

Un-Interpreted Schema Matching with Embedded Value Mapping under Opaque Column Names and Data Values

Anuj Jaiswal, David J. Miller, *Member, IEEE*, and Prasenjit Mitra

Abstract—Schema matching and value mapping across two heterogeneous information sources are critical tasks in applications involving data integration, data warehousing and federation of databases. Before data can be integrated from multiple tables, the columns and the values appearing in the tables must be matched. The complexity of the problem grows quickly with the number of data attributes/columns to be matched and due to multiple semantics of data values. Traditional research has tackled schema matching and value mapping independently. We propose a novel method that optimizes embedded value mappings to enhance schema matching in the presence of opaque data values and column names. In this approach, the fitness objective for matching a pair of attributes from two schemas depends on the value mapping function for each of the two attributes. Suitable fitness objectives include the Euclidean distance measure, which we use in our experimental study, as well as relative (cross) entropy. We propose a heuristic local descent optimization strategy that uses sorting and two-opt switching to jointly optimize value mappings and attribute matches. Our experiments show that our proposed technique outperforms earlier un-interpreted schema matching methods and thus should form a useful addition to a suite of (semi) automated tools for resolving structural heterogeneity.

Index Terms—Schema Matching, Opaque Conditions, Embedded Schema Matching with Value Mapping



1 INTRODUCTION

DATA integration becomes a necessity in order to answer queries, make decisions that require information from multiple sources. However, since data sources typically are independently created or designed, massive heterogeneity between data sources exists. Integrating data from heterogeneous sources often involves solving two related subtasks: (1) matching schema attributes across the information sources - the *schema matching* problem; and (2) mapping data values across matched schema attributes - the *value mapping* problem. Before data from two columns in different tables can be integrated, we must ensure that the semantics of the data in the two columns are the same. Similarly, for a pair of matched columns, if the same datum value is encoded differently in the two columns, the correspondence between the differently encoded data values must be established and the data from one column transformed (to the same encoding) before they can be integrated. Schema and value matching are necessary in applications involving data integration, data warehousing, data interoperation and federation of databases and are considered fundamental and challenging problems that must be resolved before such systems can be successfully implemented.

While there has been extensive research on schema matching [2], [3], [4], [8], [9], [10], [13], [17], [19], [20], [21], [23], [24], [25], [27], [30] and value mapping [1], [5], [11], [12],

[14], [15], [16], [22], [29], the existing solutions mostly tackle these problems independently. In addition, most previous solutions to the schema matching problem assume that the two databases that need to be integrated use a common (standard) syntax/description language for data values or at least have some lexical similarities in their data values. However, these assumptions are often violated in practice. For example, “Four Wheel Drive Sedan” can be represented as “4WD-Sedan” or “4WD-S” or even “4DSD4W” in different data sources. An intelligent lexical matching algorithm may be able to deduce correspondences between the first three data representations, but will fail to provide a correspondence to “4DSD4W” because no semantic or syntactic relationship exists between this and the three other values. Furthermore, schema matching proves exceptionally difficult as the number of data attributes increases because the number of candidate solutions grows exponentially with the number of attributes.

Existing techniques for schema matching utilize statistics involving single attributes (e.g. Entropy) or pairs of attributes (in particular, Mutual Information [17]). These techniques are convenient when the data values are “opaque”, i.e., when it is difficult to understand the semantics of the data value from its name. However, these methods inherently assume there are distinct values for these statistics unique to each attribute/attribute pair (across the two tables). Consequently, these methods will fail when there are multiple attributes/attribute pairs with similar statistics for the same schema. We will illustrate this problem below.

We present an automated technique that utilizes value mappings to match schemas accurately. The technique is designed to work in the particularly difficult cases where *both* the column names and the data instances are “opaque” and when

- Anuj Jaiswal and Prasenjit Mitra are with the College of Information Sciences and Technology at the Pennsylvania State University. Email: {ajaiswal, pmitra}@ist.psu.edu
- David J. Miller is with the Department of Electrical Engineering at the Pennsylvania State University. Email: djmiller@engr.psu.edu

TABLE 1

Two schemas, (a) and (b), and corresponding records taken from hypothetical Automotive Databases

(a)			(b)		
Model(X)	Color (Y)	Type (Z)	A	B	C
XE	White	Sedan	X	A1	1
XLE	Red	Coupe	XL	A2	2
LE	Silver	SUV	GE	A3	3
LX			GX		

multiple attributes/attribute pairs may have similar first/second order statistics, i.e., similar entropies or mutual informations¹. Instead of characterizing a column by a single value like entropy, our technique characterizes each attribute (or attribute pair) by the probability mass function (pmf) defined over its value support set². We introduce a suitable dissimilarity measure between probability mass functions (e.g., Euclidean distance, relative entropy). Our objective function (heretofore interchangeably referred to simply as objective) is thus a sum of these dissimilarities, one per attribute (or attribute pair) match. Note that evaluation of a pmf-based dissimilarity measure for a pair of attributes to be matched requires the specification of *value mappings/value correspondences* among the matched attributes. Thus, to minimize our objective, we must perform both attribute matching and *embedded* value mappings for the matched attributes.

As some preliminary discussion to help fix our ideas, first consider the simple example in Table 1, which shows two schemas, (a) and (b), each with three attributes(columns). For each schema, there are four database records (rows). Assume that these schemas are drawn from two automotive corporations or from different divisions within a single corporation. Table 1(a) utilizes human-readable codes while Table 1(b) utilizes corporation-specific codes. Only someone with access to a semantic description of the value codes for Table 1(b) will understand what these codes mean. Suppose that ground-truth there is a one-to-one correspondence between the attributes of the two tables and between values across the matching attributes. Conventional instance-based schema-matching tools might be able to find an alignment between the attributes “Model” and “A” due to syntactic or lexical similarities between the value strings that occur under “Model” and “A”. However, no structural alignment for other attribute pairs can be found, at least on the basis of syntax. Kang and Naughton’s approach [17] could pair attributes in the two schemas based, for example, on the closeness in attribute entropies.

To illustrate a difficulty with this approach, consider a different example, but for the same (Automotive) domain, shown

1. Since these statistics are used as the basis for schema matching in methods such as those proposed by Kang and Naughton [17], this scenario poses a difficult challenge for such methods.

2. For each attribute, the probability mass function is estimated based on the frequency of occurrence counts taken over the available database records, e.g., $P(X = x_1) = \frac{N(x_1)}{N}$, where $N(x_1)$ is the number of times $X = x_1$ occurs and N is the total number of database records (assuming for each record there is a measured value for attribute X).

TABLE 2

Two schemas, (a) and (b), showing similar entropies for each of their columns. The Probability Mass Functions/Frequency counts of values, however, are quite distinctive and can be used in deciding the correct column matches

(a) Schema 1

Model _A	P(Model _A)	Color _A	P(Color _A)
XE	0.01	Red	0.06
XL	0.02	Blue	0.08
XO	0.15	Green	0.17
LE	0.16	Black	0.22
LX	0.17	White	0.23
LZ	0.20	Mauve	0.24
LO	0.29		
H(Model_A)	1.6841	H(Color_A)	1.6857

(b) Schema 2

Model _B	P(Model _B)	Color _B	P(Color _B)
XE	0.02	Red	0.07
XL	0.02	Blue	0.08
XO	0.13	Green	0.12
LE	0.14	Black	0.22
LX	0.17	White	0.24
LZ	0.20	Mauve	0.27
LO	0.32		
H(Model_B)	1.6847	H(Color_B)	1.6717

in Table 2. Table 2³ shows two different schemas in (a) and (b), the values of the two attributes under each schema, and the normalized frequency count (probability of occurrence) for the values under each attribute. With Shannon’s entropy function defined as $H(X) = -\sum_{x \in \mathcal{X}} P(X = x) \log P(X = x)$, \mathcal{X} the values taken on by X , we might have $H(Model_A) = 1.6841$, $H(Color_A) = 1.6857$, $H(Model_B) = 1.6847$ and $H(Color_B) = 1.6717$ as shown in Table 2. The best alignment (in the sense of closeness in entropy) could either pair $Model_A$ with $Model_B$ and $Color_A$ with $Color_B$ or $Model_A$ with $Color_B$ and $Model_B$ with $Color_A$. In this example, the entropy differences are not large enough to make confident matching decisions. This statement may apply even if second order statistics (e.g. mutual information), based on pairs of attributes from the same schema, are used in matching, as in [17]. Further, one cannot use value cardinalities to aid matching because even if the cardinalities of $Model_A$ and $Model_B$ are the same (as in Table 2), this does not necessarily imply the attributes ground-truth match. For example even if there are nine car colors and seven different models, some car colors may not have occurred in one table (There may only be records for seven or eight different colors). In this case, one cannot disambiguate whether an attribute with seven distinct values represents the car’s model or its color (with two values missing).

