IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4109

Employing Neural Networks to Assess Data Quality

Abdullah Al-Namlah and Shirley A. Becker
Computer Science Department, Florida Institute of Technology, 150 West University Blvd., Melbourne, Florida
Tel: 1-321-674-8149, Fax: 1-321-674-7046, alnamlah@hotmail.com, abecker@cs.fit.edu

ABSTRACT

Neural networks have been successfully employed in a wide variety of fields, such as signal processing, pattern recognition, medicine, speech recognition, and business, in order to solve complex problems. It is proposed in this paper that neural networks can also be applied to the data quality problem that is so pervasive in legacy software systems. The focus of this work is on the use of neural networks to learn how to identify duplicate records in a data source. This problem has been recognized as extremely important to many organizations, due to the size and complexity of today's database systems. The initial use of neural nets has shown that they can be trained to perform data quality checks. Our ongoing research will address larger, more complex databases systems, as well as, learning capabilities to solve other data quality problems.

INTRODUCTION

The data quality issue is of great concern to both consumers and organizations because of the skyrocketing cost associated with it. According to Pitney Bowes (1998), incorrect pricing data in retail databases, alone, cost American consumers as much as $2.5 billion annually in overcharges. Companies have lost millions of dollars each year due to poor quality customer data. Pitney Bowes points out that on average, 15 percent of the information contained in a company's customer database is flawed. This is indicative of the data quality issues associated with existing database systems.

Too many organizations are unable to determine the level of data quality in both their operational and historical databases. Operational databases and supporting applications may have been designed incorrectly such that incomplete, incorrect, or cryptic data can be entered. Poorly designed applications may actually allow for data corruption while the database is in use. In addition, data may be imported from existing sources, thus transferring the problem from one database system to another. Legacy databases have similar data quality problems, but due to the sheer volume of historical data, the problems are amplified. Legacy databases are often filled with cryptic, inaccurate, or incomplete data yielding them virtually unusable or inaccessible. The data problems in these existing systems may be the result of data design (e.g., not normalized data or lack of key, not null, and other constraints) and data representation (e.g., last name, first name, and middle initial are all stored as one text string), to name a few.

One option in addressing this problem is to rely on existing techniques provided by data mining and data cleaning whereby data is "mined" in order to extract knowledge about it. Then, it can be inspected and modified to improve its quality. Though industry practices show that this is a viable option for improving data quality, it can prove to be very expensive and time-consuming. Large amounts of data may have to be searched, often times manually, to determine the extent of the data quality problem.

What is needed is a practical yet rigorous means of assessing data quality in order to determine whether it would be cost effective to clean it or improve upon it in some other fashion (e.g., redesign or reengineer it). This paper discusses an ongoing research effort in identifying how data quality can be assessed using existing data in a database. This work is based on the use of neural networks to provide feedback on data quality aspects of an existing database system.

Data quality is briefly described in order to show the range of data quality problems that could exist in a database system. Data mining is also briefly introduced, as it provides a foundational concept of knowledge discovery from databases. Then, we discuss the use of neural networks in the assessment of data quality. The paper concludes with future research directions.

DATA QUALITY

There are many different types of data quality problems ranging from data codes that have no documented meaning to spelling and typing mistakes that resulted in incorrect or incomplete data. Data redundancy leaves a database vulnerable to data corruption when only part of the data is updated or deleted. Data may be incomplete or inconsistent with other data due to missing or incorrect use of database constraints. Data that is stored in an abbreviated format, as an acronym, or as a cryptic code may become virtually unusable when its meaning is not stored in the database.

Data quality classification schemas have been proposed by several researchers, which include *correctness, completeness, comprehension, and consistency* classes of data quality (Becker et al., 1999; Greenfield, 1996; Fox et al., 1994). Each of these is identified as:

Data consistency: Data consistency is maintained when the same data definition is used for various data structures. A foreign key in one table, for example, would be defined as having the same data type and length as its associated primary key in another table. This would include constraints on the data such as not allowing nulls, having a predefined set of values, and other constraints.

