Probabilistic Model for Accuracy Estimation in Approximate Monodimensional Analyses

CARLO DELL'AQUILA, FRANCESCO DI TRIA, EZIO LEFONS, FILIPPO TANGORRA
Dipartimento di Informatica
Università degli Studi di Bari ALDO MORO
Via Orabona 4, 70125 Bari
ITALY
{dellaquila, lefons, tangorra, francescoditria}@di.uniba.it
http://www.di.uniba.it/~dblab

Abstract: - Approximate query processing is often based on analytical methodologies able to provide fast responses to queries. As a counterpart, the approximate answers are affected with a small quantity of error. Nowadays, these techniques are being exploited in data warehousing environments, because the queries devoted to extract information involve high-cardinality relations and, therefore, require a high computational time. Approximate answers are profitably used in the decision making process, where the total precision is not needed. Thus, it is important to provide decision makers with accuracy estimates of the approximate answers; that is, a measure of how much reliable the approximate answer is. Here, a probabilistic model is presented for providing such an accuracy measure when the analytical methodology used for decisional analyses is based on polynomial approximation. This probabilistic model is a Bayesian network able to estimate the relative error of the approximate answers.

Key-Words: Analytic query processing, Approximate query answer, Polynomial approximation, Accuracy estimation, Probabilistic model.

1 Introduction

In data warehousing (DW) environments, On-Line Analytical Processing (OLAP) is devoted to extract information to be used in decision making activities. DW databases commonly store high volumes of data and, therefore, even a simple scanning for data aggregation requires answer times that may range from minutes to hours [1, 2]. Moreover, in case of a distributed system, the access to remote data can be sometimes impracticable (due to the server and/or communication line crash, for example).

The traditional query processing engines always provide decision makers with exact query answers. However, in many cases, the total precision is not always required by final users. Indeed, there are several scenarios in OLAP where it is not mandatory to obtain exact query answers. As an example, in the drill-down task, the preliminary queries are used only to determine the most important facets to consider in decision making. In fact, analytical processing has always an unpredictable and exploratory nature. As a further example, in numerical answers that require the computation of the average of a large set of data, the total precision is not needed and an approximate value will suffice. At last, in Database Management Systems, the optimizers must define a plan for the physical data access on the basis of the selectivity estimates [3].

These issues have led to define methodologies able to provide approximate query answers [4], especially in data warehousing environments [5], such that the final users get fast responses, although affected with a small quantity of error. In fact, this research topic is based on the assumption that, if the query is very time-consuming and the error in the approximation is neglectable, then it is more suitable to have an approximate answer in a short time, rather than the exact answer after a long waiting time.

What is an approximate answer? As concerns a query that involves data aggregation, the answer is a scalar value and the relative approximate answer is an estimation of this value. The most popular methodologies that represent the theoretical bases for approximate query processing are based on sampling [6], histograms [7], wavelets [8, 9], probabilistic models [10, 11, 12], distributed processing [13], clustering [14], orthonormal series approximation [15], genetic programming [16], and graph-based models [17]. Most of them need to perform a reduction of the data stored in database relations. In fact, these methodologies provide approximate answers using small and pre-computed data synopses, obtained by compressing the original data [18].

According to [19], the criteria for comparing the methodologies for approximate query processing

are: (a) *coverage*, or the kind of queries for which it is possible to provide approximate answers; (b) *response time*, or the time needed for the computation of the approximate value; (c) *accuracy*, or the confidence degree in the approximation; (d) *update time*, or the time to compute the data synopsis; and (e) *footprint*, or the space used to store data synopses.

In this paper, we consider the well-known methodology based on polynomial approximation [20], and we extend it in order to provide the accuracy along with the approximate answer. The accuracy is computed via a probabilistic inferential process based on a Bayesian network. This allows us to model the relationships among the main stochastic variables involved in the approximate query processing that determine the relative error.

The paper is organized as follows. Section 2 recalls the methodology for polynomial approximation. Section 3 introduces the Bayesian networks. Section 4 presents our novel probabilistic model able to estimate the relative error in the approximate query processing. Section 5 reports experimental results. Section 6 presents future work based on the experimental evidence. Finally, Section 7 contains our conclusions.

2 Analytic Methodology

The analytic methodology consists of using orthonormal polynomial series to approximate the univariate data distribution function of an attribute X. The utilized approximation polynomial is the Legendre orthogonal polynomial series and its calculated coefficients carry synthetic information about the univariate data distribution of the attribute X. A complete discussion of this approach and the extension to the multivariate data distribution can be found in [20].

2.1 Coefficients Computation

Let R(X) be a relation of cardinality n on schema X, and let $X \in\in$X. We assume $dom(X) = [a, b]$ be the numeric interval denoting the domain of the attribute X. Finally, let $pdf(x)$ be the probability density function of $R.X$, where we use the dot notation to denote the attribute X of the relation R(X). We denote with $g(x)$ its polynomial approximation up to degree d. Since the Legendre orthogonal polynomials are defined on the interval $[-1, +1]$, each value $x \in dom(X)$ is suitably mapped to the corresponding value $x' \in [-1, +1]$.

Then, for each $x \in X$, it results that

$$pdf(x) \approx g(x) = \frac{1}{2}\sum_{i=0}^{d}(2i+1)c_i P_i(x'), \quad (1)$$

where, for $i = 0, 1, \ldots, d$,

- $x \to x'$ is the opportune map from $x \in X$ to $x' \in [-1, +1]$,
- $P_i(x')$ is the Legendre polynomial up to degree i, and
- $c_i = \sum_{x \in X} P_i(x') / n$ is the mean value of $P_i(x')$ on the n tuples of $R.X$.

