ON STATISTICAL SCHEMA MATCHING WITH EMBEDDED VALUE MAPPINGS

A Dissertation in
Information Sciences and Technology
by
Anuj Jaiswal

© 2012 Anuj Jaiswal

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2012
The dissertation of Anuj Jaiswal was reviewed and approved* by the following:

Prasenjit Mitra
Associate Professor of Information Sciences and Technology
Dissertation Co-Advisor, Co-Chair of Committee

David J. Miller
Professor of Electrical Engineering
Dissertation Co-Advisor, Co-Chair of Committee

James Z. Wang
Professor of Information Sciences and Technology

Wang-Chien Lee
Associate Professor of Computer Science and Engineering

Madhu Reddy
Associate Professor of Information Sciences and Technology
Graduate Program Chair

*Signatures are on file in the Graduate School.
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. The complexity of the problem grows quickly with the number of attributes to be matched and also due to multiple semantics of data values used in the real-world. Traditional research has tackled schema matching and value mapping independently, and, mainly, for categorical (discrete-valued) attributes. In this thesis, novel methods that leverage value mappings to enhance schema matching in the presence of opaque column names for schemas consisting of both continuous and discrete-valued attributes are discussed. Additional sources of confounding are that a) a discrete-valued attribute in one schema could in fact be a quantized version of a continuous-valued attribute in the other schema, and b) a continuous-valued attribute in one schema could in fact be a transformed version of a continuous-valued attribute in the other schema. In this approach, the fitness objective for matching a pair of attributes from two schemas exploits the statistical distribution over values within the two attributes. Suitable fitness objectives are based on Euclidean-distance and data log-likelihood, both of which are applied in experimental evaluations. A heuristic local descent optimization strategy that uses two-opt switching to optimize attribute matches, while simultaneously embedding value mappings, is applied for one-to-one or onto matching. A top-K schema matching strategy is developed where a top-K set of schema matches is first generated and statistical hypothesis testing is then applied to identify confident matches. This strategy is further utilized to handle partial schema matchings. Experimental results show that the proposed techniques achieve mixed continuous and discrete-valued schema matching with high accuracy and, thus, should be useful additions to a framework of (semi) automated tools for data alignment.
# Table of Contents

**List of Figures** | ix  
**List of Tables** | xii  
**List of Symbols** | xiii  
**Acknowledgments** | xiv  

### Chapter 1

**Introduction** | 1  
1.1 Uninterpreted Schema Matching for Discrete-valued Attributes | 2  
1.2 Proposed Representation for Continuous-valued Attributes | 6  
1.2.1 Matching two continuous-valued attributes | 8  
1.2.2 Matching a continuous-valued attribute and a discrete attribute | 9  
1.2.3 Matching two continuous-valued attributes with affine pdf transformations | 10  
1.3 Schema matching strategies and proposed solutions | 11  
1.3.1 Onto matching | 11  
1.3.2 Top-K matching | 12  
1.4 Research contributions | 13  
1.5 Organization of this dissertation | 15  

### Chapter 2

**Related Work** | 16  
2.1 Schema matching | 16  
2.2 Value mapping | 19
Chapter 3

Schema Matching Criterion for Categorical-valued Attributes

3.1 Introduction ........................................... 20
3.2 Schema matching framework .......................... 20
   3.2.1 Dissimilarity objectives for categorical-valued attributes ... 21
      3.2.1.1 First-order (FO) Categorical Dissimilarity Metric Model ... 22
   3.2.2 Second-order (SO) Categorical Dissimilarity Metric ... 24
   3.2.3 Matching and mapping strategy ................... 25
   3.2.4 Two-opt switching ................................ 27
   3.2.5 Heuristic value mapping strategy ................. 28
3.3 Experimental validation .............................. 30
   3.3.1 Categorical-valued dataset ....................... 30
   3.3.2 Experimental setup ................................ 31
   3.3.3 Evaluation metrics ................................ 31
   3.3.4 Monotonic nature of the algorithm ............... 31
   3.3.5 One-to-one schema matching ..................... 32
      3.3.5.1 Random initialization ....................... 32
      3.3.5.2 Computational time ......................... 34
      3.3.5.3 Data sampling effects ...................... 34
      3.3.5.4 Ground-truth initialization ................. 35

Chapter 4

Schema Matching Criterion for Continuous-valued Attributes

4.1 Introduction ........................................... 37
4.2 The Schema Matching Framework .................... 38
   4.2.1 Dissimilarity objectives for continuous-valued attributes ... 38
      4.2.1.1 First-order (FO) Continuous Dissimilarity Metrics ....... 38
      4.2.1.2 Second-order continuous dissimilarity metrics ........... 40
   4.2.2 Schema matching strategy ....................... 43
4.3 Experimental validation .............................. 44
   4.3.1 Continuous-valued attributes dataset ............ 44
   4.3.2 Experimental setup ................................ 45
   4.3.3 Evaluation Metrics ................................ 47
   4.3.4 Monotonic Nature of the Algorithm ............... 48
   4.3.5 One-to-one schema matching for continuous-valued attributes 49
      4.3.5.1 Random initialization ....................... 49
      4.3.5.2 Computational time ......................... 50
      4.3.5.3 Data sampling effects ...................... 51
Chapter 5

Schema Matching Criterion for Mixed Continuous and Categorical attributes

5.1 Introduction ........................................... 56
5.2 Schema matching framework ......................... 57
  5.2.1 Log-likelihood for discrete-valued data ........... 57
  5.2.2 First-order (FO) Mixed Dissimilarity Metrics ....... 58
  5.2.3 Schema matching strategy ....................... 65
5.3 Experimental validation ............................. 65
  5.3.1 Mixed categorical and continuous attributes datasets .... 66
  5.3.2 Experimental setup ................................ 67
  5.3.3 One-to-one schema matching for Mixed Continuous and Categorical Attributes .... 68
    5.3.3.1 Random initialization ....................... 68
    5.3.3.2 Effect of quantized continuous-valued attributes .... 70
    5.3.3.3 Computational time ....................... 71
    5.3.3.4 Data sampling effects ..................... 72
    5.3.3.5 Ground-truth initialization ............... 73
    5.3.3.6 Fixed initialization ....................... 73

Chapter 6

Schema Matching Criterion for Continuous-valued attributes with Transformations

6.1 Introduction ........................................... 75
6.2 Expectation-Maximization for transformation learning .... 76
  6.2.1 Expectation-step ................................ 78
  6.2.2 Maximization-step ................................ 78
  6.2.3 EM initialization ................................ 82
  6.2.4 EM algorithm ................................... 82
6.3 The Schema Matching Framework .................... 82
  6.3.1 First-order dissimilarity objectives for continuous-valued attributes with transformation learning .... 83
  6.3.2 First-order dissimilarity objectives for continuous-valued attributes with fixed pdf transformations .... 85
  6.3.3 Schema matching strategy ....................... 87
6.4 Experimental validation ............................. 88
  6.4.1 Transformed Continuous-valued dataset .......... 88
List of Figures

1.1 The pdf of a continuous attribute modeled by a 3-component Gaussian mixture model. The solid curve is the mixture density, while the dashed curves are the three component densities. The component masses, means and standard deviations are \( \{ \alpha_1 = 0.35, \alpha_2 = 0.25, \alpha_3 = 0.4 \} \), \( \{ \mu_1 = -3.0, \mu_2 = 0.0, \mu_3 = 2.5 \} \) and \( \{ \sigma_1 = 0.5, \sigma_2 = 1.0, \sigma_3 = 0.75 \} \), respectively. ........................................... 7

3.1 2-Opt switching applied to an initial schema match over two hypothetical schemas. The methods for obtaining an initial schema match are discussed in Section 3.3.2 ........................................... 27

3.2 Second-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 .................. 32

3.3 Schema matching results for categorical attributes when randomly initialized .................................................. 33

3.4 Schema matching results for categorical attributes when randomly initialized .................................................. 34

3.5 Schema matching results for categorical attributes when initialized from ground-truth. .................................................. 35

4.1 BIC scores for different model sizes, density estimates and histograms for the “Age” attribute for the CA and NY Datasets. The best BIC score for the “Age” attribute for the CA and TX datasets were obtained for a 5-component GMM and 4-component GMM, respectively, when component variances were freely adjustable parameters. .................................................. 46

4.2 First-order dissimilarity metric (negative Log-likelihood) and precision as a function of accepted two-opt switches for one-to-one matching of 10 attributes, starting from a random initialization ... 47

4.3 First-order schema matching results for continuous-valued attributes for random initializations. .................................................. 48
4.4 Effect of rows on precision for first-order schema matching for continuous-valued attributes. ........................................... 50
4.5 First-order schema matching results for continuous-valued attributes when initialized from ground-truth. ................................. 51
4.6 First-order dissimilarity scores for schema match search space for the California and New York Census datasets when 5 columns were matched. The ground-truth is iteration 97. ........................................... 52
4.7 First-order dissimilarity scores for schema match search space for the California and Texas Census datasets when 5 columns were matched. The ground-truth is iteration 98. ................................. 53
4.8 First-order schema matching results for continuous-valued attributes when initialized from a fixed initialization. ........................ 54

5.1 First-order schema matching results for mixed categorical and continuous attributes ......................................................... 69
5.2 First-order schema matching results for mixed categorical and continuous attributes in the presence of quantized columns. ................. 70
5.3 Effect of rows on precision for first-order schema matching results for mixed attributes. ......................................................... 71
5.4 First-order schema matching results for mixed categorical and continuous attributes when initialized from ground-truth. ................. 72
5.5 First-order schema matching results for mixed categorical and continuous attributes when initialized from a fixed initialization. ................. 74

6.1 Two hypothetical continuous-valued attributes from a schema ... 81
6.2 Negative log-likelihood as a function of iterations for EM algorithm for transformation learning ............................................ 90
6.3 Pdf transformation parameter estimates as a function of iterations ................................................................. 90
6.4 First-order schema matching results for continuous-valued attributes with transformation learning ............................................ 91
6.5 First-order schema matching results for transformed continuous-valued attributes with fixed pdf transformations .................. 94

7.1 The null distribution for dissimilarity scores of non-matches between two Schemas. The null distribution here is a normal distribution with $\mu = 1$ and $\sigma = 0.25$. The cutoff dissimilarity $d = 0.588$ is computed such that $P(D \leq d) = 0.05$, i.e., p-value=0.05. Three accepted dissimilarity scores (diamonds) and three rejected dissimilarity scores (circles) are also shown. ................................. 100
7.2 Estimated and true null distributions for EP and LL dissimilarity scores. ............................................. 104
7.3 First-order Top K schema matching results for categorical-valued attributes at 0.05 significance level .......................... 107
7.4 Average top-K accept rate for first-order top-K schema matching for categorical-valued attributes at 0.05 significance level .......... 108
7.5 Effect of rows on first-order Top K schema matching results for categorical-valued attributes at 0.05 significance level ............ 108
7.6 First-order Top K schema matching results for continuous-valued attributes at 0.05 significance level .......................... 110
7.7 Average top-K accept rate for first-order top-K schema matching for continuous-valued attributes at 0.05 significance level ........ 111
7.8 Effect of rows on first-order Top K schema matching results for continuous-valued attributes at 0.05 significance level .......... 111
7.9 First-order partial schema matching results for categorical attributes at 0.01 significance level ....................................... 112
7.10 First-order partial schema matching results for continuous-valued attributes at 0.01 significance level ............................ 114
## List of Tables

1.1 Two schemas, (a) and (b), with opaque column names and values and with both categorical and quantitative attributes .......................... 3

1.2 Two schemas, (a) and (b), showing similar entropies for each of their columns. The probability mass functions/frequency counts of values, however, which are more descriptive than the entropy, are quite distinctive and can be used in deciding the correct column matches. ................................................................. 4

1.3 Two schemas, (a) and (b), from the Automotive domain with both categorical and quantitative attributes ................................. 11

3.1 Brief description of the Census Bureau PUMS 5% dataset for year 1990 for the states of New York, California and Texas .......................... 30

9.1 Qualitative summary of “one-to-one” schema matching accuracy for the different algorithms ................................................................. 119

9.2 Qualitative summary of “top-K” schema matching accuracy for the different algorithms ................................................................. 119
List of Symbols

\( C \) Discrete component random variable

\( D \) Dissimilarity metric

\( K \) Number of components of a Gaussian mixture model

\( \theta \) Parameters of a Gaussian mixture model

\( \alpha, \beta \) Component masses of a Gaussian mixture model

\( \mu \) Means of a Gaussian mixture model

\( \sigma \) Variance of a univariate Gaussian mixture model

\( \Sigma \) Co-variance matrix of a bivariate Gaussian mixture model

\( \mathcal{L} \) Data-loglikelihood

\( N \) Number of records in a schema

\( M \) Schema match matrix

\( V \) Value mapping matrix

\( \psi \) An affine transformation

\( a, b \) Parameters of an affine transformation
Acknowledgments

This dissertation is a culmination of successful collaboration. I would, whole heartedly, like to thank my advisors Prof. Prasenjit Mitra and Prof. David J. Miller. Prof. Mitra introduced me to the world of databases, knowledge representation, machine learning and information retrieval and has significantly influenced me as a researcher. His support, financial and technical, and patience with me has been a major factor in the completion of this work. He has also been a source of constant encouragement in both my successful and unsuccessful research ventures at Penn State. I cannot be grateful enough to him. Prof. Miller has taught me a great deal through all my interactions, professional and otherwise, that I have had with him. I have not yet met someone with such enthusiasm and penchant for applying statistics and probability theory to real-world applications as him. Some of his enthusiasm, passion and working style has influenced me significantly for which I am greatly indebted to him.

Profs. James Z. Wang and Wang-Chien Lee have been kind enough to serve on my doctoral committee. I would like to gratefully acknowledge their advice and patience.

Prof. Alan M. MacEachren has been instrumental in introducing me to the world and wonder of geography. His passion, patience and vision have influenced my interest in next-generation social media systems and geographic systems. Thank you for your trust and advice. I would also like to acknowledge Prof. Mark Gahegan for his support and direction, partly due to which I pursued this endeavor.

My fellow colleagues at Penn State, past and present, have contributed a great deal to my growth as a researcher and as a human being. I would, thus, like to acknowledge Dr. Ritesh Agrawal, Dr. Tawan Banchuen, Dr. Justine Blanford, Ying Chen, Dr. Frank Hardisty, Scott Pezanowski, Dr. Anthony Robinson, Alexander Savelyev and Sen Xu. Scott Pezanowski deserves a special mention since most of my research work with the Geovista center was done with his collaboration. I would also like to thank him for introducing me to mapping UI’s, OpenLayers, and other very interesting tools.
I would like express my sincere appreciation to Shannon Johnson and Jeff Buechler, System Administrators at the College of Information Sciences and Technology and College of Earth and Mineral Science, respectively, for their continual help in the setup of all the computer systems that I have used and crashed. In addition, I would like to thank the Penn State High Performance Computing (HPC) and the Computer Science High Performance (CSE-HPC) groups for their hard work in maintaining the various high performance computing clusters that I have used during this research. Michelle Hill, Sue Kelleher, Betty Blair, Sherry Rogers and Lynn Yecina have been wonderful student advisors who have time and again made my life very easy.

I would also like to acknowledge and thank both the National Geospatial-Intelligence Agency and the Visual Analytics for Command, Control, and Interoperability Environments (VACCINE) project, a center of excellence of the Department of Homeland Security for funding projects which supported my research assistantship. This work was also supported in part through instrumentation funded by the National Science Foundation through grant OCI–0821527.

I would also like to take this opportunity to thank my friends and colleagues for their encouragement and love which has made my stay at Penn State, one which I shall cherish for a long time to come. Thank you Aarti for putting up with my eccentricities and for your support, motivating speeches, love, cooking and the constant encouragement. I would also like to thank my closest friends Ambuj, Ritesh and Sonali who have constantly provided encouragement and support.

Please forgive me if I have missed some names.
Dedication

This dissertation is dedicated to my mother, Mrs. Manju Jaiswal, and my father, Cdr. K. K. Jaiswal (retd), both of whom believe that one must pursue their dreams. Without their support, love and encouragement, this endeavor would not have been possible.
Chapter 1

Introduction

Schema matching, i.e., the matching of data fields (attributes) with similar semantics across two or more heterogeneous information sources and value mapping, i.e., the identification of a mapping function between value sets with similar semantics for a pair of matched attributes, are critical tasks for applications such as data integration \[18, 29\] and data interoperation \[28\]. Matching needs to be performed before data from multiple tables (usually from different databases) can be integrated or interopereated. Schema matching is in general hard because there is often no documentation present with columns that tell us the semantics of the columns, i.e., column names and values may be opaque. When the number of columns is large, manual matching of schemas has very high labor costs. Even when documentation is available to describe the columns, identifying these matches automatically is challenging because accurate automatic interpretation of textual descriptions is a hard problem despite advances in natural language processing and text mining.

In this paper, we present our work on schema matching for databases containing both some attributes that are categorical-valued, i.e., \(X \in (a_1, \ldots, a_N)\), and some attributes that are continuous-valued, i.e., \(X \in \mathbb{R}\). While the techniques developed in this work can be extended to ordinal attributes, matching schemas containing ordinal-valued attributes is left as future work.

Traditional research has tackled schema matching \[6, 16, 17, 25, 24, 42, 43, 45, 54, 56, 59, 60, 62\], and value mapping \[2, 27, 32, 36, 37, 46, 69\] independently. 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 since data values can be encoded or encrypted; therefore these techniques will not be effective in general. For example, Table 1.1 shows two schemas possessing several characteristics which make schema matching challenging. First, lexical matching techniques will be unable to match the column names across the two tables because the column names are uninformative (opaque). Second, the data values are also encoded opaquely, and differently, in the two tables. Thus, the data value matching cannot be exploited to match the columns. Finally, some column attributes are categorical, e.g., $A \in \{\text{"XO"}, \text{"XL"}, \text{"GE"}, \text{"GX"}\}$, while others are quantitative, e.g., $X \in \mathbb{R}$. In some domains this fact may simplify schema matching, because one can perhaps then restrict to matching attributes of the same type (categorical with categorical and quantitative with quantitative). However, it is also possible that a quantitative attribute was quantized and encoded, prior to its inclusion in one of the databases. To accommodate this possibility, one should in general also evaluate matches between categorical attributes in one schema and quantitative attributes in the other. Accordingly, one should have sound matching criterion suitable for matching attributes of different types. To summarize, important challenges for schema matching are: a) opaque column names and data values, and b) attributes of different types. The first difficulty has been addressed in several works using what are called uninterpreted schema matching techniques [35, 34, 38, 39], whereas the latter challenge has not been well-addressed by prior works. In this dissertation, we will both develop and experimentally evaluate techniques that simultaneously address both of these challenges. We will first review prior work on uninterpreted schema matching, which we will then extend in developing our novel methods.

