamic databases can change over time. This feature can complicate the privacy problem considerably, because frequent releases of new versions may enable users to make use of the differences among the versions in ways that are difficult to foresee. References [15] and [11] deal with security of dynamic statistical databases when records can be inserted or deleted from the databases; reference [11], in particular, considers the context of a partitioned database.

In this paper, we deal with on-line max auditing in dynamic SDBs. More specifically, we consider the case in which some sensitive values change their value and some records are deleted or inserted. In the following, we provide examples that show the importance to consider, in the on-line max auditing, a dynamic database rather than a sequence of static databases.

EXAMPLE 1. *Given two sensitive values $y_1 = 8$ and $y_2 = 7$, we suppose that the user asks the max between the two values and the auditor provides the answer 8; if a new record is inserted and a new static database is considered, then the information about y_1 and y_2 is lost. As a consequence, if the user submits a new query, for instance the max values among y_1, y_2 and y_3, and the auditor provides the answer 10, then y_3 is disclosed.*

In order to deal with the insertion of new records in dynamic SDBs, we will build a BN as shown in Section 3.1.

EXAMPLE 2. *Given two sensitive values $y_1 = 8$ and $y_2 = 7$, we suppose that the user asks the max between the two values and the auditor provides the answer 8; if the record corresponding to y_1 is deleted and a new static database is considered, then the information about y_2 is lost, even if the user knows that $y_2 \leq 8$. As a consequence, if the user submits a new query, for instance the max values between y_2 and y_3, and the auditor provides the answer 10, then y_3 is disclosed.*

The deletion of records in dynamic databases is discussed in Section 3.2.

EXAMPLE 3. *Given two sensitive values $y_1 = 8$ and $y_2 = 7$, we suppose that the user asks the max between the two values and the auditor provides the answer 8; if y_1 changes into 10 and a new static database is considered, then if the user asks again the max value between y_1 and y_2 and the auditor provides the answer, that is 10, the user infers that either y_1 or y_2 is increased. Moreover, if he has prior knowledge, and knows that the value of y_1 is increased, then y_1 is disclosed.*

Thus, in our approach, we consider dynamic databases and we assume at first that the user does not know if a sensitive value changes or not (Section 3.2.1), then we assume that the user has prior knowledge about a value increase or decrease (Section 3.2.2).

The paper is organized as follows: Section 2 introduces notions and definitions useful in the sequel of the work; in particular, in Section 2.1 we summarize the Bayesian approach for on-line max auditing introduced in our previous works [2], [3] and [4] for static SDBs. Section 3 extends the previous model to dynamic databases. Section 4 discusses the results of a preliminary set of experiments and Section 5 provides conclusion and future work.

2. PRELIMINARIES

We assume that:

- T is a table with n records;
- $K = \{1, 2, ..., n\}$;
- X and Y are two fields of T such that the elements of X represented by x_i, with $i \in K$, are distinct among them (each x_i identifies uniquely a subject) and the elements of Y, represented by y_i, are real numbers;
- the sensitive field Y has r distinct values ($r \leq n$);
- the private information takes the form of an association, $(x_i, y_i) \subseteq X \times Y$, that is a pair of values in the same tuple;
- a *l-query* q is a subset of K, that is $q = \{i_1, ...i_l\} \subseteq K$;
- the answer corresponding to a max query q is equal to $max\{y_{i_j} | i_j \in q\}$;
- m is the answer to a max query;

Table 1: $n = 4$, $r = 3$. The sensitive field Y is ordered in decreasing way.

X	Y
x_1	9
x_2	8
x_3	8
x_4	5

- $l = |q| > 1$, because if $q = \{j\}$, clearly, y_j is breached irrespective of the value of *m* and the association (x_j, m) is disclosed.

In Section 2.1, we will describe the approach used in references [2], [3] and [4], focusing only on max on-line auditing.

We consider the following definition of probabilistic compromise:

DEFINITION 1. *A privacy breach occurs if and only if a private association is disclosed with probability greater or equal to a given tolerance probability tol. If a private association is disclosed with $tol = 1$, then the SDB is fully compromised.*

Finally, we recall the definition of upper bound provided in references [9] and [12]; the authors define, for each element y_j, with $j \in K$, the upper bound μ_j as follows:

DEFINITION 2. *$\forall y_j$, $\mu_j = min\{m_k | j \in q_k$ with q_k a max query and m_k the answer$\}$ is the minimum over the answers to the max queries containing j.*

In other words, μ_j is the best possible upper bound for y_j that can be obtained from the answers to the max queries.

2.1 A Bayesian approach to on-line max auditing

A BN is a probabilistic graphical model that represents a set of variables and their probabilistic dependencies [14]. A BN, also called belief net, is a directed acyclic graph (DAG) which consists of nodes, to represent variables, and arcs, to represent dependencies between variables. Arcs, or links, also represent causal influences among the variables. The strength of an influence between variables is represented by the conditional probabilities which are summarized in a conditional probability table (CPT). If there is an arc from node *A* to another node *B*, *A* is called a parent of *B*, and *B* is a child of *A*. The set of parent nodes of a node X_i is denoted $parents(X_i)$.

The size of the CPT of a node X_i depends on the number *s* of its states, the number *n* of $parents(X_i)$, and the number s_j of parent states, in the following way:

$$size(CPT) = s \cdot \prod_{j=1}^{n} s_j.$$

For every possible combination of parent states, there is an entry listed in the CPT. Notice that for a large number of parents the CPT will expand drastically.

