Enhancing Database Correctness : A Statistical Approach

Wen-Chi Hou, **Zhongyang Zhang**

Department of Computer Science,
Southern Illinois University at Carbondale, IL 62901
E-mail: hou@cs.siu.edu.

Abstract

In this paper, we introduce a new type of integrity constraint, which we call a statistical constraint, and discuss its applicability to enhancing database correctness. Statistical constraints manifest embedded relationships among current attribute values in the database and are characterized by their probabilistic nature. They can be used to detect potential errors not easily detected by the conventional constraints. Methods for extracting statistical constraints from a relation and enforcement of such constraints are described. Preliminary performance evaluation of enforcing statistical constraints on a real life database is also presented.

1. Introduction

Integrity constraints can be regarded as conditions that specify correct states of databases. Much research has been done in the last two decades in enhancing correctness of data through integrity checking, e.g., [Codd 70, EsCh 75, HaSa 78, Morg 83, HsIm 85, Sell 88, McCa 89, Agra 89, Ston 90, Beer 91, Lohm 91, Gaha 92, Hans 92, Bran 93, etc.]. In this paper, we introduce a new type of integrity constraint, which we call a *Statistical (Integrity) Constraint*, and discuss its applicability to enhancing database correctness.

A database, as a model of some part of the real world, is expected to faithfully reflect reality. Therefore, it is likely to see statistical relationships existing among attributes of a relation as they are often embodied in the real world. Statistical relationships are quite similar to conventional constraints in that they both describe relationships (or properties) on data. However, despite their common existence, statistical relationships have seldom been exploited in the databases. This is mainly because the extraction and formulation of such relationships are not straightforward. In this paper, we describe how to extract statistical relationships and use them to enhance database correctness. Hereafter, as far as enhancing database correctness is concerned, we will use statistical relationship and statistical constraint interchangeably.

Depending on the enterprise a relation (or a database) tends to model, different relationships may be found therein. Throughout this paper, we will use the sample relation scheme EMP(Name, Salary, Dept, Exp(erience), Rank, Major) for a university to illustrate the use of statistical constraints. As you may notice in the sample relation scheme EMP, statistical relationships, such as "Salary is related to one's Dept, Rank, and Exp", "Dept is related to one's Major most of the time", "Rank has a lot to do with Exp and Salary", etc., often exist. Unlike conventional integrity constraints, which specify the ranges of legal attribute values (perhaps without regard to legitimate combinations of attribute values), statistical integrity constraints manifest embedded relationships among attribute values. They may be used to detect potential errors not easily detected by the conventional constraints. For example, consider the tuple (Jackson, 52K, ME, 1, Assistant-Prof, ME) of EMP. While each individual attribute value may fall within its respective attribute range, the tuple may still be incorrect. In fact, the salary seems to be unusually high for an assistant professor with one year experience in the Mechanical Engineering (ME) department according to the relationship found among Salary, Dept, Exp and Rank from the database. That is, the combination of the attribute values (52K, ME, 1, Assistant-Prof) seems to be quite unlikely, although the salary may still look reasonable for more senior personnel, or for people in other depart-

SIGMOD ' 95,San Jose , CA USA

ments or other ranks. As observed, the correctness of data can be greatly enhanced if we can incorporate those commonly found statistical constraints into the routine integrity constraint checking mechanisms. In the following discussion, we will make a more formal comparison between statistical and conventional constraints.

Statistical constraints describe relationships that hold on the current data, while the conventional constraints describe relationships that must always hold. The former can be regarded as time-variant constraints, while the latter as time-invariant constraints. Therefore, it may be necessary to update the statistical constraints after the database undergoes major changes.

Despite the difference in the persistency, statistical constraints are in fact quite similar to functional dependencies (FDs) and constraints that specify the ranges of attribute values in many respects. Let's first compare functional dependencies with statistical constraints. Assume that X is a set of attributes and Y is an attribute standing in relationship R with X. If R is an FD, then two tuples should have the same value for Y, by definition, if they agree on the X values. On the other hand, if R is a statistical relationship, one can only say that two tuples are likely to have the same value on Y if they have the same value on X. (Here we have left out statistical statements that quantify the likelihood, i.e., how likely they will have the same value). From this point of view, FDs can be regarded as a special case of statistical constraints in which the likelihood becomes definite. For example, consider the statistical relationship "Major determines Dept most of the time". It should be evident that the strongest form of such relationship may read as "Major determines Dept". That is, "Major determines Dept" is an FD as well as a statistical constraint (of maximum strength). Other statistical constraints, such as "Salary is related to one's Dept, Rank and Exp", can also find their analogies in conventional constraints that specify the ranges of attribute values. For example, one may express constraints on Salary using rules like "if Dept=... (and Rank=... and x_1 < Exp < x_2...) then y_1 < Salary < y_2". Instead of specifying the absolute range of Salary like y_1 < Salary < y_2, a statistical constraint usually indicates the most probable Salary value for that Dept (Rank, and Exp) and the associated variance. (Detailed representation of statistical constraints will be presented in Section 2.) It should not be difficult to see that a rigid range such as y_1 < Salary < y_2 can in fact be obtained by assuming the largest possible variations in a statistical constraint. Consequently, conventional constraints like this may be considered as a special case of statistical constraints. Since the ranges specified in the conventional constraints have to be large enough to accommodate all possible correct values, they usually are very loose (not to mention they have to be time-invariant) and are not able to detect unlikely values as effectively as statistical constraints.