### 1.1 Uninterpreted Schema Matching for Discrete-valued Attributes

**Definition** *Uninterpreted vs. Interpreted Matching:* Let $M_1 = \text{match}(X(X_1, X_2, \ldots, X_n), Y(Y_1, Y_2, \ldots, Y_m))$ and $M_2 = \text{match}(X(X_1, X_2, \ldots, X_n), Y(f_1(Y_1), f_2(Y_2), \ldots, f_n(Y_n)))$, where $M_i$ is a match result returned by a schema matching
Table 1.1: Two schemas, (a) and (b), with opaque column names and values and with both categorical and quantitative attributes

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>XE</td>
<td>W</td>
<td>SE</td>
<td>2.1</td>
<td>51.2</td>
</tr>
<tr>
<td>XL</td>
<td>R</td>
<td>CO</td>
<td>1.2</td>
<td>42.1</td>
</tr>
<tr>
<td>LE</td>
<td>S</td>
<td>SU</td>
<td>3.0</td>
<td>61.1</td>
</tr>
<tr>
<td>LX</td>
<td>B</td>
<td>CM</td>
<td>5.1</td>
<td>55.1</td>
</tr>
</tbody>
</table>

...continued...

algorithm, \( \mathcal{X} \) is a source schema of \( n \) attributes, \( \mathcal{Y} \) is a source schema of \( m \) attributes, and \( f_i \) is an arbitrary one-to-one transformation applied to the data values of attribute \( i \) in the target schema. We call the schema matching algorithm an uninterpreted 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.

Uninterpreted schema matching is a challenging problem that involves combinatorial optimization because one opaque column may potentially match with a number of other opaque columns in the other table. Even in the cases where values in columns are interpreted, the methods described in this paper, which match schemas based on the similarities of the distributions of values in columns across tables, can be used in conjunction with syntactic value matching based on string similarity or pattern matches. Thus, the uninterpreted schema matching methods developed in this paper, which give improved performance compared to previous methods, can be considered to be a complementary set of tools adding to the arsenal of existing matching methods.

Previous works on uninterpreted schema matching [34, 39, 38] considered only categorical attributes and matched categorical attributes across schemas by finding attributes ([attribute pairs) with similar statistics. The probability mass function (pmf) for each discrete-valued attribute in a schema can be 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 mea-
Table 1.2: Two schemas, (a) and (b), showing similar entropies for each of their columns. The probability mass functions/frequency counts of values, however, which are more descriptive than the entropy, are quite distinctive and can be used in deciding the correct column matches.

<table>
<thead>
<tr>
<th>ModelA</th>
<th>P(ModelA)</th>
<th>ColorA</th>
<th>P(ColorA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XE</td>
<td>0.05</td>
<td>Red</td>
<td>0.25</td>
</tr>
<tr>
<td>XL</td>
<td>0.10</td>
<td>Blue</td>
<td>0.25</td>
</tr>
<tr>
<td>XO</td>
<td>0.20</td>
<td>Green</td>
<td>0.25</td>
</tr>
<tr>
<td>LE</td>
<td>0.20</td>
<td>Black</td>
<td>0.25</td>
</tr>
<tr>
<td>LX</td>
<td>0.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H(ModelA)</td>
<td>1.995</td>
<td>H(ColorA)</td>
<td>2.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ModelB</th>
<th>P(ModelB)</th>
<th>ColorB</th>
<th>P(ColorB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XE</td>
<td>0.06</td>
<td>Red</td>
<td>0.24</td>
</tr>
<tr>
<td>XL</td>
<td>0.11</td>
<td>Blue</td>
<td>0.24</td>
</tr>
<tr>
<td>XO</td>
<td>0.18</td>
<td>Green</td>
<td>0.26</td>
</tr>
<tr>
<td>LE</td>
<td>0.19</td>
<td>Black</td>
<td>0.26</td>
</tr>
<tr>
<td>LX</td>
<td>0.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H(ModelB)</td>
<td>2.009</td>
<td>H(ColorB)</td>
<td>1.998</td>
</tr>
</tbody>
</table>

sured value for attribute X). Furthermore, since it is quite possible that records in a database have missing/null values, however, one can handle such records by either treating missing/null values as a unique category or by entirely dropping such records. For pmfs, one useful (characteristic) statistic is the pmf’s entropy. Entropy measures the uncertainty or information in a discrete random variable and is defined as  

\[
H(X) = -\sum_{x \in X} P(X = x) \log P(X = x),
\]

where \(X\) the set of values taken on by attribute \(X\). An attribute with high entropy has values that are ~uniformly distributed while an attribute with low entropy has its probability mass concentrated on only a few of the values in \(X\). A matching algorithm could, thus, generate a schema match by finding pairs of attributes across the two schemas with similar entropies. Furthermore, this technique can be extended to utilize second-order statistics (e.g., mutual information [39, 38]), based on pairs of attributes from the same schema, to provide improved matching. This is the strategy employed in Kang and Naughton [39, 38]. However, entropy and mutual-information based methods [39, 38] assume that there are distinct values for these statistics unique to each attribute/attribute pair (across the two tables). Consequently, such methods
will fail when there are multiple attributes/attribute pairs with similar statistics for the same schema. To illustrate the difficulty with this approach, consider a different example. Table 1.2 shows two different schemas in (a) and (b) modeled after an automotive domain, the values and probability mass functions defined on these values for each of the attributes under each schema, and the entropies for each of these attributes. Suppose that the column names and values are in factopaquely coded – we are showing non-opaque (informative) values in Table 1.2 merely to assist the reader’s understanding of this example. For example, in Schema (a), the probability that the car model is of type “XE” is 0.05. Likewise, the probability that the car’s color is “Red” is 0.25. Furthermore, the entropy of the attributes are $H(\text{Model}_A) = 1.995$, $H(\text{Color}_A) = 2.000$, $H(\text{Model}_B) = 2.009$ and $H(\text{Color}_B) = 1.998$ as shown in Table 1.2. Since the entropies are all very similar, one cannot reliably pair Model$_A$ with Model$_B$ and Color$_A$ with Color$_B$, i.e., 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 [39, 38]), based on pairs of attributes from the same schema, are used in matching. However, note that the pmfs of the columns being matched, which are more descriptive than the entropy (which is just a function of the pmf) are quite distinctive and can be potentially used to make more confident matches between attributes if the value mappings are known. In particular for the example in Table 1.2 it is easy to see that, if one makes accurate value mapping correspondences, the two “Color” pmfs are quite similar to each other and quite different from the two “Model” pmfs. Likewise, the two “Model” pmfs are quite similar to each other.

Previous work [35, 34] presented an automated technique that gives more accurate results than those produced by the entropy and mutual-information based methods [39, 38] in matching schemas with opaque categorical attributes. This improvement was achieved by exploiting the more informative, distinctive description of attributes offered by the attribute’s probability mass function and by simultaneously performing schema matching and embedded value mapping in order to evaluate the degree of similarity between the pmfs of the candidate-matched attributes. Such a technique is effective in the particularly difficult cases where both the column names and data values are opaque and where multiple attributes/attribute pairs may have similar first/second order statistics (entropies or mutual information...
tions) - in such cases Kang and Naughton [39, 38] will fail. Instead of characterizing a column by a single value like entropy, this technique characterizes each attribute (or attribute pair) by the pmf defined over its value support set. However, this method inherently assumes that attributes within two schemas are discrete-valued (categorical or ordinal). While the pmf-matching technique [35, 34] is effective in performing schema matching for discrete-valued attributes, this approach is not applicable when attributes are continuous-valued or when some attributes are discrete and some continuous-valued.

1.2 Proposed Representation for Continuous-valued Attributes

The techniques presented in this thesis are designed to work in the particularly difficult cases wherein attributes have opaque column names and opaque data values for categorical attributes, but where there may also be quantitative attributes. For example, schemas can have categorical attributes like “Sex”, “Race” etc. as well as quantitative attributes like “Age”, “Weight” etc. Furthermore, one cannot necessarily assume that attributes should be strictly matched by type, i.e., a quantitative attribute in one schema could have a quantized, encoded (discrete) matching attribute in the other schema. For example, one schema can have a quantitative attribute “Age” while the matching attribute in the other schema could be a quantized, encoded attribute “Age” which was attained by data binning. Similarly, some attributes could be quantized, encoded by other common preprocessing techniques such as data grouping, clustering, compression etc. It is natural to characterize a continuous-valued attribute by a probability density function (pdf) defined over its support. In this work, we will choose to model all continuous-valued attributes using Gaussian mixture models (GMM) [19, 50, 52]. There are several reasons for this choice. First GMMs are ubiquitous for density modeling in engineering, scientific, and business applications [58, 64, 65], and they accurately model continuous-valued data in numerous such domains. Second as we will make explicitly clear in what follows, a mixture model representation for continuous-valued attributes is a flexible choice, one that can be used both for
matching two continuous-valued attributes across schemas as well as for matching a continuous-valued attribute in one schema with a discrete attribute in another (thus accounting for the possibility that the discrete attribute was in fact the result of quantizing a continuous-valued attribute).

For a continuous-valued attribute $X$, the GMM density function is:

$$f_X(x; \theta) = \sum_{j=1}^{K} P[C = j]f_{X|M=j}(x; \theta_j) = \sum_{j=1}^{K} \alpha_j f_{X|j}(x; \theta_j), \quad (1.1)$$

where $C$ is a discrete component random variable, $\theta_j = (\mu_j, \sigma_j^2)$ with $\mu_j$ the mean and $\sigma_j^2$ the variance, and with $f_{X|j}(x; \theta_j)$ the $j^{th}$ component Gaussian density function given by:

$$f_{X|j}(x; \theta_j) = \frac{1}{\sigma_j \sqrt{2\pi}} e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}}. \quad (1.2)$$

Also $0 \leq \alpha_j \leq 1$ are the probability masses of the components, constrained to satisfy the pmf constraint $\sum_{j=1}^{K} \alpha_j = 1$. Since the masses sum to 1 and since $f_{X|j}(x; \theta_j)$ is a density function, $\forall j$, $f_X(x; \theta)$ is also a valid (mixture-based) density function, integrating to 1. A mixture density example (with $K = 3$) is shown in Fig. 1.1. With proper choice of $K$ and the parameters $\theta = \{\theta_i, \alpha_i, i = 1, \ldots, K\}$, GMMs can accurately approximate many multimodal density functions. How to learn the model parameters from a given dataset and how to estimate the number of components, $K$, in practice are topics discussed in Appendix B.
We noted earlier that mixture modeling of continuous-valued attributes allows one to match a continuous-valued attribute to another continuous-valued attribute, but it also enables matching a continuous-valued attribute to a discrete attribute. Moreover, mixture modeling also enables matching between two continuous-valued attributes when an affine pdf transformation must be applied before the two attributes can be matched. Let us explain how these types of matchings can be achieved.

### 1.2.1 Matching two continuous-valued attributes

First, suppose there are two continuous-valued attributes, \( X \) and \( Y \), in different databases that are candidates for being matched. Let \( \mathcal{X} = \{x_1, \ldots, x_{N_x}\} \) be the row values (data entries) for \( X \)'s column and let \( \mathcal{Y} = \{y_1, \ldots, y_{N_y}\} \) be the data entries for \( Y \)'s column. Further, suppose that the models \( \theta^{(x)} = \{\alpha_1^{(x)}, \theta_1^{(x)}, i = 1, \ldots, K_x\} \) and \( \theta^{(y)} = \{\alpha_1^{(y)}, \theta_1^{(y)}, i = 1, \ldots, K_y\} \) have been learned using the Expectation-Maximization (EM) algorithm \([14, 50, 49]\) with mixture model order selection (see Appendix B on how this is achieved). If \( X \) and \( Y \) are in fact matching attributes, it is reasonable to suppose that \( \theta^{(x)} \) is a good model for “explaining” (fitting) the observed data \( \mathcal{Y} = \{y_1, \ldots, y_{N_y}\} \); and, likewise that \( \theta^{(y)} \) is a good model for “explaining” \( \mathcal{X} \). This degree of fit can be objectively evaluated, assuming that the rows (data entries) are stochastically generated in a statistically independent fashion, by measuring the data’s log-likelihood under the assumed density model.

Specifically, we can evaluate the data log-likelihoods

\[
\mathcal{L}_X(\mathcal{Y}) = \log \prod_{i=1}^{N_y} f_X(y_i, \theta^{(x)}) = \sum_{i=1}^{N_y} \log \sum_{j=1}^{K_x} \alpha_j^{(x)} f_{X|j}(y_i; \theta_j^{(x)}) \tag{1.3}
\]

\[
\mathcal{L}_Y(\mathcal{X}) = \log \prod_{i=1}^{N_x} f_Y(x_i, \theta^{(y)}) = \sum_{i=1}^{N_x} \log \sum_{j=1}^{K_y} \alpha_j^{(y)} f_{Y|j}(x_i; \theta_j^{(y)}) \tag{1.4}
\]

and define the quality of the match by the sum \( \mathcal{L}_X(\mathcal{Y}) + \mathcal{L}_Y(\mathcal{X}) \) with greater log-likelihood indicating a better match.

In Section 5.2.2 we will show how the data log-likelihood can in fact also be used for matching two discrete-valued attributes (see Section 5.2.1). We will
also exploit this and the above log-likelihood measures for modeling continuous-valued attributes for matching two databases, each potentially containing both continuous-valued and categorical-valued attributes. For this purely likelihood-based global objective, we will restrict continuous-valued attributes to be matched against continuous-valued attributes and discrete attributes to be matched against discrete attributes. Effectively, this approach will assume that there is prior knowledge that no discrete attributes were obtained by quantizing continuous-valued attributes.

1.2.2 Matching a continuous-valued attribute and a discrete attribute

Suppose, however, that some discrete attributes may have been obtained by quantizing continuous-valued attributes. In this case, we may need to evaluate a candidate match between a continuous-valued attribute $X$, and a discrete attribute $A \in \{1, \ldots, L\}$. Mixture modeling naturally facilitates such matching, as we detail next. The discrete attribute, $A$, has a probability mass function $P[A = m], m = 1, \ldots, L$, which can be estimated based on frequency counts from the given database (see also Section 5.2.1 description). While the continuous-valued attribute $X$ does not explicitly have a pmf representation, the mixture component random variable $C$ used to model $X$ does have such a representation. Specifically, we have the probability mass function on the mixture components $P[C = j] = \alpha_j, j = 1, \ldots, K_x$. Furthermore, as shown in Appendix B the random variable $C$ is essentially obtained by probabilistic (soft) quantization of $X$, with $P[C = j] = \alpha_j$ an estimate of the number of data entries that are (probabilistically) assigned to “quantization interval” $j$. An advantage of mixture modeling (soft quantization) over hard-threshold quantization is that soft quantization preserves more information (estimating probabilities of membership in a component rather than hard-thresholded $\{0-1\}$ membership). Thus, if $A$ was truly obtained by quantizing a continuous-valued attribute that is the counterpart of $X$, then it is reasonable to match the pmfs for $C$ and $A$ to assess match quality. In the sequel, we will exploit the methodology and objective function from [34], which performs embedded value mapping while matching pmfs of discrete-valued attributes, for
this purpose.

1.2.3 Matching two continuous-valued attributes with affine pdf transformations

Lastly, suppose that the statistical assumption that the data from two matching continuous-valued attributes were generated by sampling i.i.d. from a common distribution $f_X(x)$ is violated. For example, consider two schemas describing sensor data collected by meteorological agencies in the United States and Europe containing a matching attribute “Temperature”. The “Temperature” column could store the values in Fahrenheit or Celcius, thus potentially requiring a transformation to be applied to data values before schema matching is performed. Mixture modeling can facilitate matching under affine transformation scenarios, i.e., when there exists an affine transformation between the two continuous-valued attributes. Let $X = \{x_1, \ldots, x_{N_X}\}$ be the row values (data entries) for $X$’s column and let $Y = \{y_1, \ldots, y_{N_Y}\}$ be the data entries for $Y$’s column. Further, suppose that the models $\theta(x) = \{\alpha_j^{(x)}, \mu_j^{(x)}, \sigma_j^{(x)}, i = 1, \ldots, K_X\}$ has been learned using the Expectation-Maximization algorithm [14, 50, 49] with mixture model order selection (see Appendix B on how this is achieved). If $X$ and $Y$ are in fact matching attributes such that $Y = aX + b$, then, this transformation will result in mean shifts and variance scaling, i.e., $\mu_i^{(y)} = a\mu_i^{(x)} + b$ and $\sigma_i^{(y)} = a\sigma_i^{(x)}$, but with the density functional form (Gaussian) preserved. Therefore, the log-likelihood for observations $Y$ can be written as:

$$\log L_X(Y, a, b) = \sum_{i=1}^{N_Y} \log \sum_{j=1}^{K_X} \frac{\alpha_j}{\sqrt{2\pi a^2 \sigma_j^{(x)^2}}} e^{-\frac{(y_i - (a\mu_j^{(x)} + b))^2}{2a^2\sigma_j^{(x)^2}}}. \quad (1.5)$$

An EM algorithm that monotonically increases in $\log L(a, b)$ can then be used to estimate the transformation, i.e., estimate the scaling factor $a$ and the shift $b$. Similarly, the transformation $X = cY + d$ can also be estimated. Furthermore, the quality of the match can be then measured by the sum $L_X(Y, a, b) + L_Y(X, c, d)$ with greater log-likelihood indicating a better match.

Several such objective functions, handling schemas with continuous-valued at-
Table 1.3: Two schemas, (a) and (b), from the Automotive domain with both categorical and quantitative attributes

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Retail Price</th>
<th>Dealer Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>XE</td>
<td>Coupe</td>
<td>21999.99</td>
<td>21099</td>
</tr>
<tr>
<td>XL</td>
<td>Sedan</td>
<td>24299.99</td>
<td>23299</td>
</tr>
<tr>
<td>LE</td>
<td>Sedan</td>
<td>24999.99</td>
<td>24299</td>
</tr>
<tr>
<td>LX</td>
<td>Luxury</td>
<td>51099.99</td>
<td>50000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>Corporation 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Retail Price</th>
<th>Dealer Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>XO</td>
<td></td>
<td>20399.99</td>
<td>20999.99</td>
</tr>
<tr>
<td>XL</td>
<td></td>
<td>21299.99</td>
<td>22999.99</td>
</tr>
<tr>
<td>GE</td>
<td></td>
<td>22199.99</td>
<td>23999.99</td>
</tr>
<tr>
<td>GX</td>
<td></td>
<td>41099.99</td>
<td>46999.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>Corporation 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

attributes (with or without pdf transformations), as well as both continuous-valued and categorical attributes, will be developed here. To optimize our proposed objective functions, and thus, to determine the schema matching, we propose a locally optimal iterative search strategy, similar to the approach taken in [35, 34], that monotonically descends in our objective functions. We have found this technique to be quite effective at producing accurate schema matches, as will be shown in Section 3.3, 4.3, and 5.3.