Therefore, $g(x)$ is the orthogonal polynomial approximation to $pdf(x)$ up to degree d and the coefficients $\{c_i \mid i = 0, \ldots, d\}$ carry information about the univariate data distribution of the attribute X. These coefficients are the so-called *Canonical Coefficients* of X and they can be used in order to perform monodimensional analyses of X, by calculating aggregate functions, such as *count*, *sum*, and *average* (*see*, Subsection 2.2).

Assuming that:

$$cdf(x) = \int_{a}^{x} pdf(y)dy \quad (2)$$

and

$$\int_{-1}^{x'}(2i+1)P_i(y)dy = P_{i+1}(x') - P_{i-1}(x') \equiv Q_i(x'), \quad (3)$$

for $i = 0, 1, \ldots, d$, where $Q_0(x') \equiv x'$, it is possible to compute the cumulative density function $cdf(x)$ of X in the following way:

$$\begin{aligned} cdf(x) &\approx G(x) \equiv G(x') \\ &= \int_{-1}^{x'} g(y)dy = \frac{1}{2}\sum_{i=0}^{d} c_i Q_i(x'), \end{aligned} \quad (4)$$

where, for all $x \in X$, $G(x)$ is the polynomial approximation of the *cdf* of X.

2.2 Monodimensional Analysis

The analytical process based on the set of precomputed Canonical Coefficients provides an approximation of typical aggregate functions, such as *sum*, *average*, and *count*. Let $I = [x, y] \subseteq [a, b]$ be the generic query range used for the computation of the aggregate function and let $I' = [x', y'] \subseteq [-1, +1]$ be the corresponding interval on the domain of the Legendre function. Given such an interval I, the main aggregate function, namely *percent* (or *selectivity*), is $p(x \le X \le y)$ and it can be estimated by

$$percent(I) \approx G(y') - G(x') = \frac{1}{2}\sum_{i=0}^{d} c_i Q_i(I'), \quad (5)$$

where $Q_i(I') = Q_i(y') - Q_i(x')$, for $i = 0, 1, \ldots, d$.

Then, the *count* aggregate function on the query range I can be estimated as

$$count(I) \approx n \times percent(I), \quad (6)$$

where n is the cardinality of the relation R.

Moreover, the *average* and the *sum* functions can be estimated as

$$\begin{aligned} &average(I) = H(I) / percent(I), \text{ and} \\ &sum(I) \equiv average(I) \times count(I) \approx n \times H(I), \end{aligned} \quad (7)$$

where $H(I) \equiv \int_I xg(x)dx$.

3 Overview of Probability Concepts

Let $\{V_1, V_2, \ldots, V_k\}$ be a set of k stochastic variables and let v_i denote the value taken on by the variable V_i, for $i = 1, \ldots, k$. If the variable V_i indicates a proposition, then its possible values are *true* and *false*. If V_i represents a measure or physical entity (such as age, weight, or speed), then its values are numbers that may range in discrete or continuous domains. If V_i represents a category, then its values are categorical and the variable is defined multinomial. As an instance, the weather can be *sunny*, *cloudy*, *snowfall*, and *rainy*.

Given the stochastic variable E, $p(E)$ denotes the *a priori* probability of E. As an instance, if E is the *weather* multinomial variable above, then the probability function assigns reals in [0, 1] to all values of E: $p(weather=sunny) = 0.7$, $p(weather=rainy) = 0.2$, $p(weather=cloudy) = 0.08$, and $p(weather=snowfall) = 0.02$, for example. Therefore, *weather* = ⟨*sunny, rainy, cloudy, snowfall*⟩ and $p(weather)$ denotes the vector ⟨0.7, 0.2, 0.08, 0.02⟩ of probabilities associated to the corresponding categories of the stochastic variable. This vector defines the probability distribution of the variable *weather*.

In general, a discrete probability function is such that:

$$\sum_{l=1}^{r} p(V = v_l) = 1, \quad (8)$$

where V is a multinomial variable that assumes r distinct values.

There are several definitions to assign *a priori* probabilities to stochastic variables [21].

According to the *classic* definition, given an event E, $p(E)$ is the ratio of the number of cases favourable to its occurrence to the total numbers of cases, all equally possible and mutually exclusive:

$$p(E) = \frac{\text{number of positive cases}}{\text{number of possible cases}}. \quad (9)$$

As an example, in a coin launch, each face of the coin has probability 0.5 to be verified.

According to the *frequentist* definition, when an experiment is repeated r times, if an event E happens s_r times, then the ratio of s_r to r provides the relative frequency of E in reference to the given repetitions of the experiment. Therefore, $p(E)$ is the limit of the relative frequency, when the number of repetitions of the experiment increases indefinitely:

$$p(E) = \lim_{r\to\infty} \frac{s_r}{r}. \quad (10)$$

At last, according to the *subjective* definition, given the event E, $p(E)$ is the degree of confidence that a person assigns to the occurrence of E.

The axioms of the probability theory are:

- if E is a stochastic variable, then $0 \le p(E) \le 1$,
- necessarily true propositions have probability value 1, while the unsatisfiable ones have probability 0 (*ie*, $p(true) = 1$ and $p(false) = 0$), and
- the probability of a disjunction is given by

$$p(A \vee B) = p(A) + p(B) - p(A \wedge B). \quad (11)$$

We denote the joint probability that $v_1, v_2, \ldots, v_k$ be the respective values of $V_1, V_2, \ldots, V_k$ simply by $p(V_1=v_1, V_2=v_2, \ldots, V_k=v_k)$ or $p(v_1, v_2, \ldots, v_k)$.

The function that assigns a number in [0, 1] to the set of stochastic variables is called *joint probability function* and its obvious properties are:

- $0 \le p(v_1, v_2, \ldots, v_k) \le 1$, and
- $\sum_{\mathbf{V}} p(v_1, v_2, \ldots, v_k) = 1$, where $(v_1, v_2, \ldots, v_k)$ range over $\mathbf{V} = V_1 \times V_2 \times \ldots \times V_k$ or the set of values of the stochastic variables.