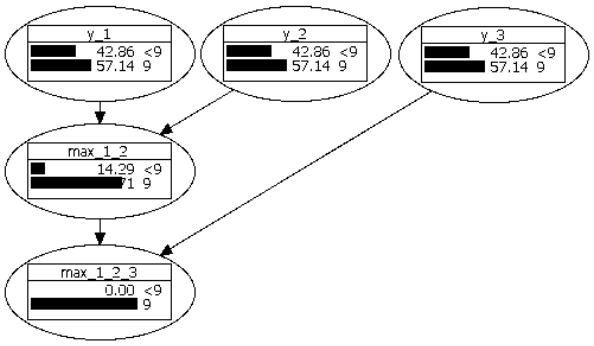

Figure 1: Temporal transformation for a max 3-query. $y_1 = 9$ **is the max value.**

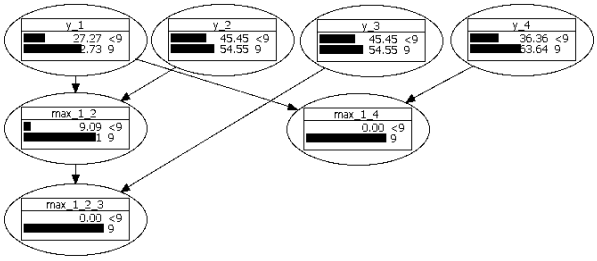

Figure 2: The private association $(x_1, 9)$ **is disclosed with probability equal to** 0.7273**.**

If X_i has no parents, its local probability distribution is said to be *unconditional*, otherwise it is *conditional*. If the value of a node is observed, then the node is said to be an *evidence* node.

In our approach, we use a BN to encode user knowledge about the private associations after a sequence of max queries. The BN contains nodes encoding the sensitive values and nodes encoding the max queries; by adding evidence on nodes encoding the max queries, the BN is able to compute the probability of disclosing the sensitive values.

Independence of causal influence (ICI) [16] among local parent-child or cause-effect relationship allows for further factoring. ICI has been used to reduce the complexity of knowledge acquisition. The size of conditional distribution that encodes the max (or min) operator can be reduced when the *n*-ary max (resp. min) operator is decomposed into a set of binary max (resp. min) operators. Two well known approaches to the decomposition are: parent divorcing [13] and temporal transformation [7]. We use temporal transformation, which constructs a linear decomposition tree where each node encodes a binary operator (see Example 4).

EXAMPLE 4. *Given Table 1 and let* $q = \{1, 2, 3\}$ *be a max query, then the 3-query is decomposed into a set of binary max queries by means of a temporal transformation as shown in Figure 1. At first we build the binary max node encoding the max between* y_1 *and* y_2*, then we build the binary max node encoding* q*.*
We can see that if we insert evidence on node encoding q*, we obtain for* $i = 1, 2, 3$*, the probabilities* $P(y_i = 9|m = 9)$*.*

REMARK 1. *In this paper, we assume that the user has not prior knowledge about the probability distribution of the sensitive field; for instance if the user knows that a sensitive value* y_i *is such that* $y_i \leq m$ *then we assume that* $P(y_i < m|y_i \leq m) = P(y_i = m|y_i \leq m) = \frac{1}{2}$*. In our previous work [4], we have also considered the case in which the probability distribution of the sensitive field is known.*

REMARK 2. *In references [2], [3] and [4], we assume that the sensitive field is ordered in a decreasing way; in Section 3, in order to consider the insertion of new records, we will remove this assumption.*

REMARK 3. *The size of the CPT for a BN encoding a temporal transformation grows linearly with the size of the query [4].*

We build the BN for the **on-line max auditing** problem at runtime, that is we execute a temporal transformation after each max user query and decide whether or not to answer the query.

EXAMPLE 5. *We continue Example 4. We suppose that the user submits the max query* $q_2 = \{1, 4\}$ *with* $m_2 = 9$*. The BN in Figure 1 changes in the BN in Figure 2. Since the private association* $(x_1, 9)$ *is disclosed with probability equal to* 0.7273*, by Definition 1, the privacy is preserved if and only if we choose a tolerance value greater than* 0.7273*.*

The answer to a query is denied if (see [4]):

1. the privacy is breached (see Definition 1);
2. the probability that a sensitive variable is equal to a value is greater or equal to a given tolerance threshold (even if this value is not the actual value of the sensitive data item);
3. for a possible answer to q_t, the probability that a sensitive variable is equal to a value is greater or equal to a given tolerance threshold (even if this value is not the actual value of the sensitive data item).

Item 2 and 3 allow us to deal with the case in which denial leaks information.

In the following, we recall some details of the BN (see [5]) that are useful to understand the sequel. In order to reduce total CPT size of the BN, it is optimized in the following way:

- **an evidence node has not children**. Let $q_1 = \{i_1, ..., i_l\}$ and $q_2 = \{i_1, ..., i_l, ..., i_{l+k}\}$ be two max queries of size l and $l + k$ respectively, with $q_1 \subset q_2$ and $m = m_1 = m_2$. Thus, the temporal transformation for q_2 is such that the first $l-1$ max nodes overlap with the nodes of the temporal transformation for q_1, and the other nodes have the same states of the the first $l - 1$ max nodes. Thus, after q_1 and q_2, each max node has two states: r_1 encoding the case in which the node value is less than m, and r_2 encoding the case in which the node value is equal to m. Because the last node in the temporal transformation of q_1, that is the node encoding the binary *max* operator between $max\{y_1, ..., y_{l-1}\}$ and y_l, is

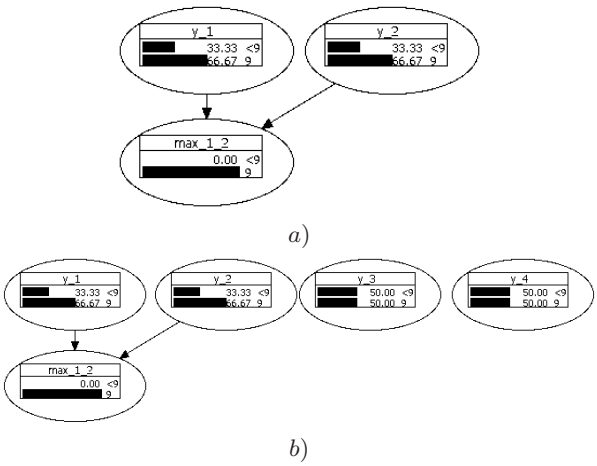

Figure 3: On-line max auditing. An evidence node has not children. a) $q_1 = \{1, 2\}$**. b)** $q_2 = \{1, 2, 3, 4\}$

an evidence node and its value is equal to m with probability 1, then the other nodes of temporal transformation of q_2 have the same states and, obviously, their value is equal to *m* with probability equal to 1, without inserting evidence. Thus, it is not needed to store these nodes: it is sufficient, for each sensitive variable $y_j \in \{y_{l+1}, y_{l+2}, ..., y_{l+k}\}$, by Remark 1, to set $Pr(y_j < m | y_j \leq m) = \frac{1}{2}$ and $Pr(y_j = m | y_j \leq m) = \frac{1}{2}$.