To make progress in such difficult scenarios, our technique utilizes the value-mapping dimension to enhance schema matching. In particular, we will capitalize on the frequency

3. Table 1 and Table 2 illustrate two different examples, involving different schemas, even though both are from the Automotive domain. Table 1 is presented to highlight difficulties in applying syntactical similarity to match schemas. Table 2 highlights issues for schema matching algorithms that align based on statistics such as entropy.

of occurrence of an attribute’s values to enhance attribute matching. The probability mass function on the attribute values is in general much more distinctive than the attribute’s entropy (or mutual information). In particular, there are uncountably infinite number of probability mass functions that possess the same (as well as similar) entropies. It is much more likely by random chance that different attributes possess similar entropies than that they possess similar probability mass functions on their value support. As an illustration of this, in Table 2, even though all the columns have similar entropies, the attribute $Model_A$ of Schema 1 (Table 2(a)) can be best matched to attribute $Model_B$ in Schema 2 (Table 2(b)) since both these attributes have similar probability mass functions. Likewise, for the same reason, $Color_A$ is best matched to $Color_B$. Thus, matching attribute probability mass functions (pmfs) across schemas should be more reliable than solely matching attribute entropies (or mutual information). We propose a heuristic, iterative search strategy that monotonically descends in our pmf matching objective function, seeking to minimize this objective. We have found this technique to be quite effective at producing accurate schema matches as will be shown in Section 3.

Another advantage of embedding value mapping to enhance schema matching is that such an approach does not require data interpretation; i.e., even if different encodings are used between two data sources, we can still utilize the statistical characteristics of the data. We refer to schema matching techniques which are independent of data interpretation as *un-interpreted matching* techniques [17], and next provide a more formal and precise definition.

Definition *Un-Interpreted vs. Interpreted Matching*: Let $M_1 = \text{match}(\chi(X_1, X_2, \dots, X_n), \gamma(Y_1, Y_2, \dots, Y_m))$ and $M_2 = \text{match}(\chi(X_1, X_2, \dots, X_n), \gamma(f_1(Y_1), f_2(Y_2), \dots, f_m(Y_m)))$, where M_i is a match result returned by the schema matching algorithm, χ is a source schema of size n , γ is a source schema of size m , and f_i is an arbitrary one-to-one transformation function applied to the data values of attribute i in the target schema. We call the schema matching algorithm an *un-interpreted matching* algorithm if and only if the two match results M_1 and M_2 are identical, i.e. if the transformation function f_i does not alter the matching results produced by the algorithm. Conversely, if the two results are not equal the matching is referred to as an *interpreted* matching.

We focus on finding a complete one-to-one mapping where each column of a table is mapped to one and only one column of another table. Our method can easily be adapted to find a partial mapping where the number of columns in Table A is larger than in Table B, and where every column in Table B must be matched to a distinct column in Table A. Finding many-to-one mappings is beyond the scope of this work. However, mapping multiple columns (like firstName, lastName) to a single column (like Name) can be done using a pre-processing step of concatenating the two columns, firstName and lastName, and then finding a one-to-one mapping of the pre-processed table with the second table. Determining which columns to concatenate is an orthogonal problem.

To address the schema-matching problem, we propose a

global objective function based on a dissimilarity metric that provides a confidence score for a fixed schema match and set of value mappings. We minimize this global objective over the space of schema matches and value mappings, seeking to arrive at the best possible schema matching and value mappings. The primary contributions of this paper are as follows:

- We introduce a novel iterative-descent algorithm that embeds value mapping to enhance schema matching.
- Our algorithm treats both problems within a common framework that involves minimizing a common (global) objective function.
- We demonstrate the effectiveness of our approach with an experimental study, where we compare our results with that of the Kang-Naughton [17] mutual information based schema matching method that utilizes statistical dependencies between attributes and with a first-order variant of their method based on entropies of individual attributes. We also evaluate a second-order variant of their method based on conditional entropy. Our experiments show that our technique outperforms [17] and its variants in matching accuracy and thus should be a useful addition to a suite of (semi) automatic schema matching techniques.

The rest of the paper is organized as follows: Section 2 outlines our matching algorithm. Section 3 presents our experimental results. Section 4 discusses related work. Section 5 discusses assumptions and limitations of this work. Section 6 identifies and describes the scope for future work. We then conclude the paper in Section 7.

2 OUR APPROACH

In this section, we describe in detail our un-interpreted schema matching with embedded value mapping algorithms. The inputs to our algorithms are two instances of table. The algorithm execution outputs a schema match and value mappings among values in matching columns. In the pseudocode of Algorithm 1, the outer “while” loop searches over possible schema matches, and the inner loop performs *embedded* search over possible value mappings given a candidate schema match. A more detailed description of the method will be given in Section 2.4.

2.1 Preliminaries

We provide the definitions of two distance measures that we utilize in our algorithms. The first is the squared Euclidean dissimilarity measure [6] between probability mass functions defined as follows.

Definition The *Euclidean Distance* between two probability mass functions, $P = \{p_1, p_2, \dots, p_n\}$ and $Q = \{q_1, q_2, \dots, q_n\}$ is defined as:

$$D_E = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (1)$$

The second measure is the relative entropy between two pmfs defined as:

Algorithm 1 Overview of our Approach

Input: Schemas, $T1$ and $T2$
Output: Schema Match s , $tmpS$
Value Mapping v , $tmpV$, $v1$
 $sMatchScore \leftarrow \infty$ { Best Dissimilarity Score }
 $sSpace \leftarrow getSchemaMatchSearchSpace()$
while $sSpace$ is not empty **do**
 $tmpS \leftarrow getNextSchemaMatch(sSpace)$
 $vSpace \leftarrow getValueMappingSpace()$
 $vMapScore \leftarrow \infty$ { Stores dissimilarity score for fixed
schema match and value mapping }
while $vSpace$ is not empty **do**
 $tmpV \leftarrow getNextValueMapping(vSpace)$
 $score \leftarrow computeDissimilarity(tmpS, tmpV, T1, T2)$
if $score < vmapScore$ **then**
 $vmapScore \leftarrow score$ { Save current dissimilarity
score }
 $v1 \leftarrow tmpV$ { Save current value mapping }
end if
removeValueMapping($vSpace$, $tmpV$)
end while
if $vmapScore < smatchScore$ **then**
 $s \leftarrow tmpS$ { Store current schema match }
 $v \leftarrow v1$ { Store the best value mapping for this schema
match }
 $smatchScore \leftarrow vmapScore$
end if
removeSchemaMatch($sSpace$, $tmpS$)
end while

Definition The *Relative Entropy* or *Kullback-Leibler* Distance between probability mass functions P and Q is defined as:

$$D(p \parallel q) = \sum_{i=1}^n p_i \log \frac{p_i}{q_i}, \quad (2)$$

where it is assumed $q_i > 0$ whenever $p_i > 0$.

2.2 Dissimilarity Objectives

In Algorithm 1 the `computeDissimilarity()` operation produces a dissimilarity score for a fixed schema match and value mapping. The dissimilarity metric provides an evaluation of how well the probability mass functions (joint probability mass functions in our second-order methods) across attributes (attribute pairs) align for a fixed schema match and value mapping.

2.2.1 First Order Dissimilarity Metric Model

The first order dissimilarity metric model measures how well the probability mass functions of matching attributes align for a fixed schema match and value mapping across tables. To measure the dissimilarity metric for two tables for a fixed schema match and value mapping, we have considered several dissimilarity metrics that utilize first order statistics: One based on the squared Euclidean squared distance as shown in Eq. (3)

and the second based on Relative entropy as shown in Eq. (4).

$$D_{AB}^{EU} = \sum_{i=1}^{N_{attr_1}} \sum_{i'=1}^{N_{attr_2}} \delta(i - m_a(i')) \sum_{j=1}^{N_{values(i)}} \left[p_i(j) - \left\{ \sum_{j'=1}^{N_{values(i')}} \delta(j - M_v^{(i,i')}(j')) p_{i'}(j') \right\} \right]^2 \quad (3)$$

$$D_{AB}^{CE} = \sum_{i=1}^{N_{attr_1}} \sum_{i'=1}^{N_{attr_2}} \delta(i - m_a(i')) \sum_{j=1}^{N_{values(i)}} \left[p_i(j) \times \log \frac{p_i(j)}{\left[\sum_{j'=1}^{N_{values(i')}} \delta(j - M_v^{(i,i')}(j')) p_{i'}(j') \right]} \right] \quad (4)$$

Where:

D_{AB}^{EU} = Euclidean Squared Based Dissimilarity Measure for a fixed schema match and value mapping

D_{AB}^{CE} = Relative Entropy Based Dissimilarity Measure for a fixed schema match and value mapping

m_a = matched attribute function (across schemas)