1.3 Schema matching strategies and proposed solutions

In this dissertation, we investigate and address both “onto” and “top-K” schema matching, which we describe below.

1.3.1 Onto matching

Onto matching, on two schemas, implies that there exists for every attribute in the first schema, a matching column in the second schema, i.e., given two schemas $S_1$ and $S_2$ with $N_{C_1}$ and $N_{C_2}$ attributes, respectively, such that $N_{C_1} \leq N_{C_2}$, then there exists for all $N_{C_1}$ attributes in $S_1$, a matching attribute in $S_2$. “One-to-one” or “bijective” matching is a special case of “onto” matching when $N_{C_1} = N_{C_2}$. To address the “onto” schema matching problem, we propose global objective functions based on dissimilarity metrics that provide a confidence (similarity) score for
a fixed schema match while taking into account the value distributions of matching attributes, i.e., it embeds value mapping optimization. Several different dissimilarity metrics are developed to accommodate both continuous-valued and categorical attributes. We minimize our global objective functions over both the space of schema matches and value mappings, seeking to arrive at the best schema matching. We have found this technique to be quite effective at producing accurate schema matches, as will be shown in Sections 4.3, 5.3 and 6.4.

1.3.2 Top-K matching

Schema matching algorithms performing “onto” matching return a single best (or top-1) match per attribute for a set of attributes being matched across two schemas, i.e., each attribute in the first schema is best matched to another attribute in the second schema. However, such state-of-the-art automatic schema matching techniques are often not used in business applications since business users do not trust automatically computed best mappings [60]. This lack of trust can be attributed to the fact that the automatically generated top-1 mapping does not provide an insight into the imprecision inherent in the schema matching process. To illustrate this issue, consider Table 1.3 which shows two different schemas, in Tables 1.3 (a) and 1.3 (b) modeled after the automotive domain. Suppose that attribute “Retail Price” in the first schema can be matched to either attribute “Price 1” or “Price 2” in the other schema based on some metric (minimum dissimilarity or maximum string similarity) by a schema matching algorithm. However, current top-1 schema matching algorithms will return only one of these matches to the user as its best guess. Thus, if the top-1 matching algorithm returned an incorrect guess, the human will need to manually perform schema matching to determine the correct schema match. While it is possible for the human to perform manual schema matching for incorrect matches returned by a matching algorithm for the schemas in Table 1.3, this is not an easy task when (a) both column names and values in the two schemas are opaque to encoded, and, (b) one or both schemas has a large number of attributes. For industrial schema matching algorithms, it is, therefore, preferable that the matching algorithm return a set of top-K ranked candidate matches [15, 23, 24, 60], i.e., for the example the algorithm should return both
“Price 1” and “Price 2” as the best candidates for matching. The human overseeing the schema matching can then make an informed matching decision for each attribute based on the set of ranked top-K matches for that attribute.

Previous works [15, 25, 24, 60] have investigated top-K schema matching and shown that a choice of a large $K$ (e.g., $K = \{5, \ldots, 10\}$) results in high schema matching recall ($>0.90$) [60]. However, choosing a large $K$ significantly increases human cost involved in deciding the correct schema matches, especially for large schemas. For example, for two schemas each having 20 attributes will require a user to evaluate at least 100 or 200 schema matches when top-5 or top-10 schema matches per attribute are returned, respectively. Rather than returning all top-K schema matches per attribute to a user, a statistical test to accept only the most significant top-K schema matches can be performed, thereby significantly reducing the number of total schema matches that a user will need to evaluate. We illustrate this strategy in Chapter 7. Experimental evaluation in Section 7.3 shows that such a strategy is effective in reducing the total number of schema matches returned to a user without compromising schema matching recall.

\section{1.4 Research contributions}

The primary contributions of this dissertation are as follows:

- To the best of our knowledge, this work is the first to propose a novel, integrated framework for un-interpreted schema matching that can
  - match columns with continuous valued attributes by modeling continuous valued columns using a Gaussian Mixture Model,
  - automatically match continuous-valued columns to discrete-valued columns where one column may have been generated by quantizing continuous-valued columns, without requiring any knowledge about how the quantization was done.

- We investigate several different objective functions for matching continuous-valued attributes, suitable for scenarios where there are a) only continuous-valued attributes, and, b) both continuous and categorical-valued attributes.
We also consider the scenario where some continuous-valued attributes have been quantized. Our experiments are conclusive about the best objective function choice for each case. Empirically, we show that

- datasets with purely continuous-valued attributes can be matched with > 75% accuracy. Of the dissimilarity metrics we tested, our method based on log-likelihood produces the best results.
- datasets with both continuous-valued attributes and categorical attributes can be matched with > 85% accuracy. Of the dissimilarity metrics we tested, our method based on Euclidean-distance produces the best results.

- We introduce an iterative descent algorithm to handle continuous-valued attributes that embeds value mappings to enhance schema matching accuracy. This naturally extends the approach used in Jaiswal et al. [34] for categorical-valued attributes.
- We develop a novel Top-K schema matching strategy that integrates statistical hypothesis testing to improves schema matching accuracy. We have found this method achieves high recall (>80%) for top-5 schema matching on both categorical and continuous-valued attributes.
- We demonstrate the effectiveness of our methods through experimental studies, where we compare our results with mutual information based [39, 38] and first-order pmf-based [34] schema matching methods. Our experiments show that the proposed techniques significantly outperform earlier methods, and thus, they should be a useful addition to a suite of (semi) automatic schema matching techniques.
- Lastly, we demonstrate the effectiveness of a second-order pmf-based schema matching method [34] with an experimental study. Our experiments show that the second-order methods produce more fine-grained matching and improved accuracy, and thus should be considered during schema matching.
1.5 Organization of this dissertation

The rest of the dissertation is organized as follows: Chapter 2 discusses the related work and provides a background to the existing techniques on schema matching and value mapping. Chapter 3 outlines pmf-based schema matching with embedded value mapping algorithms for discrete/categorical attributes. Schema matching algorithms for continuous-valued, mixed (continuous-valued and categorical) attributes and continuous-valued attributes with pdf transformations are described in Chapters 4, 5 and 6. Top-K schema matching and its application to partial schema matching is discussed in Chapter 7. Chapter 8 outlines the potential directions for future work. We conclude this dissertation in Chapter 9.
Jaiswal et al. [34, 35] was the first work in which embedded value mapping was used to improve schema matching accuracy. Other prior work had separately tackled either (1) Schema matching [3, 4, 6, 7, 8, 9, 13, 16, 17, 25, 24, 26, 31, 39, 41, 42, 43, 45, 41, 47, 51, 53, 54, 56, 57, 59, 60, 62, 68, 70], or, (2) Value/Object mapping [2, 10, 20, 27, 32, 36, 37, 46, 69]. Most of these techniques except Kang and Naughton [39, 38] and Jaiswal et al. [34] rely on identifying similarity in the 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 jointly perform schema matching and value mapping to produce improved schema matching, handling continuous, mixed continuous/categorical, and, transformed continuous-valued data attributes. Our work is complementary to traditional schema matching or value mapping, which rely on data interpretation of column names or data values respectively. Our work also substantially extends prior work Jaiswal et al. [34, 35], which only treated discrete, categorical-valued attributes.

2.1 Schema matching

Various techniques have been proposed to resolve structural heterogeneity. For good surveys and comparisons of schema matching methods, see Do et al. [16], Rahm et al. [62] and Bernstein et al. [5]. Kang and Naughton [39, 38] proposed an uninterpreted schema matching technique for categorical-valued attributes that employs mutual information to construct dependency graphs followed by graph
matching. However, their method does not integrate value mapping to improve schema matching. Jaiswal et al. [34] proposed pmf matching with embedded value mapping and demonstrated that this approach significantly improves matching accuracy for categorical-valued attributes. Cruz et al. [13] have adapted the Kang and Naughton [39] technique for schema matching applications while preserving privacy. Since, the metrics presented in this thesis are more discriminative than attribute-level metrics, such as mutual information and entropy, therefore they would be more effective for privacy preserving applications.

Machine learning based techniques have been proposed (e.g. LSD [17], Automatch [4]) and Neural Networks (e.g. SemInt [42]), but such systems rely on data interpretation for learning and are therefore complementary to our method. SMDD [43] is a semi-supervised schema matching technique based on neural networks where attribute data distributions are used as input. For example, in SMDD, numeric attributes are discretized into a fixed number of categories. Unlike SMDD, our methods can be applied to tables with both continuous and discrete-valued attributes and do not need ad-hoc discretization of continuous-valued attributes. Other machine learning techniques used for database schema matching include Automatch [4] where a knowledge-base called the attribute dictionary is used to find the optimal schema matching. The attribute dictionary is generated by bayesian learning from examples of schemas which have been mapped by a “domain expert”.

sPLMap [57] is a probabilistic framework where schema matching rules are automatically learned from given instances of both schemas. Different machine learning algorithms are used to find suitable matching candidates. The machine learning algorithms use semantic relatedness such as those based on attribute names and properties of data instances. SMatch [26] is a similar system designed for matching XML graphs based on semantic relations between them which are discovered by analyzing meaning from the schemas.

Bin et. al. [31] propose a holistic technique for statistical schema matching for integrating data sources across the “deep Web”. Their method attempts to find an underlying generative schema model that can match all input schemas since attributes for web query interfaces from similar domains typically consist of a finite set of keywords. These methods again rely on data interpretation and thus are not suitable for uninterpreted schema matching. Other techniques (e.g.
Cupid \cite{15}) perform graph matching based on the structural similarity between graphs derived for the schemas being matched. Such methods cannot be applied to uninterpreted schema matching. Other techniques include rule-based methods (e.g TranScm \cite{54} and ARTEMIS \cite{9}) which can be used in conjunction with our techniques to improve schema matching.

Bernstein et al. \cite{7, 6} present a customizable schema matcher using multiple schema matching techniques/ algorithms. Similarly, COMA++ \cite{3} is an integrated schema and ontology matching tool which provides a composite framework to combine several match algorithms. COMA++ is an extension of their earlier prototype system, COMA \cite{15}. Nandi et al. \cite{56} propose a schema matching method to integrate multiple source schemas into a search engine schema. This method matches two schema elements when the distribution of search engine keyword queries that cause click-throughs are similar. Our techniques could be a useful input within these frameworks. Wick et. al. \cite{68} demonstrate a discriminatively trained model that integrates schema matching, coreference and canonicalization jointly. Their technique uses a conditional random field (CRF) based on data (schema and record) clustering to integrate these tasks. Integrating our techniques to tackle the coreference problem would an interesting area for future research.

Gal \cite{24, 25} extends current semantic/ syntactic schema matching methods by utilizing top-K schema mappings simultaneously to automatically identify useful mappings. In Chapter \cite{7} we propose a top-K schema matching technique that leverages statistical hypothesis testing to generate confident top-K schema matches. This technique would be especially useful to previous top-K schema matching methods which use adhoc thresholding to generate top-K schema matches. McCann \cite{47} studies the role of community volunteers in aiding user-driven schema matching for Web 2.0 communities. Our techniques can form the baseline schema matching from where users can begin integration of multiple data sources. Peukert et al. \cite{60} proposes a weighted token similarity measure that decreases the overlap between top-K sets and a top-K selection operator to improve recall in schema matching. Such syntactic approaches rely on data semantics and are therefore complementary to our method. Lastly, Peukert et al. \cite{59} evaluate most commonly used similarity combination methods for schema matching which could utilize our techniques as input.
2.2 Value mapping

Past work on value/ object mapping relies on finding syntactic similarities or semantic interpretation of the values. Typically, these methods assume that the schema matching across tables is known. Kang et al. [36, 37] identify value mappings when two objects have few syntactic similarities using (semi) automatic statistical techniques. Other work on the object-mapping problem, known by various names in diverse contexts, includes; e.g., record linkage [20, 69], citation matching [46], merge-purge [32], duplicate detection [2] and approximate string join [10, 27]. These techniques find similar objects (e.g. records, tuples, citations, values) by relying on syntactic similarities between objects under comparison. Such methods cannot be easily applied to un-interpreted matching. While our methods exploit embedded value mapping to enhance schema matching accuracy, the learned value mappings, in addition to schema matches are outputs of our algorithms.
Chapter 3

Schema Matching Criterion for Categorical-valued Attributes

3.1 Introduction

Our past work [35, 34] demonstrated a first-order pmf matching algorithm [34] which was suitable for schema matching for discrete and categorical attributes. This improvement was achieved by exploiting the more informative, distinctive description of attributes offered by the attribute’s probability mass function and by simultaneously performing schema matching and embedded value mapping in order to evaluate the degree of similarity between the pmfs of the candidate-matched attributes. We introduce pmf-based matching algorithms in this chapter. In Section 3.3 we demonstrate the effectiveness of using pair-wise pmf matching over other methods for categorical attributes.

3.2 Schema matching framework

In this section, we describe in detail our uninterpreted 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 3.2.3.

Algorithm 1 Overview of our approach

Input: Schemas, $T_1$ and $T_2$
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($tmpS$, $tmpV$, $T_1$, $T_2$)
    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

3.2.1 Dissimilarity objectives for categorical-valued attributes

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.
3.2.1.1 First-order (FO) Categorical Dissimilarity Metric Model

The first order dissimilarity metric model measures how well the probability mass functions (pmfs) of matching attributes align for a fixed schema match and value mapping across tables. Therefore, given a schema match matrix $M$ defined as,

$$M = [M_{ij}], \text{ where } M_{ij} = \begin{cases} 1, & \text{if } i^{th} \text{ attribute matches to } j^{th} \text{ attribute across Schemas 1 and 2} \\ 0, & \text{otherwise} \end{cases}$$ \tag{3.1}$$

, and, a value mapping matrix $V$ defined as,

$$V = [v_{kl}^{(ij)}], \text{ where } v_{kl}^{(ij)} = \begin{cases} 1, & \text{if the } k^{th} \text{ value in the } i^{th} \text{ attribute in Schema 1 matches to } l^{th} \text{ value in the } j^{th} \text{ attribute in Schema 2} \\ 0, & \text{otherwise} \end{cases}$$ \tag{3.2}$$

We can, now, define two dissimilarity metrics that utilize first order statistics: One based on the squared Euclidean distance as shown in Eq. (3.3) and the second based on Relative entropy [11] as shown in Eq. (3.4).

$$D_{EU}(M, V) = \sum_{i=1}^{N_{C_1}} \sum_{j=1}^{N_{C_2}} M_{ij} \sum_{k=1}^{N_{V_i}} \left[p_k^{(i)} - \left\{ \sum_{l=1}^{N_{V_j}} v_{kl}^{(ij)} p_l^{(j)} \right\} \right]^2$$ \tag{3.3}$$

$$D_{RE}(M, V) = \sum_{i=1}^{N_{C_1}} \sum_{j=1}^{N_{C_2}} M_{ij} \sum_{k=1}^{N_{V_i}} p_k^{(i)} \times \log \frac{p_k^{(i)}}{\sum_{l=1}^{N_{V_j}} v_{kl}^{(ij)} p_l^{(j)}}$$ \tag{3.4}$$

Where:

$D_{EU} = $ Euclidean distance based dissimilarity for a fixed schema match and value mapping

$D_{RE} = $ Relative Entropy based dissimilarity for a fixed schema match and value mapping

$N_{C_1}, N_{C_2} = $ Number of categorical attributes in Schemas 1 and 2, respectively

$N_{V_i}, N_{V_j} = $ Number of values in $i^{th}$ and $j^{th}$ attributes in Schemas 1 and 2, respectively

$p_k^{(i)} = $ probability of the $k^{th}$ value for $i^{th}$ attribute in Schema 1
\[ p_l^{(j)} = \text{probability of the } l^{th} \text{ value for } j^{th} \text{ attribute in Schema 2} \]

In Eqs. (3.3) and (3.4), we have assumed \( \mathcal{N}_{C_1} \leq \mathcal{N}_{C_2} \), i.e. each column in schema 1 matches to a unique column in schema 2. This is equivalent to “onto matching”. The dissimilarity metric based on relative entropy has difficulties when the probability of a value used in matching in Eq. (3.4) is zero. This is problematic because \( p \times \log \frac{p}{0} = \infty \). This issue arises especially when \( \mathcal{N}_{V_i} \neq \mathcal{N}_{V_j} \) for attributes \( i \) and \( j \) 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 \( \epsilon \) 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 value.\(^1\)

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 \([11]\). The squared Euclidean 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 \( \epsilon \) 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

\(^1\)Some care may be needed in the choice of the small probability value \( \epsilon \) ensuring, e.g., that this probability value is much smaller than the smallest probability value of the original alphabet.
another suitable metric, if future evidence shows that metric to be a better distance function for our application.

3.2.1.2 Second-order (SO) Categorical Dissimilarity Metric

Finer-grained schema matching with embedded value mapping can be achieved by utilizing second-order statistics. We developed a second-order dissimilarity metric model for categorical-valued attributes which we illustrate below. The second-order dissimilarity objective for categorical-valued attributes measures how well the pairwise probability mass functions (joint probabilities) for all pairs of attributes from one schema align for a fixed schema match and value mapping with those of another table, i.e., the second-order objective seeks to match \( \{P_{ij}[k,l], i = 1, \ldots, N_{C_1}, j = 1, \ldots, N_{C_1}, i \neq j, k = 1, \ldots, N_{v_i}, l = 1, \ldots, N_{v_j}\} \) measured for one table with corresponding pairwise pmfs for another table. Here, \( N_{C_1} \) is the number of attributes in the first schema, \( N_{v_i} \) is the number of values in the \( i^{th} \) attribute and \( N_{v_j} \) is the number of values in the \( j^{th} \) attribute. Joint probabilities (pairwise pmfs) will capture and thus exploit statistical dependencies that may exist between all pairs of categorical-valued attributes. Thus, given a schema match matrix \( M \) (Eq. (3.1)) and value mappings \( V \), the second-order Euclidean distance dissimilarity metric can be written as shown in Eq. (3.2):

\[
D_{EU}(M, V) = \sum_{i=1}^{N_{C_1}} \sum_{j=1}^{N_{C_1}} \left[ \sum_{i' \neq i}^{N_{C_2}} \sum_{j' = 1}^{N_{C_2}} M_{ii'} \sum_{j' = 1}^{N_{C_2}} M_{jj'} \left[ \sum_{k=1}^{N_{v_i}} \sum_{l=1}^{N_{v_j}} \left[ v_{k,k'}^{(i,j)} v_{l,l'}^{(j,j')} P_{k,l}^{(i,j)} - \sum_{k'=1}^{N_{v_i}} \sum_{l'=1}^{N_{v_j}} v_{k,k'}^{(i,i')} v_{l,l'}^{(j,j')} P_{k,l}^{(i,i')} \right]^2 \right] \right]
\]

\( M_{ii'} = \) matched attribute function (across schemas), (Eq. (3.1))
\( N_{C_1}, N_{C_2} = \) Number of categorical attributes in Schemas 1 and 2, respectively
\( N_{v_i}, N_{v_j} = \) Number of values in \( i^{th} \) and \( i^{th} \) attributes in Schemas 1 and 2, respectively
\( V = \{v_{k,k'}^{(i,j)}\} \) is the value mappings (Eq. (3.2))
\( v_{k,k'}^{(i,i')} = 1 \) if \( k^{th} \) value in \( i^{th} \) attribute in Schema 1 matches to \( k^{th} \) value in \( i^{th} \) attribute in Schema 2, =0 otherwise
\( p_{k,l}^{(i,j)} = \) Joint probability of \( k^{th} \) value in \( i^{th} \) attribute co-occurring with \( l^{th} \) value in
$j^{th}$ attribute in Schema 1

Here, in Eq. (3.5), it is assumed $N_{C_1} \leq N_{C_2}$, i.e., the “onto matching” criterion. The term inside the outer bracket is the squared Euclidean distance between the pairwise probability mass function (pmf) \( \{P_{kl}^{(i,j)}\} \) 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 joint pmfs over all pairs of attributes from the first schema. Experimental evaluation in Section 3.3 shows that if the databases have a sufficiently large number of data records, such that accurate second-order pmf estimation is possible, second-order schema matching outperforms schema matching methods that utilize first-order statistics.