The reasoning is analogous if $q_1 = \{i_1, ..., i_l, ..., i_{l+k}\}$ and $q_2 = \{i_1, ..., i_l\}$.

EXAMPLE 6. *Given Table 1 and* $q_1 = \{1, 2\}$*, the corresponding BN is shown in Figure 3 a). After the answer to* $q_2 = \{1, 2, 3, 4\}$*, the BN is updated as shown in Figure 3 b). Because related to* q_1 *there is an evidence node, this node has not children and the nodes encoding* y_3 *and* y_4 *have probability distribution equal to* $(\frac{1}{2}, \frac{1}{2})$*.*

- **each child of y_j has the same states of y_j.** If the sensitive variable is in more than one query then only the queries with max value equal to μ_j are useful to compute the probability that y_j is equal to μ_j.

  Given $j \in K = \{1, ..., n\}$, let q_1 and q_2 be two max queries such that $j \in q_1 \cap q_2$ and $\mu_j = m_1 < m_2$. Then, the states of y_j are: r_1 encoding the case in which y_j is less than m_1; r_2 encoding the case in which y_j is equal to m_1. Moreover, the max node in the temporal transformation of q_2 with parent y_j is deleted. Finally, if j is the last element of q_2, that is $q_2 = \{i_1, ..., i_l\}$ with $i_l = j$, then it is needed to insert evidence on node encoding $max\{i_1, ..., i_{l-1}\}$.

  The reasoning is analogous if $m_1 > m_2 = \mu_j$.

  If there is a set of *m* queries, such that $j \in \bigcap_{k=1}^{k=m} q_k$, it is possible the reasoning in an analogous way.

  EXAMPLE 7. *Given Table 1 and* $q_1 = \{1, 2, 3, 4\}$*, the corresponding BN is shown in Figure 4 a). Given* $q_2 = \{3, 4\}$*, since* $m_1 = 9 > m_2 = \mu_3 = \mu_4 = 8$*, the max node encoding* q_2 *is added, the nodes in temporal transformation of* q_1 *with parents* y_3 *and* y_4 *are deleted and evidence is inserted on node encoding* $max\{y_1, y_2\}$*. The corresponding BN is shown in Figure 4 b).*

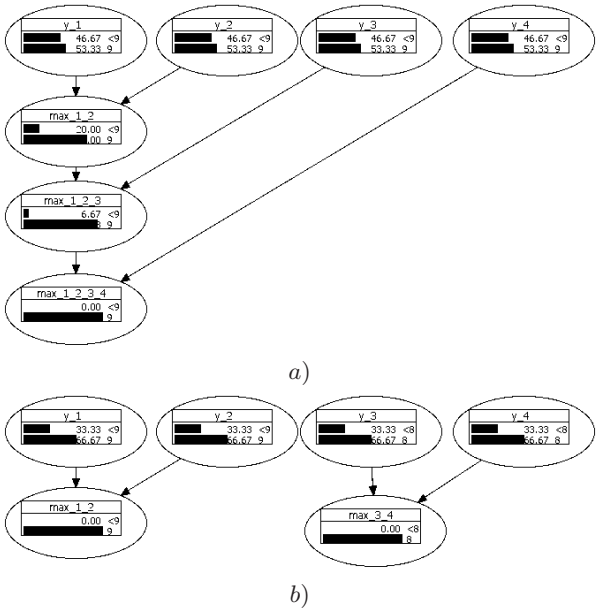

Figure 4: On-line max auditing. Each child of y_j has the same states of y_j. a) $q_1 = \{1, 2, 3, 4\}$**. b)** $q_2 = \{3, 4\}$**.**

In conclusion, each node has only children with the same states and an evidence node has not children.

REMARK 4. *We can see that there is not a max node without evidence and without children; each leaf, that is a max node, is an evidence node.*

3. A BAYESIAN APPROACH FOR DYNAMIC DATABASES

In this section we assume that the database is updated, and insertions, deletions, and changes in the sensitive values are possible. In particular, we assume that the user knows if:

1. a new record is inserted;
2. a record is deleted.

If a sensitive values changes, then at first we suppose that the user does not know if a sensitive value is modified or not (Section 3.2.1), then we suppose that the user knows if a sensitive value increases or decreases (Section 3.2.2).

3.1 Inserting records

The insertion of new records in our model is very straightforward. Since we have to insert new records in the DB, in contrast with the previous models in [2],[3], [4], we do not order the DB by sensitive field in decreasing way, but when an user submits a max query q with answer equal to m, we select an element $\bar{i} \in q$ such that $y_{\bar{i}} = m$ and the first node in the temporal transformation will be the

Table 2: $n = 3$, $r = 3$. The sensitive field Y is not ordered in decreasing way.

X	Y
x_1	5
x_2	4
x_3	9

Table 3: $n = 6$, $r = 5$. The sensitive field Y is not ordered in decreasing way.

X	Y
x_1	4
x_2	5
x_3	8
x_4	2
x_5	1
x_6	5

node encoding $y_{\bar{i}}$. This ensures that the temporal transformation is well posed.

EXAMPLE 8. *Given Table 2, if the user submits the max query $q = \{1, 2, 3\}$, the corresponding BN is similar to BN in Figure 1 but the first node in the temporal transformation is the node encoding y_3 and not the node encoding y_1. Thus, at first we build the binary max node encoding the max between y_3 and y_2, then we build the binary max node encoding q.*

In this way, it is possible to add new records in the database and to store the user knowledge about the other records.

