

Chapter I

An Algebraic Approach to Data Quality Metrics for Entity Resolution Over Large Datasets

John Talburt, University of Arkansas at Little Rock, USA

Richard Wang, Massachusetts Institute of Technology, USA

Kimberly Hess, CASA 20th Judicial District, USA

Emily Kuo, Massachusetts Institute of Technology, USA

Abstract

This chapter introduces abstract algebra as a means of understanding and creating data quality metrics for entity resolution, the process in which records determined to represent the same real-world entity are successively located and merged. Entity resolution is a particular form of data mining that is foundational to a number of applications in both industry and government. Examples include commercial customer recognition systems and information sharing on “persons of interest” across federal intelligence agencies. Despite the importance of these applications, most of the data quality literature focuses on measuring the intrinsic quality of individual records than the quality of record grouping or integration. In this chapter, the authors describe current research into the creation and validation of quality metrics for entity resolution, primarily in the context of customer recognition systems. The approach is based on an algebraic view of the system as creating a partition of a set of

entity records based on the indicative information for the entities in question. In this view, the relative quality of entity identification between two systems can be measured in terms of the similarity between the partitions they produce. The authors discuss the difficulty of applying statistical cluster analysis to this problem when the datasets are large and propose an alternative index suitable for these situations. They also report some preliminary experimental results and outline areas and approaches to further research in this area.

Introduction

Traditionally, data quality research and practice have revolved around describing and quantifying the intrinsic quality of individual data records or rows in a database table. However as more and more organizations continue to embrace the strategies of customer relationship management (CRM), new issues are raised related to the quality of integrating or grouping records, especially as it related to the process of entity resolution.

Most current approaches to data integration quality are rooted in the evaluation of traditional data matching or duplicate detection techniques, such as precision and recall graphs (Bilenko & Mooney, 2003). However, these techniques are inadequate for modern knowledge-based entity resolution techniques where two records for the same entity may present entirely different representations, and can only be related to each other through a priori assertions provided by an independent source of associative information.

The authors propose that casting data integration problems in set theoretic terms and applying well-developed definitions and techniques from abstract algebra and statistics can lead to productive approaches for understanding and addressing these issues, especially when applied to very large datasets on the order of 10 to 100 million records or more. The chapter also describes the application of algebraic techniques for defining metrics for grouping accuracy and consistency, including measurement taken on real-world data.

Background

Entity resolution is the process in which records determined to represent the same real-world entity are successively located and merged (Benjelloun, Garcia-Molina, Su, & Widom, 2005). It can also be viewed as a special case of heterogeneous system interoperability (Thuraisingham, 2003). The attributes that are used to determine whether records related to two entities are the same are called “indicative information.” A basic problem is that the indicative information for same entity can vary from record to record, and therefore does not always provide a consistent way to represent or label the entity. Although the specific techniques used to implement a particular entity resolution system will vary, in almost all cases the end result is that the system assigns each entity a unique “token,” a symbol or string of symbols that is a placeholder for the entity. Token-based entity resolution systems fall into two broad classes, based on how the tokens are created: hash tokens and equivalence class tokens.

Hash Tokens

The simplest method for associating a token with an entity is to use an algorithm to calculate or “derive” a value for the token from the primary indicative information for the entity. The derived value is called a “hash token.” For example, if the indicative information for a customer were “Robert Doe, 123 Oak St.,” then the underlying binary representation of this string of characters can be put through a series of rearrangements and numeric operations that might result in a string of characters like “r7H5pK2.”

The use of hash tokens for entity resolution has two drawbacks: hash collisions and lack of consistency. Hash collisions occur when the hash algorithm operating on two different arguments creates the same hash token, thus creating a many-to-one mapping from indicative information to the token representations. There are a number of mitigations for hash collisions, and this does not present a major obstacle for entity resolution.

On the other hand, a more serious problem related to the use of hash tokens is their lack of consistency. Hash algorithms are notoriously sensitive to very small changes in the argument string. For example, even though “Robert Doe, 123 Oak St.” and “Bob Doe, 123 Oak St.” may represent the same customer, most hash algorithms will produce very different hash values for each. Although some systems go to great lengths to “standardize” the argument string before the algorithm is applied (Frederich, 2005) such as changing “Bob” to “Robert,” in the real world the indicative information for the same entity can often change dramatically. For example, “Jane Doe, 123 Pine St.” can marry John Smith and move to a new address, resulting in “Jane Smith, 345 Elm St.” as valid indicative information for the same person. In cases like this, no amount of name and address standardization could enable these two records to produce the same hash token.

Equivalence Class Tokens

One way to improve the consistency of token assignments for these kinds of situations is to use a knowledge base approach (Morgan, McLaughlin, et al., 2000; Morgan, Talley, et al., 2003). As knowledge is acquired that indicative information for an entity has

Table 1. Two equivalence classes

	Token	Representation
Equivalence Class xH45nT	xH45nT	Jane Doe, 123 Pine St.
	xH45nT	Jane Smith, 345 Elm St.
	xH45nT	J S Smith, 345 Elm St.
Equivalence Class y7Bw6	y7Bw6	Robert Doe, 123 Oak St.
	y7Bw6	Bob Doe, 123 Oak St.

changed, the new representation is stored along with other valid representations in a list, called an “equivalence class.” Each equivalence class is assigned an arbitrary, but unique, token value that is not derived from a particular representation of the entity. Table 1 shows how both examples described earlier can easily be accommodated using an equivalence class approach.