### 3.2.2 Matching and mapping strategy

Our global objective can now be formulated as a two dimensional minimization problem as shown in Eq. (3.6), where the resultant minimal score would correspond to the best (declared) schema matching/alignment and the corresponding best set of value mappings.

$$
D_{\text{Overall}}(M, V) = \min_{M \in \mathcal{M}} \left\{ \min_{V \in \mathcal{V}} [D^E(M, V)] \right\} 
$$

(3.6)

Where:
- $D_{\text{Overall}}$ = minimum dissimilarity score obtained
- $M$ = a fixed schema match (Eq. (3.1))
- $\mathcal{M}$ = the search space of different schema matches
- $V$ = a fixed value mapping for a fixed schema match (Eq. (3.2))
- $\mathcal{V}$ = the search space of different value mapping for a fixed schema match
- $D^E$ = dissimilarity metric utilized ($D^{EU}$ / $D^{RE}$)

As Eq. (3.6) 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 categorical 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.

### 3.2.3 Heuristic search strategy

In order to make the algorithm computationally tractable, we implemented a local minimization based on Eq. 3.6. We describe this approach utilizing the first-order dissimilarity metric.

Eq. 3.6 and Eq. 3.3 can be further rewritten as shown in Eq. 3.7. 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. Note that to compute the VMM elements requires the minimization over the value mapping dimension. We discuss a heuristic optimal value mapping strategy based on sorting pmfs in Section 3.2.5. VMM computation over all pairs of attributes in the two schemas 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 [12] (described in Section 3.2.4 and depicted in Figure 3.1) starting from an initial schema match. Schema match initialization is discussed in Section 3.3.2. The search space is limited by performing a finite number of two-opt switches. Furthermore, a sub-optimal 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 [40 55] on the Value Mapping Minimization (VMM) matrix. In addition, minimization via the Hungarian algorithm is possible for the first-order algorithm, however, it is not possible to extend this technique 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.
Based on the VMM matrix, we can re-express Eq. (3.6) as:

\[
D_{\text{Overall}}(M,V) = \min_{M \in \mathcal{M}} \left\{ \sum_{i=1}^{N_{C_1}} \sum_{j=1}^{N_{C_2}} M_{ij} \left[ \min_{V_{ij} \in V_{ij}} VMM(i,j) \right] \right\} \tag{3.7}
\]

Where:
- \(D_{\text{Overall}}\) = minimum dissimilarity score obtained
- \(M\) = a fixed schema match
- \(\mathcal{M}\) = the search space of different schema matches
- \(V_{ij}\) = a fixed value mapping for an attribute pair (i.e. fixed \(i, j\))
- \(V_{ij}\) = the search space of different value mapping for a fixed attribute pair
- \(D_{AB}^E\) = dissimilarity metric utilized (currently \(D^{EU}\))
- \(VMM(i,i')\) = Euclidean distance between probability mass functions associated with the best value mapping for the \((i, i')\) matching attribute pair

### 3.2.4 Two-opt switching

Two-opt switching is a simple local search algorithm that we utilize in our heuristic search strategy (Section 3.2.3). 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 3.1. Algorithm 2 presents the pseudocode for the overall heuristic search strategy for attribute matching (as discussed in Section 3.2.3) when using two-opt switching and the first order dissimilarity metric model based on Euclidean distance.
Algorithm 2 Pseudocode for the heuristic search strategy when using two-opt switching.

```
best_match $\leftarrow$ get Initial Schema Match()
D_{Overall} $\leftarrow$ $D_{EUIL}(\text{best_match})$
repeat
    new_match $\leftarrow$ Two-Opt-Switch(best_match)
    if $D_{EUIL}(\text{best_match}) > D_{EUIL}(\text{new_match})$ then
        best_match $\leftarrow$ new_match
        D_{Overall} $\leftarrow$ $D_{EUIL}(\text{best_match})$
    end if
until no further improvement or a specified number of iterations
```

3.2.5 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, j) (and thus how we compute the scores in the VMM matrix). We do so in the following way. Given a fixed candidate pair (i, j) we separately sort the two pmfs $\{P_{i}^{(k)}\}$ and $\{P_{j}^{(l)}\}$ and map the value with the highest probability in the first sorted pmf to the 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 $p_{i}$ and $q_{j}$. 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 $\mu$ map the $i^{th}$ element of $X_{o}$ to the $i^{th}$ element of $Y_{o}$. The mapping $\mu$ minimizes the squared Euclidean distance (Eq. (3.3)), i.e., for any value mapping $\phi$ between the values in X and those in Y, the Euclidean distance between pmfs using $\phi$ is greater than or equal to the distance corresponding to $\mu$.

**Proof.** Let X and Y be two discrete random variables with probability mass functions $P(X = x_{j}) = p_{j}$ and $P(Y = y_{j}) = q_{j}$, $j = 1, \ldots, n$.

Let there be a value mapping $\mu_{1}$ 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 $\mu_{1}$ 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 $\mu_1$ 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 $\mu_1$.

Consider the mapping $\mu_2$ that maps $x_1, x_2$ to $y_2$ and $y_1$ respectively, with the rest of the value mappings the same as $\mu_1$. Let $D_2$ be the squared Euclidean distance corresponding to the value mapping $\mu_2$.

We have (for some $D_0$)

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

$$= D_0 + (p_1 - q_1)^2 + (p_2 - q_2)^2 \quad (3.8)$$

and

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

$$= D_0 + (p_1 - q_2)^2 + (p_2 - q_1)^2 \quad (3.9)$$

Subtracting, Eq. (3.9) from Eq. (3.8),

$$D_1 - D_2 = -(2p_1q_1 + 2p_2q_2) + (2p_1q_2 + 2p_2q_1) \quad (3.10a)$$

$$= 2p_1(q_2 - q_1) + 2p_2(q_1 - q_2) \quad (3.10b)$$

$$= 2(p_1 - p_2)(q_2 - q_1) \quad (3.10c)$$

$$\therefore p_1 > p_2 \Rightarrow (p_1 - p_2) > 0 \quad (3.10d)$$

$$q_1 < q_2 \Rightarrow (q_2 - q_1) > 0 \quad (3.10e)$$

$$\therefore D_1 - D_2 > 0 \quad (3.10f)$$

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.
Table 3.1: Brief description of the Census Bureau PUMS 5% dataset for year 1990 for the states of New York, California and Texas

<table>
<thead>
<tr>
<th>State Name</th>
<th># Attributes (n)</th>
<th>Row Size (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>241</td>
<td>960K</td>
</tr>
<tr>
<td>California</td>
<td>241</td>
<td>1500K</td>
</tr>
<tr>
<td>Texas</td>
<td>241</td>
<td>935K</td>
</tr>
</tbody>
</table>

3.3 Experimental validation

In this section, we present the results of second-order pmf algorithms for schema matching on categorical attributes compared with several alternative methods on real-world Census Bureau datasets.

3.3.1 Categorical-valued dataset

We used the Census Bureau Public Usage Micro data Sample (PUMS) 5% datasets for the states of California (CA), New York (NY) and Texas (TX) for the year 1990. The common schema describing each of these tables consists of 241 attributes. Table 3.1 summarizes these datasets. First, we chose 50 attribute pairs from the two tables, selecting 50 attributes that are close in entropy. 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. 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. We chose $K = 20$ and averaged matching accuracy results over 50 trials where, in each trial the columns to be matched were randomly chosen ($K$ categorical-valued out of a total of 50 categorical-valued attributes closest in entropy). These 50 trial sets comprised our categorical-valued attribute data sets and were used to evaluate the second-order pmf-based dissimilarity metric discussed in Section 3.2.1.2.
3.3.2 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 two starting initial schema matches: (1) Random Initialization: a random schema match is chosen, and (2) Ground-truth Initialization: the ground-truth schema match.

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 [39]) (MI), ii) between conditional entropies, for matching attribute pairs, (CE) and iii) between entropies, for a pair of matching attributes (EN). All three of these methods are consistent with the Kang-Naughton approach. The first method explicitly minimizes the objective proposed by Kang and Naughton [38, 39] (For details of our implementation of this approach, please refer to Appendix C)

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 subset of $n$ ($2 \leq n \leq 20$) categorical attributes as discussed in Section 3.3.1. We randomly permuted the columns in the second schema to ensure that the algorithms cannot trivially stumble upon the correct schema matching result.

3.3.3 Evaluation metrics

To measure the accuracy of the match result, we use $\text{Precision} = \frac{m}{n}$, 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.

3.3.4 Monotonic nature of the algorithm

Figure 3.2.1 shows the evolution of the dissimilarity score as a function of accepted iterations when 10 attributes were matched between the two schemas. An accepted
3.2.1 Dissimilarity Score vs. Accepted Iteration

3.2.2 Precision vs. Accepted Iteration

Figure 3.2: Second-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.

Iteration is a schema match obtained by accepting a two-opt switch, i.e., a switch which reduced the objective function. Fig. 3.2.2 shows that the dissimilarity score (Eq. (3.5)) decreases monotonically with an increase in the number of algorithm steps. As Eq. (3.5) decreases, there is also a general trend of increasing precision (though not strictly monotonically).

3.3.5 One-to-one schema matching

3.3.5.1 Random initialization

Figure 3.3 presents the results of bijective (one-to-one) matching for the categorical dataset where the initial schema match was randomly selected. It can be observed from Figure 3.3 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. Figs. 3.3.1 and 3.3.3 show the match accuracy for the second-order pmf based dissimilarity metric [34] (labeled SO) for the CA vs. TX and CA vs. NY datasets, respectively, in comparison with other methods. As discussed in Section 3.3.2 we increased the number of attributes in the two input tables from 2 to 20. The attributes were randomly chosen from a set of 30 columns closest in entropy as described in Section 3.3.1. For each schema, we randomly chose 250K tuples. We repeated the experiment 50 times while randomly...
3.3.1: Average Precision for CA vs. TX dataset

3.3.2: Computation Time for CA vs. TX dataset

3.3.3: Average Precision for CA vs. NY dataset

3.3.4: Computation Time for CA vs. NY dataset

Figure 3.3: Schema matching results for categorical attributes when randomly initialized

choosing attributes from the set of 30 columns and measured the average precision over the 50 experiments. For comparison, the first-order pmf (FO), Entropy only (EN), Mutual Information (MI) and Conditional Entropy (CE) algorithm results are presented. All methods were initialized from the same random schema match.

The second-order pmf based method performs better than all other methods for all attribute sizes being matched for both datasets as illustrated in Figs. 3.3.1 and 3.3.3. Because the second-order method exploits statistical dependencies between columns, provided that there is accurate frequency-count estimation of the second-order pmfs, the second-order method should perform best and this is borne out by our results. For example, for the CA vs. TX dataset the second-order pmf based method performed at least 10% better than the first-order pmf method when greater than 10 columns were being matched. The first-order pmf method had acceptable matching accuracy (≥ 60%) while Mutual Information (MI), Con-
### 3.3.5.1 Computational time

Figs. 3.3.2 and 3.3.4 illustrate the computation time for executing the schema matching algorithms for the three state files when tackling categorical datasets. As expected, the second-order pmf (SO) based dissimilarity algorithm was the most expensive due to the increased computation required for calculating the second-order statistics. The first-order pmf (FO) and entropy-only algorithms (EN) have the least computation time.

### 3.3.5.3 Data sampling effects

Figs. 3.4.1 and 3.4.2 illustrate the effect of row sizes on matching accuracies produced during schema matching for the second-order pmf based schema matching method. The match accuracies for 50K tuples (labeled “SO 50K”), 100K tuples (labeled “SO 100K”) and 250K tuples (labeled “SO 250K”) are shown for the second-order pmf-based dissimilarity metric. For comparison, the match accuracies for first-order ("FO 250K") and MI ("MI 250K") methods generated using 250K tuples is also presented. The second-order pmf-based metric performed better than both the first-order pmf and MI based methods under all conditions. Interestingly (and somewhat counterintuitively), for the CA and NY dataset, the match accuracy was
3.5.1: Average Precision for CA vs. TX dataset

3.5.2: Average Precision for CA vs. NY dataset

Figure 3.5: Schema matching results for categorical attributes when initialized from ground-truth.

highest when 50K tuples were used for schema matching while for the CA and TX dataset the match accuracy of the second-order method improved with the number of K-tuples used.

3.3.5.4 Ground-truth initialization

We next demonstrate that the objective function of the first-order and second-order dissimilarity algorithms 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 3.5 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 20. The attributes themselves were randomly chosen from a set of 30 columns as described in Section 3.3.1. The two input datasets were generated by randomly selecting 250K tuples from the two datasets. We repeated the experiment 50 times while randomly choosing attributes from the set of 30 columns. We measured the average precision over the 50 experiments for both the CA vs. TX (Figure 3.5.1) and CA vs. NY datasets (Figure 3.5.2).
For both datasets, the second-order and first-order dissimilarity-metric based algorithms converged to a solution closer to the correct solution in all 50 experimental trials. The entropy-only algorithm performed the worst in both cases. Kang-Naughton Mutual Information algorithm did not always converge to the correct schema match unlike our dissimilarity metric algorithms. This illustrates that minimizing the objective function of the second-order dissimilarity algorithm is much more consistent with the matching goal than the objective functions of the other methods.
4.1 Introduction

As discussed in the previous chapter, the pmf-based schema matching is suitable for categorical-valued attributes. However, since continuous-valued attributes are naturally characterized by probability density functions (pdf), this approach cannot be directly applied in matching such attributes. However, as discussed in Chapter 1 there are two plausible metrics that could be used for matching continuous-valued attributes, namely:

- The Log-likelihood metric, which measures dissimilarity between two attributes as the negative log-likelihood of values in one attribute according to the other attribute’s density function, i.e., it measures a goodness of fit between two attributes.

- The Euclidean distance metric, which measures dissimilarity between two attributes as the Euclidean distance between mixture component pmfs for the two continuous-valued attributes.

Note that since the mutual information for two continuous-valued attributes $X$ and $Y$ is defined as,

$$I(X; Y) = \int_X \int_Y p(x, y) \log \frac{p(x, y)}{p(x)p(y)},$$

(4.1)
one could extend the Kang-Naughton [39] to handle continuous-valued attributes but this technique would require accurate estimation of both univariate and bivariate density models (e.g., GMM’s). Our experimental results, discussed in Section 4.3.5, will conclusively demonstrate that the Log-likelihood metric is best suited for matching continuous-valued attributes.

4.2 The Schema Matching Framework

In this section, we describe our uninterpreted schema matching with embedded value mapping algorithms for continuous-valued attribute spaces. The inputs to our algorithms are two tables while the output is a schema match. Our algorithms perform iterative hill climbing on a global objective function based on a dissimilarity metric, as elaborated in this section. Suitable dissimilarity metrics (log-likelihood or Euclidean distance) for continuous-valued attribute spaces are discussed below.

4.2.1 Dissimilarity objectives for continuous-valued attributes

The dissimilarity metrics developed in this section provide an evaluation of how well the probability density functions (joint pdfs) align across attributes (attribute pairs) for a fixed schema match while embedding attribute (pairwise) value distributions.

4.2.1.1 First-order (FO) Continuous Dissimilarity Metrics

In the previous chapter, we introduced dissimilarity objectives which used Euclidean and relative entropy between pmfs for categorical-valued attributes. Continuous attributes, however, must be characterized by pdfs, not pmfs. However, we can still derive pmfs for continuous-valued attributes and then utilize Euclidean distance for matching as hinted in Section 1.2.1. We describe in detail how pmfs are derived in what follows.

Consider matching two continuous-valued attributes $X$ and $Y$, from different schemas. Let attribute $X$, with a set of observations $\mathcal{X} = \{x_1, \ldots, x_{N_x}\}$, be modeled by a mixture model given by $f_X(x; \theta^{(x)}) = \sum_{k=1}^{K_x} \alpha_k^{(x)} f_{X|k}(x, \theta_k^{(x)})$, where $K_x$ is
the number of components, \( \theta_k^{(x)} = (\mu_k^{(x)}, \sigma_k^{(x)}) \) are the parameters, \( \mu_k^{(x)} \) is the mean and \( \sigma_k^{(x)} \) is the standard deviation of the \( k \)th component respectively, \( \alpha_k^{(x)} \geq 0 \forall k \) and \( \sum_k \alpha_k^{(x)} = 1 \). We can then write the a posteriori probability of each mixture component, given a data sample \( x_i \) as:

\[
P[C_x = k|x_i; \theta^{(x)}] = \frac{\alpha_k^{(x)} f_{X|k}(x_i; \theta_k^{(x)})}{\sum_{j=1}^{K_x} \alpha_j^{(x)} f_{X|j}(x_i; \theta_j^{(x)})}, \quad K = 1, \ldots, K_x. \tag{4.2}
\]

Let attribute \( Y \), with a set of observations \( Y = \{y_1, \ldots, y_{N_y}\} \), be modeled by a mixture model given by

\[
f_Y(y; \theta^{(y)}) = \sum_{k=1}^{K_y} \beta_k^{(y)} f_{Y|k}(y; \theta_k^{(y)}),
\]

where \( K_y \) is the number of components, \( \theta_k^{(y)} = (\mu_k^{(y)}, \sigma_k^{(y)}) \) are the parameters, \( \mu_k^{(y)} \) is the mean and \( \sigma_k^{(y)} \) is the standard deviation of the \( k \)th component respectively, \( \beta_k^{(y)} \geq 0 \forall k \) and \( \sum_k \beta_k^{(y)} = 1 \). We can then estimate the component masses for \( Y \) under \( X \)’s density model, which we will use in defining one of our dissimilarity objectives, as:

\[
\hat{\alpha}_k^{(y)} = \frac{1}{N_y} \sum_{n=1}^{N_y} P[C_x = k|y_n; \theta^{(x)}], \quad k = 1, \ldots, K_x. \tag{4.3}
\]

Likewise, we can estimate the component masses for \( X \) under \( Y \)’s density model as,

\[
\hat{\beta}_k^{(x)} = \frac{1}{N_x} \sum_{n=1}^{N_x} P[C_y = k|x_n; \theta^{(y)}], \quad k = 1, \ldots, K_y. \tag{4.4}
\]

Thus, given a schema match matrix \( M \) defined as,

\[
M = [M_{ij}], \quad M_{ij} = \begin{cases} 
1 & \text{if } i^{th} \text{ attribute matches to } j^{th} \text{ attribute across Schemas 1 and 2} \\
0 & \text{otherwise}
\end{cases}
\]

we can define a dissimilarity metric utilizing Euclidean distance based on (mixture model) derived pmfs for continuous-valued attributes as,

\[
D^{EP}(M) = \sum_{i=1}^{N_{C_1}} \sum_{j=1}^{N_{C_2}} M_{ij} \left[ \sum_{k=1}^{K_x} (\alpha_k^{(i)} - \hat{\alpha}_k^{(i)})^2 + \sum_{k=1}^{K_y} (\beta_k^{(j)} - \hat{\beta}_k^{(j)})^2 \right], \tag{4.6}
\]

where “EP” stands for “Estimated PMF” and where,
$N_{C_1}, N_{C_2} =$ Number of attributes (here assumed to be all continuous-valued) in Schema 1 and 2, respectively.