EXAMPLE 9. *Given Table 1, we suppose that the user knows the max value between y_1 and y_2 and the corresponding BN, encoding user knowledge, is shown in Figure 3 a). If a new record, with sensitive field $y_5 = 20$, is added in the table and the user asks the max value among y_1, y_2 and y_5, then our model is able to use the stored user knowledge, that is $max\{y_1, y_2\} = 9$, to deny the answer.*

3.2 Deleting and modifying records

We deal with the deletion of records under the two following conditions:

Condition 1 If a record is deleted then it will be never inserted into the database again;

Condition 2 If a record is deleted then the value of its sensitive field can be disclosed.

We suppose that the record corresponding to $x_{\bar{i}}$ (each x_i identifies uniquely a subject) is deleted from the DB or the sensitive value $y_{\bar{i}}$ changes, in particular it increases or decreases.

Then, if $Q = \{q_1, ..., q_t\}$ is the set of queries already submitted, two cases are possible:

- there is not a query $q_j \in Q$ such that $\bar{i} \in q_j$;

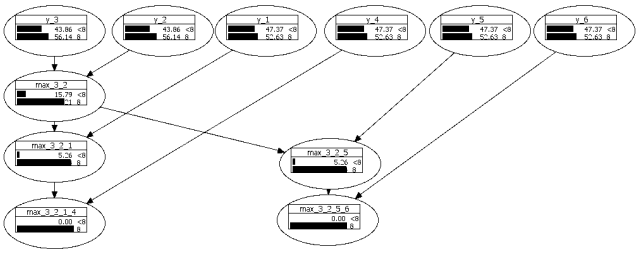

Figure 5: BN encoding user knowledge after max queries $q_1 = \{1, 2, 3, 4\}$ and $q_2 = \{2, 3, 5, 6\}$.

- there is one or more queries containing $\bar{i}$.

Only in the second case, we have to update the BN.

Therefore, we consider the following cases:

1. the record corresponding to $x_{\bar{i}}$ is deleted;
2. $y_{\bar{i}}$ increases;
3. $y_{\bar{i}}$ decreases.

In the sequel, $\mu_{\bar{i}}$ denotes the upper bound of $y_{\bar{i}}$ (see Definition 2).

3.2.1 The user does not know if a sensitive value has changed

In this section, we assume that if a record is deleted or a sensitive value changes, then the BN encoding user knowledge is updated in such way that only the user information that remains valid for the new version of the database is stored. For instance, if a sensitive value changes its value and the previous user information, about this value and the max queries including it, is false, then this information is deleted. As a matter of fact, this false information does not help the auditor to preserve the privacy and to store it requires memory.

Under the hypotheses that the user does not know if a sensitive value $y_{\bar{i}}$ increases or decreases, then:

1. let q_j be a max a query such that $\bar{i} \in q_j$. If the record corresponding to $x_{\bar{i}}$ is deleted then the user knows that $m_{q_j \setminus \{\bar{i}\}} \leq m_j$;
2. let q_j be a max a query such that $\bar{i} \in q_j$. If the value of $y_{\bar{i}}$ increases more than its upper bound then a part of the BN provides false information and it must be deleted; else, if the value of $y_{\bar{i}}$ increases less or equal to its upper bound, then the BN is not updated;
3. let q_j be a max a query such that $\bar{i} \in q_j$. If the value of $y_{\bar{i}}$ decreases then: if the node encoding $y_{\bar{i}}$ is the first node in a temporal transformation then a part of the BN provides false information (the leaf node in the corresponding temporal transformation is not an evidence node) and a part of the BN can be deleted; else, if there is not a temporal transformation such that the node encoding $y_{\bar{i}}$ is the first node, then the BN is not updated.