If we consider all of the possible entity representations as the underlying set S , then the rule that “two representations are assigned the same token if, and only if, they represent the same entity” defines an equivalence relation on S that partitions S into equivalence classes, that is, all representations associated with the same token. Equivalence classes, equivalence relations, partitions, and other concepts from abstract algebra are not only descriptive, but they also provide important new analysis tools for problems related to data integration and entity resolution (Talburt, Kuo, Wang, & Hess, 2004).

Customer Recognition

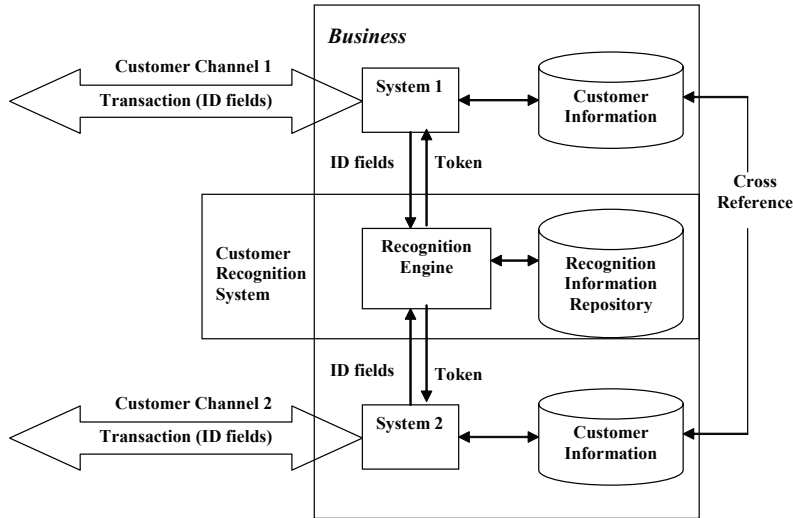
An important commercial application of entity resolution is customer recognition, where the entities in question are the customers of a business, usually an individual or a business (Hughes, 2005). For several years, businesses have realized that in a highly competitive environment they must not only gain market share, but they must also retain and maximize the value of the customers they have. A company will have multiple interactions with the same customer at different times, locations, or lines of business. Each failure to make these connections is a lost opportunity to discover knowledge about a customer’s behavior — knowledge that can make these and future interactions more profitable for the business and more satisfying for the customer. The collection of strategies around maximizing the value of these customer interactions is called customer relationship management, or CRM (Lee, 2000)

Most modern businesses interact with their customers through several channels that carry transactions to and from internal systems. Channels may represent different lines of business (homeowners vs. auto for an insurance company), different sales channels (inbound telephone sales vs. online sales for a retailer), or different geographic locations. It is common for each channel to have its own form of internal customer recognition based on one or more items of identification information. The identification information in the transaction may include some type of internally assigned customer key specific to that particular channel. Even within a single channel, key-based recognition is not perfect. The same customer may be assigned different identifying keys for a number of reasons. The white paper, “Customer-Centric Information Quality Management” (Talburt, Wang, et al., 2004), published through the MITIQ program gives a more complete discussion of the factors that impact the quality of customer recognition.

In a multichannel business the problem is further compounded by the need to recognize and profile customers across channels and synchronize the keys assigned by different channels. Figure 1 shows a typical configuration for a customer recognition system that manages recognition across channels. Note that in this diagram and discussion, the indicative attributes are called “ID fields.”

In Figure 1, the customer transactions coming through the channels include one or more items of identifying information. The two channels are connected to a recognition

Figure 1. Block diagram of a multichannel recognition system



engine, which has access to a repository of recognition information that has been collected from both channels. The information in the repository is organized in such a way that the transactions belonging to the same customer are assigned a unique token as shown in the diagram. The token represents the customer's single, enterprise identity, and is used to bring together the various internal (system) keys the customer may have been assigned at different times or through different channels. For this reason, customer recognition tokens are sometime referred to as "cross-reference identifiers."

Despite the fact that customer recognition is a critical factor in successful CRM solutions, there is little guidance in the literature on metrics specific to entity resolution in general, and customer recognition quality in particular. This chapter attempts to describe a formal approach to quality metrics for entity resolution similar to what has been done by Wang, Lee, and others to develop a data quality algebra for database systems (Wang, Ziad, & Lee, 2001) and information products in general (Huang, Lee, & Wang, 1999).

An Algebraic Model for Entity Resolution

Despite the complexity involved in an actual entity resolution system implementation, its function can be described relatively simply in terms of "equivalence relation" from basic abstract algebra. In this model there are three critical elements. Let $T = \{t_1, t_2, \dots, t_n\}$ represent a finite set of "n" entity transactions that have been processed in a particular order through a given resolution engine. As shown in Figure 1, the engine will assign to each transaction a token.

Definition 1: For a given resolution engine E , and a given order of the transactions T , define the **binary relation** R_E on the set of transactions T by:

$R_E \subset T \times T$, such that

$(t_i, t_j) \in R_E \Leftrightarrow$ The resolution engine E assigns t_i and t_j the same token.