A second dissimilarity metric between two continuous-valued attributes can be measured based on the negative data log-likelihood. Intuitively, a model best fit to observed data will have least negative data log-likelihood. Assuming the observations $\mathcal{Y} = \{y_i, i = 1, \ldots, N_y\}$ are generated independent and identically distributed, the negative log-likelihood of $\mathcal{Y}$ under $X$'s density model, which we will use to define the negative log-likelihood based dissimilarity objective, is:

$$- \log L(\mathcal{Y}; \theta^{(x)}) = - \sum_{n=1}^{N_y} \sum_{k=1}^{K_x} \log \alpha^{(x)}_k f_{X|k}(y_n; \theta^{(x)}_k)$$  \hspace{1cm} (4.7)

Thus, given a schema match matrix $M$ (as defined in Eq. (4.5)), a dissimilarity metric based on negative log-likelihood for continuous-valued attributes can be defined as,

$$D^{LL}(M) = - \sum_{i=1}^{N_{C_1}} \sum_{j=1}^{N_{C_2}} M_{ij} \left[ \sum_{n=1}^{N_x} \log \sum_{k=1}^{K_j} \beta^{(j)}_k f_{Y|k}(x_n; \theta^{(j)}_k) + \sum_{n=1}^{N_y} \log \sum_{k=1}^{K_i} \alpha^{(i)}_k f_{X|k}(y_n; \theta^{(i)}_k) \right]$$ \hspace{1cm} (4.8)

where,

$N_{C_1}, N_{C_2} =$ Number of attributes (here assumed to be all continuous-valued) in Schema 1 and 2, respectively

$N_x, N_y =$ Number of rows in Schema 1 and 2, respectively

In Eqs. (4.6) and (4.8) we have assumed that $N_{C_1} \leq N_{C_2}$, i.e., each continuous-valued column in Schema 1 matches to a unique column in Schema 2. This is equivalent to “onto matching”. The negative log-likelihood metric (Eq. (4.8)) outperforms the Euclidean distance metric (Eq. (4.6)) for databases with purely continuous-valued attributes, as will be shown in Section 4.3.5.

4.2.1.2 Second-order continuous dissimilarity metrics

Finer-grained schema matching with embedded value mapping can be achieved by utilizing second-order statistics. The second-order dissimilarity objective for
continuous-valued attributes measures how well the pairwise pdfs (joint densities) for all pairs of continuous-valued attributes from one schema align for a fixed schema match with those of another table. We can extend the two metrics discussed in the previous section for matching continuous-valued attribute pairs. The two metrics are:

- The *Log-likelihood* metric, which measures dissimilarity between two attribute pairs as the negative log-likelihood of values in one attribute pair according to the other attribute pair’s density function, i.e., it measures a goodness of fit between two attribute pairs.

- The *Euclidean distance* metric, which measures dissimilarity between two attribute pairs as the Euclidean distance between mixture component pmfs for the two continuous-valued attribute pairs.

Joint densities (pairwise pdfs) will capture and thus exploit statistical dependencies that may exist between all pairs of continuous-valued attributes. To develop these metrics, we utilize the GMM density function for a multivariate random variable $X$ which is defined as:

$$f_X(x; \theta) = \sum_{j=1}^{K} P[C = j] f_{X|M=j}(x; \theta_j) = \sum_{j=1}^{K} \alpha_j f_{X|j}(x; \theta_j), \quad (4.9)$$

where $C$ is a discrete component random variable, $\theta_j = (\mu_j, \Sigma_j)$ with $\mu_j$ the mean and $\Sigma_j$ the covariance, and with $f_{X|j}(x; \theta_j)$ the $j^{th}$ component Gaussian density function given by:

$$f_{X|j}(x; \theta_j) = \frac{1}{2\pi|\Sigma_j|^{1/2}} e^{-(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)}. \quad (4.10)$$

Now, suppose that $X^{(ij)}$ denote the $i^{th}$ and $j^{th}$ attributes (i.e., an attribute pair) from one schema with a set of observations (data entries) $X^{(ij)} = \{x_n^{(ij)}\}_{n=1}^{N_y'}$, where $x_n^{(ij)} = [x_n^{(i)} x_n^{(j)}]^T$, $\forall n = 1, \ldots, N_y$. Let, $Y^{(i'j')}$ denote the $i^{th}$ and $j^{th}$ attributes from the second schema with a set of observations (data entries) $Y^{(i'j')} = \{y_n^{(i'j')}\}_{n=1}^{N_y}$, where $y_n^{(i'j')} = [y_n^{(i')} y_n^{(j')}]^T$, $\forall n = 1, \ldots, N_y$. Now, suppose that $X^{(ij)}$ and $Y^{(i'j')}$ represent attribute pairs that are candidates for being matched.
Let \( X^{(ij)} \) be modeled by a mixture model given by \( f_X(x^{(ij)}; \theta^{(ij)}) = \sum_{k=1}^{K_{ij}} \alpha_k^{(ij)} f_{X|k}(x^{(ij)}; \theta_k^{(ij)}) \), where \( K_{ij} \) is the number of components, \( \theta_k^{(ij)} = (\mu_k^{(ij)}, \Sigma_k^{(ij)}) \) are the parameters, \( \mu_k^{(ij)} \) is the mean and \( \Sigma_k^{(ij)} \) is the covariance of the \( k^{th} \) component respectively, \( \alpha_k^{(ij)} \geq 0 \ \forall k \) and \( \sum_k \alpha_k^{(ij)} = 1 \). We can then write the a posteriori probability of each mixture component, given a data sample \( x_n^{(ij)} \) as:

\[
P(C_{ij} = k | x_n^{(ij)}; \theta^{(ij)}) = \frac{\alpha_k^{(ij)} f_{X|k}(x_n^{(ij)}; \theta_k^{(ij)})}{\sum_{i=1}^{K_{ij}} \alpha_i^{(ij)} f_{X|j}(x_n^{(ij)}; \theta_j^{(ij)})}, \quad K = 1, \ldots, K_{ij}. \quad (4.11)
\]

Let attribute \( Y^{(i'j')} \) be modeled by a mixture model given by \( f_Y(y; \theta^{(i'j')}) = \sum_{k=1}^{K_{i'j'}} \beta_k^{(i'j')} f_{Y|k}(y; \theta_k^{(i'j')}) \), where \( K_{i'j'} \) is the number of components, \( \theta_k^{(i'j')} = (\mu_k^{(i'j')}, \sigma_k^{(i'j')}) \) are the parameters, \( \mu_k^{(i'j')} \) is the mean and \( \Sigma_k^{(i'j')} \) is the covariance of the \( k^{th} \) component respectively, \( \beta_k^{(i'j')} \geq 0 \ \forall k \) and \( \sum_k \beta_k^{(i'j')} = 1 \). We can then estimate the component masses for \( Y^{(i'j')} \) under \( X \)'s density model, which we will use in defining one of our dissimilarity objectives, as:

\[
\hat{\alpha}_k^{(i'j')} = \frac{1}{N_{i'j'}} \sum_{n=1}^{N_{i'j'}} P(C_{ij} = k | y_n; \theta^{(ij)}) \quad k = 1, \ldots, K_{ij}. \quad (4.12)
\]

Likewise, we can estimate the component masses for \( X^{(ij)} \) under \( Y \)'s density model as,

\[
\hat{\beta}_k^{(ij)} = \frac{1}{N_{ij}} \sum_{n=1}^{N_{ij}} P(C_{i'j'} = k | x_n; \theta^{(i'j')}) \quad k = 1, \ldots, K_{i'j'}. \quad (4.13)
\]

Thus, given a schema match matrix \( M \) defined in Eq. [4.5] we can define a dissimilarity metric utilizing Euclidean distance based on (mixture model) derived pmfs for continuous-valued attribute pairs as,

\[
D^{EP}(M) = \sum_{i=1}^{N_C1} \sum_{j=1}^{N_C2} \sum_{i' \neq i} \sum_{j' \neq j} M_{iw} M_{jj'} \left[ \sum_{k=1}^{K_{ij}} (\alpha_k^{(ij)} - \hat{\alpha}_k^{(i'j')})^2 + \sum_{k=1}^{K_{i'j'}} (\beta_k^{(i'j')} - \hat{\beta}_k^{(ij)})^2 \right],
\]

where,

\( N_{C1}, N_{C2} = \) Number of attributes (here assumed to be all continuous-valued) in Schema 1 and 2, respectively
A second dissimilarity metric between two continuous-valued attribute pairs can be measured based on the negative data log-likelihood which is the natural second-order extension of the first-order log-likelihood metric discussed in Section 4.2.1.1. Assuming the observations \(Y(i'j') = \{y_1(i'j'), \ldots, y_{N_y}(i'j')\}\) are generated independent and identically distributed, the negative log-likelihood of \(Y(i'j')\) under \(X(ij)\)'s density model, which we will use to define the negative log-likelihood based dissimilarity objective, is:

\[
- \log L(Y; \theta(ij)) = - \sum_{n=1}^{N_{i'j'}} \log \sum_{k=1}^{K_{ij}} \alpha_{k(ij)} \cdot f_{X|k}(y_{n}(i'j'); \theta_{k(ij)})
\] (4.15)

Thus, given a schema match matrix \(M\) (as defined in Eq. (4.5)), a dissimilarity metric based on negative log-likelihood for continuous-valued attributes can be defined as,

\[
D_{LL}(M) = - \sum_{i=1}^{N_{C_1}} \sum_{j=1}^{N_{C_2}} \sum_{i' \neq i} \sum_{j' \neq j} M_{ii'} M_{jj'} \left[ \sum_{n=1}^{N_x} \log \sum_{k=1}^{K_{ij'}} \beta_{k(i'j')} \cdot f_{Y|k}(x_{n}; \theta_{k(i'j')}) \right] + \sum_{n=1}^{N_y} \log \sum_{k=1}^{K_{ij}} \alpha_{k(ij)} \cdot f_{X|k}(y_{n}; \theta_{k(ij)})
\]

(4.16)  \(4.17\)

where,

\(N_{C_1}, N_{C_2} = \) Number of attributes (here assumed to be all continuous-valued) in Schema 1 and 2, respectively
\(N_x = N_{ij}, N_y = N_{i'j'} = \) Number of rows in Schema 1 and 2, respectively

In Eqs. (4.14) and (4.17) we have assumed that \(N_{C_1} \leq N_{C_2}\), i.e., each continuous-valued column in Schema 1 matches to a unique column in Schema 2. This is equivalent to “onto matching”.

### 4.2.2 Schema matching strategy

Our algorithm’s goal is minimization of the dissimilarity objective (Eqs. (4.6), (4.8), (4.14), or (4.17)) over the space of schema matches, i.e., minimization of the
global objective over all possible pairs of attributes (or attribute pairs) across the
two schemas. However, running a naïve exhaustive search to achieve a global mini-
imization is computationally infeasible since the total number of schema matches
that must be evaluated is $n!$, where $n$ is the number of attributes being matched.
In order to make the optimization computationally tractable, we implemented a lo-
cal minimization strategy. Our heuristic method first pre-computes a dissimilarity
cost matrix where each element corresponds to the dissimilarity (log-likelihood/
Euclidean-distance metric) associated with each attribute in Schema 1 paired to
every attribute in Schema 2. Then, the algorithm iterates over the schema-match
search space starting from an initial schema-match, with the objective of mini-
mizing the global dissimilarity score obtained based on this cost matrix. Improved
schema matches are obtained using two-opt switching [12] (depicted in Figure 3.1),
starting from an initial schema match. Here, at each step, one chooses a pair of
matches (A,P) and (B,Q) and exchanges the matching attributes between the pair
giving matches (A,Q) and (B,P). The exchange is retained if it reduces the global
dissimilarity objective, i.e., 2-opt switching performs iterative hill-climbing on the
global cost objective. Initial schema match choices are discussed in Section 4.3.2.
The search is practically limited by performing a finite number of two-opt switches.

4.3 Experimental validation

In this section, we present the results of our algorithms for schema matching on
continuous-valued attributes compared with several alternative methods on real-
world Census Bureau datasets.

4.3.1 Continuous-valued attributes dataset

We used the Census Bureau Public Usage Micro data Sample (PUMS) 5% datasets
for the states of California (CA), New York (NY) and Texas (TX) for the year
1990. The common schema describing each of these tables consists of 241 at-
ttributes. Table 3.1 summarizes these datasets. In the Section 3.3.5 it was shown
(unsurprisingly) that the accuracy of schema matching algorithms degrades as the
number of columns to be matched increases. Thus, for example, matching all 241
attributes would likely yield very poor accuracy for all methods and not allow us to clearly distinguish the relative performance of evaluated methods. Thus, in our experiments we worked with an attribute subset sufficiently large so as to make the matching problem nontrivial, while not so large that the matching accuracies of all methods will be poor. We chose $K = 20$ and averaged matching accuracy results over 50 trials where, in each trial the columns to be matched were randomly chosen ($K$ continuous-valued out of a total of 26 continuous-valued attributes). These 50 trial sets comprised our continuous-valued attribute data sets and were used for the evaluation of the methods developed in Section 4.2.1.1

### 4.3.2 Experimental setup

The schema matching with embedded value mapping algorithm is called to perform schema matching on continuous-valued attributes. The matching algorithm is initialized from one of the following three starting initial schema matches: (1) Random Initialization: a random schema match is chosen, and (2) Ground-truth Initialization: the ground-truth schema match, and (3) Fixed initialization: a schema match where the same column index was assumed as a match, i.e., column 1 in schemas A and B are assumed as matching and so on. We compared our schema matching algorithms against four methods that, respectively, do two-opt switching of attributes to minimize squared Euclidean distance between i) pmfs, for a pair of matching attributes (The objective function in Eq. (3.3)) ii) mutual informations, for matching attribute pairs (The objective function in Kang and Naughton [39, 38]) (MI), iii) conditional entropies [34], for matching attribute pairs, (CE) and iv) entropies, for a pair of matching attributes (EN). The last three methods are consistent with the Kang and Naughton [39, 38] approach in that these three cost function do not exploit (and require) embedded value mappings, in matching a pair of (mixture model) discrete-valued attributes. The second method explicitly minimizes the objective function proposed by Kang and Naughton (For details of our implementation of this approach, please refer to Appendix C). Since these methods assume categorical-valued attributes, derived pmfs for all continuous-valued attributes (using mixture component a posteriori probabilities) were input to these methods. All matching algorithms are initialized from the same schema match.
4.1.1: BIC Scores for the “Age” attribute for the TX dataset when a) all components had same tied variance (E), and b) component variances were freely adjustable parameters (V)

4.1.2: BIC Scores for the “Age” attribute for the TX dataset when a) all components had same tied variance (E), and b) component variances were freely adjustable parameters (V)

4.1.3: Density Estimate for the “Age” attribute for the CA dataset

4.1.4: Density Estimate for the “Age” attribute for the TX dataset

4.1.5: Histogram for the “Age” attribute for the CA dataset

4.1.6: Histogram for the “Age” attribute for the TX dataset

Figure 4.1: BIC scores for different model sizes, density estimates and histograms for the “Age” attribute for the CA and NY Datasets. The best BIC score for the “Age” attribute for the CA and TX datasets were obtained for a 5-component GMM and 4-component GMM, respectively, when component variances were freely adjustable parameters.
4.2.1: Dissimilarity Score vs. Accepted Iteration  
4.2.2: Precision vs. Accepted Iteration

Figure 4.2: First-order dissimilarity metric (negative Log-likelihood) and precision as a function of accepted two-opt switches for one-to-one matching of 10 attributes, starting from a random initialization.

We implemented our schema matching with embedded value mapping algorithms using R [61] 2.11 and MCLUST 3.4 [23, 21, 22]. Fig. 4.1 presents the Bayesian Information Criterion (BIC) scores (see Appendix B) for different model sizes, the estimated pdf and the histogram for the “Age” attribute from the CA and TX datasets when 50K random rows were sampled from both datasets. Five and four component model sizes, respectively, minimized BIC, assuming all components have different variances. Since the density estimation and the log-likelihood algorithm are computationally intensive, they were executed on a 224 node (Dual Intel® Xeon @ 2.67GHz) cluster. Up to 10% of the cluster CPU resources were allocated by the cluster scheduler to execute the algorithms.

We ran our experiments over a randomly chosen subset of \( n \) (\( 2 \leq n \leq 20 \)) continuous-valued attributes. We randomly permuted the columns in the second schema to ensure that the algorithms cannot trivially stumble upon the correct schema matching result.