Conventional constraints are definite constraints in that any violations of the constraints imply, without doubt, incorrectness of the tuples. On the other hand, statistical constraints are indefinite (or probabilistic) constraints in that violations of statistical constraints do not necessarily imply incorrectness of tuples, but indicate the likelihood of incorrectness. An unlikely tuple may well be a correct tuple in a relation. Therefore, the mechanism for determining the correctness of data for statistical constraints may be different from that of ordinary constraints. By issuing warnings when unlikely combinations of attributes values are encountered, statistical constraints can be used to detect potential errors.

Statistical constraints can be very useful in many other areas. For example, we have used them to estimate unknown attribute values in incomplete databases [HZZ 93]. Statistical constraints can also be used to monitor statistical properties of sensor data in a scientific, engineering, or real-time database environment. Yet from another point of view, statistical constraints represent an important type of knowledge embedded in the database, in which researchers in the area of data mining or knowledge discovery [Piat 91, Agra 93] are very interested. Methods used in this paper for identifying relevant attributes and extracting relationships among attributes can certainly be used as mining tools in knowledge discovery.

The rest of the paper is organized as follows. In Section 2, we first present examples of such constraints and discuss their enforcement. Section 3 introduces statistical techniques for deriving statistical relationships. Section 4 presents the performance evaluation of the enforcement of statistical constraints on a real life database. Section 5 is the conclusion.

2. Enforcement of Statistical Constraints

In this section, we discuss mainly two issues of the research, the representation and the enforcement of statistical constraints. In Section 2.1, we first briefly explain some terminology that will be used in the discussion. Then in Sections 2.2 and 2.3, we present concrete examples of statistical constraints and discuss the mechanism for determining the correctness of tuples.

2.1 Terminology

A statistical relationship is usually specified as one attribute, Y, dependent on (or correlated to) another set of attributes, X. Y is usually called the *dependent* attribute of the relationship and X is called the set of *explanatory* attributes of the relationship as it can help "explain" the values of dependent attribute Y. It should be noted that X represents the set of attributes that best explains Y. Any subsets of X will not be our interest here as they cannot explain Y as well as the entire X. Hereafter, a relationship is named after its dependent attribute.

Attributes of a relation can be classified as either *numerical* or *categorical*. A numerical attribute is one in which subjects differ in amount or degree (or whose domain is integers or reals), e.g., Salary and Exp of EMP. A categorical attribute is one on which the subjects differ in type or kind (or whose domain is character strings), e.g., Dept, Rank, and Major of EMP. A categorical attribute can have a infinite domain, however, with finite values represented in the database. Different methods (to be discussed in Section 3) will be used to derive relationships with different types of dependent attributes.

Once the relationship is substantiated, given a set of X attribute values, one can estimate the corresponding Y value and draw inference about whether a combination of X and Y values is likely or unlikely to be correct. An estimator of Y, denoted $\hat{Y}$, is often accompanied by a *confidence interval* and a *confidence level*. A confidence interval is an interval of plausible values for the parameter being estimated, while a confidence level is the degree of plausibility of such an interval.

2.2 Representation of Statistical Constraints

By taking advantage of a statistical software package, called SAS [SAS 91], statistical constraints are extracted from databases and stored in a system catalog, called STAT. STAT is consulted whenever a tuple is inserted or modified. This is also true for tuples arriving in batch fashion, i.e., checked with STAT individually. An updated tuple is treated as a newly inserted tuple. By comparing the attribute values calculated from STAT with the attribute values of a tuple, the system determines whether a tuple is likely to be correct or incorrect. However, since an unlikely tuple may well be a correct one, such tuples are still stored into relations just like correct ones, but with warning messages. It is user's ultimate responsibility to determine the correctness of the tuple based on the additional information at hand, e.g., context information, knowledge on the data, etc. When the database has gone through major changes, reextraction of constraints may be necessary.

A prototype DBMS, called CASE-DB [HoOz 93], is used as the underlying DBMS. While the physical implementations of STAT, such as linked lists, trees, relations, etc., are not our concern here, we present it as two tables shown in Figure 2.1, one for relationships with numerical dependent attributes, the other for relationships with categorical dependent attributes, as it will become clear that each type of relationship requires a different set of attributes. In the following, we demonstrate some sample relationships extracted from EMP.

Assume that it has been found that Salary (i.e., the dependent attribute) is closely related to Exp, Dept, and Rank (i.e., the explanatory attributes), and furthermore the Salary relationship, named after the dependent attribute, can be well approximated by

$$log(\text{Salary}) = \beta_1 \text{ Exp} + \beta_2 \text{ Dept} + \beta_3 \text{ Rank} \qquad (1)$$

where log is the natural logarithm function, and salary is expressed in thousands (K) of dollars. The reason for using a log function on Salary is that usually pay raise is calculated by the percentage, e.g., $Salary := Salary \times 1.05^{Exp}$ for an average 5% raise every year.