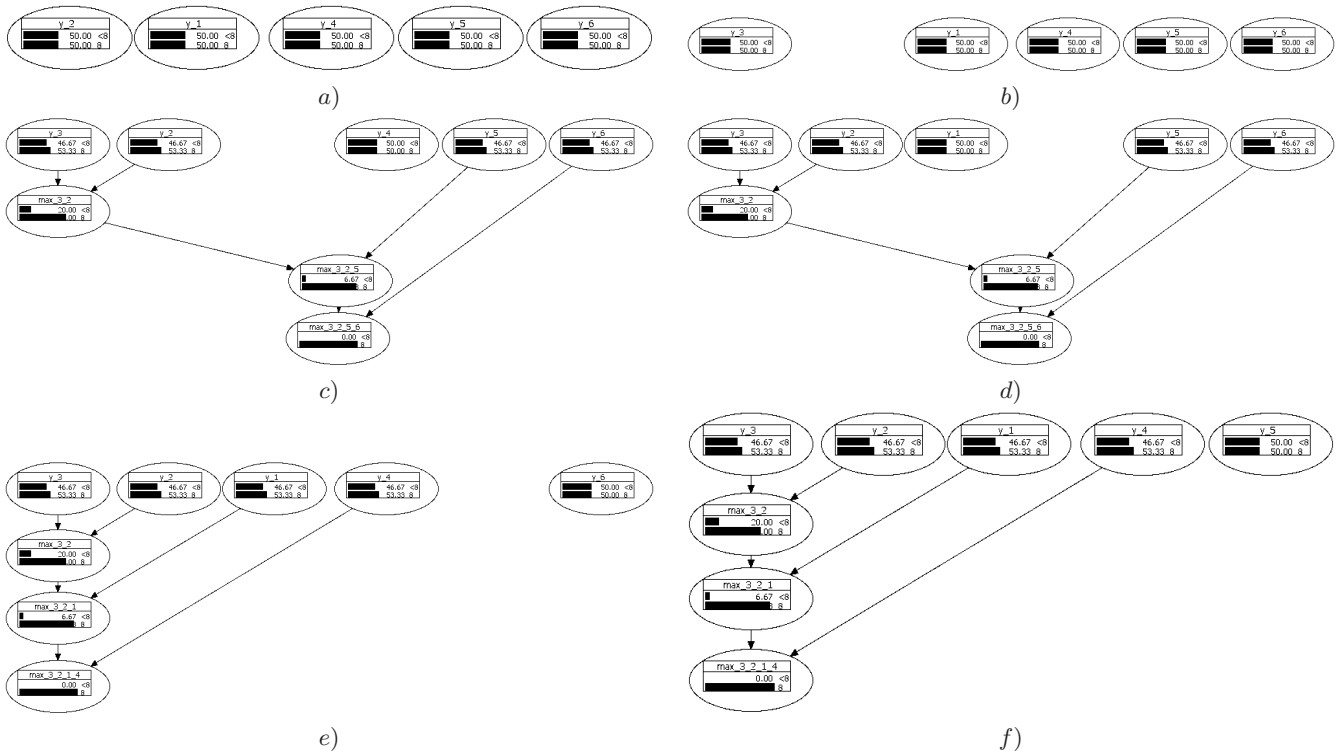

Figure 6: We update the BN in Figure 5 when: a) the record corresponding to x_3 is deleted (by means of Algorithm 1 or Algorithm 2) or y_3 increases (by means of Algorithm 1 or Algorithm 2); b) the record corresponding to x_2 is deleted (by means of Algorithm 1 or Algorithm 2) or y_2 increases more than 8 (by means of Algorithm 1) or y_2 increases (by means of Algorithm 2); c) the record corresponding to x_1 is deleted (by means of Algorithm 1 or Algorithm 2) or y_1 increases more than 8 (by means of Algorithm 1) or y_1 increases (by means of Algorithm 2); d) the record corresponding to x_4 is deleted (by means of Algorithm 1 or Algorithm 2) or y_4 increases more than 8 (by means of Algorithm 1) or y_4 increases (by means of Algorithm 2); e) the record corresponding to x_5 is deleted (by means of Algorithm 1 or Algorithm 2) or y_5 increases more than 8 (by means of Algorithm 1) or y_5 increases (by means of Algorithm 2); f) the record corresponding to x_6 is deleted (by means of Algorithm 1 or Algorithm 2) or y_6 increases more than 8 (by means of Algorithm 1) or y_6 increases (by means of Algorithm 2).

EXAMPLE 10. *Given Table 3, if the user gets the answers to the max queries $q_1 = \{1, 2, 3, 4\}$ and $q_2 = \{2, 3, 5, 6\}$, with $m_1 = m_2 = 8$, then the BN encoding user knowledge is shown in Figure 5.*

If the record corresponding to x_3 (resp. to x_2) is deleted from the DB, then the user does not know if m_1 and m_2 are equal or not to 8, but he knows that each y_i with $i \neq 3$ (resp. $i \neq 2$) is less or equal to 8. The corresponding BN is shown in Figure 6 a) (resp. Figure 6 b)).

If the record corresponding to x_1 (resp. to x_4) is deleted from the DB, then the user does not know if m_1 is equal or not to 8, but he knows that each y_i with $i \neq 1$ (resp. $i \neq 4$) is less or equal to 8 and he knows the information derived from evidence on m_2=8. The corresponding BN is shown in Figure 6 c) (resp. Figure 6 d)).

If the record corresponding to x_5 (resp. to x_6) is deleted from the DB, then the user does not know if m_2 is equal or not to 8, but he knows that each y_i with $i \neq 5$ (resp. $i \neq 6$) is less or equal to 8 and he knows the information derived from evidence on m_1=8. The corresponding BN is shown in Figure 6 e) (resp. Figure 6 f)).

EXAMPLE 11. *Given Table 3, if the user gets the answers to the max queries $q_1 = \{1, 2, 3, 4\}$ and $q_2 = \{2, 3, 5, 6\}$, with $m_1 = m_2 = 8$, then the BN encoding user knowledge is shown in Figure 5.*

If y_3 increases (resp. y_2 increases more than 8), then m_1 and m_2 are not equal to 8; thus, we store only user knowledge about y_i with $i \neq 3$ (resp. $i \neq 2$), that is $y_i \leq 8$. See Figure 6 a) (resp. Figure 6 b)).

If y_1 (resp. y_4) increases more than 8, then m_1 is not equal to 8; thus, we store only user knowledge about y_i with $i \neq 1$ (resp. $i \neq 4$), that is $y_i \leq 8$, and the information derived from evidence on m_2=8. See Figure 6 c) (resp. Figure 6 d)).

If y_5 (resp. y_6) increases more than 8, then m_2 is not equal to 8; thus, we store only user knowledge about y_i with $i \neq 5$ (resp. $i \neq 6$), that is $y_i \leq 8$, and the information derived from evidence on m_1=8. See Figure 6 e) (resp. Figure 6 f)).

If y_3 decreases, then it can be that m_1 or m_2 is not equal to 8; moreover, if $y_2 < 8$, then the temporal transformations for q_1 and q_2 are not well posed. Thus, we store only user knowledge about y_i ($i = 1, 2, 3, 4, 5, 6$) but not about the max nodes. See Figure 7.

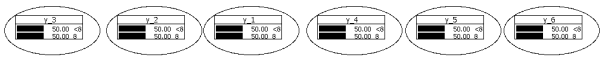

Figure 7: Update BN in Figure 5 by means of Algorithm 1 when y_3 decreases.

If y_2, y_1, y_4, y_5, or y_6 increases less than 8 (or decreases), then m_1 and m_2 are still equal to 8 and we do not need to update the BN; it remains as in Figure 5.

Algorithm 1 describes how to update the BN. In the algorithm, we use the following notation:

- $MaxNode$ is a node encoding a max query or a max subquery.
- $TypeModify$ indicates the kind of update. It is equal to 'M' if the sensitive value $y_{\bar{i}}$ changes; it is equal to 'D' if the record corresponding to $x_{\bar{i}}$ is deleted from the database;
- if $TypeModify$ is equal to 'M' then $newValue$ is the new value of $y_{\bar{i}}$;
- if $MaxNode$ has not children then $children.size = 0$;
- if $MaxNode$ is not an evidence node then $evidenceIsEntered = false$;
- $\mu_{\bar{i}}$ is the upper bound of $y_{\bar{i}}$ before of its modification.