$\delta(i - m_a(i')) = 1$ if i^{th} attribute in Schema 1 matches to i^{th} attribute in Schema 2, = 0 otherwise

$M_v(i, i') =$ value mapping function

$\delta(j - M_v^{(i,i')}(j')) = 1$ if j^{th} value for i^{th} attribute matches to j^{th} value for matching i^{th} attribute, = 0 otherwise

N_{attr_1} = Number of attributes in schema 1

N_{attr_2} = Number of attributes in schema 2

$N_{values(i)}$ = Number of values in i^{th} attribute

$N_{values(i')}$ = Number of values in i'^{th} attribute

In equations (3) and (4), we have assumed $N_{attr_1} \leq N_{attr_2}$, i.e. each column in schema 1 matches to a unique column in schema 2. The dissimilarity metric based on relative entropy has difficulties when the probability of a value used in matching in Eq. (4) is zero. This is problematic because $p \times \log \frac{p}{0} = \infty$. This issue arises especially when $N_{values(i)} \neq N_{values(i')}$ for attributes i and i' being matched. There are several strategies to deal with this:

- 1) Introduce extra symbols for the attribute having the smaller alphabet and assign them a small probability value ϵ followed by renormalization of the probability mass function of this reconstructed alphabet. This ensures both attributes have alphabets of equal size and $p \times \log \frac{p}{0} = \infty$ will never occur for any value⁴.
- 2) Effectively reduce the size of the alphabet of the larger attribute (deleting values with the smallest probabilities) followed by renormalization. This method achieves the result that both attributes will have alphabets of the same size, though by discarding some information from the larger alphabet.

Another issue with the relative-entropy based dissimilarity measure is its asymmetric nature. However, there are symmetric versions of relative entropy [6]. The squared Euclidean

4. Some care may be needed in the choice of the small probability value ϵ ensuring, e.g., that this probability value is much smaller than the smallest probability value of the original alphabet.

dissimilarity measure, though, is free from both these issues: it is symmetric and does not require a (somewhat arbitrary) reconstruction of attribute alphabets. If two attributes across tables have alphabets of different sizes, we can assume that, for a value x with probability p in one attribute that does not have a corresponding value for the matching attribute, the probability of the matching value is zero. In other words, in the Euclidean distance approach we essentially introduce extra probability values $\epsilon = 0$, which causes *no* difficulties in this case, unlike the relative entropy case. Since the results for a method based on relative entropy may sensitively depend on the choice for ϵ and since Euclidean distance (naturally) handles matching attributes with different cardinalities by assigning zero probabilities, we only consider the Euclidean metric in the sequel. However, our framework is such that the Euclidean metric can be replaced with another suitable metric, if future evidence shows that metric to be a better distance function for our application.

2.2.2 Second Order Dissimilarity Metric Model

We extend the first order dissimilarity metric further to provide finer-grained schema matching with embedded value mapping by utilizing *second-order* statistics. The objective of the second order dissimilarity metric model is to measure how well the pairwise probability mass functions (joint probabilities) for *all* pairs of attributes align for a fixed schema match and value mapping across tables, *i.e.* the second order objective seeks to match $\{P_{ij}[k, l] \mid i = 1, \dots, N_{attr_1}, j = 1, \dots, N_{attr_1}, k = 1, \dots, N_v(i), l = 1, \dots, N_v(j)\}$, *i.e.*, where N_{attr_1} is the number of attributes in schema 1, $N_v(i)$ is the number of values in the i^{th} attribute and $N_v(j)$ is the number of values in the j^{th} attribute. This will capture pairwise statistical dependencies between all pairs of attributes/columns. We propose the following second-order dissimilarity metric based on squared Euclidean distance as shown in Eq. (5), which is the natural second order extension of (3):

$$D_{AB}^{PEU} = \sum_{i=1}^{N_{attr_1}} \sum_{j=1}^{N_{attr_1}} \left[\sum_{\substack{i'=1 \\ i' \neq i}}^{N_{attr_2}} \delta(i - m_a(i')) \sum_{\substack{j'=1 \\ j' \neq j}}^{N_{attr_2}} \delta(j - m_a(j')) \left[\sum_{k=1}^{N_v(i)} \sum_{l=1}^{N_v(j)} \left[p_{ij}(k, l) - \left\{ \sum_{k'=1}^{N_v(i')} \sum_{l'=1}^{N_v(j')} \delta(k' - M_v^{(i,i')}(k)) \delta(l' - M_v^{(j,j')}(l)) p_{i'j'}(k', l') \right\} \right]^2 \right] \right] \quad (5)$$

Where:

D_{AB}^{PEU} = Pair-wise Squared Euclidean Based Dissimilarity Measure for a fixed schema match and value mapping

m_a = matched attribute function (across schemas)

$\delta(i - m_a(i')) = 1$ if i^{th} attribute in Schema 1 matches to i^{th} attribute in Schema 2, = 0 otherwise

$M_v(i, i')$ = value mapping function

$\delta(k' - M_v(i, i')(k)) = 1$ if k^{th} value for i^{th} attribute matches to k^{th} value for matching i^{th} attribute, = 0 otherwise

N_{attr_1} = Number of attributes in schema 1

N_{attr_2} = Number of attributes in schema 2

$N_V(i)$ = Number of values for i^{th} attribute

$N_V(i')$ = Number of values for i'^{th} attribute

$P_{ij}(k, l)$ = Joint Probability of k^{th} value for i^{th} attribute co-occurring with l^{th} value for j^{th} attribute

Here, in Eq. (5) we have assumed $N_{attr_1} \leq N_{attr_2}$, just as for the first-order dissimilarity metric. The term inside the outer bracket is the squared Euclidean distance between the pairwise probability mass function (pmf) $\{P_{ij}[k, l]\}$ from the first schema and the corresponding pairwise pmf from the second schema, based on the attributes in the second schema that match the first schema's attributes i and j . The outer double sum thus adds the Euclidean distances between pmfs over all pairs of attributes from the first schema.

2.3 Matching and Mapping Strategy

Our global objective can now be formulated as a two dimensional minimization problem as shown in Eq. (6), where the resultant minimal score would correspond to the best (declared) schema matching/alignment and the corresponding best set of value mappings.

$$D_{Overall} = \min_{x \in S} \left\{ \min_{y \in V} [D_{AB}^M] \right\} \quad (6)$$

Where:

$D_{Overall}$ = minimum dissimilarity score obtained

x = a fixed schema match

S = the search space of different schema matches

y = a fixed value mapping for a fixed schema match

V = the search space of different value mapping for a fixed schema match

D_{AB}^M = dissimilarity metric utilized ($D_{AB}^{EU} / D_{AB}^{PEU}$)

As Eq. (6) indicates, our algorithm employs minimization of the dissimilarity metric over both value mapping and schema matching dimensions. Running a naïve exhaustive search to achieve global minimization over both dimensions is computationally infeasible since the complexity for One-to-One Mapping is $O(n! \times n \times m!)$, where n is the number of attributes and m is the maximum cardinality of any of the attributes (for the first order dissimilarity metric). In the following section, we discuss heuristic methods that we have utilized to reduce the search space.

2.4 Heuristic Search Strategy

In order to make the algorithm computationally tractable, we implemented a local minimization based on Eq. 6. We describe this approach utilizing the first-order dissimilarity metric.

Eq. 6 and Eq. 3 can be further rewritten as shown in Eq. 7. The inner bracket corresponds to the value mapping for a fixed attribute pair. Our method first pre-computes the VMM (Value Mapping Minimization) elements for each attribute in schema 1 paired with every attribute in schema 2. This computation results in a VMM cost matrix where each element corresponds to the squared Euclidean distance associated with the best

(minimum distance) value mapping for a given attribute pair. Then the algorithm iterates over the schema-match search space starting from an initial schema match, with the objective of minimizing the dissimilarity score obtained based on this VMM cost matrix. New schema matches are obtained by using *two-opt switching* [7] (described in Section 2.5 and depicted in Figure 1) starting from an initial schema match. Schema match initialization is discussed in Section 3.1. The search space is limited by performing a finite number of two-opt switches. Furthermore, an exact solution for the first-order dissimilarity objective (minimization over the schema-match search space) can also be computed in $O(n^3)$ time by using the Hungarian algorithm [18], [26] on the Value Mapping Minimization (VMM) matrix. While an exact minimization via the Hungarian algorithm is possible for the first-order algorithm, it is not possible to extend this to the second-order algorithms. Thus, to give a fair comparison of all methods, we used a common (local) optimization strategy for schema matching across all the methods, i.e. the two-opt local switching approach. In addition, we have also conducted experiments minimizing our first order dissimilarity cost function using the Hungarian algorithm, as well as with 2-opt switching. In practice, we did not observe that local optimization (2-opt switching) introduces significant loss in accuracy in comparison to the Hungarian global optimization method (Section 3.4 in “Optimization Strategies”).