Because E will assign one and only one token to each transaction it processes, it follows that the binary relation R_E defined in this way is an equivalence relation, that is:

1. R_E is reflexive, $(t_i, t_i) \in R_E \quad \forall t_i \in T$
2. R_E is symmetric, $(t_i, t_j) \in R_E \Rightarrow (t_j, t_i) \in R_E$
3. R_E is transitive, $(t_i, t_j) \in R_E, (t_j, t_k) \in R_E \Rightarrow (t_i, t_k) \in R_E$

Definition 2: If P is a set of subsets of a set T , that is, $A \in P \Rightarrow A \subseteq T$, then P is said to be a partition of T if and only if:

$A \in P$ and $B \in P \Rightarrow$ either $A \cap B = \phi$ or $A = B$,

and, $\bigcup_{A \in P} A = T$

Because the binary relation R_E defined on particular ordering of T by a Resolution Engine E is an equivalence relation, the set of all equivalence classes of R is a partition P_R of T , that is, if $P_i = \{t_j \mid (t_j, t_i) \in R\}$, then $P_E = \{P_i \mid 1 \leq i \leq n\}$ is a partition of T .

Each equivalence class P_i represents all of the transactions belonging to the same entity as determined by the resolution engine.

Definition 3: If E is an entity resolution engine, T is a set of transactions, α is a particular ordering of T , and P_E is the partition of T generated by the equivalence relation R_E , then $\{E, T, \alpha, P_E\}$ is an entity resolution model.

Different resolution engines, different transactions sets, or even different orderings of the same transaction set will produce different models. However, the models are considered equivalent if they produce the same partition of the transaction set.

Definition 4: Two entity resolution models, $\{R, T, \alpha, P_R\}$ and $\{S, T, \beta, P_S\}$, are equivalent over the same transaction set T if and only if $P_R = P_S$.

Note that Definition 4 requires the models be defined over the same set of transactions. However, different engines and different orderings of the transactions comprise different models, which may or may not be equivalent.

As a simple example, suppose that R assigns an incoming entity transaction a token that is the same as the first previously processed transaction where the last names are no more than one character different, and the street numbers are the same.

Table 2 shows that the four transactions processed in the order shown would be clas-

Table 2. Classification into two partition classes

Order (α)	Transactions (T)	Token
T ₁	(Smithe, 101 Oak St.)	A
T ₂	(Smith, 101 Elm St.)	A
T ₃	(Smith, 202 Oak St.)	B
T ₄	(Smythe, 101 Pine St.)	A

Table 3. Classification into three partition classes

Order (β)	Transactions (T)	Token
T ₄	(Smythe, 101 Pine St.)	A
T ₃	(Smith, 202 Oak St.)	B
T ₂	(Smith, 101 Elm St.)	C
T ₁	(Smithe, 101 Oak St.)	A

sified into two partition classes: $\{T_1, T_2, T_4\}$ and $\{T_3\}$. The first transaction would be assigned a token of “A”. The second transaction would be compared to the first, and because “Smithe” and “Smith” are only one character different, and the street numbers are the same, it would also be assigned “A”. The third transaction has a street number that does not match either the first or second transaction, and would therefore receive a different token of “B”. Finally, the fourth transaction would be assigned “A” because when compared to the first transaction, “Smythe” is only one character different than “Smithe” and the street numbers are the same.

On the other hand, Table 3 shows the outcome of processing the same set of transactions with the same resolution rules, but reversing the order of processing. In this case, the four transactions are classified into three partition classes: $\{T_1, T_4\}$, $\{T_2\}$, and $\{T_3\}$. In this processing order, the third transaction processed (T₂) does not match the first transaction (T₄) because “Smythe” and “Smith” differ by two characters, and does not match the second transaction (T₃) because the street numbers are different.

Definition 5: A resolution engine R is said to be order invariant over a set of transactions T if and only if R produces the same partition for every ordering of T.

Partition Similarity

Definition 4 relates the equality (equivalence) of two resolution models to the equality of the partitions they produce. In the same way, the relative similarity of two resolution models can be based on the relative similarity of the partitions they produce. However in this case, the definition of similarity between partitions is less clear. A number of similarity “indices” have been developed in statistics in connection with cluster analysis. The primary

consideration in selecting a particular index for an application is the extent to which it provides adequate discrimination (sensitivity) for a particular application. As a starting point in the initial research, the authors have chosen to test three indices, the Rand index (Rand, 1971) and the adjusted Rand index (Yeung & Ruzzo, 2001), in the initial research, and the TW index developed by the authors and described in this chapter.

The Talburt-Wang index was designed by the authors to provide an easily calculated baseline measure. The Rand index and adjusted Rand index have been taken from the literature on cluster analysis and recommended for cases where the two partitions have a different number of partition classes (Hubert & Arabie, 1985). These indices have a more complex calculation than the Talburt-Wang index, involving the formula for counting the combinations of n things taken two at a time, $C(n,2)$. Because transaction sets can be on the order of hundreds of thousands or even millions of records, the combination calculations for the Rand and adjusted Rand indices can exceed the limits of single precision for some statistical packages. Moreover, the lack of symmetry in the calculations for these indices requires that either a very large amount of main memory be available to make all of the calculations in a single pass of the transactions, or that the transactions be sorted and processed twice.

Talburt-Wang Index

Definition 6: If A and B are two partitions of a set T , define $\Phi(A,B)$, the partition overlap of A and B , as follows:

$$\Phi(A,B) = \sum_{i=1}^{|A|} |\{B_j \in B \mid B_j \cap A_i \neq \phi\}|$$

For a given partition class of partition A , it counts how many partition classes of partition B have a non-empty intersection with it. These are summed over all partition classes of A .

Theorem 1: If A and B are two partitions of a set T , then $\Phi(A,B) = \Phi(B,A)$.