Algorithm 1

for each MaxNode containing $\bar{i}$ **do**
  if $y_{\bar{i}}$ is the first node in the temporal transformation of MaxNode **then**
    delete MaxNode
  else
    if (TypeModify=='M' AND $\mu_{\bar{i}} < newValue$) OR (TypeModify=='D') **then**
      delete MaxNode
    end if
  end if
end for
while exists MaxNode such that:
children.size=0 AND evidenceIsEntered=false **do**
  delete MaxNode
end while
if (TypeModify=='M' AND $\mu_{\bar{i}} < newValue$) OR (TypeModify=='D') **then**
  delete node encoding $y_{\bar{i}}$
end if

3.2.2 The user knows if a sensitive value increases or decreases

Under the hypotheses that the user knows if a sensitive value $y_{\bar{i}}$ increases or decreases, then:

1. let q_j be a max a query such that $\bar{i} \in q_j$. If the user knows that the record corresponding to $x_{\bar{i}}$ is deleted, then he knows that $m_{q_j \setminus \{\bar{i}\}} \leq m_j$ (as in Section 3.2.1);
2. let q_j be a max a query such that $\bar{i} \in q_j$ and $m_j = \mu_{\bar{i}}$. If the user knows that the value of $y_{\bar{i}}$ increases, then he does not if $y_{\bar{i}} \leq m_j$ or $y_{\bar{i}} > m_j$;
3. let q_j be a max a query such that $\bar{i} \in q_j$ and $m_j = \mu_{\bar{i}}$. If the user knows that the value of $y_{\bar{i}}$ decreases, then he knows that $y_{\bar{i}} < m_j$.

EXAMPLE 12. *If the user gets the answers to the max queries $q_1 = \{1, 2, 3, 4\}$ and $q_2 = \{2, 3, 5, 6\}$, with $m_1 = m_2 = 8$, then the BN encoding user knowledge is shown in Figure 5.*

If the user knows that y_3 increases (resp. y_2), then he has not information about y_3 (resp. y_2), because one the following cases are possible: $y_3 < 8$ (resp. $y_2 < 8$); $y_3 = 8$ (resp. $y_2 = 8$); $y_3 > 8$ (resp. $y_2 > 8$). As a consequence, he has not information about m_1 and m_2, he knows only that $y_i \leq 8,\ \forall i \neq 3 (resp.\ i \neq 2)$. See Figure 6 a) (resp. Figure 6 b)).

If the user knows that y_1 increases (resp. y_4), then he has not information about y_1 (resp. y_4). As a consequence, he has not information about m_1, he knows only that $y_i \leq 8,\ \forall i \neq 1 (resp.\ i \neq 4)$ and the information derived from evidence on $m_2 = 8$. See Figure 6 c) (resp. Figure 6 d)).

If the user knows that y_5 increases (resp. y_6), then he has not information about y_5 (resp. y_6). As a consequence, he has not information about m_2, he knows only that $y_i \leq 8,\ \forall i \neq 5 (resp.\ i \neq 6)$ and the information derived from evidence on $m_1 = 8$. See Figure 6 e) (resp. Figure 6 f)).

If the user knows that y_3 decreases (resp. y_2), then he knows that $y_3 < 8$ (resp. $y_2 < 8$) and that $m_1 \leq 8$ and $m_2 \leq 8$. Thus, we add evidence on the node encoding y_3 (resp. y_2), and remove evidence on the max nodes encoding q_1 and q_2. In alternative to removing evidence on the max nodes, we obtain the same probabilities for nodes $y_i\ \forall i \neq 3 (resp.\ i \neq 2)$ if we delete all max nodes and we store only node encoding $y_i \leq 8,\ \forall i \neq 3 (resp.\ i \neq 2)$. See Figure 8 a) (resp. Figure 8 b)).

If the user knows that y_1 decreases (resp. y_4), then he knows that $y_1 < 8$ (resp. $y_4 < 8$), that $y_i \leq 8,\ \forall i \neq 1 (resp.\ i \neq 4)$ and information derived from evidence on $m_2 = 8$. See Figure 8 c) (resp. Figure 8 d)).

If the user knows that y_5 decreases (resp. y_6), then he knows that $y_5 < 8$ (resp. $y_6 < 8$), that $y_i \leq 8,\ \forall i \neq 5 (resp.\ i \neq 6)$ and information derived from evidence on $m_1 = 8$. See Figure 8 e) (resp. Figure 8 f)).

Algorithm 2 describes how to update the BN. We use the same notation used in Algorithm 1, moreover $oldValue$ denotes the old value of $y_{\bar{i}}$ before the update of the DB.

4. EXPERIMENTATION

The experimentation is conducted on a computer with the following properties: HP Compaq dc7100; Pentium(R) 4 CPU 2.80 GHz; 2 GB of RAM. In the experimentation, set tolerance (see Definition 1) equal to tol = 0.8 and we run sequences of 100 queries. We consider a baseball dataset in [17]; it consists of 377 records. We have added a field ID, in such way that $(ID, Salary)$ is the private association, with ID the field identifying the baseball player