Based on the VMM matrix, we can re-express equation (6) as:

$$D_{Overall} = \min_{x \in S} \left\{ \sum_{i=1}^{N_{attr1}} \sum_{i'=1}^{N_{attr2}} \delta(i - m_a(i')) \left[\min_{\lambda \in V_A} VMM(i, i') \right] \right\} \quad (7)$$

Where:

$D_{OverAll}$ = minimum dissimilarity score obtained

x = a fixed schema match

S = the search space of different schema matches

λ = a fixed value mapping for an attribute pair (i.e. fixed i, i')

V_A = the search space of different value mapping for a fixed attribute pair

D_{AB}^M = dissimilarity metric utilized (currently D_{AB}^{EU})

$VMM(i, i')$ = Euclidean distance between probability mass functions associated with the best value mapping for the (i, i') matching attribute pair

2.5 Two-Opt Switching

Two-opt switching is a simple local search algorithm that we utilize in our heuristic search strategy (Section 2.4). The basic step of two-opt switching is to swap attribute matches for a pair of attributes from the two schemas, as shown in Figure 1. Algorithm 2 presents the pseudocode for the overall heuristic search strategy for attribute matching (as discussed in Section 2.4) when using two-opt switching and the first order dissimilarity metric model based on Euclidean distance.

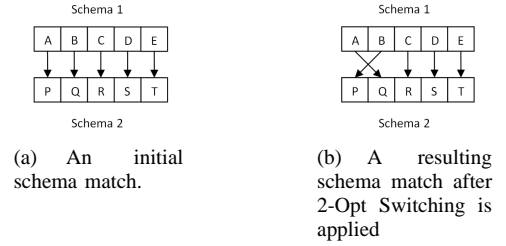


Fig. 1. 2-Opt Switching applied to two hypothetical schemas. The methods for obtaining an initial schema match are discussed in Section 3.1

Algorithm 2 Pseudocode for the heuristic search strategy when using two-opt switching.

```

best_match ← get_Initial_Schema_Match()
DOverall ← DABEU(best_match)
repeat
  new_match ← Two-Opt-Switch(best_match)
  if DABEU(best_match) > DABEU(new_match) then
    best_match ← new_match
    DOverall ← DABEU(best_match)
  end if
until no further improvement or a specified number of iterations

```

2.6 Heuristic Value Mapping Strategy

In the previous section, we only explain how we search over possible attribute matches. We have not yet explained how we determine the best value mapping for a candidate pair (i, i') (and thus how we compute the scores in the VMM matrix). We do so in the following way. Given a fixed candidate pair (i, i') we separately sort the two pmfs $\{P_i(j)\}$ and $\{P_{i'}(j)\}$ and map the value with the highest probability in the first sorted pmf to value with the highest probability in the second sorted pmf, and so on. The following theorem shows that the value mapping obtained by sorting the pmfs is in fact optimal.

Theorem 1: Let X and Y be two equal-sized sets of values with probability mass functions pmf_X and pmf_Y . Sort the values in X and Y based on their probabilities to produce ordered sets X_o and Y_o , respectively. Let the value mapping μ map the i^{th} element of X_o to the i^{th} element of Y_o . The mapping μ minimizes the squared Euclidean distance (Eq. (1)), i.e., for any value mapping ϕ between the values in X and those in Y , the Euclidean distance between the two mapped sets using ϕ is greater than or equal to the distance corresponding to μ .

Proof: Let X and Y be two discrete random variables with probability mass functions $P(X = x_j) = p_j$ and $P(Y = y_j) = q_j, j = 1, \dots, n$.

Let there be a value mapping μ_1 that maps the two sets of values so as to minimize the Euclidean distance between the probability mass functions of the values in the two sets. Furthermore, suppose that this mapping μ_1 does not map the values in the two sets according to the sorted order of their probabilities. Thus, there exist values x_1, x_2 mapped by μ_1 to y_1 and y_2 respectively, such that $p_1 > p_2$ but $q_1 < q_2$. Let D_1

be the squared Euclidean distance obtained by the mapping μ_1 .

Consider the mapping μ_2 that maps x_1, x_2 to y_2 and y_1 respectively, with the rest of the value mappings the same as μ_1 . Let D_2 be the squared Euclidean distance corresponding to the value mapping μ_2 .

We have (for some D_0)

$$D_1 = \sum_{j=1}^n (p_j - q_j)^2 \quad (8)$$

$$= D_0 + (p_1 - q_1)^2 + (p_2 - q_2)^2$$

and

$$D_2 = \sum_{j=3}^n (p_j - q_j)^2 + (p_1 - q_2)^2 + (p_2 - q_1)^2 \quad (9)$$

$$= D_0 + (p_1 - q_2)^2 + (p_2 - q_1)^2$$

Subtracting, Eq. (9) from Eq. (8),

$$D_1 - D_2 = -(2p_1q_1 + 2p_2q_2) + (2p_1q_2 + 2p_2q_1) \quad (10a)$$

$$= 2p_1(q_2 - q_1) + 2p_2(q_1 - q_2) \quad (10b)$$

$$= 2(p_1 - p_2)(q_2 - q_1) \quad (10c)$$

$$\because p_1 > p_2 \Rightarrow (p_1 - p_2) > 0 \quad (10d)$$

$$q_1 < q_2 \Rightarrow (q_2 - q_1) > 0 \quad (10e)$$

$$\therefore D_1 - D_2 > 0 \quad (10f)$$

Thus D_1 is not the value mapping which minimizes the squared Euclidean distance between the two probability mass functions (a contradiction). Therefore sorting the two probability mass function minimizes the Euclidean distance between them. \square

3 VALIDATING THE FRAMEWORK

In this section, we present the results of our first-order dissimilarity-metric-based algorithm for schema matching compared with several alternative methods. First, we chose 50 attribute pairs from the two tables, selecting 50 attributes that are close in entropy⁵. Our choice of the ‘‘close’’ attributes to evaluate a matching method based on the distributions of values in the attributes makes the schema matching problem challenging. For example, in Table 3, if we had to select three attributes to match, we would choose B, D and E. Since their entropies are quite close (in each schema) this problem will be very challenging for a method that matches just on the basis of first-order statistics (entropy) without fine-grained pmf matching. From the chosen 50 columns, we randomly select n columns ($2 < n < 30$) to match.

5. Specifically, we first calculated the entropy differences between all matching attribute pairs in the two tables. All matching attribute pairs which had an entropy difference that was less than 10% of the entropy for both individual attributes in the pair were put onto a list of possible candidate attribute pairs. All matching attribute pairs in the two tables where either or both attributes had zero entropy were rejected. 50 attribute pairs were then randomly selected from the list of possible candidate attribute pairs.

TABLE 3
Hypothetical Entropy Distribution in two schemas

	A	B	C	D	E
Schema 1	0	2.55	1.01	2.25	2.20
Schema 2	0	2.40	1.02	2.40	2.23

3.1 Experimental Setup

The schema matching with embedded value mapping algorithm is called to perform schema matching and value mapping. The matching algorithm is initialized from one of the following four starting initial schema matches: (1) Random Initialization: a random schema match is chosen, (2) Kang-Naughton Initialization: the schema match result of the heuristic Kang-Naughton implementation [17], and (3) CE-modified Kang-Naughton Initialization: the schema match result from the heuristic Kang-Naughton implementation where we replaced mutual information with conditional entropy, given by $H(Y|X) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(y|x)$ or (4) Entropy-only initialization: the schema match result from the variant of the Kang-Naughton technique that matches attributes solely on the basis of the first-order attribute entropy statistics. We compared our schema matching results against three methods that, respectively, do two-opt switching of attributes to minimize squared Euclidean distance i) between mutual informations, for matching attribute pairs (The objective function in [17]) (KNMI), ii) between conditional entropies, for matching attribute pairs, (KNCE) and iii) between entropies, for a pair of matching attributes (KNE). All three of these methods are consistent with the Kang-Naughton approach. The first method explicitly minimizes the objective proposed by Kang and Naughton.

We implemented our own version of the Kang-Naughton algorithm for un-interpreted schema matching. Kang and Naughton used a naïve exhaustive search algorithm to find the best schema match. We modified the Kang-Naughton method by using the undirected Eigen-decomposition graph matching algorithm [28], which reduces two mutual information matrices into a cost matrix. The Hungarian Algorithm [18], [26] was then run over this cost matrix, resulting in an *initial* schema match. We then ran two-opt switching (hill climbing) starting from this initial schema match while minimizing the Mutual Information (entropy, conditional entropy) metric. This resulted in the optimized schema match obtained based on the criterion from the work by Kang and Naughton [17].

We implemented our schema matching with embedded value mapping algorithm using GCC 3.4.6. Since all the algorithms are computationally expensive, they were executed on a 192 processor (AMD[®] Opteron[™]250 running at 2.4 GHz) cluster. Up to 30% of the cluster CPU resources were allocated by the system to execute the algorithms.