In Figure 2.1, the first column "Dattr" stores the dependent attribute name (e.g., Salary). The second column "Fun" stores the function (e.g., log) performed, if any, on the dependent attribute. In fact, one should consider log(Salary), instead of Salary alone, as the dependent attribute. The third column "Expln" contains information about the explanatory attributes with attribute names stored under the subcolumn Eattr, coefficients (i.e., β values) under the subcolumn "Coef", and values (if categorical) or function (if numerical) performed on the explanatory attribute under the subcolumn "VF". The last column "Stderr" stores the standard error of the estimation. For simplicity of presentation, in Figure 2.1.(a) we list only Rank values Prof, AsoP, AstP, Inst, and Lect, (standing for Professor, Associate Professor, Assistant Professor, Instructor, and Lecturer, respectively) for some of Dept values CS, EE, and ME. It has been calculated from a sample of the relation EMP that $\hat{\beta}_1$ (i.e., an estimate of β_1) is 0.003835. $\hat{\beta}_2$s are 0.142614, 0.145317, and 0.110216 for CS, EE, and ME, respectively, and $\hat{\beta}_3$s are 3.85258, 3.68999, 3.58775, 3.49429, and 3.06755 for Prof, AsoP, AstP, Inst, and Lect respectively. The standard error of the entire estimation is 0.1138. With these data, the expected salary of an

Dattr	Fun	Expln			Stderr
		Eattr	Param		
			Coef	VF	
Salary	log	Exp	.003835		.1138
		Dept	.142614	CS	
			.145317	EE	
			.110216	ME	
			.	.	
			.	.	
		Rank	3.85258	Prof	
			3.68999	AsoP	
			3.58775	AstP	
			3.49429	Inst	
			3.06755	Lect	
			.	.	
			.	.	

(a) Salary Relationship

Dattr	Population*					
	Pattr	Pval	Num		Covar	Prob
			Nattr	Mean		
Rank	Dept	ME			13.34, -2.11, -2.11, 34.67	
	Rank	Prof	Salary	58.7		.4667
			Exp	20.4		
	Rank	AsoP	Salary	46.2		.3333
			Exp	9.6		
	Rank	AstP	Salary	43.5		.1333
			Exp	2.5		
	Rank	Inst	Salary	28.0		.0667
			Exp	9.0		
	Dept	CS			2.83, 4.27, 4.27, 31.42	
	Rank	Prof	Salary	55.0		.1818
			Exp	22.5		
	Rank	AsoP	Salary	47.3		.2727
			Exp	11.3		
	Rank	AstP	Salary	44.0		.5455
			Exp	4.9		
		.	.			
		.	.			

(b) Rank Relationship

Figure 2.1 Structure of the STAT Catalog.

assistant professor (i.e., $\hat{\beta}_3 = 3.58775$) with one year experience (i.e., $\hat{\beta}_1 = 0.003835$) in ME department (i.e., $\hat{\beta}_2 = 0.110216$) is thus computed as

$$log\ (\text{Salary}) = 0.003835 \times 1 + 0.110216 + 3.58775 = 3.70180 \quad (2)$$

which is 40.52K.

Figure 2.1.(b) demonstrates the Rank relationship with dependent attribute Rank and explanatory attributes Dept, Exp, and Salary. Unlike the Salary relationship, this type of relationship (i.e., with categorical dependent attributes) can hardly be formulated as equations. As will be discussed in Section 3, categorical explanatory attributes in such relationships serve the purpose of subdividing the relation (the population) into subrelations (subpopulations). Therefore, we store all categorical explanatory attributes under the attribute "Population*" in Figure 2.1.(b), which may nest within itself because there might be multiple categorical explanatory attributes subdividing the population repeatedly. The dependent attribute values further divide each subpopulation into groups. For example, the categorical explanatory attribute Dept first divides the entire relation into subrelations based on the Dept values. Then each department is further divided into Prof, AsoP, AstP, Inst, and Lect groups based on the dependent attribute Rank. Each group is characterized by its mean numerical explanatory attribute values, that is, the mean Salary and mean Exp in this case. For instance, from the table one can observe that the professor group in ME has an average salary of 58.7K and an average experience of 20.4 years. With the mean characteristics for each group and the covariance matrix for each subpopulation (i.e., the column "Covar"), one can compute the dissimilarity coefficient of a tuple with respect to a group (see equation (7) of Section 3). The dissimilarity coefficient of the tuple <Wright, 53K, ME, 7, AstP, ME> with respect to the Prof group in ME, for example, is obtained as 8.391. The dissimilarity coefficient can be viewed as the squared distance of a tuple from the mean characteristics of a reference group. The dissimilarity coefficients with respect to other groups, AsoP, AstP, and Inst, in ME are obtained in the same way as 3.535, 7.815, 27.719.

Finally, based on the computed dissimilarity coefficients and the prior probabilities, for Prof, AsoP, AstP, Inst, respectively), the probabilities of the tuple belonging to the individual groups are obtained as .10359, .85616, .04025, and .000, respectively, using Bayes' Theorem [Tats 88, JoWi 92].