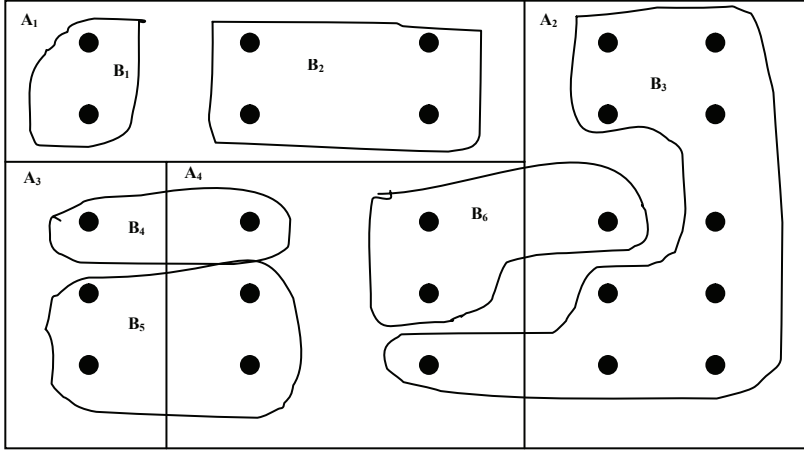
Proof: It is easy to see that the definitions of $\Phi(A,B)$ and $\Phi(B,A)$ are symmetric.

Definition 7: If A and B are two partitions of a set T , define $\Delta(A,B)$, the Talburt-Wang index between A and B , as follows:

$$\Delta(A,B) = \frac{|A| \cdot |B|}{(\Phi(A,B))^2}$$

Figure 2 shows a 5-by-5 array of 25 points that represents an underlying set T . The four partition classes of partition A are represented as rectangles labeled A_1 through A_4 , and the six partition classes of partition B are represented by the oval shapes labeled B_1 through B_6 .

Figure 2. Array diagram of two partitions A and B



The calculation of the overlap of A and B for this example is:

$$\begin{aligned}
 & \Phi(A, B) \\
 &= |\{B_j \in B \mid B_j \cap A_1 \neq \phi\}| + |\{B_j \in B \mid B_j \cap A_2 \neq \phi\}| + |\{B_j \in B \mid B_j \cap A_3 \neq \phi\}| \\
 & \quad + |\{B_j \in B \mid B_j \cap A_4 \neq \phi\}| \\
 &= |\{B_1, B_2\}| + |\{B_3, B_6\}| + |\{B_4, B_5\}| + |\{B_3, B_4, B_5, B_6\}| \\
 &= 2 + 2 + 2 + 4 = 10
 \end{aligned}$$

Therefore,

$$\Delta(A, B) = \frac{|A| \cdot |B|}{(\Phi(A, B))^2} = \frac{4 \cdot 6}{10^2} = 0.24$$

Corollary 1: If A and B are partitions of the set T, then $\Delta(A, B) = \Delta(B, A)$.

Definition 8: If A and B are partitions of the set T, partition A is said to be a “refinement” of partition B, if and only if

$$A_i \in A \Rightarrow A_i \subseteq B_j \text{ for some } j, 1 \leq j \leq |B| ;$$

that is, every partition class of partition A is a subset of some partition class of partition B.

Theorem 2: If A and B are partitions of the set T, and A is a refinement of B, then:

$$\Delta(A, B) = \frac{|B|}{|A|}$$

Proof: If A is a refinement of B, then every partition class of A will intersect only one partition class of B. Therefore:

$$\Phi(A, B) = \sum_{i=1}^{|A|} |\{B_j \in B \mid B_j \cap A_i \neq \emptyset\}| = \sum_{i=1}^{|A|} (1) = |A|$$

and

$$\Delta(A, B) = \frac{|A| \cdot |B|}{(\Phi(A, B))^2} = \frac{|A| \cdot |B|}{|A|^2} = \frac{|B|}{|A|}$$

From Definition 6, it is easy to see that:

$$\Phi(A, B) \geq \max(|A|, |B|).$$

Consequently, by Definition 7:

$$\Delta(A, B) \leq 1.$$

The following theorem shows that the Talburt-Wang index is equal to one, only when the partitions are identical.

Theorem 3: A and B are identical partitions of T, if and only if $\Delta(A, B) = 1$.

Proof: Suppose the A and B are identical partitions of T. Then A must be a refinement of B. By Theorem 2:

$$\Delta(A, B) = \frac{|B|}{|A|}.$$

However, because A and B are identical, $|A| = |B|$. Consequently, $\Delta(A, B) = 1$.

The converse can be demonstrated by observing that Definition 6 requires that:

$$\Phi(A, B) \geq \max\{|A|, |B|\}.$$

Any difference between partitions A and B will mean that either $\Phi(A, B) > |A|$ or $\Phi(A, B) > |B|$ and, consequently:

$$\Delta(A, B) = \frac{|A| \cdot |B|}{(\Phi(A, B))^2} < 1.$$

Corollary 2: If A is any partition of T, and B is the “trivial partition” of T, that is, $B = \{T\}$, then:

$$\Delta(A, B) = \frac{1}{|A|}.$$

Proof: Every partition is a refinement of the trivial partition. Therefore by Theorem 2:

$$\Delta(A, B) = \frac{|B|}{|A|} = \frac{1}{|A|}.$$

Corollary 3: If A is the “point partition” of T, that is, $A = \{\{t_1\}, \{t_2\}, \dots, \{t_n\}\}$ where each partition class of A contains only one element of T, and B is any partition of T, then:

$$\Delta(A, B) = \frac{|B|}{|T|}.$$

Proof: The “point partition” is a refinement of every partition. Again by Theorem 2:

$$\Delta(A, B) = \frac{|B|}{|A|} = \frac{|B|}{|T|}.$$

Corollary 4: If A is the “point partition” of T, and B is the trivial partition of T, then:

$$\Delta(A, B) = \frac{1}{|T|}.$$

Proof: Apply Corollaries 2 and 3 together.