We ran our experiments over a randomly chosen set of up to 30 attributes as described in Section 3.1.

We also randomly permuted the columns in the second schema to ensure that the algorithms cannot trivially stumble upon the correct schema matching result due to fortuitous initialization.

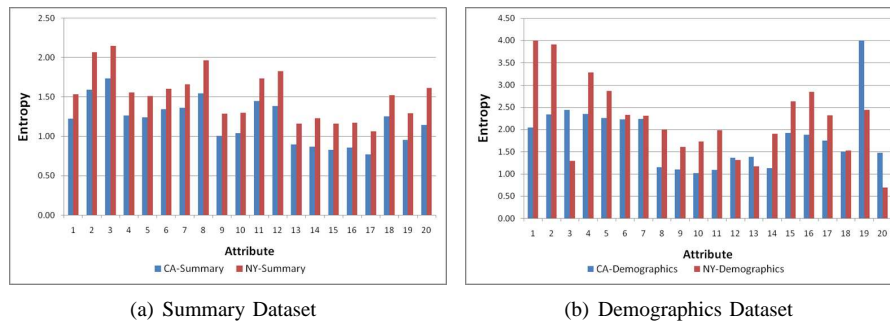
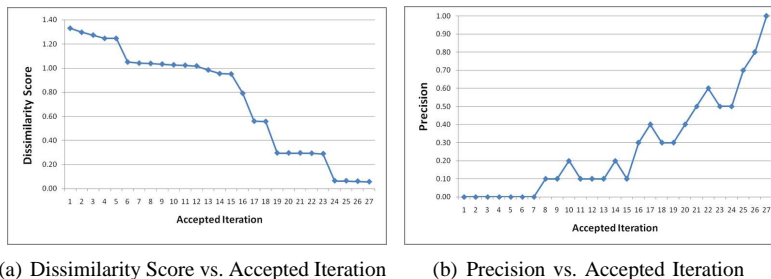


Fig. 2. Attribute Entropies from US Census Summary and Demographics files for the states of California and New York

TABLE 4
First five columns from Census Datasets

(a) CA Summary Dataset					(b) CA Demographics Dataset				
C1	C2	C3	C4	C5	C1	C2	C3	C4	C5
uSF1	CA	0	12	1	u108_H	CA	0	37	1
uSF1	CA	0	12	2	u108_H	CA	0	37	2
uSF1	CA	0	12	3	u108_H	CA	0	37	3
uSF1	CA	0	12	4	u108_H	CA	0	37	4
uSF1	CA	0	12	5	u108_H	CA	0	37	5
uSF1	CA	0	12	6	u108_H	CA	0	37	6
uSF1	CA	0	12	7	u108_H	CA	0	37	7
uSF1	CA	0	12	8	u108_H	CA	0	37	8
uSF1	CA	0	12	9	u108_H	CA	0	37	9
uSF1	CA	0	12	10	u108_H	CA	0	37	10



(a) Dissimilarity Score vs. Accepted Iteration (b) Precision vs. Accepted Iteration

Fig. 3. First Order Dissimilarity Metric (Euclidean distance) and Precision as a Function of Accepted Two-opt Switches for One-to-One Matching of 10 attributes, starting from Random Initialization

3.2 Evaluation Metrics

To measure the accuracy of the match result, we use $Precision = \frac{m}{n}$, where n is the total number of columns to match and m is the total number of correct matches that are returned by the schema matching algorithm.

3.3 Monotonic Nature of the Algorithm

Figure 3(a) shows the evolution of the dissimilarity score as a function of *accepted iterations* when 10 attributes were matched between the two schemas. An accepted iteration is a schema match obtained by accepting a two-opt switch, i.e., a switch which reduced the objective function. This figure shows that the dissimilarity score (Eq. (3)) decreases monotonically with an increase in the number of algorithm steps. As (3) decreases, there is also a general trend of increasing precision (though not strictly monotonic).

3.4 One-to-One Schema Matching

Random Initialization

Figure 4 presents the results of one-to-one matching where the initial schema match was randomly selected. We increased the number of attributes in the two input tables from 2 to 30. For each pair of schemas, we randomly chose 20K tuples. For each of these sub-datasets, we repeated the experiment 50 times, each time randomly choosing attributes from the set of 50 columns. We measured the average precision over the 50 experiments. For comparison, the Entropy only (KNE), Kang-Naughton Mutual Information (KNMI) and modified Kang-Naughton algorithm (labeled KNCE) results are also presented. All methods started from the same random initial schema match.

Figure 4(a) shows the precision of the schema matching obtained for the Census summary data set. Clearly, our first-

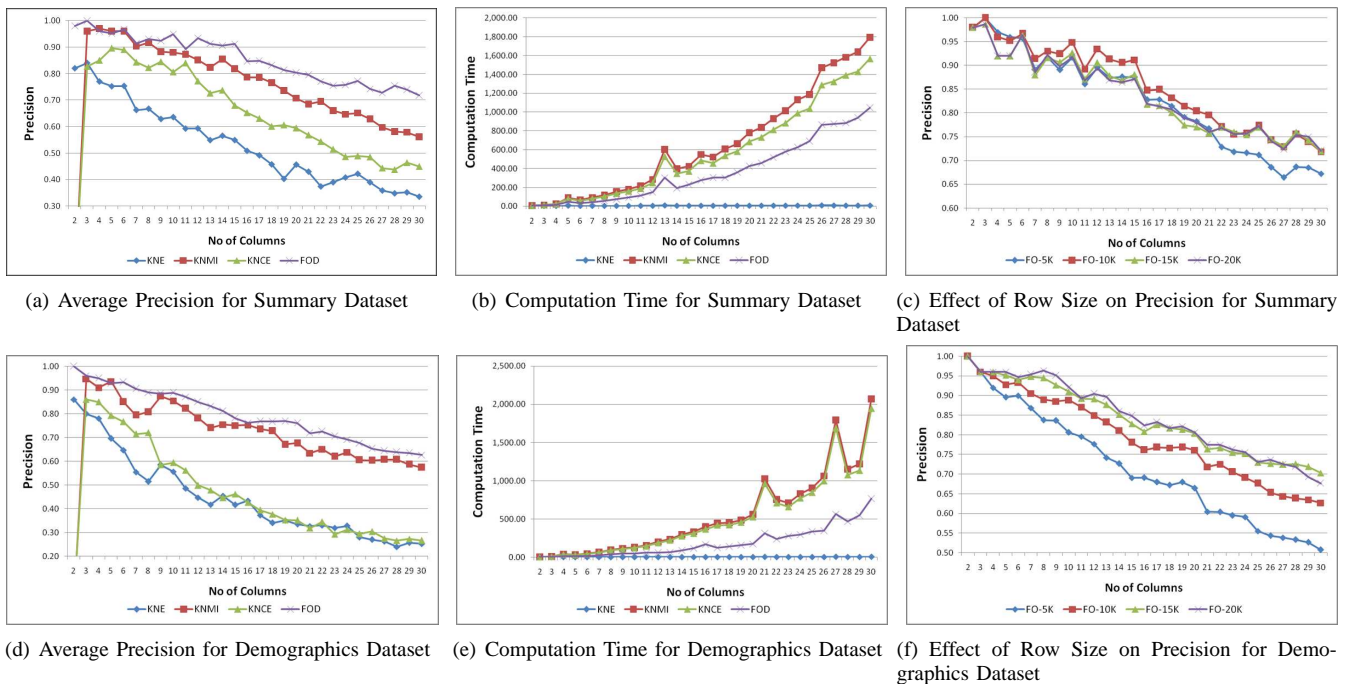


Fig. 4. First Order Dissimilarity Metric Algorithm Results for Random Initializations

order dissimilarity algorithm performs better than the mutual information, conditional entropy and entropy based schema matching techniques. It can be observed from Figure 4(a) that, as one would expect, overall match results for all the methods are better for narrow tables (fewer attributes to match) in comparison to wider tables, i.e., more attributes increases the problem difficulty and degrades matching accuracy for all methods. Comparing the four matching techniques, we see that entropy-only schema matching results (KNE) deteriorates extremely fast in comparison to the first-order dissimilarity metric algorithm as the number of attributes increases. Though both algorithms are only using first-order pmf statistics, we see that embedded value mapping allows the first-order dissimilarity algorithm to achieve much better match results. The mutual information [17] and conditional entropy techniques are also not as accurate as our first-order dissimilarity algorithm even though these methods utilize second-order statistics. This demonstrates the benefit of embedded matching of values, in determining each attribute match.