4.3.3 Evaluation Metrics

We use \( \text{Precision} = \frac{m}{n} \) to measure the accuracy of matching, where \( n \) is the total number of columns to match and \( m \) is the total number of correct matches.
4.3.4 Monotonic Nature of the Algorithm

Figure 4.2.1 shows the evolution of the dissimilarity score (Eq. (4.8)) as a function of accepted iterations when 10 attributes were matched between the two schemas. Both schemas, in this case, contained purely continuous-valued attributes. An accepted iteration is a schema match obtained by accepting a two-opt switch, i.e., a switch that decreases the objective function. Accordingly, Fig. 4.2.1 shows that the dissimilarity score (Eq. (4.8)) decreases monotonically with the number of algorithm steps. As the dissimilarity score (Eq. (4.8)) decreases, generally the precision increases (though not strictly monotonically).
4.3.5 One-to-one schema matching for continuous-valued attributes

4.3.5.1 Random initialization

Fig. 4.3 presents schema matching results for continuous attributes datasets where the initial schema match was randomly selected. All algorithms were initialized from the same initial schema match. We increased the number of attributes in the two input tables from 2 to 20. The attributes for the continuous-valued datasets were randomly chosen from the set of 26 columns. For each pair of schemas, we first randomly chose 50K rows (samples). For each of these sub-datasets, we repeated the experiment 50 times, each time randomly choosing attributes from the set of 26 (continuous). Fig. 4.3 shows the average precision (over 50 trials) for Euclidean PMF matching (FO EP) and the log-likelihood (FO LL) metrics for continuous attributes datasets. For comparison, the first-order discrete Euclidean metric (FO DM) (Eq. (3.3)) [34], the Entropy only (EN), Kang-Naughton Mutual Information (MI) [39] and modified Kang-Naughton algorithm (labeled CE) results are also presented.

Figs. 4.3.1 and 4.3.3 show the precision for the CA vs. NY and CA vs. TX continuous-valued attributes datasets, respectively. Clearly, the FO LL dissimilarity algorithm performs best ($\approx 10\%$ better accuracy over the next best method when $\geq 10$ columns are matched). The FO EP algorithm performs second best. Both these methods perform much better in comparison to the other methods with typical precision gains being $\geq 25\%$ when $\geq 5$ columns are matched for both datasets. Mutual information, conditional entropy and entropy methods all perform poorly even though they are supplied with soft-discretized versions of the dataset. These methods have less than 20\% matching accuracy when $> 7$ attributes are matched. All methods that treat attributes as discrete and do not embed value mapping (e.g., MI, EN, CE) perform poorly in comparison to methods that embed value mappings (FO EP and FO DM). This observation demonstrates the benefit of embedding value mapping in determining each attribute match (Note that FO LL treats attributes as continuous-valued and does not require embedded value mappings). The first-order pmf-based dissimilarity (FO DM, Eq. (3.3)) [34] performs poorly, with schema matching precision ranging between 40\% and 50\%
Figure 4.4: Effect of rows on precision for first-order schema matching for continuous-valued attributes.

when \( \geq 10 \) columns are matched. This is consistent with the results of the earlier study \([33, 34]\), which involved only categorical-valued attributes. This method requires similar pmfs of the latent component random variables across two attributes being matched, which does not necessarily occur in real-world datasets. Moreover, FO DM requires embedded value mapping optimization, whereas FO LL and FO EP do not. Value mapping inaccuracy may be one significant source of error in FO DM relative to FO LL and FO EP. It can be observed from Figs. 4.3.1 and 4.3.3 that, as one would expect, match results for all methods are better for narrow tables (fewer attributes to match) in comparison to that for wider tables.

### 4.3.5.2 Computational time

Figs. 4.3.2 and 4.3.4 illustrate the computation time for executing the schema matching algorithms for the three state files when tackling continuous attributes datasets. The first-order Euclidean distance methods (FO EP and FO EU) and entropy-only algorithms (EN) have the least computation times for continuous-valued attributes. The FO LL dissimilarity algorithm was the most expensive. This increase in computation cost for the Log-Likelihood based algorithms is expected due to higher computation cost of calculating the log-likelihood score. In addition, a significant component of the computation time of all the schema matching algorithms is spent in learning the probability density functions for all the continuous-valued attributes in both schemas in a preprocessing step.
4.3.5.3 Data sampling effects

Figs. 4.4.1 and 4.4.2 illustrate the effect of the number of rows on matching accuracies produced during schema matching. The match accuracies for 10K tuples (labeled “FO EP 10K” and “FO LL 10K”), 25K tuples (labeled “FO EP 25K” and “FO LL 25K”), and, 50K tuples (labeled “FO EP 50K” and “FO LL 50K”) are shown for the two dissimilarity metrics when continuous-valued attributes were matched. The match accuracies were highest when \( \geq 50K \) tuples were used for schema matching and it can be observed that the schema matching accuracies improve with number of K-tuples used, as expected. Both methods have similar performance variations \( \approx 5\% \) as the number of K-tuples is varied. As in earlier experiments, the log-likelihood dissimilarity metric performed best (Figs. 4.4.1, 4.4.2).

4.3.5.4 Ground-truth initialization

We next demonstrate that the objective function of the first order dissimilarity algorithms for continuous-valued attributes captures the schema matching objective better than the other schema matching criteria (first-order pmf, 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?). Fig. 4.5.1 and Fig. 4.5.2 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 20. We chose 50K tuples from both matching schemas. The attributes themselves were randomly chosen from a set of 26 continuous-valued attributes as discussed in Section 4.3.1. We repeated the experiment 50 times while randomly choosing attributes from the set of 26 columns. We measured the average precision over the 50 experiments for both the California and New York (Fig. 4.5.1) and California and Texas demographics data sets (Fig. 4.5.2).

For both datasets, the entropy-only schema matching algorithms performs best. The log-likelihood (Eq. (4.8), labeled FO LL), estimated pmf (Eq. (4.6), labeled FO LP) and pmf-only (labeled FO DM) dissimilarity algorithms when initialized from ground-truth have similar performance as random initializations. While entropy-only matching had best performance, however the entropy-only algorithm

Figure 4.6: First-order dissimilarity scores for schema match search space for the California and New York Census datasets when 5 columns were matched. The ground-truth is iteration 97.
Figure 4.7: First-order dissimilarity scores for schema match search space for the California and Texas Census datasets when 5 columns were matched. The ground-truth is iteration 98.

does not have a unique global minimum. To illustrate this lack of a unique solution for entropy-only matching, Figs. 4.6 and 4.7 present the schema matching dissimilarity score for entropy-only, log-likelihood, estimated pmf and first-order pmf metrics as a function of the schema match space when 5 columns are matched using 50K tuples from both schemas. For example, for entropy-only matching $\geq 10\%$ of all possible schema match configurations have the same global minimum dissimilarity score, as illustrated in Figs. 4.6.1 and 4.7.1 which in no way ensures correct matching. Both estimated pmf (FO EP) (Figs. 4.6.3 and 4.7.3) and log-likelihood (FO LL) metric (Figs. 4.6.4 and 4.7.4) have one global minimum for both datasets (iteration 97 for CA and NY dataset and iteration 98 for CA and TX dataset) being matched. For this case, the global minimum for both these dissimilarity algorithms corresponded to ground-truth.
In the previous experiment illustrated in Section 4.3.5.1, a random trial meant a random selection of rows and initial solution. In this section, we evaluated schema matching accuracy for all algorithms when initialized from a fixed initial schema match. Figs. 4.8.1 and 4.8.2 present the results of schema matching from fixed schema initialization. The initial schema match is generated by assigning the same column index for attributes across schemas, i.e., column $i$ in schemas A and B is a matching attribute pair and so on. Two-opt switching was then performed to minimize the global objective. We ran the experiments while increasing the number of attributes in the two input tables from 2 to 20. We repeated the experiment 50 times for the same fixed initialization. In each experiment, 50K different tuples were randomly selected while the attributes were randomly chosen from a set of 26 continuous-valued attributes as discussed in Section 4.3.1. We measured the average precision over the 50 experiments for both the California and New York (Fig. 4.8.1) and California and Texas demographics data sets (Fig. 4.8.2). For comparison, the First-order pmf (labeled FO DM), Entropy-only (labeled EN), Kang-Naugton Mutual Information (labeled MI), and modified Kang-Naughton algorithm (labeled CE) results are also presented.

For both data sets, the schema matching accuracy for fixed initialization was similar to those for random initialization. Clearly, the FO LL dissimilarity algorithm performs best followed by the FO EP algorithm. Both these methods
perform much better in comparison to the other methods with typical precision gains being $\geq 25\%$ when $\geq 5$ columns are matched for both datasets. Mutual information, conditional entropy and entropy methods all perform poorly. This clearly shows that the first-order log-likelihood and estimated pmf dissimilarity algorithms are robust and largely unaffected by tuple selection.
Chapter 5

Schema Matching Criterion for Mixed Continuous and Categorical attributes

5.1 Introduction

In the previous chapters, we discussed dissimilarity objectives for schema matching for tackling categorical and continuous-valued attributes separately. However, real-world datasets typically contain a mix of both categorical and continuous-valued features. In some domains this fact may simplify schema matching, because one can perhaps then restrict to matching attributes of the same type (categorical with categorical and quantitative with quantitative). However, it is also possible that a quantitative attribute was quantized and encoded, prior to its inclusion in one of the databases. Therefore, one cannot necessarily assume that attributes should be strictly matched by type, i.e., a quantitative attribute in one schema could have a quantized, encoded (discrete) matching attribute in the other schema. To accommodate this possibility, one should in general also evaluate matches between categorical attributes in one schema and quantitative attributes in the other. Accordingly, one should have sound matching criterion suitable for matching attributes of different types. In this chapter, we address schema matching in the even more difficult case where some attributes are categorical and others are continuous-valued.
5.2 Schema matching framework

In this section, we describe our uninterpreted schema matching with embedded value mapping algorithms for mixed continuous-valued and categorical attribute spaces. The inputs to our algorithms are two tables. Our algorithms output a schema match, with associated value mappings for pairs of matched attributes that are both discrete-valued. Our algorithms perform iterative hill climbing on a global objective function based on a dissimilarity metric, as elaborated in this section. Suitable dissimilarity metrics (log-likelihood or Euclidean distance) for mixed continuous-valued and categorical attribute spaces are discussed below. We, first, introduce the log-likelihood for categorical data which we will use to define the log-likelihood based metric for mixed categorical and continuous-valued attributes.

5.2.1 Log-likelihood for discrete-valued data

Let $p$ be a probability mass function (pmf) over a set of finite cardinality defined by $1, \ldots, k$. Suppose, $n$ samples $x_1, \ldots, x_n$ are generated independently and identically distributed (i.i.d.) according to $p$. The pmf can then be estimated as:

\[ p(X = j) = \frac{1}{n} \sum_{t=1}^{n} 1(x_t = j) \equiv \frac{n_j}{n}, \quad j \in \{1, \ldots, k\} \tag{5.1} \]

where $n_j$, $j = 1, \ldots, k$, is the number of times the value $j$ appears in the sample, $n = \sum_{j=1}^{k} n_j$ and $1(.)$ is an indicator function. Furthermore, the probability of observing $x_1, \ldots, x_n$ is,

\[ p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i) = \prod_{j=1}^{k} p_j^{n_j}. \tag{5.2} \]

This is the likelihood function for discrete-valued data. The Log-Likelihood is then

\[ \log \prod_{j=1}^{k} p_j^{n_j} = \sum_{j=1}^{k} n_j \log p_j. \tag{5.3} \]
5.2.2 First-order (FO) Mixed Dissimilarity Metrics

In the previous chapters, we introduced dissimilarity measures suitable for either continuous-valued or categorical-valued attributes. Here we address schema matching in the even more difficult case where some attributes are categorical and others are continuous-valued. What makes this problem especially formidable is the fact that it may be unknown whether a discrete attribute is inherently categorical or whether it is the result of quantizing a continuous-valued attribute. Accordingly, in the most general setting, one must allow for candidate matches between all pairs of attributes from the two schemas, including both pairs of the same type (categorical with categorical, continuous with continuous) and also different types (continuous and categorical). Accommodation of all these types of matches requires some care in crafting a global matching objective function. Specifically, it may be natural to measure the negative log-likelihood distance between density functions for a candidate pair of matching continuous-valued attributes (as discussed in Section 4.2.1.1) and the Euclidean distance between probability mass functions for a candidate pair of matching categorical-valued attributes (as discussed in Section 3.2.1.1), respectively. However, for candidate matching of a continuous-valued and a categorical attribute, which metric (log-likelihood or Euclidean distance) is most suitable? Moreover, a global matching objective based on an unweighted sum of Euclidean distance terms (for discrete-valued attributes), negative log-likelihood distances (for continuous-valued ones) and a chosen metric for continuous-to-discrete matches might not give proper relative emphasis in the global objective to the three types of candidate matches. Some type of cross validation for tuning the coefficients of a weighted sum could be used, but such tuning may not be robust to different environments and contexts. Such tuning can in fact be reasonably avoided if the same dissimilarity function is used for all the attribute matches irrespective of whether the attributes in the pair are continuous-valued or categorical.

We thus introduce global schema matching objectives for mixed continuous and categorical attribute spaces by restricting to using the same dissimilarity function for all attributes. We can do so in two possible ways, namely by exclusively using either a) Euclidean-distance, or, b) negative log-likelihood as the metric for all matches. In Chapter 3 [35, 34] introduced a dissimilarity measure based on Euclidean distance which is suitable for categorical-valued attributes (Eq. (3.3))
and is also compatible with a mixture model representation of continuous-valued attributes (Eq. (4.6)). Let the schema match tensor $M$ be defined as,

$$M = [M_{ij}], \text{ where, } M_{ij} = [M_{ij}^{dd} M_{ij}^{cd} M_{ij}^{dc} M_{ij}^{cc}] \quad (5.4)$$

and,

- $M_{ij}^{dd} = 1$ if $i^{th}$ and $j^{th}$ attributes are matched and are both categorical-valued, =0 otherwise
- $M_{ij}^{dc} = 1$ if $i^{th}$ and $j^{th}$ attributes are matched and are categorical and continuous-valued, respectively, =0 otherwise
- $M_{ij}^{cd} = 1$ if $i^{th}$ and $j^{th}$ attributes are matched and are continuous-valued and categorical, respectively, =0 otherwise
- $M_{ij}^{cc} = 1$ if $i^{th}$ and $j^{th}$ attributes are matched and are both continuous-valued, =0 otherwise

Let us now define dissimilarity metric based on Euclidean distance. Therefore, given a schema match tensor $M$ (as defined in Eq. (5.4)), we need to consider the following four cases, namely:

1. If the $i^{th}$ and $j^{th}$ attributes are categorical, then we can measure the Euclidean distance between pmfs, using Eq. (3.3), as:

$$M_{ij}^{dd} \sum_{k=1}^{N_{V_i}} \left( p_k^{(i)} - \sum_{k'=1}^{N_{V_j}} v_{vk}^{(i,j)} p_{k'}^{(j)} \right)^2, \quad (5.5)$$

where,

- $N_{V_i}, N_{V_j}$ = Number of values in $i^{th}$ (categorical) attribute in Schema 1 and $j^{th}$ (categorical) attribute in Schema 2, respectively
- $p_k^{(i)} =$ probability of $k^{th}$ value for $i^{th}$ attribute in Schema 1
- $p_{k'}^{(j)} =$ probability of $k'^{th}$ value for $j^{th}$ attribute in Schema 2
- $V = \{v_{vk}^{(i,j)}\}$ is the value mappings

2. If the $i^{th}$ and $j^{th}$ attributes are continuous-valued, then we can measure the
Euclidean distance between derived pmfs, using Eq. (4.6), as:

\[ M_{cc}^{ij} = \frac{1}{2} \left[ \sum_{k=1}^{K_i} (\alpha_k^{(i)} - \hat{\alpha}_k^{(j)})^2 + \sum_{k=1}^{K_j} (\beta_k^{(j)} - \hat{\beta}_k^{(i)})^2 \right], \tag{5.6} \]

where,

\( K_i, K_j = \) Number of mixture components for the \( i^{th} \) and \( j^{th} \) continuous-valued attributes, respectively

3. If the \( i^{th} \) attribute is categorical and the \( j^{th} \) attribute is continuous-valued, then we can measure the Euclidean distance between a pmf and a derived pmf, using Eq. (3.3), as:

\[ M_{dc}^{ij} = \frac{1}{2} \sum_{k=1}^{N_{Vi}} \left[ p_k^{(i)} - \sum_{k' = 1}^{K_j} v^{(i,j)}_{kk'} \beta_k^{(j)} \right]^2, \tag{5.7} \]

where,

\( N_{Vi} = \) Number of values in \( i^{th} \) (categorical) attribute in Schema 1
\( p_k^{(i)} = \) probability of \( k^{th} \) value for \( i^{th} \) attribute in Schema 1
\( V = \{v^{(i,j)}_{kk'}\} \) is the value mappings
\( K_j = \) Number of mixture components for the \( j^{th} \) continuous-valued attribute in Schema 2
\( \beta_k^{(j)} = \) probability of the \( k^{th} \) mixture component (latent value) for \( j^{th} \) attribute in Schema 2.

4. Lastly, if the \( i^{th} \) attribute is continuous-valued and the \( j^{th} \) attribute is categorical, then we can measure the Euclidean distance between a derived pmf and a pmf, using Eq. (3.3), as:

\[ M_{cd}^{ij} = \sum_{k=1}^{K_i} \left[ \alpha_k^{(i)} - \sum_{k' = 1}^{N_{Vj}} v^{(i,j)}_{kk'} p_k^{(j)} \right]^2, \tag{5.8} \]

where,

\( N_{Vj} = \) Number of values in \( j^{th} \) (categorical) attribute in Schema 2
\( p_{k'}^{(j)} \) = probability of \( k^{th} \) value for \( j^{th} \) attribute in Schema 2

\( V = \{ v_{kk'}^{(i,j)} \} \) is the value mappings

\( K_i = \) Number of mixture components for the \( i^{th} \) continuous-valued attribute in Schema 1

\( \alpha_k^{(i)} = \) probability of the \( k^{th} \) mixture component (latent value) for \( i^{th} \) attribute in Schema 1.

Therefore, the composite dissimilarity metric based on Euclidean distance for mixed (categorical and continuous-valued) attributes is:

\[
D_{EU} (M, V) = \sum_{i=1}^{N_{C1}} \sum_{j=1}^{N_{C2}} \left\{ M_{ij}^{dd} \sum_{k=1}^{N_{V_i}} \sum_{k'=1}^{N_{V_j}} \left[ p_k^{(i)} - \sum_{k'} v_{kk'}^{(i,j)} p_{k'}^{(j)} \right]^2 + M_{ij}^{dc} \sum_{k=1}^{N_{V_i}} \sum_{k'=1}^{N_{V_j}} p_{k}^{(i)} [\alpha_k^{(i)} - \sum_{k'} v_{kk'}^{(i,j)} \beta_{k'}^{(j)}]^2 \right. \\
- \left. \sum_{k'=1}^{K_i} v_{kk'}^{(i,j)} \beta_{k'}^{(j)} \right]^2 + M_{ij}^{cc} \sum_{k=1}^{K_i} \sum_{k'=1}^{K_j} \left[ \alpha_k^{(i)} - \sum_{k'} v_{kk'}^{(i,j)} \beta_{k'}^{(j)} \right]^2 \\
+ \sum_{k=1}^{K_i} \sum_{k'=1}^{K_j} \left[ \alpha_k^{(i)} - \beta_{k'}^{(j)} \right]^2 + \sum_{k=1}^{K_j} \left[ \beta_{k'}^{(j)} - \beta_{k}^{(i)} \right]^2 \right\} \tag{5.9}
\]

where:

\( N_{C1}, N_{C2} = \) Number of attributes in Schema 1 and 2, respectively.