In the current design, we store STAT as two non-first normal form relations, which CASE-DB already supported. STAT can be retrieved and modified (i.e., insertion, deletion, and update) just like ordinary relations in the database. STAT is consulted whenever tuples are inserted or modified and should be updated after major changes in the database. Incidentally, the cost of extracting relationships, such as the ones shown in Figure 2.1 (a) and (b), from a sample of 200 tuples using SAS is about 0.4 and 0.1 seconds, respectively. Retrieval of a constraint may require 1 to 2 disk accesses. As the number of constraints increases, one may consider

adding an index structure on the constraints. For efficiency, one may also consider loading (a large portion of) STAT into memory for frequent references.

2.3 Enforcing Statistical Integrity Constraints

Since statistical constraints are probabilistic constraints, the determination of correctness can be practiced at various degrees of strictness α, $0 \leq \alpha \leq 1$. The higher the α value, the more strict the system is or the smaller the discrepancy between the attribute value and the expected value will be tolerated. Essentially, given an α, the system computes a most plausible interval (for numerical dependent attribute) or a most probable set of values (for categorical dependent attribute). This interval or set is expected to cover the correct values with a probability $1 - \alpha$. As α value increases, the corresponding interval or set narrows, and the system becomes more strict. For example, given an α value of 0.05 (i.e., 5%), the system will issue a warning message if an attribute value of a tuple, assumed numerical, falls outside the 95% (i.e., 1 – 0.05) confidence interval. The confidence interval here can be viewed as a range of values that the system is willing to tolerate or consider correct. The smaller the confidence interval (i.e., a higher α value), the more strict the system is. In the following section, we formally discuss how the system determines whether a value is correct or incorrect for a given α value.

Decision Rule for Relationships with Numerical Dependent Attributes

Let's continue to use the Salary relationship as an example. Let n be the number of sample tuples, say 200, that we draw from a relation to extract the statistical relationship, and let m be the number of explanatory attributes involved in the relationship (i.e., 3 in this example). Let $\hat{y}$ be the estimated dependent attribute value, perhaps with some function performed on it (e.g., log(Salary)), and $\sigma_{\hat{y}}$ is the standard error of the $\hat{y}$. Given an α, the confidence interval is computed as $\hat{y} \pm t_{\alpha/2,n-m}\sigma_{\hat{y}}$ (as will be discussed in Section 3), where $t_{\alpha/2,n-m}$ is the t distribution with parameters $\alpha/2$ and $n - m$. The decision rule is that if the attribute value y of a tuple does not fall within the interval, a warning message shall be issued. That is,

if $y > \hat{y} + t_{\alpha/2,n-m}\sigma_{\hat{y}}$ or $y < \hat{y} - t_{\alpha/2,n-m}\sigma_{\hat{y}}$
then issue-warning;

At one extreme $\alpha = 1$, the system accepts a tuple only if $y = \hat{y}$. At another extreme $\alpha = 0$, all possible values are considered correct by the system.

Let's consider how this rule can be applied to check whether the the tuple <Jackson, 52K, ME, 1, AstP, ME> is likely to be correct. Assume that we have chosen a strictness level .05. The corresponding $t_{\alpha/2,n-m}$ value can be easily found from the t distribution table to be 1.96. Therefore, the confidence interval for log(Salary), where the Salary is expressed in thousands, is $3.70180 \pm 1.96 \times 0.1138$, i.e., 3.70180 ± 0.2238, using equation (5). Since log(52) (= 3.95124) falls outside the interval 3.70180 ± 0.2238, a warning message should be issued to alert the user.

Decision Rule for Relationships with Categorical Dependent Attributes

In Section 2.2, we briefly described how the probability of a dependent attribute having y_i as its value can be derived. However, unlike the numerical case above, it is not clear how one could determine the correctness of data given a strictness level α. We generalize the idea of confidence interval for numerical values to accommodate categorical values so that consistent decision rules can be established in the system. By considering a confidence interval as a range of most probable values, we can easily construct a "confidence interval" for categorical values as the set of most probable categorical values. That is, a $100 \times (1 - \alpha)\%$ categorical "confidence interval", say S, is the smallest set of most probable attribute values, and those values make up no less than $100 \times (1 - \alpha)\%$ of the probability. Formally speaking, let $p(y_i)$ be the probability that the dependent attribute has y_i as its value. Then, S is the smallest subset of dependent attribute values such that $\sum_{y_j \in S} p(y_j) \geq 1 - \alpha$, and $p(y_j) \geq p(y_k)$, for any $y_k \notin$ S. Let y be the actual dependent attribute value of the tuple. Then, the decision rule is

if $y \notin$ S then issue-warning;

At one extreme $\alpha = 1$, we define S to be a set containing only the most probably value (i.e., with the largest probability). At another extreme $\alpha = 0$, S contains all domain values.