If the joint probabilities of all the values of a set of stochastic variables are known, then the so-called *marginal probability* of each variable can be calculated. As an instance, the marginal probability of V_1 at v_1 is

$$p(V_1 = v_1) = \sum_{\mathbf{V}} p(v_1, V_2, \ldots, V_k), \quad (12)$$

where $p(V_1, V_2, \ldots, V_k)$ is the vector of the joint probabilities.

Example 1. Let A and B be two propositional variables. For simplicity, we shall write $p(A)$ instead of $p(A=true)$ and $p(\neg A)$ for $p(A=false)$.

Assuming the following joint probabilities:

$p(A, B) = 0.2$,
$p(A, \neg B) = 0.3$,
$p(\neg A, B) = 0.4$, and
$p(\neg A, \neg B) = 0.1$,

the marginalization of A is

$p(A) = p(A, B) + p(A, \neg B) = 0.5$, and
$p(\neg A) = p(\neg A, B) + p(\neg A, \neg B) = 0.5$. □

The probability that $V_i = v_i$ when $V_j = v_j$ is called *conditional probability of* V_i *given* V_j and it is defined as

$$p(V_i = v_i \mid V_j = v_j) = \frac{p(V_i = v_i, V_j = v_j)}{p(V_j = v_j)}. \quad (13)$$

It is possible to define the joint probability in terms of conditional probability, using the following expression, known as *product rule*:

$$p(V_i = v_i, V_j = v_j) = p(V_i = v_i \mid V_j = v_j)\, p(V_j = v_j). \quad (14)$$

The generalization of the product rule allows expressing the joint probability of a set of stochastic variables in terms of conditional probabilities. Thus, the general form of the product rule is

$$p(v_1, v_2, \ldots, v_k) = \prod_{i=1}^{k} p(v_i \mid v_{i-1}, \ldots, v_1). \quad (15)$$

Let A and B be two stochastic variables. Then, with reference to the product rule, we obtain that

$$p(A, B) = p(A \mid B)\, p(B), \text{ and } p(B, A) = p(B \mid A)\, p(A).$$

Since

$$p(A, B) = p(B, A),$$

then

$$p(A \mid B)\, p(B) = p(B \mid A)\, p(A).$$

Thus,

$$p(B \mid A) = \frac{p(A \mid B)\, p(B)}{p(A)}. \quad (16)$$

Equation (16) is known as *Bayes' rule* and it allows to perform a probabilistic reasoning [22, 23].

3.1 Probabilistic Inference

The general form of the probabilistic inference is based on a set $V = \{V_1, V_2, \ldots, V_k\}$ of k stochastic variables and the evidence e (*ie*, the 100% of truthness) that the variables of a set $E \subseteq V$ assume defined values. In this way, it is possible to compute the conditional probability $p(V_i = v_i \mid E = e)$ of V when E is known. This process is called *probabilistic inference*. As an example, if we know that the probability it rains is 80% when the weather is cloudy, that is, there exists the conditional probability $p(\text{rainy} = \textit{true} \mid \text{cloudy} = \textit{true}) = 0.8$, and we have the evidence that today it is cloudy, then the probability that today it rains is 80%.

To apply the probabilistic inference using joint probabilities requires the knowledge of the full joint distribution. Therefore, in case of k propositional variables (*ie*, stochastic variables whose values are *true* or *false*), we need the list of 2^k values of the joint probability $p(V_1, V_2, \ldots, V_k)$. Indeed, for many problems of interest, we could never obtain such a distribution. Owing to this kind of intractability, a more efficient probabilistic reasoning is used, based on the conditional independence existing among stochastic variables.

The variable A is *conditionally independent from the variable* B, *given* C, if it holds that

$$p(A \mid B, C) = p(A \mid C). \quad (17)$$

Intuitively, the conditional independence of A from B, given C, states that B does not provide information about A, but all information is given by C.

Conditional independences can be represented by graphs, or the so-called *Bayesian Networks* (or *belief networks*). In detail, these graphs allow modelling the cause-effect relationships existing among stochastic variables and, thus, they are very useful in the probabilistic inference, for they allow computing the conditional probabilities in a very fast way.

3.2 Bayesian Networks

A Bayesian Network is a Machine Learning technique used to perform probabilistic reasoning [24]. The network is represented by an acyclic, oriented graph, whose nodes are labelled with the names of the stochastic variables and the edges represent the cause-effect relationships between variables. On the basis of the conditional independence, such a network requires that each node is influenced only by its parents. In a Bayesian Network, the following properties hold:

- a set of stochastic variables form the nodes of the graph,
- a set of oriented edges connect couples (parent, descendant) of nodes,
- each node has the associated table of the conditional probabilities that summarize the effects the parents have on the node itself, and
- the graph has not cycles.

Therefore, the existence of an edge from the node A to B states that A influences B directly.

Let $V_1, V_2, \ldots, V_k$ be a set of k nodes of a Bayesian Network. Under the assumption of conditional

independence, we can compute the joint probabilities of each node of the network as:

$$p(V_1, V_2, \ldots, V_k) = \prod_{i=1}^{k} p(V_i \mid parents(V_i)) , \quad (18)$$

where $parents(V_i)$ is the set of the parents of V_i.

Notice that we must know the conditional probabilities of each node with respect to its parents, in order to compute the joint probabilities. Nodes without parents are not conditioned by any other node and their *a priori* probabilities must be provided.