Although the Talburt-Wang index will always be greater than zero, Corollary 4 shows that it approaches zero for the point partition of an arbitrarily large set T. Therefore, the Talburt-Wang index takes on values in the half open interval $(0, 1]$.

Rand Index and Adjusted Rand Index

The Rand index (Rand, 1971) and the adjusted Rand index (Yeung & Ruzzo, 2001) are both commonly used indices to compare clustering results against external criteria (Hubert

Table 4. Intersection matrix for partitions A and B

A\B	B ₁	B ₂	...	B _n	Sums
A ₁	C ₁₁	C ₁₂	...	C _{1n}	S _{1*}
A ₂	C ₂₁	C ₂₂	...	C _{2n}	S _{2*}
...
A _m	C _{m1}	C _{m2}	...	C _{mn}	S _{m*}
Sums	S _{*1}	S _{*2}	...	S _{*n}	S _{mn}

& Arabie, 1985). The computation of these indices is best explained using a tabular representation of the overlap between two partitions.

If A and B are two partitions of the set T, the overlap between A and B can be represented in Table 4.

In Table 4, the row and column entry C_{ij} represents the count of elements in the intersection between partition class A_i of partition A and the partition class B_j of partition B. Each row sum S_{i*} is equal to the number of elements in the partition class A_i , and the column sum S_{*j} is equal to the number of elements in the partition class B_j . The sum S_{mn} is equal to the number of elements in the underlying set T.

The calculation of both the Rand index and adjusted Rand index can be expressed in terms of four values: x, y, z, and w, defined as follows:

$$x = \sum_{i,j} \left(\frac{C_{ij}}{2} \right), \text{ where } \binom{N}{2} = \frac{N \cdot (N-1)}{2}$$

$$y = \sum_i \left(\frac{S_{i*}}{2} \right) - x$$

$$z = \sum_j \left(\frac{S_{*j}}{2} \right) - x$$

$$w = \left(\frac{S_{mn}}{2} \right) - x - y - z$$

Based on these values:

$$\text{Rand index} = \frac{x + w}{x + y + z + w}$$

Table 5. Intersection matrix for the partitions of Figure 2

A\B	B ₁	B ₂	B ₃	B ₄	B ₅	B ₆	Sums
A ₁	2	4	0	0	0	0	6
A ₂	0	0	9	0	0	1	10
A ₃	0	0	0	1	2	0	3
A ₄	0	0	1	1	2	2	6
Sums	2	4	10	2	4	3	25

$$\text{adjusted Rand index} = \frac{x - \left(\frac{(y+x) \cdot (z+x)}{x+y+z+w} \right)}{\frac{(y+z+2x)}{2} - \left(\frac{(y+x) \cdot (z+x)}{x+y+z+w} \right)}$$

The primary difference is that the adjusted Rand takes on a wider range of values thus increasing its sensitivity.

Transforming the example of Figure 1 into tabular form yields Table 5.

Based on these counts:

$$x = 1 + 6 + 36 + 1 + 1 + 1 = 46$$

$$y = 15 + 45 + 3 + 15 - 46 = 32$$

$$z = 1 + 6 + 45 + 1 + 6 + 3 - 46 = 16$$

$$w = 300 - 46 - 32 - 16 = 206$$

$$\text{Rand index} = (46+206)/(46 + 32 + 16 + 206) = 0.84$$

$$\text{adjusted Rand index} = (46 - (78*62)/300)/((78 + 62)/2 - (78*62)/300) = 0.5546$$

By contrast:

$$\text{Talbert-Wang index} = 0.24$$

An important aspect of the preliminary research is to determine which one, or which possible combination, of these indices provides an appropriate level of discrimination in comparing the partitions actually generated by entity resolution applications involving large volumes of transactions.

Entity Resolution Quality Metrics

Given that entity resolution system outcomes can be represented as partitions, and that an appropriate index has been selected to assess the degree of difference between partitions, the next step is to investigate the use of the index to create data quality metrics relevant to

entity resolution systems. Having measurements appropriate for critical touch points in a data process flow is an important aspect of any total data quality strategy (Campbell & Wilhoit, 2003). For purposes of this discussion, we will simply refer to it as the “similarity index.” The following suggests how a partition similarity index could be applied.

Metric for Entity Resolution Consistency

The following describes three contexts in which a similarity index could provide a type of consistency metric for cases where the entity resolution is customer recognition. The first is a comparison between two different recognition systems, and the second is an assessment of changes to a single recognition system. In both cases we hold the transaction set fixed. Experiments 1 and 2 illustrate these two applications, respectively. A third example (Experiment 3) considers the case where the engine is held fixed and the transaction set changes in quality.

Experiment 1: Different Engines

In this experiment, the first recognition system R is a CDI product based on traditional “merge/purge” approximate string matching technology, and the second system S is a newer customer data integration (CDI) product using both matching and a knowledge base of external information about occupancy associations. Both R and S are used as the recognition engine in Customer Recognition applications. T is a fixed set of ordered customer transaction.

Tables 6 and 7 show a comparison of the partitions A and B created by R and S respectively.