Let's now consider how the system determines the correctness of the tuple <Wright, 53K, ME, 7, AstP, ME> using the Rank constraint. Again, we assume that the strictness level α is .05. As illustrated earlier, the expected probabilities of the Rank values being Prof, AsoP, AstP, Inst are .1036, .8562, .0403, .0000, respectively. As a result, the 95% "confidence interval" consists of Prof and AsoP only (because 0.1036+0.8562 >

0.95), and thus the tuple <Wright, 53K, ME, 7, AstP, ME> is considered as potentially incorrect at level .05.

3. Statistical Modeling of Relationships Among Attributes

In this section, we show how any relationships, with all types of dependent explanatory attributes can be extracted from the raw data. We have utilized several of the most commonly used statistical methods to develop this methodology. In this research, regression modeling is used for relationships with numerical dependent attributes, while classification analysis is used for relationships with categorical dependent attributes. To avoid tedious statistical discussions, we will present these methods more from a practical point of view. Interested readers are referred to [Devo 84, JoWi 92, Chou 75, Tsts 88] for more details of these methods. For illustrative purpose, we will make different assumptions on the relationships existing among attributes wherever appropriate.

3.1 Regression Models for Numerical Dependent Attributes

In general, a regression model can be specified as either *a linear regression model* or *a nonlinear regression model*. They can be used to model various numerical relationships.

3.1.1 Linear Regression Models

In a linear regression model, a relationship is specified as

$$Y = \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m \qquad (3)$$

where Y is the dependent attribute and X_i's are the explanatory attributes, assuming that both X_i's and Y are numerical. If the Salary of an employee is related solely to Exp and yet an approximately linear relationship exists, using the equation (3) with Salary as the dependent attribute and Exp as the explanatory attribute may yield a good approximation. Estimates of β_i, $1 \le j \le m$, can be obtained from (a sample of) the population and be then used to derive the expected dependent attribute value, denoted $\hat{y}$, from equation (3). Given a desired confidence level α, the associated 100 (1- α)% confidence interval is

$$\hat{y} \pm t_{\alpha/2,n-m}\hat{\sigma}_{\hat{y}} \qquad (4)$$

where $\hat{\sigma}_{\hat{y}}$ is an estimate of the standard error of the estimator $\hat{y}$, n is the sample size, and $t_{\alpha/2,n-m}$ is the t distribution value with parameters $\alpha/2$ and $n - m$.

The above regression model can be easily generalized to accommodate multiple categorical explanatory attributes by representing each distinct value as a dummy binary attribute. For example, assume that Salary is not only related to Exp but also related to Dept with a domain, for simplicity, of {CS, EE, ME}. To incorporate Dept into the regression model, dummy attributes D_1, D_2, and D_3 are introduced for CS, EE, and ME, respectively. D_i has the value 1 if the tuple of concern has the corresponding value; otherwise D_i has a value 0. That is, if an employee works in CS department, then $D_1 = 1, D_2 = D_3 = 0$. With dummy attributes, the regression model can be expressed as

$$Y = \beta_1 D_1 + \beta_2 D_2 + \beta_3 D_3 + \beta_4 Exp \qquad (5)$$

Earlier discussions on the derivation of $\hat{y}$ and the associated confidence interval remain valid here.

3.1.2 Nonlinear Regression Model

A nonlinear model is generally specified as

$$Y = f(X) \qquad (6)$$

where Y and X are defined as before and f is an arbitrary function. Other discussions, such as derivation of the estimation of $\hat{y}$ and the associated confidence interval, remain the same as for the linear regression modeling.

3.2 Classification Analysis for Categorical Dependent Attribute

Classification is a grouping method based on the measure of resemblance or dissimilarity [Tats 88] of the characteristics of the objects. In general, a classification analysis can be carried out by first dividing the tuples into distinct groups based on their dependent attribute values. Then the dissimilarity (or resemblance) coefficient of a tuple with respect to each individual group is calculated. Based on the dissimilarity coefficients, we can estimate the membership probability of a tuple belonging to a group.

First, we discuss the situations where the explanatory attributes are numerical, e.g., the Rank relationship with Exp as the only explanatory attribute. Let $X_1, X_2, \ldots, X_m$ be the set of numerical explanatory attributes and $(x_{1i}, x_{2i}, \ldots, x_{mi})$ (x_i for short) be the $X_1, X_2, \ldots, X_m$ values of a given tuple t_i. Let $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m)$ be the *centroid* of the k^{th} group (for simplicity, without specifying the k), where $\bar{x}_j$, $j = 1, 2, \ldots, m$, are the mean X_j values of the k^{th} group. The centroid vector indicates the average characteristics

of the concerned group. The dissimilarity of a tuple t_i with respect to the k^{th} group is usually measured by the "squared distance", defined as

$$D_{ik}^2 = (x_{1i} - \bar{x}_1, \cdots, x_{mi} - \bar{x}_m)\ C_k^{-1} \cdot (x_{1i} - \bar{x}_1, \ldots, x_{mi} - \bar{x}_m)^T \qquad (7)$$

where C_k^{-1} is the inversed covariance matrix of the k^{th} group. Equation (7) is often referred to as the Chi-square (χ^2)-statistic. Both the centroid ($\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m$) and the covariance matrix C_k of a group can be obtained from (a sample of) the relation. It should be noted that the larger the χ^2 statistic, the farther away, in the generalized distance sense, the point ($x_{1i}, x_{2i}, \ldots, x_{mi}$) is from the centroid ($\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m$) of the reference group. Thus, the tuple may be said to be more deviant from the average member of the group. Conversely, a small χ^2 statistic indicates that the tuple resembles the group closely. With the calculated χ^2 value and the frequency of each group in the population, one can further calculate the membership probability of a tuple using Bayes' theorem [Tats 88, JoWi 92].