Example 2. Let us consider the network shown in Figure 1. The nodes B, S, I and M are propositional variables that recall a well-known case [22]: a block can be moved (M) whether the battery is charged (B) or whether the item itself can be elevated (S); moreover, an indicator (I) states whether the battery is charged or not. There are given the *a priori* probabilities of B and S, and the conditional probabilities of I and M with respect to their own parents. The joint probability $p(B,S,I,M)$ is better computed as

$$p(B,S,I,M) = p(B)\, p(S)\, p(I \mid B)\, p(M \mid B,S),$$

rather than using the complex expression

$$p(B,S,I,M) = p(B)\, p(S \mid B)\, p(I \mid B,S)\, p(M \mid B,S,I)$$

obtained by the product rule.

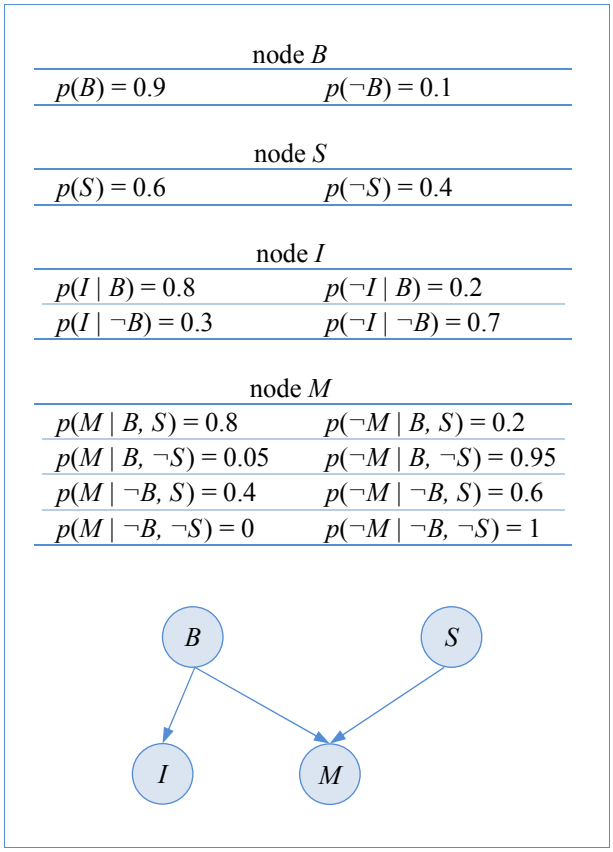

Fig. 1. *Bayesian network.*

Notice that only 8 probabilities are essential—*eg*, the ones on the left column—as the probabilities on the right column are easily derived. As an instance,

$$p(B) = 0.9 \Rightarrow p(\neg B) = 1 - p(B) = 0.1.$$

On the other hand, the product rule expression would require the specification of all the $2^4 = 16$ joint probabilities. □

In a Bayesian Network, we distinguish three types of categories that allow different probabilistic reasonings, depending on how the evidence is propagated in the network: *causal inference* (from causes to effects) or top-down, *intercausal inference* (among causes of a shared effect) or explaining away, and *diagnostic inference* (from effects to causes) or bottom-up. Further, the types can be merged in order to produce mixed inference.

The types of probabilistic inference among nodes A, B and C are represented as graphs in Figure 2.

The activity of assigning probabilities to nodes of a Bayesian Network is called *learning process* and the effectiveness of the network depends on both its ability to represent the knowledge of the domain (*ie*, all and only the relationships among the variables), and the goodness of the dataset used for the learning process (*ie*, the training set).

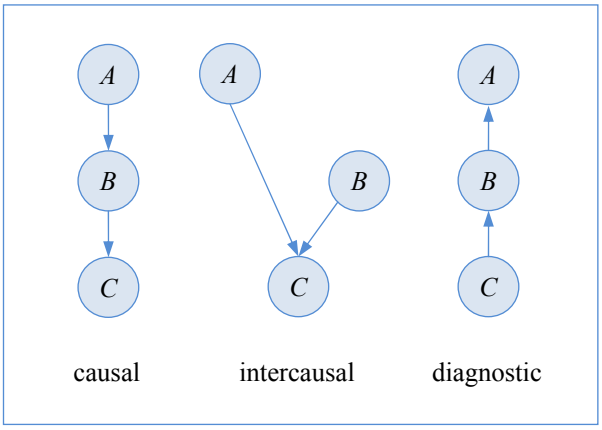

Fig. 2. *Types of probabilistic inference.*

3.3 Graphical Modelling

Bayesian Network tools in Java (BNJ) [25] is a graphical tools suite for developing Bayesian Networks. In this subsection, we show a complete example of probabilistic inference executed with BNJ. The example regards the well-known case study describing the relationships among the sky (that can be cloudy or not), the weather (that can be rainy or not), the sprinkler (that can be turned on or off), and the grass (that can be wet or not). The network is devoted to model the knowledge that the state of the sky influences the state of the weather and the one

of the sprinkler, that on turn determine the state of the grass. Once defined the network topology, the probabilities must be provided for each node on the basis of the experts' knowledge of the domain and/ or empirical studies.

Figure 3 reports such Bayesian Network in BNJ. Running the network with no evidence (*cf*, Figure 4), we discover that

$$p(\text{rainy}=\textit{true}) =$$
$$p(\text{rainy}=\textit{true} \mid \text{cloudy}=\textit{true}) \times p(\text{cloudy}=\textit{true}) +$$
$$p(\text{rainy}=\textit{true} \mid \text{cloudy}=\textit{false}) \times p(\text{cloudy}=\textit{false})$$
$$= 0.8 \times 0.3 + 0.2 \times 0.7 = 0.38.$$

In the same way, we discover that the probability the grass is wet is 60.8%. Putting the evidence on a node, the probabilities are updated. As an example, if we are certain that it is cloudy, then it is possible to infer that the probability the grass is wet is equal to 74.5% (*cf*, Figure 5).