Table 6. Results of Experiment 1

Statistic	A	B
Record Cnt	673,003	673,003
Class Cnt	175,527	136,795
Single Cnt	112,857	62,839
Avg Class	3.83	4.92
Max Class	110	80

Table 7. Similarity index results

Index	Value
Talburt-Wang	0.4339
Rand	0.9998
Adj Rand	0.8104

In this experiment, the second partition B shows more grouping, in that it has fewer partition classes than the partition A created by engine R that relies entirely on string matching. On average the partition classes created by the knowledge-assisted engine S are larger, and there are fewer singleton classes. These all indicate that the knowledge-assisted recognition engine S groups more transactions. Presumably this can be attributed to the additional knowledge that allows some of the “match only” classes of R to be consolidated into a single class using external knowledge. For example, partition A may contain two classes: one with two transactions, {“John Jones, 123 Main”, “J. Jones, 123 Main”}, and another with one transaction {“John Jones, 345 Oak”}. However if external knowledge indicates that “John Jones” has moved from “123 Main” to “345 Oak”, then these three transactions would be in the same class of partition B, that is, the class {“John Jones, 123 Main”, “J. Jones, 123 Main”, “John Jones, 345 Oak”}.

Although this may be an expected result, the indices only indicate the degree to which R and S generate different partitions, with the profile showing that R makes fewer associations (on average) than S. The measurement does not indicate which, if either, makes more correct associations. Furthermore, the three indices vary widely on the degree of similarity with the Rand indicating a rather strong similarity, the Talburt-Wang index a fairly strong difference, and the adjusted Rand somewhere in the middle.

Experiment 2: Changes to the Same Engine

Having a way to measure the impact of changes to the recognition engine can also be very useful in assessing recognition quality, especially in the initial phases of a system implementation. In this scenario, the input transactions are held fixed, and the grouping is performed twice, once before the change (R), and once after the change (S). The similarity index provides a metric for assessing the change in groupings that can be attributed to the change in the recognition engine.

In this experiment, R is the April release of a knowledge-based CDI product that is released monthly and used in customer recognition applications. S is the May release of the same product. T is a fixed set of ordered customer transactions.

Tables 8 and 9 show a comparison of the partitions A and B created by R and S, respectively.

Table 8. Results of Experiment 2

Statistic	A	B
Record Cnt	17,778	17,778
Class Cnt	3,218	3,223
Single Cnt	1,271	1,222
Avg Class	5.53	5.52
Max Class	63	63

Table 9. Similarity index results

Index	Value
Talburt-Wang	0.9972
Rand	0.9999
Adj Rand	0.9989

Although the partition of the new release (B) shows increased clustering in terms of fewer singleton classes and fewer classes overall, the average class size has slightly decreased. This would be an expected result if we believe that in a knowledge-based approach, knowledge about the entities in a fixed set of transactions increases over time; that is, there is a time-latency in knowledge gathering. Under this assumption and given that the transactions are held fixed in time, one could expect that knowledge about these transactions (customers) will increase over time, and that the engine's ability to connect transactions for the same customer will improve. In this particular measurement, all three indices point to a very high degree of similarity (consistency) between the partitions produced by the two releases, and the second release brings together slightly more transactions. However this measurement only points to stability between the two releases and does not prove that the second release is more or less accurate in grouping than the first.

Experiment 3: Changes in Input Quality

Here the Recognition Engine is held fixed and the transaction set is intentionally degraded in quality. For experimental purposes, the change (error) can be introduced at a fixed rate.

In this experiment, R is a knowledge-based CDI product used in customer recognition applications and is held fixed. R identifies individual customers based on name and address (occupancy). First, R processes the ordered transaction set T to create the partition A. Next, the quality of T is deliberately degraded by removing all vowels from the names in 800 of

Table 10. Results of Experiment 3

Statistic	A	B
Record Cnt	17,788	17,788
Class Cnt	3,218	3,332
Single Cnt	1,271	1,675
Avg Class	5.53	5.34
Max Class	63	60

Table 11. Similarity index results

Index	Value
Talbur-Wang	0.6665
Rand	0.9998
Adj Rand	0.8782

the 17,788 transaction records (4.5%), and R processes the degraded transactions to create the second partition B.

Tables 10 and 11 show a comparison of the partitions A and B created by R and S respectively.

In this scenario, the effect of quality degradation is evident. Even though more classes are created from the degraded transactions, the number of singleton classes has increased dramatically. These represent records that were formerly integrated into larger classes, but due to degradation, cannot be matched and become outliers. The average size of the classes has also decreased significantly. Again, the Talbur-Wang index is the most sensitive to this change, whereas the Rand indicates almost complete similarity.

Metric for Customer Recognition Accuracy

If A and B are both partitions of the same ordered transaction set T, and if A represents the “correct partition of T” (i.e., is a benchmark), and B represents the partition of T imposed by some recognition system R, then the similarity index can provide a quantifiable and objective measure of the accuracy of the recognition system R. Because all of the indices described previously have the characteristic of taking on the value of 1 when the partitions are identical, and values less than 1 as the partitions become dissimilar, then the value of the similarity index times 100 (or some normalized transformation of the similarity index) can be used as an accuracy metric.

Even though it is evident how one could create an accuracy metric for customer recognition using a similarity index, it is less obvious how to create the benchmark of correct groupings. In practice, this can be very difficult to do. The authors have experience in using the following methods to create a benchmark.