The above discussion can be easily generalized to include categorical explanatory attributes. For instance, consider the Rank relationship above with an additional assumption that Dept is also an explanatory attribute. In this case, Dept first subdivides the population (or the relation) into subpopulations (or subrelations) based upon the Dept values. Then the calculation of similarity measures, membership probabilities, and prediction of Rank values (i.e., group membership) are practiced in the same way within each subpopulation.

4 Preliminary Experimental Results

In this section, we present the performance evaluation of statistical constraints in detecting potential errors. The data of EMP relation is collected on the faculty members of a university, which consists of 1,200 tuples. It should be noted that the size of data should not be a factor of the performance evaluation since the statistical methods discussed earlier suit large data sets as well as small data sets (as long as they are not too small to collect meaningful statistics). In addition, the effectiveness of statistical constraints in detecting potential errors is determined by the properties of the data, not by the size of data. That is, the stronger the relationship, the more effective the constraint.

Two statistical relationships have been extracted from EMP and stored in STAT for the following experiments, one with a numerical dependent attribute, the other with categorical dependent attributes. The explanatory attributes consist of both numerical and categorical attributes. The experiments have been designed to show how statistical constraints can help detect potential errors that are not easily detected by the conventional integrity constraints. We made up tuples by replacing the attribute values (e.g., Salary, Rank) of tuples in EMP. Hereafter, we shall call these newly created tuples *incorrect* tuples, while the original tuples of EMP the *correct* tuples. Mixed with correct tuples, incorrect tuples are run through our system to see if they can be captured. Note that all the attribute values of either correct or incorrect tuples fall within their respective legal ranges or domains. We believe such incorrectness is quite common yet difficult to detect by conventional constraints.

As mentioned earlier, statistical constraints determine whether a tuple is likely to be correct based upon statistical measures. If an attribute value deviates largely from the expected value, it may be considered as incorrect. As a result, correct tuples, as well as incorrect tuples, may be regarded as unlikely tuples if their attribute values are significantly different from the typical values. Therefore, two indicators, called *type one* (T_1) and *type two* (T_2), are set up to measure the degrees of correctness and incorrectness, respectively, of a system in making judgements. Let n_1 be the number of the incorrect tuples in the test and n_1^w be the number of incorrect tuples detected by the system (i.e., warning messages are issued). Then, type one indicator is defined as

$$T_1 = \frac{n_1^w}{n_1}$$

Let n_2 be the number of correct tuples tested in our experiment and n_2^w be the number of such tuples misjudged as incorrect ones. Then, type two indicator is

$$T_2 = \frac{n_2^w}{n_2}$$

Both n_1 and n_2 are 1,200 in our experiments as each incorrect tuple is created from a correct tuple.

Ideally, a system is to issue warning messages upon encountering any incorrect tuples, i.e., a high T_1 value, while accepting all correct tuples, i.e., a low T_2 value. Unfortunately, it is usually not possible to meet both ends unless all correct tuples follow very closely the relationships postulated, and yet all incorrect tuples deviate greatly from the relationship. By controlling the degree of strictness α, an acceptable compromise may be

achieved, that is, a reasonably high T_1 and a relatively low T_2. Another factor of determining the performance is the magnitude of the incorrectness itself. That is, the larger the discrepancies (from the correct values), the easier it is to detect. In the following section, we will present the experimental results based on the strictness levels and the discrepancies in the incorrect tuples.

4.1 Experiments on the Numerical Dependent Attribute

The Salary relationship is between Salary and Exp, Rank, and Dept, with Salary (more correctly log(Salary)) as the dependent attribute. We use the linear regression expression shown in equation (1) of Section 2 to model it, that is,

$$\log(Salary) = \beta_1 \text{ Exp} + \beta_2 \text{ Dept} + \beta_3 \text{ Rank}$$

Some of the β_j values, $1 \le j \le 3$, have been shown in Figure 2.1.(a) as an example.

For numerical attributes, the discrepancy is specified by the percentage. For example, given an employee with a salary of 50K, the corresponding incorrect salary is calculated as 50K+5K (or 50K−5K) if the discrepancy is specified as 10%. For categorical attributes, we first rank the values (if possible), and then represent the discrepancy in levels of ranking. For example, Rank values are first ordered as (Prof, AsoP, AstP, Inst, Lect). A one-degree difference in Rank corresponds to moving the original rank values up (or down) one level, that is, to replace Prof by AsoP, AsoP by AstP, and so on. Two types of experiments are run, one with Salary values being replaced, the other with Rank values being switched. Experimental results are shown in Tables 4.1.(a) and 4.1.(b), respectively.