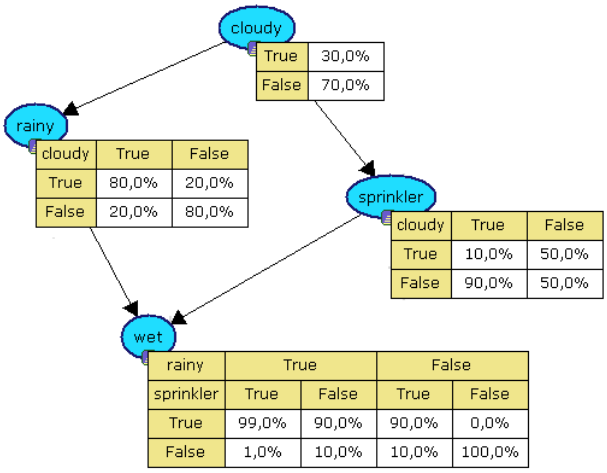

Fig. 3. Bayesian network in BNJ.

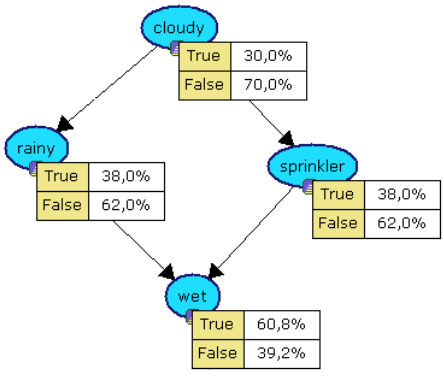

Fig. 4. *Marginalization.*

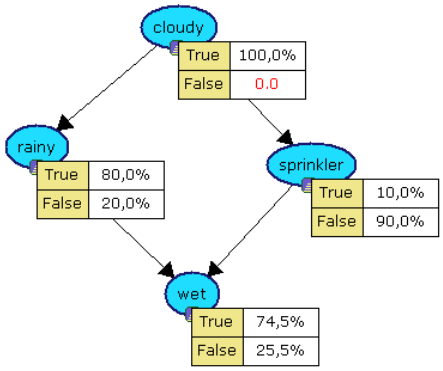

Fig. 5. *Probabilistic inference.*

4 Probabilistic Modelling

In this subsection, we introduce the main variables affecting the relative error in the polynomial approximation. Then, we use the relative error as a measure of the accuracy of the approximate answer.

The main factors that influence the relative error are (1) the degree of polynomial approximation, (2) the cardinality of relations, and (3) the query range (*ie*, the width of the interval of the queries).

The first factor does not need explanation, since it is well-known that the higher the polynomial degree the better the approximation to the probability density function of data attributes. However, for opportunity, we fixed the min, medium, and max degrees equal to 7, 17, and 27, respectively. Indeed, the polynomial degree we used for computing the canonical coefficients is the max degree only and, therefore, we assume that the computed coefficients are not affected by approximation errors. On the contrary, when computing aggregate functions in approximate analyses, we allow choosing the approximation function with the min, medium, or max degrees.

The cardinality of relations has influence on the relative error for it determines the quality of the coefficients used to compute the aggregate functions. In fact, every coefficient is the mean of n quantities, where n is the cardinality of the data. The error in computing the coefficients is related to the finite arithmetic and machine precision. So, when the cardinality of the relation is very high (*eg*, of the order of 10^7 tuples), the mean value is affected by a truncation error. Figure 6 shows how the truncation error increases with the cardinality of relations.

Figure 7 reports the trend of the relative error of approximate answers in reference to the approximation degree and width of the query range. The chart shows that the smaller the query range, the higher the relative error. Moreover, as the query range

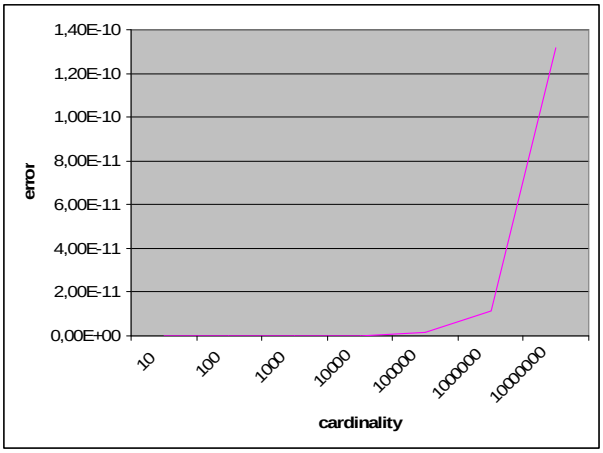

Fig. 6. *Truncation error of approximation coefficients*.

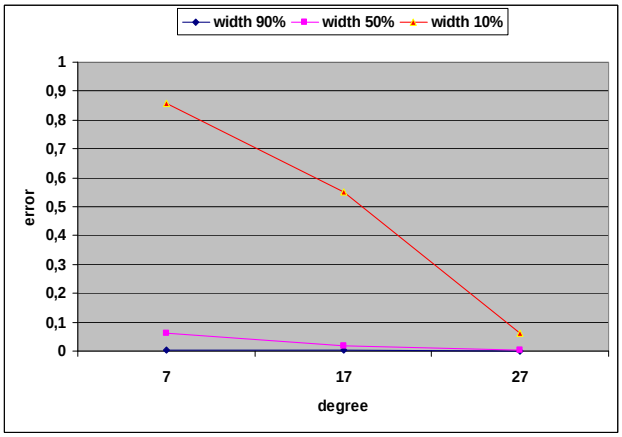

Fig. 7. *Relative error per degree and query range width.*

approaches 10%, good results are obtained only by using the maximum degree.

Figure 8 shows the probabilistic model that we have defined for the measure of the accuracy, after the learning process of the network. To assign the *a priori* probabilities to the influence factors, we have adopted the subjective definition.

This model shows that there is a high probability the approximate answers yields a medium relative error (that is, a relative error in the range [0.001, 0.01[) and a very low probability to have a high relative error (that is, greater than or equal to 0.01).