Figure 4(d) shows the match results obtained when using the Census Demographics data sets. Precision results for the summary data sets are better overall in comparison to that of demographics datasets. This may possibly be due to the fact that some attribute statistics have large differences across the two tables, as shown in Figure 2, which degrades the performance of all four methods. The first-order dissimilarity algorithm clearly outperforms the mutual information, conditional entropy and entropy-only techniques. Interestingly conditional entropy performs much worse in this case in comparison to its performance on the summary datasets. Moreover, conditional entropy performance for both datasets was poor in comparison to our first-order dissimilarity algorithm. In both datasets the clear best performer was the first-order-

dissimilarity-metric-based algorithm. For the summary dataset, the first-order dissimilarity algorithm produced 95% accuracy when two 10-column datasets were matched and 85% accuracy when two 20-column datasets were matched. This corresponds to 17 columns, on average, being returned correctly for the two 20-column datasets being matched. Comparatively, the mutual information method produced approximately 88% and 78% accuracy when 10 and 20 attributes were matched respectively, conditional entropy produced accuracies of 80% and 59% respectively while the entropy-only method produced accuracy of 63% and 45%, respectively. The average gain of the first order dissimilarity algorithm over the mutual information technique was approximately 10% when less than 15 columns were being matched and increased to approximately 12% when 15 columns or greater were being matched. Similarly for the demographics datasets, the match accuracy for the first-order dissimilarity metric was 89% and 76% for 10 column and 20 column schema matching. Comparatively, the mutual information technique produced approximately 84% and 67% accuracy when 10 and 20 columns were matched respectively, conditional entropy produced match accuracies of 59% and 35% respectively while entropy-only produced match accuracies of 55% and 33% respectively.

Computational Time

The first-order dissimilarity algorithm is computationally less expensive than both the mutual information and conditional entropy based schema matching algorithms, which can be attributed to the fact that these methods require repeated calculations based on second-order (higher-order) statistics. Figure 4(b) and 4(e) illustrates the computational time as a function of the number of attributes being matched. The entropy-only technique takes the least computation time but

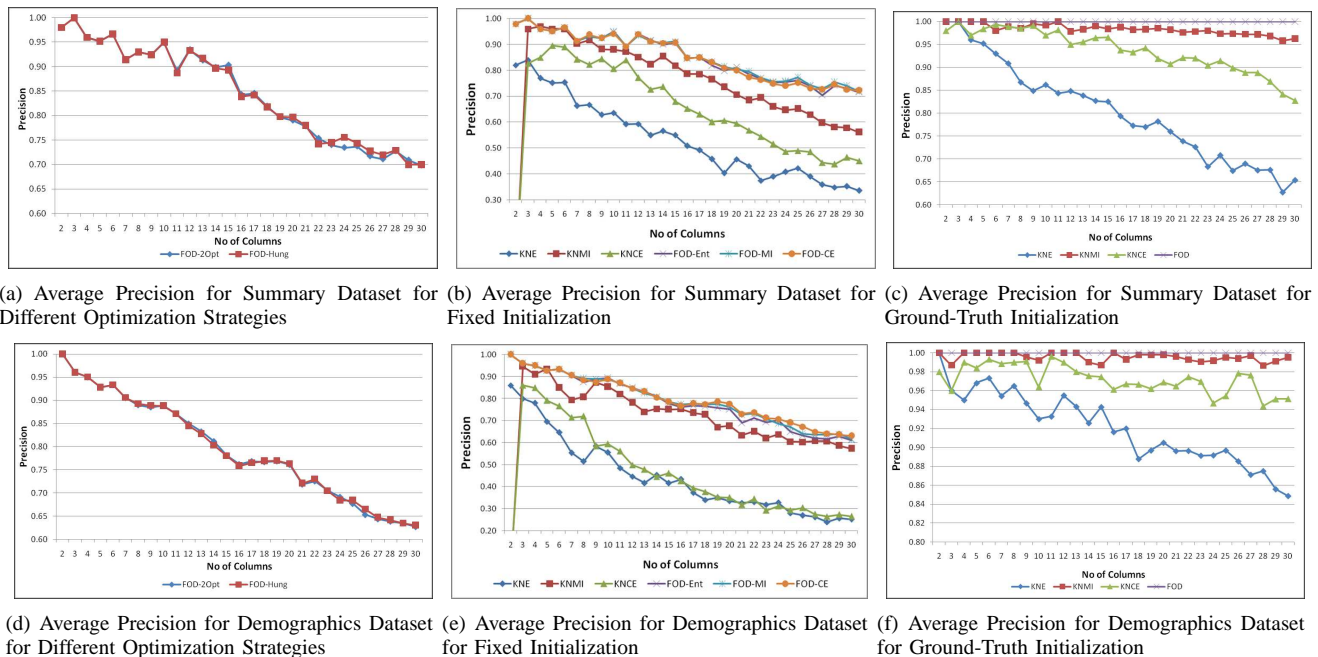


Fig. 5. First Order Dissimilarity Metric Algorithm Results for Different Optimization Strategies, Fixed Initialization and Ground-Truth Initialization

its corresponding performance is extremely poor. The first-order-dissimilarity-metric-based algorithm took approximately 424 seconds to execute a one-to-one schema matching for 20 attributes while the heuristic mutual-information-based algorithm took approximately 780 seconds to execute in order to get the respective matching accuracies achieved by the two methods for the summary dataset. In other words, for the same or *less* computation our method achieves better matching accuracies. Moreover, later we will show that increasing the allowed execution time for KNMI does not necessarily lead to improvement in its matching accuracy.

Data Sampling Effects

Figure 4(c) and 4(f) illustrate the effect of row sizes on match accuracies produced during schema matching by the first-order dissimilarity algorithm for the demographics and summary datasets. The match accuracies for 5K tuples (labelled FO-5K), 10K tuples (labelled FO-10K), 15K tuples (labelled FO-20K) and 20K tuples (labelled FO-20K) are shown. Interestingly, for the summary dataset the match accuracies were highest when 10K tuples were used for schema matching. This may be the result of a skewed sample in the 20K case. For the demographics dataset, the match accuracy of our method improved with the number of K-tuples (rows) used, as expected.

Optimization Strategies

Figure 5(a) and 5(d) illustrate the effect on the precision results of using a global minimization (Hungarian algorithm [18], [26], labelled FOD-Hung) or local minimization (2-Opt switching, labelled FOD-2Opt) strategy to minimize the first-order dissimilarity. We chose 10K tuples and considered both the demographics and summary datasets. The first-order

algorithm produced similar match accuracies irrespective of the minimization strategy employed.

Fixed Initialization

Figure 5(b) and 5(e) presents the results of our method when the initial schema match for the dissimilarity algorithm was:

- 1) Entropy-only Initialization (labelled FO-Ent)
- 2) Mutual Information Initialization (labelled FO-MI)
- 3) Conditional Entropy Initialization (labelled FO-CE)

Two-opt switching was then performed to minimize (7). We ran the experiments while increasing the number of attributes in the two input tables. The selection of attributes and tuples and the number of experiment repetitions were exactly as discussed in Section 3.4 under “Random Initialization”. For comparison, the Entropy-only (labelled KNE), Kang-Naughton Mutual-Information (labelled KNMI) and modified Kang-Naughton algorithm (labelled KNCE) results are also presented.

Figure 5(b) and 5(e) show the effects of different initializations on the final match accuracies for the summary and demographics datasets respectively. For both datasets, different initializations had virtually no effect on the final matching accuracy of the first order dissimilarity algorithm. For the demographics dataset, as shown in Figure 5(e), there was a 2-3% change in matching accuracies due to different initializations only when schema matching for a large number of attributes was performed (> 15 attributes) while for the summary dataset (Figure 5(b)) there was no effect. This clearly shows that the first-order dissimilarity algorithm is robust and largely unaffected by the initial schema match.

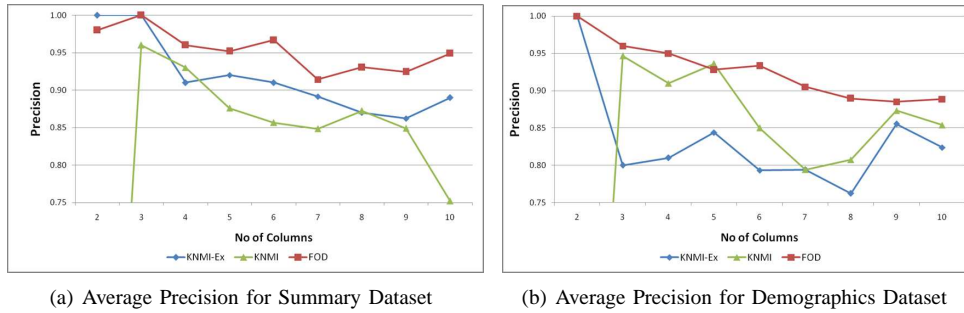


Fig. 6. First Order Dissimilarity Metric Algorithm Results vs. Kang-Naughton Exhaustive Search Based Algorithm Results for Random Initialization

Ground-Truth Initialization

We next demonstrate that the objective function of the first order dissimilarity algorithm captures the schema matching objective better than the other schema matching criteria (Mutual-Information, Conditional-Entropy and Entropy-only). To demonstrate this, we initialized all methods at the *ground-truth* schema matching (i.e., at the correct solution) and then assessed whether minimizing the matching objective function stays at this (correct) solution or deviates from it (i.e., is the ground-truth matching a local/ global minimum of the objective function?). Figure 3.4 presents the results of one-to-one mapping where the initial schema match was the correct schema match (ground-truth initialization). We increased the number of attributes in the two input tables from 2 up to 30. The attributes themselves were randomly chosen from a set of 50 columns close in entropy as described in Section 3.1. The two input datasets were generated by randomly selecting 10K tuples from one dataset to generate two sub-datasets. For each of these sub-datasets, we repeated the experiment 50 times while randomly choosing attributes from the set of 50 columns. We measured the average precision over the 50 experiments for both the CA Summary (Figure 5(c)) and CA Demographics datasets (Figure 5(f)).