Consider the first row (0.01, 7%, 16%, 44%, 79%, 4%) of Table 4.1.(a). It indicates that at $\alpha = 1\%$ (a low strictness level), 7%, 16%, 44%, and 79% of the incorrect tuples with their Salary values differing from the original values by 2.5%, 5%, 7.5% and 10%, respectively, are captured by the system. That the rest of the incorrect tuples are not detected by the system is mainly because they "look" like correct tuples at this level of strictness. Meanwhile, 4% of the actual tuples in EMP are misjudged as potential incorrect tuples. This is due to the fact that some attribute values of correct tuples themselves deviate quite largely from the normal values and thus are judged as incorrect. As α increases, the system becomes more strict. As a result, more incorrect tuples are detected; however, more misjudgements are also reported. Meanwhile, as the discrepancy increases, more incorrect tuples are detected, which is quite reasonable.

α	T_1 (percentage differences from correct Salary values)				T_2
	2.5%	5%	7.5%	10%	
0.01	7%	16%	44%	79%	4%
0.10	18%	51%	77%	96%	9%
0.20	53%	64%	92%	97%	14%
0.50	60%	89%	97%	99%	40%

(a) Salary Values Replaced

α	T_1 (degree differences from correct Rank values)			T_2
	one	two	three	
0.01	24%	46%	78%	4%
0.10	46%	77%	86%	8%
0.20	56%	87%	89%	12%
0.50	82%	89%	91%	42%

(b) Rank Values Replaced

Table 4.1. Detecting Errors Using the Salary Relationship.

We would like to point out that detecting incorrect tuples of these types in itself is not an easy task because all the values replaced are legitimate, and there are no extremely large or small values. As observed, the performance is very good even for moderate discrepancies. For instance, at strictness level 0.1, 96% of the incorrect tuples are detected, even though they differ only by 10% from the correct Salary values. Table 4.1.(b) basically shows similar results, however, with Rank values replaced in the same relationship.

4.2 Experiments on the Categorical Dependent Attribute

The Rank relationship is between Rank and Salary, Exp, and Dept, with Rank as the dependent attribute. Categorical analysis is used here to detect potential errors. The results are presented in Table 4.2.

As in the Salary relationship, we performed two types of experiments on the Rank relationship, one with Salary values replaced, the other with Rank values replaced. Again, it is observed that as α increases, more incorrect tuples are detected, but more misjudgements of correct tuples are also reported. As in the previous

experiments, as the discrepancy increases, more incorrect tuples are detected. Very good results are reported especially when Rank values are replaced.

It is worth noting that Rank is not very sensitive to the changes in Salary in the Rank relationship, though they are relevant. The main reason for this is that since the tuples are classified into only five categories based on the Rank values, each category could cover a relatively large group of tuples, which in turn implies that each group could have a large range of legitimate salary values. Therefore, minor changes in Salary can not rule out the possibility that the old rank value is still reasonable in the newly created tuples.

α	T_1 (percentage differences from correct Salary values)				T_2
	10%	20%	50%	80%	
0.01	6%	8%	23%	67%	2%
0.10	11%	23%	51%	78%	4%
0.20	17%	29%	58%	81%	5%
0.50	27%	41%	71%	87%	24%

(a) Salary Values Replaced

α	T_1 (degree differences from correct Rank values)			T_2
	one	two	three	
0.01	32%	67%	100%	2%
0.10	62%	89%	100%	5%
0.20	74%	97%	100%	6%
0.50	95%	100%	100%	27%

(b) Rank Values Replaced

Table 4.2. Detecting Errors Using the Rank Relationship.

4.3 General Discussion

The strength of relationships is the major factor affecting the performance. If the attribute values of a relation are strongly corelated, then a high T_1 and low T_2 can be expected for a given α. In fact, statistical constraints can be considered as a weaker form of functional dependency in some respect. At one extreme, where a statistical constraint expresses the same condition as a functional dependency, any violations of the statistical constraint will definitely be detected and any correct tuples will not be misjudged. That is, $T_1 = 100\%$ and $T_2 = 0\%$ are expected for the strongest statistical relationships (or FDs). At another extreme, where a significant relationship does not exist among attribute values, $T_1 \approx T_2 \approx 50\%$ is expected. (Of course, we would not have stored such a "relationship".) In general, the stronger the relationship (or dependence), the higher the T_1 and the lower the T_2 values are.

We should also point out that statistical constraints may fall short in detecting incorrect tuples that are statistically sound. That is, when incorrect data fits well into the relationship postulated, statistical constraints may not be able to detect. In fact, the major reason that some incorrect tuples were not captured in our experiments is that they simply "look" great, and sometimes even fit better the statistical constraints than some of the correct tuples. However, such a deficiency is not unique in statistical constraints; it can also be found in a similar form in ordinary integrity constraints. That is, one can make up a tuple that satisfies all value constraints (e.g., $0 \leq$ Salary $\leq$ 1,000,000, etc.,) and functional dependencies (e.g., Major -> Dept, etc.,), and yet it is not a correct (or original) tuple in the relation. Nevertheless, statistical constraints can help detect many of the conventional constraint sound but incorrect tuples. Thus, statistical constraints should work hand in hand with conventional constraints to ensure higher quality of data.