For each query answer, the network can be used at run time to estimate its relative error. For example, if we run the network with the evidence that the cardinality of the relation is about 10^5 records, the width of the query range is about 90% of the domain, and the degree is the maximum one, then the model infers that the probability to have a low relative error (that is, an error less than 0.001) is 50%.

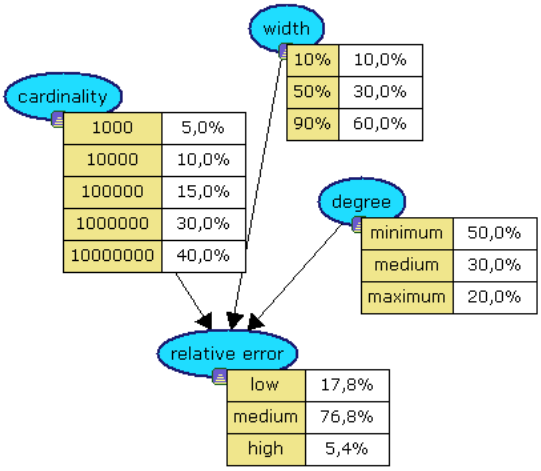

Fig. 8. *Accuracy probabilistic model.*

5 Experimental Set-up

An accuracy measure is composed of (a) error bounds, that is, the interval I which the real value is assumed to belong to, and (b) the confidence degree $p(I)$, that is, the probability that the real value falls in that interval [26].

Let r and r_α be the real query answer and its computed approximate value, respectively, and let δ be the relative error of r_α w.r.t. r. We define

$$\delta :\stackrel{\text{def}}{=} \begin{cases} \text{is } low & \text{if } \delta < 0.001, \\ \text{is } medium & \text{if } 0.001 \le \delta < 0.01, \\ \text{is } high & \text{otherwise } (\delta \ge 0.01). \end{cases}$$

The probabilistic model provides an accuracy measure, giving an estimation of the relative error according to the confidence degree. This can be used to estimate the real query answer.

As an example, suppose that $r_\alpha = 100$ and we do not know the real query answer r, but we know the conditions under which the query was executed: cardinality 10^4, width 50%, and degree *minimum*. These conditions are the factors affecting r_α. Therefore, the probabilistic model infers that $p(\delta$ is *low*$) = 10\%$, $p(\delta$ is *medium*$) = 85\%$, and $p(\delta$ is *high*$) = 5\%$. Thus, the user understands that $99.9 < r < 100.1$ with 10% of probability, $99.01 < r \le 99.9$ or $100.1 \le r < 101.01$ with 85% of probability, and $r \le 99.01$ or $r \ge 101.01$ with 5% of probability.

For the present experimentation, we executed 10 launches while varying the influence factors, namely, the cardinality of the relation, the query width, and the approximation degree. For each launch, we computed the aggregate functions *sum*, *count* and

Table 1. *Experimental set-up of probabilistic accuracy measure.*

launch #	data cardinality	query range width	approx. degree	relative error per function			error probabilistic estimation		
				sum	*count*	*avg*	*low*	*medium*	*high*
1	10^4	50 %	27	medium	medium	low	21 %	79 %	0 %
2	10^3	90 %	7	medium	medium	medium	40 %	60 %	0 %
3	10^4	10 %	7	high	high	medium	0 %	30 %	70 %
4	10^4	50 %	7	medium	medium	low	10 %	85 %	5 %
5	10^3	10 %	27	high	high	medium	5 %	40 %	55 %
6	10^6	50 %	17	medium	medium	low	0 %	100 %	0 %
7	10^5	10 %	7	medium	medium	low	5 %	55 %	40 %
8	10^6	50 %	27	medium	medium	low	0 %	100 %	0 %
9	10^5	50 %	17	medium	medium	low	25 %	75 %	0 %
10	10^3	50 %	7	medium	low	medium	10 %	75 %	15 %

avg in both approximate and non-approximate way and, then, the relative error of the approximate answer to the real one. Finally, we compared the relative error with the estimation of the relative error given by the probabilistic model. The results of the experimentation are summarized in Table 1.

As to the first launch in detail, we have obtained a medium relative error in computing both the *sum* and *count* functions, whereas the *avg* function produced a low relative error. The Bayesian network infers that the probability to observe a medium relative error is 79%, while the probability to obtain a low relative error is only 21%. So, we point out that the probabilistic model provides a good estimate of the relative error in the case of the *sum* and *count* functions. On the contrary, this estimate does not agree with the relative error obtained when computing the *avg* function.

Evidently, this probabilistic model is not quite powerful, as the relative error depends also on the kind of function.

The non-reliability of the estimation in the case of the *avg* function is due to the training set we have utilized, which does not take into account this kind of function. (Indeed, in the training set, the function used for the computation was always the *sum* function.)

Now, we analyse more deeply the experimental data in order to highlight the trend of the probabilistic model in reference to the three aggregate functions.

As concerns the *sum* function, we notice that the probabilistic model has always provided a good estimate of the relative error. In fact, in every launch, the highest estimated confidence degree exactly corresponds to the value of the relative error obtained in computing that function. Therefore, we have obtained the 100% of good estimates for the *sum* function (*cf*, Figure 9a).