Data completeness: Data completeness means that the data stored in the database is wholly representative of the real world. Data may be considered incomplete when part of it is missing, corrupted, or inaccessible.

Data correctness: Data correctness has to do with storing and maintaining the correct data value for all database objects. Data correctness is also associated with data manipulation such that data is not corrupted due to queries, normalization, views, data constraints, and other factors.

Data comprehension: Data comprehension is a significant and often overlooked factor in the quality of a database system. Far too often, cryptic codes are used that may be currently understood by personnel. Over time, the meanings of these codes may change or may be lost.

These classification schemas are important in understanding the breadth and depth of potential data quality problems. They provide the foundation upon which an organization can assess data quality. However, innovative techniques are needed to apply these classification schemas because of the complexity associated with today's database systems.

DATA MINING CONCEPTS

A foundational aspect of this research is data mining in order to understand and react to knowledge discovered about data quality. Han and Kamber (2001, p.7) define data mining as "the process of discov-

This paper appears in *Issues and Trends of Information Technology Management in Contemporary Organizations,* the proceedings of the Information Resources Management Association International Conference.

ering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories." This definition is appropriate to our work because of the opportunity to assess data quality for large databases through knowledge discovery.

One of the important aspects of data mining, in terms of this research, is that a conditioned sample size is correlated with multiple sources of information. This will allow us to use a sample of data from a database system, and draw conclusions about the quality of its data. Statistical inferencing, based on the results of data mining efforts, must be sufficiently rigorous and reliable in order to draw valid conclusions about the newly discovered knowledge. Without accurate identification of duplicated information, frequency distributions and various other aggregations will produce false or misleading statistics leading to perhaps untrustworthy new knowledge.

DATA QUALITY ASSESSMENT USING NEURAL NETWORKS

A neural network is an information processing system that can be used to store and recall data or patterns and then be able to classify them. It has the capability to learn by example in order to be able to recall and classify data. The internal architecture of the neural net consists of nodes that are highly connected. Each connection has a weight, and as the neural network is trained, it adjusts those weights. When these weights no longer need to be adjusted during the training phase, the neural net has learned from provided examples. Then, it should be able to recognize exact (memorize) and similar (generalize) patterns when it sees them in future applications (Fausett, 1994).

Neural networks have proven to be quite effective for a broad range of problems, but are especially useful for predicting events when there is a large pool of data to use during the learning process. Because neural nets are very useful in recognizing complex patterns in existing data, it is quite appropriate for assessing data quality in existing database systems.

It is proposed that neural nets be used in this research in order to provide a learning environment based on a sample of data. The more data that is assessed, the more information that will be discovered about similarities and differences in a particular data source(s). This learning process, over time, could be extended to include more than one database within and across software systems. It could also be extended to include new technologies such as XML such that comparisons could be done between different data structures.

The Learning Process

We illustrate how data quality can be assessed using neural nets via a simple algorithm that selects a set of records, based on some condition, in order to calculate data equivalence within a set of records. We use the data equivalence percentages as a learning mechanism for our neural net. In this way, the net can be trained to discover duplicate records. (Note: our initial work is focusing on the discovery of duplicate records, though future research will focus on other data quality issues.)

A three-step approach is proposed in order to use neural nets to provide feedback on the quality of a data source. We are applying this approach in a limited fashion in order to specifically train the net on what constitutes duplicate records in a database system. The output of the neural net is used to draw conclusions based on its statistical input. In this case, the inputs to the net are based on existing data quality, as defined by pattern matching in the existing data.

Step 1: Find similar records based on a matching condition. This first step of knowledge discovery deals with the database as a whole. The database is used to find records that are similar in terms of a specified criterion (or it could be a set of criteria). This may include searching within and across tables based on record matching conditions (e.g., similar SSN and names). For illustrative purposes, let's say that the following records were retrieved based on having the same social security number. Notice that in these records, some of the data in the columns match completely, while other data may have a partial or no match in the other records.