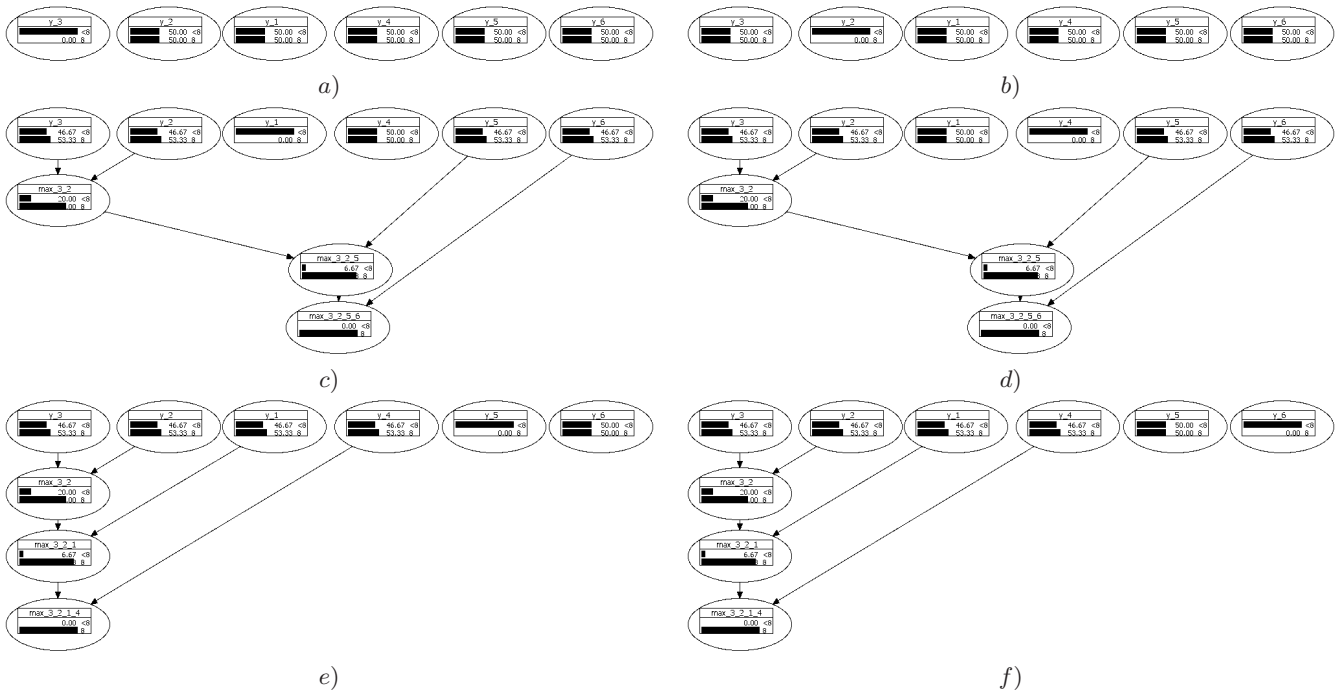

Figure 8: We update the BN in Figure 5 by means of Algorithm 2, when: a) y_3 decreases, b) y_2 decreases, c) y_1 decreases, d) y_4 decreases, e) y_5 decreases, f) y_6 decreases.

Algorithm 2

for each MaxNode containing $\bar{i}$ **do**
  delete MaxNode
end for
while exists MaxNode such that:
children.size=0 AND evidenceIsEntered=false **do**
  delete MaxNode
end while
if TypeModify=='M' **then**
  if newValue-oldValue>0 **then**
    delete node encoding $y_{\bar{i}}$
  end if
  if newValue-oldValue<0 **then**
    set evidence on the first state of the node encoding $y_{\bar{i}}$
  end if
end if
if TypeModify=='D' **then**
  delete node encoding $y_{\bar{i}}$
end if

and $Salary$ the sensitive field. The dataset comprises 210 distinct values of $Salary$. Each max query is generated in random way with length in the range $[2, ..., n]$. We have conducted experimentation about Algorithm 1 and Algorithm 2; in the following order we have:

1. executed 50 random max queries;
2. updated the dataset with 7 insertions, 7 deletions and 56 modification of the sensitive values;
3. executed 50 random max queries.

Figure 9 a) shows that if we run Algorithm 1 then the CPT size decreases to around 960 after the updates (insertions, deletions and modification of the sensitive values) else if we run Algorithm 2 then it decreases to around 730 (see Figure 9 b). Moreover the CPT size increases again around 1260 and 1290 after 100 queries respectively by means of Algorithm 1 and Algorithm 2.

Thus, the results, in Figure 9 a) and 9 b), suggest that the hypothesis of the additional user knowledge, about the modifications of the sensitive values, allows us to optimize the CPT size after the DB updates; however, after 100 queries, the difference between the two CPT size is very small.

Finally, in order to analyze the utility of our auditor model, we consider the probability to deny; intuitively, it seems that the more an auditing scheme denies, the less useful it is. From a comparison between the probability to deny, we can see from Figure 9 c) and Figure 9 d) that it rises around to 0.5 with Algorithm 1 and around to 0.4 with Algorithm 2, after some 100 queries.

We can see that, after the updates, the probability to deny in Figure 9 c) increases more speedily than the probability to deny in Figure 9 d).

This preliminary results suggest that it is reasonable to consider additional user prior knowledge, that is to consider the case in which the user knows if a sensitive value increases or decreases, thus dealing with the on-line max auditing in dynamic databases by means of Algorithm 2. However, further experimentation is needed.

5. CONCLUSIONS AND FUTURE WORK

We propose a method to reasoning under uncertainty in on-line auditing of dynamic statistical databases; in particular, we consider the case in which records are added or deleted, or some sensitive