In the case of recognition systems that rely only on matching, it is possible to create correct groupings by manually inspecting the records and making an expert judgment about which records belong in each class. The primary limitation of this method is the effort required to create a benchmark of any significant size. In addition, experts do not always agree, and this method may require some type of arbitration, such as a voting scheme.

However in the case of knowledge-based recognition systems using equivalence class tokens, manual inspection is not enough. For example, the mere inspection of two occupancy records, such as “Jane Smith, 123 Oak” and “Jane Jones, 456 Elm,” cannot establish if they should or should not be in the same class without a priori knowledge that these represent the same customer who has married and moved to a new address. In this situation, creating a benchmark requires accurate information about changes in addresses and changes in

names that is best obtained from the customers. Such a benchmark can be both expensive and difficult to create, even for a relatively small sample (Talburt & Holland, 2003). Even attempts to create these by having company employees volunteer this information have been largely abandoned due to privacy and legal concerns.

Future Trends

Although there are a number of interesting problems related to the quality of entity resolution, two have the most importance: grouping accuracy and the time-to-failure problem, also known as the system entropy problem.

Grouping Accuracy Problem

As discussed earlier, the problem in determining accuracy of a particular grouping is the difficulty in establishing the correct benchmark. To overcome this problem, at least two approaches have been suggested: verification of match exceptions and synthetic data generation.

Match Exceptions Approach

This is an approach that could apply to those entity resolution systems in which the majority of associations between entity records are established through matching. However, this is often the case in commercial applications such as customer recognition. Consider the example of a knowledge-based customer recognition system where the indicative information is customer name and address. A typical grouping generated by the system would predominately comprise records of the same name and same address within the tolerance of a given set of matching rules. For example, the matching rules might allow for single edit differences such as a missing letter or transposed characters. It also might allow for aliases and abbreviations, such as Bob for Robert, or St for Street.

Because the system is knowledge based, there may also be some classes that contain associations where the name and/or the address are different. This basis of approach would be to factor out the matching associations and focus only on the smaller number of cross-name and cross-address associations. For a large dataset, it would still be necessary to work with only a sample of these cases.

The objective is to reduce the amount of verification to manageable amount of time and effort. The assumptions would be that the records established through matching are essentially correct, and the error found in the verification of the cross-name and cross-address sample can be extrapolated to the entire dataset.

Because this method begins by looking at the correctness of associations that have already been made, it is biased against estimating errors for failing to make associations. What is lacking here is any empirical understanding of the expected rate of cross-name and cross-address association for a particular kind of dataset.

This approach is more complex, but somewhat akin to the estimation of a population mean through sampling. In this case, there is a correct, but unknown, partition of the dataset, and we assume that the given partition is an approximation of the correct partition. Is

it possible to develop a methodology to estimate how similar the given partition is to the correct partition from the known errors of association found in some sample of the partition classes?

Synthetic Data Approach

An alternative approach is to generate a correct partition with supporting associations, extract a partial transaction set, resolve the transaction set into a partition, then apply the similarity index to measure the accuracy of the resolution engine. The issue here is how closely the target population of entities can be simulated. The use of synthetically generated occupancy information has been used in the context of partitioning data over nodes in a grid application where the validation of performance optimization is dependent upon the data reflecting real-world name and address characteristics (White & Thompson, 2005).

Again using the example of a knowledge-based customer recognition system, the generation of the synthetic data would have to account for numerous factors that could affect the performance of the resolution system, such as distribution of name frequencies, rates of change in address individually and by household, and rates of change in name from marriage and divorce.

After the synthetic entity population has been created, the second step is to extract a set of transactions to process through the resolution engine. The partition created from the extracted transactions can then be compared to the correct partition to determine the accuracy of resolution.

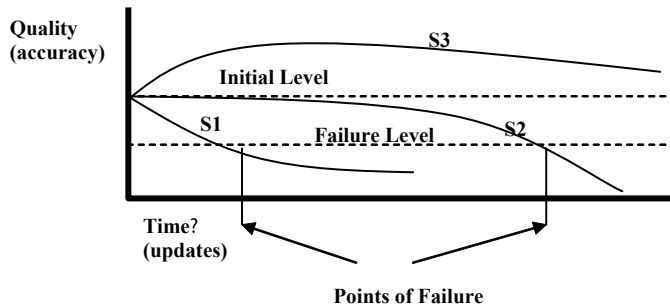
This approach is a direct test of the performance of the resolution engine and leads to a number of interesting experiments. If the transaction set used is the entire synthetic dataset including associative transactions, then the resolution engine should recreate exactly the correct partition. Another series of experiments can be set up that will test the sensitivity of the resolution engine to error and omissions, or even the order of processing, of the transaction set.

For example, if the assumption is that only 80% of individual moves can typically be found through associative sources, then extracting only 80% of the change-of-address assertion from the full universe into the transaction set would indicate the amount of error in the resolution attributable to this cause. Using a degraded extraction of the correct dataset to create the transaction dataset could be used to assess the impact of types of errors and incompleteness in the transaction dataset.

Time-to-Failure Problem

In physics the Law of Entropy states that a system left on its own will spontaneously tend toward disorder; that is, “entropy”, the measure of disorder, will increase. Energy must be expended to increase the system’s organization, that is, decrease entropy. An automobile is a highly organized, statistically improbable configuration of materials that requires a great deal of energy to produce. Accidents and everyday wear and tear tend to make it less like its original “new” configuration, and at some point, unusable. Energy, in the form of repairs, must be expended to move it back toward its original configuration. In general we do not expect cars to spontaneously look new again. Everything that happens tends to make it less organized, that is, less like a new car.