Step 2: Calculate the percentage of data that is equivalent for each pair of records. The second step identifies the percentage of matched data (within each data column) in one record when it is compared to the data in the other records in the set. Given the data set presented in step 1, we compare the Franklin record to the Frank record and come up with the following percentage **of matched data.**

Notice, for example, that FNAME data has a 63% match (Franklin and Frank). The middle initials in these two records are different and so the percentage match is zero. The social security numbers are exactly the same, and hence the percentage is 100%. At this point, we may not be able to draw any conclusions about whether these records represent the same individual or are actually two individuals. But, we will use this information in our neural net in order to learn from it.[1]

Step 3: Find duplicate records using a neural net learning mechanism. We build a neural network that can learn from examples of duplicate records and then rely on another technique to minimize the complexity of find more duplicate records. The most reliable way to detect duplicate records is to compare each record with every other record. In order to reduce the complexity of this comparison process, a method called the Sorted Neighborhood Method (SNM) is used. It reduces the complexity described as $O(N^2)$ by sorting the database using an application-specific key. Then, pairwise comparisons of nearby records are made by sliding a window of size w. Each new record entering the window is compared with the previous w-1 records. As a result this method requires wN comparisons. This is considered feasible, through automated support, for large database systems (Hernandez, 1995).

Our neural net uses a set of rules (previously referred to as "criteria) that decides whether two records are duplicates. User requirements in terms of data duplication are formalized as rules to be used by the net in order to learn from them. Other sources would include database experts who could use past experience in data cleaning, database reengineering, and design work in formulating duplication rules. Rules that had been previously specified during the learning process of a similar database system might also be used.

It is important to note that reliance on human assessment of data quality is minimized once the neural net is built. Instead of a person manually assessing each record for data duplication (or other data quality issues), the net would be responsible for this work. It would

Table 1: Example records based on having the same social security number

Fname	MI	Lname	SSN	DOB	Address	Sex	Salary	Superssn	D#
Franklin	T	Wong	333445555	08-dec-45	638 Voss, Houston TX	M	4000	888665555	5
Frank	S	Wong	333445555	08-dec-45	638 Voxx, Houstojn, TX	M	1800	987654321	
Francis	G	Wonson	333445555	18-dec-55	638 Lilly Lane, Houston, TX	F	4000	888665555	5

Table 2: Percentage of matched data

FNAME	MI	LNAME	SSN	DOB	ADDRESS	SEX	SALARY	SUPERSSN	D#
63	0	100	100	100	67	100	50	25	0

Figure 1: The learning process

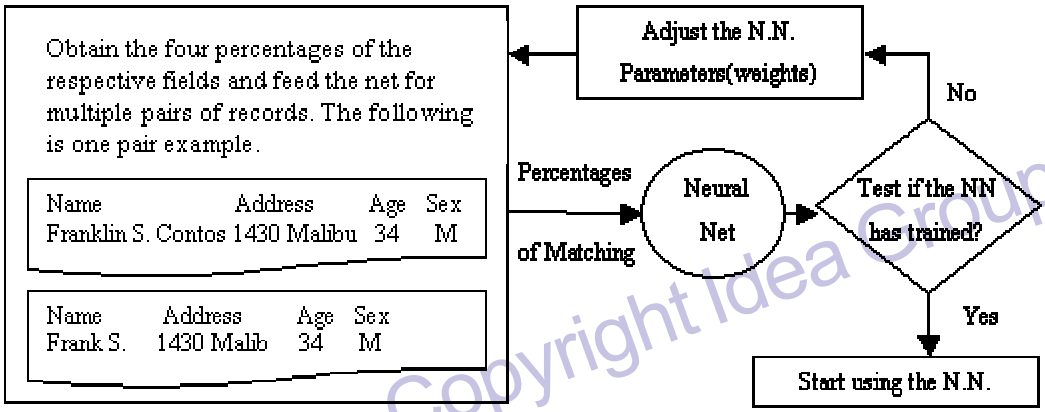

Table 3: Records retrieved (training pattern for inconsistent event-driven data)