For both datasets, the first-order-dissimilarity-metric based algorithm converged to the correct solution *in all 50 experimental trials*. The entropy-only algorithm performed the worst in both cases. Kang-Naughton Mutual Information algorithm, while producing high precision results, did not always converge to the correct schema match unlike the first-order dissimilarity metric algorithm. This illustrates that minimizing the objective function of the first-order dissimilarity algorithm is much more consistent with the matching goal than the objective functions of the other methods.

Dissimilarity Algorithm vs. Exhaustive Search on Mutual Information Objective

The Kang-Naughton [17] technique used a naïve exhaustive search algorithm to minimize their (MI-based) objective function. To determine whether improved results could be obtained for the Kang-Naughton method if more searching were allowed, we also implemented a naïve exhaustive search based implementation (labelled KNMI-Ex) for their method. For a comparison, the heuristic Kang-Naughton results (labelled

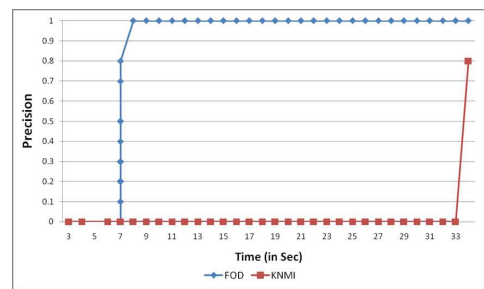


Fig. 7. Algorithm Precision as a function of Computation Time

KNMI) are also presented, along with results for our method (labelled FOD).

Due to computational time constraints for executing a naïve exhaustive search, we ran the experiments for the two datasets for attribute sizes varied from 2 up to 10. For these experiments KNMI-Ex, in performing its *exhaustive* search, took on average 3 times as much execution time as our FOD method. Figure 6(a) and Figure 6(b) clearly demonstrate that our dissimilarity algorithm had the best overall schema matching performance for both datasets, despite using much less execution time. Further, we expected the exhaustive Kang-Naughton implementation to have better matching accuracies than the heuristic implementation. However, Figure 6(b) illustrates that the heuristic Kang-Naughton implementation had better matching accuracies than the naïve exhaustive Kang-Naughton implementation for the demographics dataset. Thus, seeking the global minimum of the Kang-Naughton mutual information [17] objective is not necessarily consistent with the true (precision) objective.

Precision vs. Computational Time

Figure 7 illustrates schema matching accuracy as a function of execution time for the first-order dissimilarity algorithm performing 10-attribute schema matching on the Summary dataset. For comparison, the Kang-Naughton heuristic implementation (labelled KNMI) match accuracy as a function of time is also presented. Figure 7 shows that the initialization phase (time to compute first-order/second-order probability statistics) of both algorithms is the dominant portion of their execution time. For our first-order dissimilarity algorithm the

initialization phase was 7.9 seconds. Similarly, for the Kang-Naughton heuristic algorithm the initialization time was 33 seconds. Following initialization, both algorithms converged to their best solution for 10 attribute schema-matching in <1 second. Further, the first-order dissimilarity algorithm reached its final solution (in this case, in 8 seconds) well before the Kang-Naughton algorithm completed its initialization phase (in this case, in 33 seconds), during all our experiments.

3.5 Value Mapping

While our method utilizes value mapping specifically to enhance schema matching, we also wanted to give some preliminary assessment of the accuracy of the value mappings themselves, produced by our first order dissimilarity algorithm. We used the Census state summary dataset for New York, with 250 attributes and $> 80K$ tuples. We chose this dataset because it contained some attributes (10) with value cardinality ≤ 10 . Only these attributes, with relatively small cardinality, were used during the schema matching. Two schema datasets were created by randomly sampling rows from the original summary dataset. First 40K tuples were randomly selected without replacement to create dataset 1. Next, the remaining 40K tuples were used to form dataset 2. Using the same dataset to create two test datasets ensured that we had ground-truth on the value mappings across attributes. From the 10 attributes with cardinality ≤ 10 values, we repeated the following experiment 50 times. We randomly selected 5 attributes and randomly selected 40K rows to form dataset 1. We then measured the average schema matching accuracy and the average value mapping accuracy, considering only attribute pairs that were matched. The average schema matching accuracy was 85.6% (Kang-Naughton achieved 73.6%) and the average value mapping accuracy for our method was 56%. In future work, we will more extensively evaluate value mapping accuracy of our method.

4 RELATED WORK

Previous work has tackled *either* (1) *Schema matching* [2], [3], [4], [8], [9], [10], [13], [17], [19], [20], [21], [23], [24], [25], [27], [30] or (2) *Value/ Object mapping* [1], [5], [11], [12], [14], [15], [16], [22], [29]. Most of these techniques except Kang and Naughton [17] rely on identifying similarity in semantics of schema element names, data encoding formats or common data domains. The novelty of our work lies in the fact that we provide a framework to perform both schema matching and value mapping at the same time to produce improved schema matching. Our work can be used to complement traditional schema matching or value mapping systems.

Schema Matching

A wide variety of techniques have been proposed to resolve structural heterogeneity. For good surveys and comparisons of schema matching methods, see [9], [27]. Kang and Naughton [17] proposed an un-interpreted schema matching technique that employs mutual information to construct dependency graphs followed by graph matching. However, this method does not integrate value mapping. As demonstrated in

this work, embedding value mapping within schema matching significantly improves matching accuracy. Cruz et al. [8] have adapted the Kang and Naughton [17] technique for schema matching applications while preserving privacy.

Other techniques have been proposed that employ machine learning (e.g. LSD [10]) and Neural Networks (e.g. SemInt [19]). These systems however, rely on data interpretation for learning and therefore are not suitable for our problem. Other work includes techniques based on structural similarity (e.g. Cupid [21], Similarity Flooding [23]), which translate schemas into graphs and perform graph matching based on the structural similarity between the two graphs. These methods however require schema-based structural similarity information and therefore, cannot be applied to the domain of un-interpreted schema matching. Other techniques use a corpus of schemas [20] or documents [13] during schema matching.

Rule based schema matching methods have also been proposed (e.g. TranScm [25] and ARTEMIS [4]). Both these techniques could be used in conjunction with our algorithms to improve schema matching accuracy. Bernstein, et al., [2] discuss how to build a customizable schema matcher using multiple schema matching techniques/algorithms. Our algorithms can be a useful addition to this suite.

Value Mapping

Most previously proposed works on value/ object mapping rely on finding syntactic similarities or semantic interpretation of the values. These methods typically assume that the correspondence between columns across tables is known. Kang et al., [15], [16], have proposed (semi) automatic statistical techniques that can identify mappings when two objects have few syntactic similarities.

Other work on the object-mapping problem, known by various names in diverse contexts, includes; e.g.: record linkage [11], [29], citation matching [22], merge-purge [14], duplicate detection [1] and approximate string join [5], [12]. These techniques attempt to find similar objects (e.g. records, tuples, citations, values) by focusing on syntactic similarities between objects under comparison. Such methods cannot be easily applied to our domain.

5 ASSUMPTIONS AND LIMITATIONS OF THIS WORK

Our assumption that matching columns should have similar probability distributions is a reasonable one in many cases. First, assuming some form of statistical agreement between columns to be matched is not unique to our work – the earlier work of Kang-Naughton made a statistical matching assumption, albeit using less fine-grained statistics than the ones we propose to match. Second, suppose two columns to be matched are truly based on randomly sampling from the same probability mass function. Then, it is well-known that as the number of data samples (the number of rows) increases, frequency count estimates of the probability mass function converge to the true probability mass function in the large sample limit. Thus, the probability mass function estimate for each of the columns being matched should converge to

the same probability mass function, the true probability mass function, if we have enough data samples in each schema. So, indeed we make two assumptions: 1) that it is reasonable to assume matching columns are realizations based on the same distribution and 2) that the number of samples in each schema is large enough such that for each schema we get accurate estimates of the probability mass functions for each column. Finally, the effectiveness of our method, based on these assumptions, was borne out through our experimental results.