Furthermore, since the log-likelihood for categorical data can also be evaluated ( Eq. \ref{eq5.3}), we can also define a log-likelihood based objective for tackling mixed continuous and categorical attributes (where Eq. \ref{eq4.8} is used for the continuous-valued attributes). Let us now define dissimilarity metric based on log-likelihood. Again, given the schema match tensor \( M \) (as defined in Eq. \ref{eq5.4}), we need to consider the following four cases, namely:

1. If the \( i^{th} \) and \( j^{th} \) attributes are categorical, then we can measure the log-likelihood between pmfs (see Section \ref{section5.2.1}), as:

\[
M_{ij}^{dd} \sum_{k=1}^{N_{V_i}} \sum_{k'=1}^{N_{V_j}} v_{kk'}^{(i,j)} n_k^{(i)} \log p_k^{(i)} + \sum_{k=1}^{N_{V_i}} \sum_{k'=1}^{N_{V_j}} v_{kk'}^{(i,j)} n_k^{(j)} \log p_k^{(j)} \tag{5.10}
\]

where,
\(N_{V_i}, N_{V_j}\) = Number of values in \(i^{th}\) (categorical) attribute in Schema 1 and 
\(j^{th}\) (categorical) attribute in Schema 2, respectively

\(p_k^{(i)}\) = probability of \(k^{th}\) value for \(i^{th}\) attribute in Schema 1

\(p_k^{(j)}\) = probability of \(k^{th}\) value for \(j^{th}\) attribute in Schema 2

\(V = \{v_{kk'}^{(ij)}\}\) is the value mappings

\(n_k^{(i)}\) = number of occurrences of \(k^{th}\) value for \(i^{th}\) attribute in Schema 1

\(n_k^{(j)}\) = number of occurrences of \(k^{th}\) value for \(j^{th}\) attribute in Schema 2

2. If the \(i^{th}\) and \(j^{th}\) attributes are continuous-valued, then we can measure the log-likelihood of model fit, using Eq. (4.8), as:

\[
M_{ij}^{cc} = \left[\sum_{n=1}^{N_i} \log \sum_{k=1}^{K_i} \beta_k^{(j)} f_y(x_n^{(i)}; \theta_k^{(j)}) + \sum_{n=1}^{N_j} \log \sum_{k=1}^{K_j} \alpha_k^{(i)} f_x(y_n^{(j)}; \theta_k^{(i)})\right] \quad (5.11)
\]

where,

\(K_i, K_j\) = Number of mixture components for the \(i^{th}\) and \(j^{th}\) continuous-valued attributes, respectively

\(N_i, N_j\) = Number of entries in the \(i^{th}\) and \(j^{th}\) (continuous) attribute in Schema 1 and Schema 2, respectively

3. If the \(i^{th}\) attribute is categorical and the \(j^{th}\) attribute is continuous-valued, then we can measure the log-likelihood between a pmf and a derived pmf as:

\[
M_{ij}^{dc} = \left[\sum_{k=1}^{K_j} \sum_{k'=1}^{K_j} v_{kk'}^{(ij)} n_k^{(i)} \log \beta_k^{(j)} + \sum_{k=1}^{K_i} \sum_{k'=1}^{K_i} v_{kk'}^{(ij)} n_k^{(j)} \log p_k^{(i)}\right], \quad (5.12)
\]

where,

\(N_j\) = Number of entries in the \(j^{th}\) (continuous) attribute Schema 2

\(N_{V_i}\) = Number of values in \(i^{th}\) (categorical) attribute in Schema 1

\(K_j\) = Number of mixture components for the \(j^{th}\) continuous-valued attribute in Schema 2

\(v_{kk'}^{(ij)}\) = 1 if \(k^{th}\) value in \(i^{th}\) attribute in Schema 1 matches to \(k^{th}\) latent value (mixture component) in \(j^{th}\) attribute in Schema 2, = 0 otherwise
\( n_k^{(i)} \) = number of occurrences of \( k^{th} \) value for \( i^{th} \) attribute in Schema 1
\( \hat{n}_{k'}^{(j)} \) = number of occurrences of \( k^{th} \) latent value (mixture component) for \( j^{th} \) attribute in Schema 2
\( \hat{n}_k^{(i)} = N_j \beta_{k'}^{(j)} \)

4. Lastly, if the \( i^{th} \) attribute is continuous-valued and the \( j^{th} \) attribute is categorical, then we can measure the log-likelihood between a derived pmf and a pmf as:

\[
M_{ij}^{cd} \left[ \sum_{k=1}^{K_i} \sum_{k'=1}^{N_{V_j}} v_{kk'}^{(i,j)} \hat{n}_k^{(i)} \log p_{k'}^{(j)} + \sum_{k=1}^{K_i} \sum_{k'=1}^{N_{V_j}} v_{kk'}^{(i,j)} n_{k'}^{(j)} \log \alpha_k^{(i)} \right], \tag{5.13}
\]

where,
\( N_i \) = Number of entries in the \( i^{th} \) (continuous) attribute Schema 1
\( N_{V_j} \) = Number of values in \( j^{th} \) (categorical) attribute in Schema 2
\( K_i \) = Number of mixture components for the \( i^{th} \) continuous-valued attribute in Schema 1
\( v_{kk'}^{(i,j)} = 1 \) if \( k^{th} \) value in \( i^{th} \) attribute in Schema 1 matches to \( k^{th} \) latent value (mixture component) in \( j^{th} \) attribute in Schema 2, =0 otherwise
\( n_{k'}^{(j)} \) = number of occurrences of \( k^{th} \) value for \( j^{th} \) attribute in Schema 2
\( \hat{n}_k^{(i)} \) = number of occurrences of \( k^{th} \) latent value (mixture component) for \( i^{th} \) attribute in Schema 1
\( \hat{n}_k^{(i)} = N_i \alpha_k^{(i)} \)

Therefore, the composite dissimilarity metric based on negative log-likelihood for mixed (categorical and continuous-valued) attributes is:

\[
D_{LL}(M, V) = - \sum_{i=1}^{N_{C_1}} \sum_{j=1}^{N_{C_2}} \left[ M_{ij}^{dd} \left[ \sum_{k=1}^{K_i} \sum_{k'=1}^{N_{V_j}} v_{kk'}^{(i,j)} n_k^{(i)} \log p_{k'}^{(j)} + \sum_{k=1}^{K_i} \sum_{k'=1}^{N_{V_j}} v_{kk'}^{(i,j)} \hat{n}_{k'}^{(j)} \log \alpha_k^{(i)} \right] + M_{ij}^{dc} \left[ \sum_{k=1}^{K_i} \sum_{k'=1}^{N_{V_j}} v_{kk'}^{(i,j)} \hat{n}_k^{(i)} \log \beta_{k'}^{(j)} + \sum_{k=1}^{K_i} \sum_{k'=1}^{N_{V_j}} v_{kk'}^{(i,j)} \hat{n}_{k'}^{(j)} \log p_k^{(i)} \right] \right] + \sum_{i=1}^{N_{C_1}} \sum_{j=1}^{N_{C_2}} \sum_{k=1}^{K_i} \sum_{k'=1}^{N_{V_j}} v_{kk'}^{(i,j)} \hat{n}_k^{(i)} \log \beta_{k'}^{(j)} + \sum_{k=1}^{K_i} \sum_{k'=1}^{N_{V_j}} v_{kk'}^{(i,j)} \hat{n}_{k'}^{(j)} \log p_k^{(i)} \right]
\]


\[ + M_{ij}^{cd} \left[ \sum_{k=1}^{N_i} \sum_{k'=1}^{N_j} V_{ij}^{(i,j)} \times n_k^{(i)} \log p_k^{(j)} + \sum_{k=1}^{K_i} \sum_{k'=1}^{K_j} V_{ij}^{(i,j)} \times n_k^{(j)} \log \alpha_k^{(i)} \right] \\
+ M_{ij}^{cc} \left[ \sum_{n=1}^{N_i} \log \left( \sum_{k=1}^{K_i} \beta_k^{(j)} f_{Y|k}(x_n^{(i)}, \theta_k^{(j)}) \right) + \sum_{n=1}^{N_j} \log \left( \sum_{k=1}^{K_j} \alpha_k^{(i)} f_{X|k}(y_n^{(j)}, \theta_k^{(i)}) \right) \right] \]

(5.14)

where:

\( N_{C_1}, N_{C_2} = \) Number of attributes in Schema 1 and 2, respectively

Here, we have assumed that \( N_{C_1} \leq N_{C_2} \), i.e., “onto matching”, just as for the first-order dissimilarity metric for continuous-valued attributes. The dissimilarity metric based on negative log-likelihood (Eq. (5.14)) has difficulty when the cardinalities of the categorical-valued attributes are not equal, i.e., if the probability of a value used in matching in Eq. (5.14) is zero. This is problematic because for \( p_k^{(i)} = 0, n_k^{(j)} \log p_k^{(i)} \) is ill-defined, which results in an ill-defined global objective. There are several strategies to deal with this.

1. Introduce extra symbols for attributes with the smaller cardinality and assign them a small frequency count \( \epsilon \) followed by renormalization of the pmf of this reconstructed alphabet. This will ensure that both pmfs have alphabets of equal size and an undefined global objective will never occur.

2. Exclusively use the Log-Likelihood dissimilarity of the pmf of the larger alphabet, in this case \( n_k^{(j)} \log p_k^{(i)} \). For example, replace the first (symmetric) term in Eq. (5.14), \( \sum_{k=1}^{N_i} \sum_{k'=1}^{N_j} V_{ij}^{(i,j)} \times n_k^{(i)} \log p_k^{(j)} + \sum_{k=1}^{N_i} \sum_{k'=1}^{N_j} V_{ij}^{(i,j)} \times n_k^{(j)} \log p_k^{(i)} \), by twice the dissimilarity associated with the larger pmf, i.e., \( 2 \sum_{k=1}^{N_i} \sum_{k'=1}^{N_j} V_{ij}^{(i,j)} \times n_k^{(j)} \log p_k^{(i)} \). The latter strategy was used in our experiments which are discussed in Section 5.3.

As noted in previous work [34], the squared Euclidean dissimilarity measure, though, is free from this issue and does not require a (somewhat arbitrary) reconstruction of attribute alphabets— if two attributes across tables have different cardinalities, we can assume that for a value \( k \) with probability \( p_k \) in one attribute

\(^1\epsilon \) should be chosen much smaller than the smallest probability value of the original alphabet.
that does not have a corresponding value in the matching attribute, the probability of the matching value is zero. In other words, for the Euclidean distance measure, we essentially introduce extra probability values $\epsilon = 0$, which cause no issues, unlike the log-likelihood measure. Since the results for a method based on log-likelihood may not be robust due to different value alphabet sizes across attributes (as a result of alphabet reconstruction), and since Euclidean distance (naturally) handles matching attributes with different cardinalities by assigning zero probabilities, Euclidean distance is an attractive criterion for tackling schema matching when schemas have mixed attribute types. In fact, Euclidean distance (Eq. (5.9)) outperforms the negative log-likelihood metric (Eq. (5.14)) for mixed continuous and categorical attributes, as will be shown in Section 5.3.3.

5.2.3 Schema matching strategy

Our algorithm’s goal is minimization of the dissimilarity objective (Eqs. (5.9), or, (5.14)) over the space of schema matches, i.e., minimization of the global objective over all possible pairs of attributes across the two schemas. We implemented the same local minimization strategy as discussed in Section 4.2.2.

Furthermore, schema matching (Eqs. (5.9), or, (5.14)) that involves treating a pair of matching attributes requires specification of a value mapping for the attributes in the matching pair. In Section 3.2.5 [35, 34], it was shown that the value mapping obtained by sorting the pmfs for the matching attributes is optimal with respect to Euclidean distance between pmfs. This value-mapping solution minimizes Euclidean distance between the two pmfs. We use this same strategy (albeit now a heuristic) for log-likelihood based matching of discrete-valued attributes.

5.3 Experimental validation

In this section, we present the results of our algorithms for schema matching for mixed categorical and continuous-valued attributes compared with several alternative methods on real-world Census Bureau datasets.
5.3.1 Mixed categorical and continuous attributes datasets

We used the Census Bureau Public Usage Micro data Sample (PUMS) 5% datasets for the states of California (CA), New York (NY) and Texas (TX) for the year 1990. The common schema describing each of these tables consists of 241 attributes. Table 3.1 summarizes these datasets. In Sections 3.3.5 and 4.3.5, it was shown (unsurprisingly) that the accuracy of schema matching algorithms degrades as the number of columns to be matched increases. Thus, for example, matching all 241 attributes would likely yield very poor accuracy for all methods and not allow us to clearly distinguish the relative performance of evaluated methods. Thus, in our experiments we worked with an attribute subset sufficiently large so as to make the matching problem nontrivial, while not so large that the matching accuracies of all methods will be poor. Specifically, we worked with \( 2K \) attributes – \( K \) continuous-valued and \( K \) categorical-valued – and considered \( K = 15 \). To reliably measure matching accuracy for these data sets, we average matching accuracy results over 50 trials where, in each trial the columns to be matched were randomly chosen (\( K \) continuous-valued and \( K \) categorical-valued). These 50 trial sets comprised our mixed attribute data sets and were used for the evaluation of the methods developed in Section 5.2.2. Finally, we considered mixed data sets wherein some continuous-valued attributes were quantized. Specifically, we chose \( 3K \) attributes – 2K continuous-valued and K categorical – and considered \( K = 10 \). Furthermore, the last K continuous-valued attributes were quantized for the second schema (by using mixture component \( a \ posteriori \) probabilities). Thus the first schema contained 2K continuous-valued and K categorical attributes while the second schema contained K continuous-valued, K categorical and K quantized continuous-valued attributes. To reliably measure matching accuracy for these data sets, we average matching accuracy results over 50 trials where, in each trial the columns to be matched were randomly chosen (from the \( 3K \) attributes). These 50 trial sets comprised our mixed quantized attribute data sets and were used for the evaluation of the methods developed in Section 5.2.2.
5.3.2 Experimental setup

The schema matching with embedded value mapping algorithm is called to perform schema matching on continuous-valued attributes. The matching algorithm is initialized from one of the following three starting initial schema matches: (1) Random Initialization: a random schema match is chosen, and (2) Ground-truth Initialization: the ground-truth schema match, and (3) Fixed initialization: a schema match where the same column index was assumed as a match, i.e., column 1 in schemas A and B are assumed as matching and so on. We compared our schema matching algorithms against four methods that, respectively, do two-opt switching of attributes to minimize squared Euclidean distance between i) pmfs, for a pair of matching attributes (The objective function in Eq. (3.3)) ii) mutual informations, for matching attribute pairs (The objective function in Kang and Naughton [39, 38] (MI), iii) conditional entropies [34], for matching attribute pairs, (CE) and iv) entropies, for a pair of matching attributes (EN). The last three methods are consistent with the Kang and Naughton [39, 38] approach in that these three cost function do not exploit (and require) embedded value mappings, in matching a pair of discrete-valued attributes. Since these methods assume categorical-valued attributes, derived pmfs for all continuous-valued attributes (using mixture component a posteriori probabilities) were input to these methods. To fairly compare methods, all matching algorithms were initialized from the same random schema match (random initialization). Lastly, for the mixed attributes dataset we also compared schema matching results against the result from a method (labeled “FO EU LL”) that applies Euclidean distance between pmfs for categorical-valued attributes and log-likelihood dissimilarity between pdfs for continuous-valued attributes. This method does not allow matching continuous-valued with discrete-valued attributes; thus, this method will necessarily fail to match correctly in the most difficult setting where some discrete-valued attributes in one schema are in fact quantized continuous-valued attributes.

We implemented our schema matching with embedded value mapping algorithms using R [61] 2.11 and MCLUST 3.4 [23, 21, 22]. Example BIC scores, estimated pdfs and histogram for a continuous-valued “Age” attribute from the CA and TX datasets when 50K random row samples is illustrated in Fig. 41. Since the density estimation and the log-likelihood algorithm are computationally
intensive, they were executed on a 224 node (Dual Intel® Xeon @ 2.67GHz) cluster. Up to 10% of the cluster CPU resources were allocated by the cluster scheduler to execute the algorithms.

We ran our experiments over a randomly chosen subset of \( n \) (\( 2 \leq n \leq 20 \)) attributes (either mixed continuous and categorical or mixed continuous, categorical and quantized attributes). We randomly permuted the columns in the second schema to ensure that the algorithms cannot trivially stumble upon the correct schema matching result.

### 5.3.3 One-to-one schema matching for Mixed Continuous and Categorical Attributes

#### 5.3.3.1 Random initialization

Figs. 5.1 present results for mixed attributes datasets where the initial schema match was randomly selected. All algorithms were initialized from the same initial schema match. For all datasets, we increased the number of attributes in the two input tables from 2 to 20. The attributes for the mixed (continuous-valued and categorical) datasets were randomly chosen (an equal number of continuous and discrete-valued) from a set of 30 columns that consisted of 15 continuous-valued and 15 categorical-valued attributes. For each pair of schemas, we first randomly chose 100K rows (samples). We repeated the experiment 50 times, each time randomly choosing attributes from the set of 30 mixed (categorical and continuous) columns. Fig. 5.1 shows the average precision (over 50 trials) for Euclidean PMF matching (FO EU) and the log-likelihood (FO LL) metrics for mixed type datasets, respectively. For comparison, the first-order discrete Euclidean metric (FO DM) [34], the Entropy only (EN), Kang-Naughton Mutual Information (MI) [39] and modified Kang-Naughton algorithm (labeled CE) results are also presented.