FNAME	MI	LNAME	SSN	DOB	ADDRESS	SEX	SAL	SUPERSSN	D#
Franklin	T	Wong	333445555	08-dec-45	638 Voss, Houston TX	M	40000	888665555	5
Frank	S	Wong	333445555	08-dec-45	638 Voxx, Houstojn, TX	M	18000	987654321	

Table 4: Data matching percentages (training pattern for inconsistent event-driven data)

FNAME	MI	LNAME	SSN	DOB	ADDRESS	SEX	SALARY	SUPERSSN	D#
63	0	**100**	**100**	**100**	67	100	60	25	0

Table 5: Records retrieved (training pattern for mismatched unique identifier)

FNAME	MI	LNAME	SSN	DOB	ADDRESS	SEX	SAL	SUPERSSN	D#
Franklin	T	Wong	333445555	08-dec-45	638 Voss, Houston TX	M	40000	888665555	5
Frank	S	Wong	987343444	04-apr-70	638 Voxx, Houstojn, TX	M	18000	987654321	

Table 6: Data matching percentages (training pattern for mismatched unique identifier)

FNAME	MI	LNAME	SSN	DOB	ADDRESS	SEX	SALARY	SUPERSSN	D#
63	0	**100**	11	14	76	100	60	22	0

Table 7: Records retrieved (training pattern for missing data)

FNAME	MI	LNAME	SSN	DOB	ADDRESS	SEX	SAL	SUPERSSN	D#
Tom	R	Johnson	122111111	30-Feb-58	2203 Astle, Spring, TX	M	44000	333445555	2
Thomas	T	Johnson	1221111	30-Feb-58		M	44000	333442222	3

Table 8: Data matching percentages (training pattern for missing data)

FNAME	MI	LNAME	SSN	DOB	ADDRESS	SEX	SALARY	SUPERSSN	D#
17	0	**100**	78	**100**	<not assessed>	100	100	60	0

initially learn from a set of records that were identified as duplicates, and then it could be applied to a sample or the whole database. Figure 1 illustrates the training and learning processes of the neural net.

Training Phase

During the training phase of building the neural net, rules are identified from which learning can occur. We illustrate several of these rules that have been extracted from existing data in the database. In our research, rules are defined in terms of the percentages associated with data matching between data record, though this will be expanded upon in future research efforts.

Training Pattern for Inconsistent Event-driven Data: unique identifier type data matches 100% and event-driven data has low match rate.

The percentages of data matching for these two records are presented above. In this case, the social security, date of birth, gender, and last name are 100% matches. Because these data fields are typically used in databases as part of the unique identifier (or alternate identifi-

ers), data duplication appears to hold. Further analysis of data matching shows that the salary, supervisor, and department data differ, but this data is event-driven. As such, the data disparity (low percentages) is considered a reflection of an event occurrence. In this case, the person may have switched departments and received an increase in pay. For this scenario, the net would be trained to detect record duplication.

Training Pattern for Mismatched Unique Identifier — Unique identifier data has a low match rate, alternate key data (DOB & lname) has less than a 50% match rate, and event-driven and other personal data has a moderate match rate.

The percentages of data matching for these two records are presented above. Notice that there is a significant amount of disparity in data matching within data columns and for each record as a whole. The two unique identifiers (SSN) are significantly different, as shown by the low number of matching digits (11% match rate). Another significant data mismatch is the date of birth. Because of the significant amount of data inconsistencies between these two records, the net is trained to conclude these records are not duplicates.