Now, there are also cases where our above assumption will fail. We elaborate on a few here. First, even if two columns to be matched do in principle correspond to the same random variable, there may be some hidden factor (not observed) that differs for the two schemas and that will cause the two columns that ground-truth are indeed a matching pair to differ in their probability mass functions. For example, for two schemas that deal with census data, the probability mass function on ethnicity may be highly region-dependent, and also will strongly depend on whether the census was taken for an urban, suburban, or rural region. If this region information is available, one could condition on it, and then the two (conditional) probability mass functions, conditioning on the same value for the hidden factor, should still reasonably agree. However, if this conditioning information is unknown, then indeed our assumptions will be violated. A second case is simply one for which there is simply no strong statistical agreement between the ground-truth matching columns – the basis for “ground-truth” agreement is either semantic or syntactic, but without strong statistical agreement. As one example, two databases may both have columns one could label as “Car owned”. However, one database might be for individuals in the U.S., while the other could be for individuals in Australia we might well expect car ownership patterns (and models owned) would differ greatly between the U.S. And Australia. In this case, no strong statistical agreement between ground-truth matching columns is to be expected.

6 FUTURE WORK

In this work, we have only experimentally validated our first-order method (Section 2.2). Our second-order method, which exploits statistical dependencies between columns, should give even better matching accuracies *if* the number of database examples allows accurate frequency-count estimation of the second-order pmfs. However, the second-order methods may take substantially more time and should only give improved results when the number of database examples (rows) is sufficiently large. Nevertheless, this will be investigated in future work. We will also consider the cases i) where the number of columns in the schemas differ; ii) where only a *subset* of columns in schema A have a ground-truth matching pair in schema B; and iii) where there are missing attribute values, which may necessitate use of imputation techniques in the second-order case. Finally, in future work, we will thoroughly evaluate the accuracy of the value mappings selected by our method.

7 CONCLUSIONS

We have proposed a fine-grained un-interpreted schema matching criterion, based on matching probability mass functions between attributes, which requires *embedded* learning of a value mapping between the attributes. We proposed an iterative descent algorithm for our matching objective and demonstrated that this technique achieves greater attribute matching accuracies than previous methods and at less computational cost. Our approach has been validated with several experiments on two real-world datasets.

8 ACKNOWLEDGEMENTS

This work was performed with support from the National Visualization and Analytics Center (NVAC), a U.S. Department of Homeland Security Program, under the auspices of the Northeast Regional Visualization and Analytics Center (NEVAC). NVAC is operated by the Pacific Northwest National Laboratory (PNNL), a U.S. Department of Energy Office of Science laboratory.

REFERENCES

- [1] R. Ananthakrishna, S. Chaudhuri, and V. Ganti. Eliminating fuzzy duplicates in data warehouses. In *VLDB*, pages 586–597, 2002.
- [2] P. A. Bernstein, S. Melnik, M. Petropoulos, and C. Quix. Industrial-strength schema matching. *SIGMOD Rec.*, 33(4):38–43, 2004.
- [3] M. A. Casanova, K. K. Breitman, D. F. Brauner, and A. L. Marins. Database conceptual schema matching. *Computer*, 40(10):102–104, 2007.
- [4] S. Castano, V. D. Antonellis, and S. D. C. di Vimercati. Global viewing of heterogeneous data sources. *IEEE Transactions on Knowledge and Data Engineering*, 13(2):277–297, 2001.
- [5] W. W. Cohen. Integration of heterogeneous databases without common domains using queries based on textual similarity. In *ACM SIGMOD*, pages 201–212, New York, NY, USA, 1998.
- [6] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.
- [7] G. Croes. A method for solving traveling salesman problems. *Operations Research*, 6(6):791–812, 1958.
- [8] I. Cruz, R. Tamassia, and D. Yao. Privacy-preserving schema matching using mutual information. *Lecture Notes in Computer Science*, 4602:93, 2007.
- [9] H. H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *Revised Papers from the NODe 2002 Web and Database-Related Workshops on Web, Web-Services, and Database Systems*, pages 221–237, London, UK, 2003. Springer-Verlag.
- [10] A. Doan, P. Domingos, and A. Y. Halevy. Reconciling schemas of disparate data sources: a machine-learning approach. In *ACM SIGMOD*, pages 509–520, New York, NY, USA, 2001.
- [11] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.
- [12] L. Gravano, P. G. Ipeirotis, N. Koudas, and D. Srivastava. Text joins for data cleansing and integration in an rdbms. *ICDE*, 2003.
- [13] B. He and K. C.-C. Chang. Statistical schema matching across web query interfaces. In *SIGMOD*, pages 217–228, New York, NY, USA, 2003. ACM.
- [14] M. A. Hernández and S. J. Stolfo. The merge/purge problem for large databases. In *ACM SIGMOD*, pages 127–138, New York, NY, USA, 1995.
- [15] J. Kang, T. S. Han, D. Lee, and P. Mitra. Establishing value mappings using statistical models and user feedback. In *ACM CIKM*, pages 68–75, New York, NY, USA, 2005.
- [16] J. Kang, D. Lee, and P. Mitra. Identifying value mappings for data integration: An unsupervised approach. In *WISE*, 2005.
- [17] J. Kang and J. F. Naughton. On schema matching with opaque column names and data values. In *SIGMOD*, pages 205–216, New York, NY, USA, 2003.
- [18] H. Kuhn. The hungarian method for solving the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

- [19] W.-S. Li and C. Clifton. Semint: a system prototype for semantic integration in heterogeneous databases. In *ACM SIGMOD*, page 484, New York, NY, USA, 1995.
- [20] J. Madhavan, P. A. Bernstein, A. Doan, and A. Halevy. Corpus-based schema matching. In *ICDE*, pages 57–68, Washington, DC, USA, 2005. IEEE Computer Society.
- [21] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *VLDB*, pages 49–58, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [22] A. McCallum, K. Nigam, and L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *ACM SIGKDD*, pages 169–178, New York, NY, USA, 2000.
- [23] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *IEEE ICDE*, page 117, Washington, DC, USA, 2002.
- [24] R. J. Miller, L. M. Haas, and M. A. Hernández. Schema mapping as query discovery. In *VLDB*, pages 77–88, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [25] T. Milo and S. Zohar. Using schema matching to simplify heterogeneous data translation. In *VLDB*, pages 122–133, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
- [26] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.
- [27] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.
- [28] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988.
- [29] W. Winkler. The state of record linkage and current research problems. Technical report, US Bureau of the Census, 1999.
- [30] L. Xu and D. Embley. Using schema mapping to facilitate data integration. 2003.



Prasenjit Mitra received his Ph.D. in Electrical Engineering from Stanford University in 2004, M.S. in Computer Science from The University of Texas at Austin in 1994, and B.Tech.(Hons.) in Computer Science and Engineering from the Indian Institute of Technology, Kharagpur, India, in 1993. Since 2003, he has been an Assistant Professor of Information Sciences and Technology and Computer Science and Engineering at The Pennsylvania State University, University Park campus. From 1995 to 2000, he has served

as a senior member of the technical staff at Oracle Corporation. Dr. Mitra has also served as a senior software engineer at Narus Incorporated, and at DBWizards, Inc. He currently serves on the board of advisors of Global IDs.

His research interests are in information extraction, information integration, database systems, data mining, semantic web, and visual analytics.



Anuj Jaiswal received his M.S. degree in Electrical Engineering from The Pennsylvania State University, University Park in 2006. He is currently working towards the Ph.D. degree in Information Sciences and Technology at The Pennsylvania State University, University Park.

His research interests include machine learning, database alignment, data mining, search engines and geographic information systems.



David J. Miller (M'94) received the B.S.E. degree from Princeton University, Princeton, NJ, in 1987, the M.S.E. degree from the University of Pennsylvania, Philadelphia, PA, in 1990, and the Ph.D. degree from the University of California, Santa Barbara, in 1995, all in electrical engineering.

From January 1988 through January 1990, he was with General Atronics Corp., Wyndmoor, PA. From Sept. 1995 - July 2001 he was Assistant Professor of electrical engineering at The Pennsylvania State University, University Park campus. He is now tenured Full Professor of electrical engineering at The Pennsylvania State University. His research interests include machine learning, source coding and coding over noisy channels, image coding and processing, and bioinformatics.

Dr. Miller received the National Science Foundation CAREER Award in 1996. He was General Chair for the 2001 IEEE Workshop on Neural Networks for Signal Processing. Dr. Miller was Associate Editor for IEEE Transactions on Signal Processing from 2004-2007. He was also Chair of the Machine Learning for Signal Processing Technical Committee, within the IEEE Signal Processing Society, from 2007-2009.