As concerns the *count* function, the value of the relative error is *low* in the 10*th* launch whereas there is the highest confidence degree (*viz*, 75%) on the *medium* value in the estimate provided by the Bayesian network. Since we expected to have a

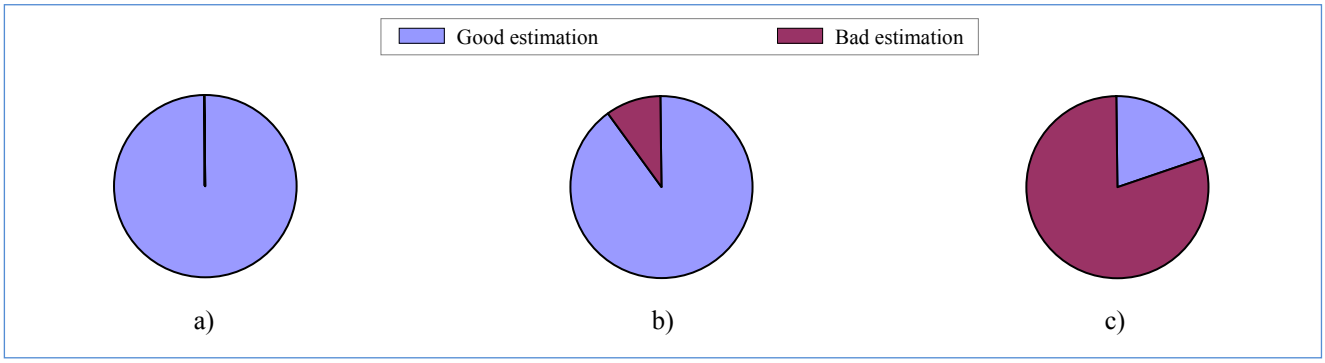

Fig. 9. *Analysis of probabilistic accuracy estimation of* a) *sum,* b) *count, and* c) *avg aggregate functions.*

medium relative error (with 75% of probability) according to the probabilistic model, but we observed a *low* relative error in the computation of the *count* function, we derive that the model fails about one prediction on ten for this function (*cf*, Figure 9b).

Finally, as concerns the *avg* function, we notice that the model is not a good estimator. In fact, only in the second and the tenth launch we obtained an estimate coherent with the observed relative error and, then, we conclude that the model fails about eight predictions on ten for this function (*cf*, Figure 9c).

6 Future Work

Future work is mainly devoted to extend the probabilistic model to take into account multidimensional distributions.

In particular, we need to include in the network the influence that the kind of aggregate function performs on the relative error. In fact, in multidimensional cases, we have experienced that the trend of the count function is quite different from that of the sum function while the sum function is very close to the average function.

Moreover, we need to introduce also query range stochastic variables, as we have more than one attribute (or dimension) and, for each attribute, the query range width can be 10%, 50%, or 90% independently of each other.

As an example, let us consider the relation $R(X_1, X_2, X_3, X_4)$ and an aggregate query that involves the 10% of (the active domains of) attributes X_1 and X_2, and the 50% of attributes X_3 and X_4. So, the query range of the aggregate query is determined by the number of dimensions (*ie*, four) and the width of the query interval involved for each attribute. Now, a question arises. Where can we put the evidence on the width node: in the 10% node value, the 50% node value, or in the 90% node value? The correct answer is none of these. In fact, traditional Bayesian networks are based on the Boolean logic, where the evidence can be either *true* or *false*. (According to this, we can only state that today is cloudy or not, for example.) Consequently, the evidence *true* can be put only on one node per time. On the other hand, Fuzzy Bayesian networks [27] are based on the fuzzy logic [28]. According to the fuzzy logic, the evidence ranges from 0 to 1. Thus, it is possible to state that today is 50% cloudy, for example. To conclude the example, we need to put the 50% of evidence on the 10% node value of the width stochastic variable (as the query involves the width 10% for attributes X_1 and X_2), and the 50% of evidence on the 50% node value of the width stochastic variable (for the query involves the width 50% for attributes X_3 and X_4). The probabilistic model of the next (Fuzzy) Bayesian Network is shown in Figure 10.

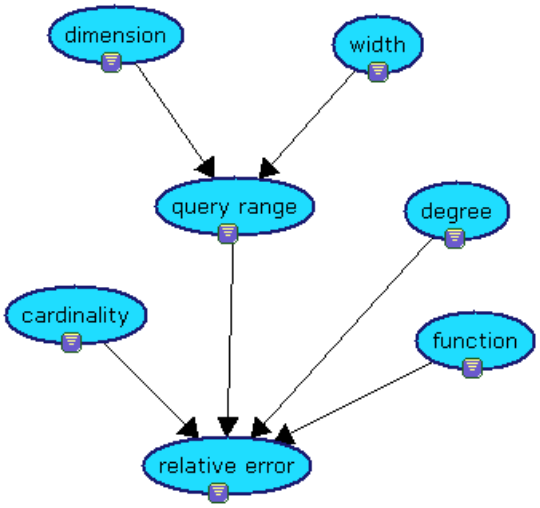

Fig. 10. *Bayesian network for multidimensional analyses.*

Another important issue is to consider stochastic variables varying on continuous domains, instead of using multinomial variables. For example, the relative error is currently classified according to three possible values (*viz*, *low*, *medium*, and *high*) whereas it may be useful to have relative errors ranging in the [0, 1] interval.

7 Conclusions

A Bayesian network is able to model the cause-effect relationships existing among the main random variables occurring in approximate query processing.

The presented probabilistic model provides an estimation of the relative error that the approximate query process yields, furnishing both the error bounds and confidence degree useful to estimate the real query answers.

Encouraging experimental results have confirmed that the probabilistic model is a good estimator of the relative error.

However, the experimentation has also highlighted the importance of how the training set to be used in the probabilistic model is constructed. For this reason, future work will address the extension of the current Bayesian network to consider further variables affecting the relative error.

These variables should include the kind of aggregate functions in constructing the training set, and the number of dimensions of the relations. In fact, for a multidimensional query, it has to be investigated how to specify the query range width relative to each dimension.

References

[1] dell'Aquila C., Lefons E., and Tangorra F., Decisional Portal Using Approximate Query Processing, *WSEAS Transactions on Computers*, Vol. 2, No. 2, 2003, pp. 486-492.