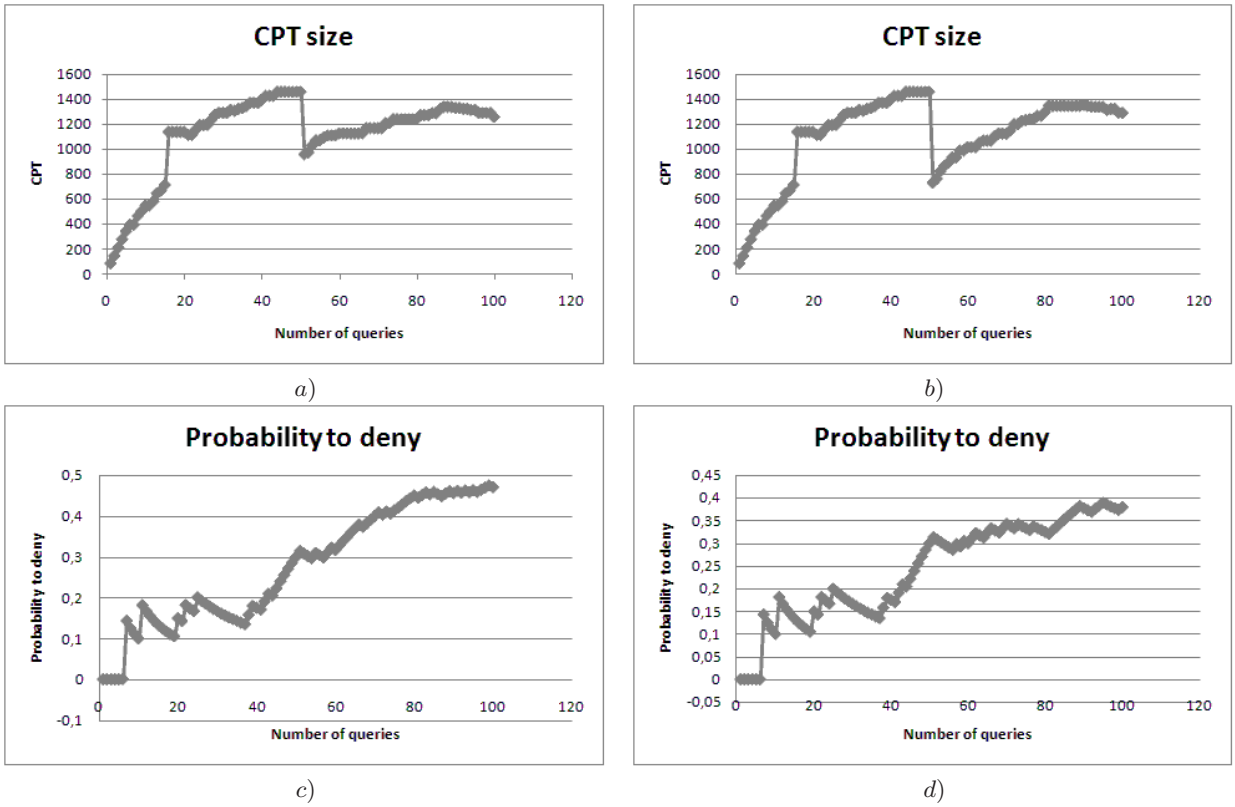

Figure 9: CPT size and probability to deny after 100 queries. a) Algorithm 1, b) Algorithm 2, c) Algorithm 1, d) Algorithm 2.

values change their value.

The method extends the Bayesian approach presented in [2], [3] and [4] for static databases and, as in the previous works, the model is able to:

- deal with on-line max auditing without maintaining query logs;
- deal with a probabilistic definition of privacy;
- provide a graphical representation of user knowledge;
- consider the case in which denial leaks information.

The paper introduces two algorithms to update the network whenever the database changes and discusses results of a preliminary set of experimental trials.

The goal of our future work is twofold:

1. to realize a further experimentation, aimed at optimizing the algorithms;
2. to deal with on-line auditing of other statistical queries, including joined auditing of max and min queries.

6. REFERENCES

[1] N. R. Adam and J. C. Wortmann. Security-control methods for statistical databases: A comparative study. *ACM Comput. Surv.*, 21(4):515–556, 1989.

[2] G. Canfora and B. Cavallo. A bayesian approach for on-line max and min auditing. In *Proocedings of International workshop on Privacy and Anonymity in Information Society (PAIS)*, pages 12–20. ACM DL, 2008.

[3] G. Canfora and B. Cavallo. A bayesian approach for on-line max auditing. In *Proocedings of The Third International Conference on Availability, Reliability and Security (ARES)*, pages 1020–1027. IEEE Computer Society Press, 2008.

[4] G. Canfora and B. Cavallo. Reasoning under uncertainty in on-line auditing. In *Privacy in Statistical Databases, Lecture Notes in Computer Science*, volume 5262, pages 257–269. Springer-Verlag Berlin Heidelberg, 2008.

[5] B. Cavallo. *Data Privacy in Statistical Databases. A Bayesian approach to deal with user uncertain knowledge in on-line auditing*. PhD thesis, University of Sannio, Software Engineering, June 2008. http://plone.rcost.unisannio.it/canfora/downloads/Bicedissertation.pdf/view.

[6] F. Y. Chin. Security problems on inference control for sum, max, and min queries. *Journal of the ACM*, 33(3):451–464, July 1986.

[7] D. Heckerman. Causal independence for knowledge acquisition and inference. *In Proceedings of Ninth Conference on Uncertainty in Artificial Intelligence*, pages 122–127, 1993.

[8] K. Kenthapadi, N. Mishra, and K. Nissim. Simulatable auditing. *In PODS*, pages 118–127, June 2005.

[9] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Auditing boolean attributes. *Journal of Computer and System Sciences*, 66(1):244–253, February 2003.

[10] F. M. Malvestuto, M. Mezzini, and M. Moscarini. Auditing sum-queries to make a statistical database secure. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):31–60, February 2006.

[11] M. McLElSH. Further results on the security of partitioned dynamic statistical databases. *ACM Transactions on Database Systems*, 14(1):98–113, March 1989.

[12] S. U. Nabar, B. Marthi, K. Kenthapadi, N. Mishra, and R. Motwani. Towards robustness in query auditing. In *International Conference on Very Large Data Bases*, pages 151–162, 2006.

[13] K. G. Olesen, U. Kjaerulff, F. Jensen, F. V. Jensen, B. Falck, S. Andreassen, and S. K. Andersen. A munin network for the median nerve - a case study in loops. *Applied Artificial Intelligence*, 3(2-3):385–403, 1989.

[14] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Francisco, CA, USA, 1998.

[15] S.-P. Shieh and C.-T. Lin. Auditing user queries in dynamic statistical databases. *Information Sciences*, 113(1-2):131–146, January 1999.

[16] S. Srinivas. A generalization of the noise-or-model. In *Ninth Annual Conference of Uncertainty on AI*, pages 208–218, 1993.

[17] M. Watnik. Pay for play: Are baseball salaries based on performance. *Journal of Statistics Education*, 6, 1998.