5. Conclusion and Future Work

In this paper, we have introduced a new type of integrity constraint, called statistical constraint, and discussed its applicability to enhancing database correctness. Unlike other integrity constraints, statistical constraints are characterized by their probabilistic nature, and the mechanism for detecting potential errors. We have also developed a complete methodology for extracting statistical constraints using several of the most known statistical methods. Statistical constraints are stored in a system catalog and consulted whenever tuples are inserted or modified. We have shown in our experiments that statistical constraints can capture many errors that are not easily detected by conventional constraints. In summary, the contributions of this paper include the introduction, derivation, and enforcement of statistical constraints.

There is still much future work to be done. For example, one can try to further identify (or rank) the incorrect attribute values once potentially incorrect tuples are detected. Another immediate extension of the research is to adapt the constraints to an ever-changing environment. Investigation in incremental constraint

modification techniques may offer some help. There are many other applications of the statistical constraints. For example, statistical constraints can be used to monitor the properties of sensor data in a scientific, engineering, real-time environment. Also, we can explore potential applications of our approach in areas such as uncertainty reasoning and data mining.

References

[Agra 89] Agrawal, R., Gehani, N., "Ode (Object Database and Environment) : The Language and the Data Model", ACM SIGMOD Conference, 1989, pp. 36-45.

[Agra 93] Agrawal, R., Imielinski, T., Swami, A., "Mining Association Rules between Sets of Items in Large Databases", ACM SIGMOD Conference, 1993, pp. 207-216.

[Beer 91] Beeri, C., Milo, T. "A Model for Active Object Oriented Database", Proc. 17th VLDB Conference, 1991, pp 337-349.

[Bran 93] Brant, D., Miranker, D.,"Index Support for Rule Activation", ACM SIGMOD Conference, 1993, pp. 42-48.

[Chou 75] Chou, Y-I. "Statistical Analysis", Holt, Rinehart and Winston, 1975.

[Coch 77] Cochran, W. "Sampling Techniques", 3rd Ed., John Wiley & Sons, 1977.

[Codd 70] Codd, E. F., "A Relational Model for Large Shared Data Banks", Communication of the ACM, Vol. 13, No. 6, 1970, pp. 377-387.

[Devo 84] Devore, J., "Probability & Statistics for Engineering and the Sciences", Brooks/Cole Publishing, 1984.

[EsCh 75] Eswaran, K., Chamberlin D. "Functional Specifications of a Subsystem for Data Base Integrity", Proc. VLDB 1975, pp. 48-68.

[Hans 92] Hanson, E., "Rule Condition Testing and Action Execution in Ariel", ACM SIGMOD Conference, 1992, pp. 49-58.

[HaSa 78] Hammer, M., Sarin, S., "Efficient Monitoring of Database Assertions", Proc. of ACM SIGMOD Conference, 1978.

[HoWo 73] Hollander, M., and Wolfe, D., "Nonparametric Statistical Methods", John Wiley, 1973.

[HoOz 93] Hou, W-C., Ozsoyoglu, G., "Processing Real-Time Aggregate Relational Queries in CASE-DB", ACM Transactions on Database Systems Vol. 18, No. 2, June, 1993.

[HsIm 85] Hsu, A., Imielinski, T., "Integrity Checking for Multiple Updates", Proc. of ACM SIGMOD Conference, 1985, pp. 152-168.

[HZZ 93] Hou, W-C., Zhang, Z., Zhou, N., "Statistical Inference of Unknown Attribute Values in Databases", Proc. CIKM 1993, pp. 21-30.

[JoWi 92] Johnson, R. and Wichern, D., "Applied Multivariate Statistical Analysis", 3rd ed. Prentice-Hall, Englewood Cliffs, 1992.

[Lohm 91] Lohman, G., etc., "Extension to Starburst : Objects, Types, Functions, and Rules", Comm. ACM, Vol. 34, No. 10, 1991, pp. 94-109.

[McCa 89] McCarthy, D., Uayal, U, "The Architecture of An Active Object-Oriented Database System, ACM SIGMOD Conference, 1989, pp. 215-224.

[Morg 83] Morgenstern, M. "Active Databases as a Paradigm for Enhanced Computing Environments", Proc. the 9th VLDB Conference, 1983, pp. 34-42.

[Piat 91] G. Piatetsky-Shapiro etc., "Knowledge Discovery in Databases", AAAI/MIT Press, 1991.

[SAS 91] "SAS/STAT User's Guide", Release 6.03 Ed., SAS Institute Inc., North Carolina.

[Sell 88] Sellis, T., Lin, C., Raschid, L., "Implementing Large Production Systems in a DBMS Environment : Concepts and Algorithms", ACM SIGMOD Conference, 1988, pp. 404-412.

[Ston 90] Stonebraker, M., etc., "On Rules, Procedures, Caching and Views in Database Systems", ACM SIGMOD Conference, 1990, pp. 281-290.

[Suit 85] Suits, D. "Statistics : An Introduction to Quantitative Economic Research", Halyburton Press, 1985.

[Tats 88] Tatsuoka, M., "Multivariate Analysis", Macmillan Publishing, 1988.