Figure 3. Change in quality over time for three systems



The concept of entropy can also be applied to an entity resolution system, in particular to the configuration of its identification repository. The question becomes whether updates to the system cause the entropy to increase (be less organized) or the entropy to decrease (become more organized). Given that the measure organization is the correct association of transactions, increasing entropy is manifest as decreasing accuracy of the system and vice versa.

A commonly occurring failure of many entity resolution implementations is typified by what is often observed in commercial customer recognition systems. They often initially perform well based on the set of transactions available during setup, but then steadily degrade in accuracy as new update requests are received. Degradation of customer recognition accuracy over time is perhaps the most overlooked and the least understood point of failure in CRM implementations.

Figure 3 illustrates the changes in quality as a function of time of three hypothetical recognition systems: S1, S2, and S3. In this case, time implies change through update requests that change the configuration of the systems' repositories. System S1 shows a rapid degradation in quality with a short time to failure, that is, accuracy below a given threshold. S2 has a longer time to failure, and S3 shows a system where quality improves as updates are processed.

Metrics and case studies of entity resolution quality over time are virtually nonexistent and are fertile ground for further study and research.

Conclusion

The algebraic approach of characterizing entity resolution systems as partitions of ordered transaction sets is proving to be useful in creating metrics for quality assessment. In addition to providing an easily understood model, it also opens the door to utilizing the research literature already available related to cluster analysis.

Although the preliminary experiments indicate that the Talburt-Wang index provides even more discrimination than the Rand or adjusted Rand indices, and is easier to calculate,

further testing on a broader range of recognition outcomes needs to be done before abandoning these or other techniques. In the end, the quality of entity resolution outcomes will probably best be expressed in terms of a number of key metrics, of which the Talburt-Wang index is only one.

There are many fertile areas of research related to the quality of entity resolution, and clearly better answers are needed to assist companies and government agencies dealing with the implementations of these systems.

References

- Benjelloun, O., Garcia-Molina, H., Su, Q., & Widom, J. (2005, March 3). *Swoosh: A generic approach to entity resolution* (Technical report). Stanford InfoLab.
- Bilenko, M., & Mooney, R. J. (2003, August 27). On evaluation of training-set construction for duplicate detection. In *Proceedings of the ACM SIGKDD-03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, Washington, DC (pp. 7-12).
- Campbell, T., & Wilhoit, Z. (2003, November 7-9). How's your data quality? A case study in corporate data quality strategy. In *Proceedings of the International Conference on Information Quality* (pp. 112-124). Massachusetts Institute of Technology.
- Frederich, A. (2005, May). *IBM DB2 Anonymous Resolution: Knowledge discovery without knowledge disclosure* (IBM White Paper). Retrieved March 27, 2006, from <http://faculty.washington.edu/kayee/pca/supp.pdf>
- Huang, K., Lee, Y. W., & Wang, R. Y. (1999). *Quality information and knowledge*. Upper Saddle River, NJ: Prentice Hall.
- Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classifications*, 193-218.
- Hughes, A. M. (2005). *Building customer loyalty by recognition*. Database Marketing Institute. Retrieved March 27, 2006, from <http://www.local6.com/news/4643968/detail.html>
- Lee, D. (2000). *The customer relationship management survival guide*. San Diego, CA: HYM Press.
- Morgan, C. D., McLaughlin, G. L., Fogata, M. G., Baker, J. L., Cook, J. E., Mooney, J. E., et al. (2000, June 6). *Method and system for the creation, enhancement, and update of remote data using persistent keys* (U.S. Patent No. 6,073,140). Washington, DC: U.S. Patent and Trademark Office.
- Morgan, C. D., Talley, T., Talburt, J. R., Bussell, C., Kooshesh, A., Anderson, W., et al. (2003, February 18). *Data linking system and method using tokens* (U.S. Patent No. 6,523,041). Washington, D.C : U.S. Patent and Trademark Office.
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66, 846-850.
- Talburt, J. R., & Holland, G. (2003). A shared system for assessing consumer occupancy and demographic accuracy. In *Proceedings of the International Conference on Information Quality* (pp. 166-177). Massachusetts Institute of Technology.
- Talburt, J. R., Kuo, E., Wang, R., & Hess, K. (2004, November 5-7). An algebraic approach to data quality metrics for customer recognition. In *Proceedings of the 9th International Conference on Information Quality (ICIQ-2004)* (pp. 234-247). Cambridge, MA.

- Talburt, J. R., Wang, R. Y., et al. (2004). *Customer-centric information quality management* (MITIQ White Paper). Retrieved March 27, 2006, from <http://mitiq.mit.edu/Documents/CCIQM/CCIQM%20White%20Paper.pdf>
- Thuraisingham, B. (2003). *Web data mining and applications in business intelligence and counter-terrorism*. Boca Raton, FL: CRC Press.
- Wang, R. Y., Ziad, M., & Lee, Y. W. (2001). *Data quality*. Norwell, MA: Kluwer Academic Publishers.
- White, J., & Thompson, D. R. (2005, June 20-23). Load balancing on a grid using data characteristics. In *Proceedings of the 2005 International Conference on Grid Computing and Applications* (pp. 184-188). Las Vegas, NV.
- Yeung, K. Y., & Ruzzo, W. L. (2001). Details of the adjusted Rand index and clustering algorithms, supplement to the paper "An empirical study on principal component analysis for clustering gene expression data". *Bioinformatics*, 17(9), 763-774.