[2] Naiman K., Kopackova H., Simonova S., and Bilkova R., Approaches of Quality Outputs from the Business Systems, *Proceedings of the 5th WSEAS Int. Conf. on Computational Intelligence, Man-Machine Systems and Cybernetics*, Venice, Italy, Nov. 20-22, 2006, pp. 282-285.

[3] Elmasri R. and Navathe S. B., *Fundamentals of Database Systems*, (6th Edition), Addison Wesley, 2010.

[4] Chaudhuri S. and Dayal U., An Overview of Data Warehousing and OLAP Technology, *ACM Sigmod Record*, Vol. 26, No. 1, 1997, pp. 65-74.

[5] dell'Aquila C., Lefons E., and Tangorra F., Approximate Query Processing in Decision Support System Environment, *WSEAS Transactions on Computers*, Vol. 3, No. 3, 2004, pp. 581-586.

[6] Acharya S., Gibbons P. B., Poosala V., and Ramaswamy S., Join Synopses for Approximate Query Answering, *ACM SIGMOD Record*, Vol. 28, No. 2, 1999, pp. 275-286.

[7] Poosala V., Ganti V., and Ioannidis Y. E., Approximate Query Answering Using Histograms, *IEEE Data Engineering Bulletin*, 1999.

[8] Chakrabarti K., Garofalakis M. N., Rastogi R., and Shim K., Approximate Query Processing Using Wavelets, *The VLDB Journal*, Vol. 10, No. 2-3, September 2001, pp. 199-223.

[9] Pears R. and Houliston B., Optimization of Multidimensional Aggregates in Data Warehouses, *Journal of Database Management*, IDEA Group Publishing, USA, Vol. 18, No. 1, 2007, pp. 69-93.

[10] Mannila H. and Smyth P., Approximate Query Answering with Frequent Sets and Maximum Entropy, *Proceedings of the Sixteenth IEEE International Conference on Data Engineering* (ICDE), San Diego, CA, March 2000.

[11] Deshpande A., Guestring C., Madden S. R., and Hellerstein J. M., Model-driven Data Acquisition in Sensor Networks, *Proceedings of the 30th VLDB Conference*, Toronto, Canada, 2004, pp. 588-599.

[12] Missaoui R., Goutte C., Kouomou Choupo A., and Boujenoui A., A Probabilistic Model for Data Cube Compression and Query Approximation, *Proceedings of the ACM Tenth International Workshop on Data Warehousing and OLAP*, Lisbon, Portugal, 2007, pp. 33-40.

[13] Bernardino J., Furtado P., and Madeira H., Approximate Query Answering Using Data Warehouse Striping, *Journal of Intelligent Information Systems*, Vol. 19, No. 2, 2002, pp. 145–67.

[14] Shanmugasundaram J., Fayyad U., and Bradley P., Compressed Data Cubes for OLAP Aggregate Query Approximation on Continuous Dimensions, *Proceedings of the ACM SIGKDD 5th International Conference on Knowledge Discovery in Databases*, San Diego, CA, August 1999, pp. 223–232.

[15] Feng Y., Wen-Chi H., Zhewei J., Cheng L., and Qiang Z., Selectivity Estimation of Range Queries Based on Data Density Approximation via Cosine Series, *Data & Knowledge Engineering*, Vol. 63, No. 3, 2007, pp. 855-878.

[16] Peltzer J. B., Teredesai A. M., and Reinard G., AQUAGP: Approximate QUery Answers Using Genetic Programming, *9th European Conference*, EuroGP 2006, Budapest, Hungary, April 10-12, 2006, pp. 49-60.

[17] Spiegel J. and Polyzotis N., TuG Synopses for Approximate Query Answering, *ACM Transactions on Database Systems* (TODS), Vol. 34, No 1, article 3, 2009.

[18] dell'Aquila C., Di Tria F., Lefons E., and Tangorra, F., Data Reduction for Data Analysis. In: C. Cepisca, G. A. Kouzaev, N. E. Mastorakis, (Eds.): *New Aspects on Computing Research*, 2008, Athens, WSEAS Press, pp. 204-210.

[19] Acharya S., Gibbons P. B., Poosala V., and Ramaswamy S., The AQUA Approximate Query Answering System, *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, Philadelphia, Pennsylvania, United States, 1999, pp. 574-576.

[20] Lefons E., Merico A., and Tangorra F., Analytical Profile Estimation in Database Systems, *Information Systems*, Vol. 20, No. 1, 1995, pp. 1-20.

[21] Hunt E. B., *The Mathematics of Behavior*, Cambridge University Press, 1st edition, 2006.

[22] Nilsson N. J., *Artificial Intelligence: a New Synthesis*, Morgan Kaufmann, 1998.
[23] Russell S. J. and Norvig P., *Artificial Intelligence: Modern Approach*, Prentice Hall, 1995.
[24] Covell R., Introduction to Inference for Bayesian Networks, *Proceedings of the NATO Advanced Study Institute on Learning in Graphical Models*, Erice, Italy, 1998, pp. 9-26.
[25] Bayesian Network Tools in Java, http://bnj.sourceforge.net.
[26] Gibbons P. B., Poosala V., Acharya S., Bartal Y., Matias Y., Muthukrishnan S., Ramaswamy S., and Suel T., *AQUA: System and Techniques for Approximate Query Answering*, Murray Hill, New Jersey, U.S.A., 1998.
[27] Fogelberg C., Palade V., and Assheton P., Belief Propagation in Fuzzy Bayesian Networks, *First International Workshop on Combinations of Intelligent Methods and Applications* (CIMA) at ECAI'08 (University of Patras, Greece, 22 July 2008), I. Hatzilygeroudis, Ed., pp. 19–24.
[28] Klir G. J. and Yuan B., *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, (1st Edition), Prentice Hall, 1995.