Training Pattern for Missing Data — Unique identifier data matches 100% in partial data that is 78% complete, alternate key data (DOB & lname) matches 100% in data that is 93% complete, and event-driven data has a moderate match rate.

The percentages of the data matching for these two records are shown above. It is important to note that the amount of data missing for this column is 22%, which is deemed acceptable for decision-making purposes about semantic equivalence between the data values. The date of birth, and gender have a data match of 100%, and the partial last name data matches 100%. The address field is missing in one of the records, so it is not used in the duplicate record assessment. In this particular data source, the missing address data is not considered a significant factor in training the net to find duplicates. For this particular scenario, the net would be trained to detect record duplication.

CONCLUSION AND FUTURE DIRECTIONS

Combining neural nets with other methods, such as the Sorted Neighborhood Method (SNM), can be a powerful mechanism to solve the duplicate records problem (business organizations call this problem the *merge/purge* problem). This is especially the case for large databases that have been built from different sources with heterogeneous representations of information. By generalizing the parameters that we have used to train the neural net, we have proposed that the neural net can be used to replace other more labor-intensive methods of solving data quality problems.

We have identified an initial set of rules from which conclusions can be drawn as to whether a record is a duplicate one. These rules are based on the percentage of data matching in existing data. After training the neural net using these rules, it was used to find duplicate records in a database system. The trained neural net recognized 100% of the duplicate records. This result holds great promise in using trained nets to recognize various types of data quality problems. If this is the case, then labor-intensive, data cleaning may be enhanced or perhaps even replaced by trained neural nets.

The set of data quality rules are being expanded upon, as we learn more about data matching in a larger pool of data. Ongoing research efforts are expanding upon this initial work by applying neural nets to assess data quality at a more granular level of detail.

Future research will examine data cleaning concepts in order to take advantage of techniques that could be employed before neural nets are trained. Basic data cleaning operations that might be employed including removing noise if appropriate, collecting information to model or account for noise, deciding on strategies for handling missing data fields, and accounting for time sequence information and known changes, among others. The combination of neural nets with data cleaning may prove to be a viable approach to the data quality problem.

ENDNOTE

1 The question might be raised, "how could two records be inserted for the same person?" There are numerous reasons for such data corruption. If, for example, the data entry was such that the social security number could be entered twice (no integrity constraint on the data) then both Frank and Franklin could be inserted as two records. When data is imported from several sources into an existing database, integrity constraints have to be turned off, thus allowing duplicate records to be inserted into a table. If the data isn't cleaned, the duplicate records may go undetected.

REFERENCES

Becker A.S. & Berkemeyer A. (1999). "The Application of a Software Testing Technique to Uncover Data Errors in a Database System," **Proceedings of the Pacific Northwest Software Quality Conference,** Portland, Oregon, October 1999.

Bartholomew, D. (1992). "The Price is Wrong," **Information Week**, September 14.

Fausett L. (1994). **Fundamentals of Neural Networks, Architectures, Algorithms, and Applications**, Prentice-Hall, Inc., Upper Saddle River, NJ.

Fox, C., Levitin, A. & Redman, T. (1994). "The Notion of Data and Its Quality Dimensions," **Information Processing and Management,** Vol. 30, No. 1.

Greenfield, L. (1997). "An (informal) Taxonomy of Data Warehouse Data Errors," http://www.dwinfocenter.org/errors.html

Han, Jiawei & Kamber, M. (2001). **Data Mining Concepts and Techniques**, Morgan Kaufmann Publishers, San Francisco, CA.

Haykin, S. (1999). **Neural Networks, A Comprehensive Foundation**, Prentice-Hall, Inc., Upper Saddle River, NJ.

Hernandez M A. and Stolfo, S., Merge/Purge Problem for large databases, Proceedings of the ACM SIGMOD International Conference on Management of Data, , May 1995.

Pitney Bowes, "Third Party Data Increases Revenue and Market share Opportunities"(1998). http://www.pitneysoft.com/new/custreun.htm.