Figs. 5.1.1 and 5.1.3 show precision of the schema matching for the CA vs. NY and CA vs. TX mixed attributes data sets, respectively. The restricted criterion matching method (FO EU LL), which minimizes Eqs. (4.6) and (4.8) while restricting to matching attributes of the same type (discrete with discrete and continuous with continuous), and the Euclidean-distance based method (FO EU), which minimizes Eq. (5.9), perform best (\( > 10\% \) better accuracy over other meth-
5.1.1: Average Precision for CA vs. NY dataset

5.1.2: Computation Time for CA vs. NY dataset

5.1.3: Average Precision for CA vs. TX dataset

5.1.4: Computation Time for CA vs. TX dataset

Figure 5.1: First-order schema matching results for mixed categorical and continuous attributes

methods for large attribute sizes). While the restricted criterion matching method has a best-case performance improvement of 2-5% over the Euclidean distance method, its computational complexity is much higher, especially for large attribute sizes. Moreover, FO EU LL will not fare so well when some discrete attributes are in fact a result of quantization (see Section 5.3.3.2) since this method cannot perform matching between continuous-valued and discrete-valued attributes. The performance of the FO LL metric is poor; this can be attributed to the inability of the log-likelihood metric to handle discrete-valued attributes with different cardinalities well (as discussed in Section 5.2.2). The first-order pmf-based dissimilarity method (FO DM) has relatively poor performance (between 50% and 70% accuracy for schema sizes ≥10) as well. The mutual information (MI), conditional entropy (CE) and entropy (EN) methods all perform poorly.
5.2.1: Average overall precision for CA vs. NY dataset

5.2.2: Average correct number of quantized continuous-valued to continuous-valued matches for CA vs. NY dataset for various methods, and total number of such ground-truth matches (TOTAL).

Figure 5.2: First-order schema matching results for mixed categorical and continuous attributes in the presence of quantized columns.

5.3.3.2 Effect of quantized continuous-valued attributes

Figure 5.2 presents the results of bijective (one-to-one) matching for the CA vs NY mixed quantized datasets. It can be observed that the first-order pmf method (FO DM) \[34\], which minimizes Eq. (3.3), performs best (≈ 10 – 20% improvement over other methods for large attribute sizes). The Euclidean-distance based method (FO EU), which minimizes Eq. (5.9), performs second best. Both these methods perform much better in comparison to other methods with typical precision gains being > 50% when ≥ 5 columns are matched. The performance of the FO LL metric is poor; this can be attributed to the inability of the log-likelihood metric to handle discrete-valued attributes with different cardinalities well (as discussed in Section 5.2.2). The mutual information (MI), conditional entropy (CE) and entropy (EN) methods all, again, perform poorly even though they are supplied with soft-discretized versions of the dataset. Note that the restricted criterion matching method (FO EU LL) method, discussed in Section 5.3.2, does not apply to such matching since it cannot match a discrete-valued attribute to a continuous-valued attribute. Figure 5.2.2 shows the average number of ground-truth quantized to continuous-valued attribute matches present (labeled “TOTAL”) as a function of schema size in the 50 trials. For comparison, the average number of true quantized to continuous-valued attribute matches detected as a function of schema size by FO
5.3.1: Effect of Rows on Precision for CA vs. NY dataset

5.3.2: Effect of Rows on Precision for CA vs. TX dataset

Figure 5.3: Effect of rows on precision for first-order schema matching results for mixed attributes.

EU, FO LL, FO DM, EN, MI, and CE methods are shown. As expected, the FO DM and FO EU methods have high precision of matching quantized to continuous-valued attributes, consistent with the overall precision of these methods on the mixed quantized data sets. All other methods, however, are unable to accurately perform such matching.

5.3.3.3 Computational time

Figs. 5.1.2 and 5.1.4 illustrate the computation time for executing the schema matching algorithms for the three state files when tackling mixed (categorical and continuous-valued) datasets. The first-order Euclidean distance methods (FO EU) and entropy-only algorithms (EN) have the least computation times for mixed (categorical and continuous-valued) attributes. The FO LL and FO EU LL dissimilarity algorithms were the most expensive. This increase in computation cost for the log-Likelihood based algorithms is expected due to higher computation cost of calculating the log-likelihood score. In addition, a significant component of the computation time of all the schema matching algorithms is spent in learning the probability density functions for all the continuous-valued attributes in both schemas in a preprocessing step.
5.4.1: Average Precision for CA vs. NY dataset

5.4.2: Average Precision for CA vs. TX dataset

Figure 5.4: First-order schema matching results for mixed categorical and continuous attributes when initialized from ground-truth.

5.3.3.4 Data sampling effects

Figs. 5.3.1 and 5.3.2 illustrate the effect of the number of rows on matching accuracies produced during schema matching. The match accuracies for 10K tuples (labeled “FO EU 10K” and “FO LL 10K”), 50K tuples (labeled “FO EU 50K” and “FO LL 50K”), and, 100K tuples (labeled “FO EU 100K” and “FO LL 100K”) are shown for the two dissimilarity metrics when mixed continuous and categorical attributes were matched. The match accuracies were highest when ≥ 50K tuples were used for schema matching and it can be observed that the schema matching accuracies improve with number of K-tuples used, as expected. Both methods have similar performance variations ≈ 5% as the number of K-tuples is varied. As in the previous sections, the Euclidean (FO EU) dissimilarity metric performed best (5.3.1 5.3.2). For mixed attributes, the restricted criterion matching method (labeled FO EU LL) is also shown, which had the best performance. However, as discussed in Section 5.3.3.2 this metric will not fare so well when some discrete attributes are in fact a result of quantization. Furthermore, as discussed in the previous section, the performance improvement (2-5%) of restricted criterion matching method (FO EU LL) over the Euclidean distance (FO EU) method requires significantly higher computational time.
5.3.3.5 Ground-truth initialization

We next demonstrate that the objective function of the first order dissimilarity algorithms for mixed categorical and continuous-valued attributes captures the schema matching objective better than the other schema matching criteria (first-order pmf, 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?). Fig. 5.4.1 and Fig. 5.4.2 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 20. We chose 100K tuples from both matching schemas. The attributes themselves were randomly chosen from a set of 30 categorical and continuous-valued attributes as discussed in Section 5.3.1. We repeated the experiment 50 times while randomly choosing attributes from the set of 30 columns. We measured the average precision over the 50 experiments for both the California and New York (Fig. 5.4.1) and California and Texas demographics data sets (Fig. 5.4.2).

For both datasets, the restricted criterion matching (FO EU LL) and estimated pmf (FO EU) methods performed best illustrating the suitability of these methods for schema matching for mixed continuous-valued and categorical attributes. The entropy-only schema matching algorithms performs second-best but, as discussed in Section 4.3.5.4, the entropy-only algorithm lacks a unique global minimum making this method unsuitable for schema matching. All other algorithms had similar performance to random initialization.

5.3.3.6 Fixed initialization

In the previous experiment illustrated in Section 5.3.3.1, a random trial meant a random selection of rows and initial solution. In this section, we evaluated schema matching accuracy for all algorithms when initialized from a fixed initial schema match. Figs. 5.5.1 and 5.5.2 present the results of our method when the initial schema match for all algorithms was fixed. The initial schema match is generated
by assigning the same column index for attributes across schemas, i.e., column \( i \) in schemas A and B is a matching attribute pair and so on. Two-opt switching was then performed to minimize the global objective. We ran the experiments while increasing the number of attributes in the two input tables from 2 to 20. We repeated the experiment 50 times for the same fixed initialization. In each experiment, 50K different tuples were randomly selected while the attributes were randomly chosen from a set of 30 categorical and continuous-valued attributes as discussed in Section 5.3.1. We measured the average precision over the 50 experiments for both the California and New York (Fig. 5.5.1) and California and Texas demographics data sets (Fig. 5.5.2). For comparison, the First-order pmf (labeled FO DM), Entropy-only (labeled EN), Kang-Naughton Mutual Information (labeled MI), and modified Kang-Naughton algorithm (labeled CE) results are also presented.

Clearly, the FO EU LL and FO EU dissimilarity algorithms perform best. Both these methods perform much better in comparison to the other methods with typical precision gains being \( \geq 20\% \) when \( \geq 5 \) columns are matched for both datasets. FO LL and FO DM perform second best with \( \geq 50\% \) accuracy when \( \geq 5 \) columns are matched for both datasets. Mutual information (MI), conditional entropy (CE) and entropy (EN) methods all perform poorly. This clearly shows that the FO EU and FO EU LL dissimilarity algorithms are robust and largely unaffected by tuple selection.
Chapter 6

Schema Matching Criterion for Continuous-valued attributes with Transformations

6.1 Introduction

In Section 4.2.1, we have made the statistical assumption that for two matching continuous-valued attributes, the data in the two columns were generated by sampling i.i.d. from a common distribution, \( f_X(x) \). That is, we have assumed no pdf transformations are necessary, when matching two continuous-valued columns across schemas. However, this assumption is quite often violated in real-world situations. For example, consider two schemas describing sensor data collected by meteorological agencies in the United States and Europe containing a matching attribute “Temperature”. The “Temperature” column could store the values in Fahrenheit or Celcius, thus potentially requiring a transformation to be applied to data values before schema matching is performed. In this chapter, we address schema matching when attributes in both schemas are continuous-valued and when pdf transformations may exist between matching attributes. Furthermore, only affine transformations such as English to metric unit conversion and related transformation (e.g., Temperature) are generally encountered during schema matching, we therefore only consider schema matching with affine pdf transformations in this
As discussed in Section 1.2.3, we can extend the EM algorithm to incorporate learning transformations. We develop the EM algorithm with embedded transformation learning in Section 6.2. We then outline schema matching with embedded transformation learning in Section 6.3.1. Lastly, the dissimilarity metrics discussed in Sections 4.2.1.1 and 4.2.1.2 can be extended to incorporate fixed transformations, which we outline in Section 6.3.2.

### 6.2 Expectation-Maximization for transformation learning

Consider matching two continuous-valued attributes $X$ and $Y$, from different schemas. Let attribute $X$, with a set of observations $X = \{x_1, \ldots, x_N\}$, be modeled by a mixture model given by:

$$f_X(x; \theta_X) = \sum_{j=1}^{K_x} \alpha_j f_{X|j}(x, \theta_{X|j}) = \sum_{j=1}^{K_x} \alpha_j \frac{e^{-(x-\mu_{X|j})^2/(2\sigma_{X|j}^2)}}{\sqrt{2\pi\sigma_{X|j}^2}}$$

(6.1)

where $K_x$ is the number of components, $\theta_{X|j} = (\mu_{X|j}, \sigma_{X|j})$ are the parameters, $\mu_{X|j}$ is the mean and $\sigma_{X|j}$ is the standard deviation of the $j^{th}$ component respectively, $\alpha_j \geq 0$ $\forall k$ and $\sum_j \alpha_j = 1$.

Let $Y$ be a matching attribute, with a set of i.i.d. observations $Y = \{y_1, \ldots, y_N\}$, such that there exists an affine transformation $\psi = g(a, b)$ between $X$ and $Y$ given by:

$$Y = aX + b.$$  

(6.2)

The transformation will result in mean shifts and variance scaling, i.e.,

$$\mu_{Y|j} = a\mu_{X|j} + b, \ j = 1, \ldots, K_x$$

(6.3)

$$\sigma_{Y|j} = a\sigma_{X|j}, \ j = 1, \ldots, K_x$$

(6.4)

but with the density functional form (Gaussian) preserved.

Therefore, the log-likelihood for $Y$ given the $K_x$ component mixture model and
the transformation $\psi$ can be written as:

$$\log \mathcal{L}(\mathcal{Y}; \psi, \theta_X) = \sum_{i=1}^{N_y} \log \sum_{j=1}^{K_x} \frac{\alpha_j e^{-(y_i - (a\mu X_j + b))^2}}{\sqrt{2\pi a^2 \sigma^2_{X_j}}}.$$  \hfill (6.5)

The maximum likelihood (ML) estimate

$$\hat{\psi}_{ML} = \arg \max_{\psi} \log f(\mathcal{Y}; \psi, \theta_X)$$  \hfill (6.6)

cannot be found analytically. However, the pdf transformation can be estimated using the Expectation-Maximization (EM) algorithm \[14, 50, 49\] for finding locally optimal maximum likelihood estimates of transformation parameters $\psi$ for a given mixture model with an a priori given number of components and parameters $\theta_X$. The EM algorithm is based on the interpretation of $\mathcal{Y}$ as incomplete data, where the missing part is a set of $N_y$ labels $\mathcal{Z} = \{z^{(1)}, \ldots, z^{(N_y)}\}$ associated with the $N_y$ observations, that indicate which component produced each sample. Each label is a binary vector $z^{(i)} = [z^{(i)}_1, \ldots, z^{(i)}_J]$, where $z^{(i)}_m = 1$ and $z^{(i)}_p = 0$, for $p \neq m$ which means that sample $y^{(i)}$ was generated by the $m^{th}$ component. Thus the complete log-likelihood (i.e., we estimate $\psi$ if complete data $\mathcal{W} = \{\mathcal{Y}, \mathcal{Z}\}$ was observed):

$$\log \mathcal{L}(\mathcal{Y}, \mathcal{Z}; \psi, \theta_X) = \sum_{i=1}^{N_y} \sum_{j=1}^{K_x} z^{(i)}_j \log \frac{\alpha_j e^{-(y_i - (a\mu X_j + b))^2}}{\sqrt{2\pi a^2 \sigma^2_{X_j}}}.$$  \hfill (6.7)

The EM algorithm generates a sequence of estimates $\psi'$ by iteratively applying the two following steps (until some convergence criterion is met):

- **Expectation (E) - step**: Compute the conditional expectation of the complete log-likelihood, given $\mathcal{Y}$ and parameters $\theta_X$ and the current estimate $\psi$ using the Q-function:

$$Q(\psi' ; \psi) = E[\log \mathcal{L}(\mathcal{Y}, \mathcal{Z}; \psi', \theta_X); \mathcal{Y}, \psi, \theta_X]$$  \hfill (6.8)

- **Maximization (M) - step**: Update $\psi'$ such that $\psi' = \arg \max_{\psi} Q(\psi'|\psi)$ under some constraints. We discuss these constraints in Section 6.2.2.

We derive the updates for $a$ and $b$ in the following sections.
6.2.1 Expectation-step

The E-step simply computes the mixture component \textit{a posteriori} probabilities, given the current parameter estimates \((a^{(t)}, b^{(t)})\), i.e.,

\[
P[C_i = j; y_i, a^{(t)}, b^{(t)}, \theta_X] = \frac{\alpha_j}{\sum_k K_x \sqrt{2\pi a^{(t)} \sigma_{X|k}^2}} e^{-\frac{(y_i - (a^{(t)} \mu_{X|j} + b^{(t)}))^2}{2a^{(t)} \sigma_{X|j}^2}}
\]

(6.9)

where, \(j = 1, \ldots, K_x\), \(i = 1, \ldots, N_y\).

6.2.2 Maximization-step

The M-step re-estimates the model parameters based on the E-step quantities. The Expected complete data log-likelihood is given by:

\[
A(a, b) \triangleq E[\log \mathcal{L}(Y, Z; \psi, \theta_X)]
\]

\[
= \sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a, b] \log \left( \frac{\alpha_j}{\sqrt{2\pi a^2 \sigma_{X|j}^2}} e^{-\frac{(y_i - (a \mu_{X|j} + b))^2}{2a^2 \sigma_{X|j}^2}} \right)
\]

(6.10)

Therefore, differentiating with respect to \(b\), we have

\[
\frac{\partial A(a, b)}{\partial b} = -\sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}] \frac{\partial}{\partial b} \left[ \frac{(y_i - (a^{(t)} \mu_{X|j} + b))}{2a^2 \sigma_{X|j}^2} \right]
\]

(6.12)

\[
= -\sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}] 2(y_i - (a \mu_{X|j} + b))(-1)
\]

(6.13)

The critical point can be found as,

\[
-\sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; x_i, a^{(t)}, b^{(t)}] \frac{2(y_i - (a \mu_{X|j} + b))(-1)}{2a^2 \sigma_{X|j}^2} = 0,
\]

(6.14)
which results in the update for \( b \) as,

\[
b^{(t+1)} = \sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}] \frac{(y_i - a^{(t)} x_{ij})}{\sigma_{x_{ij}}} \tag{6.15}
\]

Similarly, the updates for \( a \) can be found by differentiating with respect to \( a \), i.e.,

\[
\frac{\partial A(a,b)}{\partial a} = -\sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}] \frac{\partial}{\partial a} \left( \frac{(y_i - (a \mu_{xij} + b))^2}{2a^2 \sigma_{x_{ij}}^2} + \log a \right) \tag{6.16}
\]

\[
= -\sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}] \left[ \frac{-a \mu_{xij}(y_i - a \mu_{xij} - b) - (y_i - a \mu_{xij} - b)^2}{a^3 \sigma_{x_{ij}}^2} + \frac{1}{a} \right] \tag{6.17}
\]

\[
= -\sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}] \left[ \frac{-a \mu_{xij}(y_i - a \mu_{xij} - b) - (y_i - a \mu_{xij} - b)^2 + a^2 \sigma_{x_{ij}}^2}{a^3 \sigma_{x_{ij}}^2} \right] \tag{6.18}
\]

\[
= -\sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}] \left[ \frac{a^2(\mu_{x_{ij}}^2 - \mu_{x_{ij}}^2 + \sigma_{x_{ij}}^2) + a \mu_{x_{ij}}(-y_i + b + 2y_i - 2b) - (y_i - b)^2}{a^3 \sigma_{x_{ij}}^2} \right] \tag{6.19}
\]

\[
= -\sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}] \left[ \frac{a^2(\sigma_{x_{ij}}^2 + a \mu_{x_{ij}}(y_i - b) - (y_i - b)^2}{a^3 \sigma_{x_{ij}}^2} \right] \tag{6.20}
\]
The critical point can be found as,

\[- \sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}] \left[ a^2 \left( \frac{\sigma^2_{X|j}}{\sigma^4_{X|j}} \right) + a \mu_{X|j} (y_i - b) - (y_i - b)^2 \right] = 0.\]

Therefore, the updates for \( a \) are the solution to the quadratic equation given by:

\[ a^2 \left( \sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}] \right) + a \left( \sum_{i=1}^{N_y} \sum_{j=1}^{K_x} \frac{P[C_i = j; y_i, a^{(t)}, b^{(t)}]}{\sigma^2_{X|j}} \mu_{X|j} (y_i - b^{(t)}) \right) \]

\[- \sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}] \frac{(y_i - b^{(t)})^2}{\sigma^4_{X|j}} = 0. \quad (6.22)\]

The updates for \( a \) are therefore the roots to the quadratic equation, i.e.,

\[ a^{(t+1)} = \frac{-B^{(t)} \pm \sqrt{B^{(t)2} - 4A^{(t)}C^{(t)}}}{2A^{(t)}}, \quad (6.23)\]

where,

\[ A^{(t)} = \sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}], \quad (6.24)\]

\[ B^{(t)} = \sum_{i=1}^{N_y} \sum_{j=1}^{K_x} \frac{P[C_i = j; y_i, a^{(t)}, b^{(t)}]}{\sigma^2_{X|j}} \mu_{X|j} (y_i - b^{(t)}), \quad (6.25)\]

\[ C^{(t)} = - \sum_{i=1}^{N_y} \sum_{j=1}^{K_x} P[C_i = j; y_i, a^{(t)}, b^{(t)}] \frac{(y_i - b^{(t)})^2}{\sigma^4_{X|j}}. \quad (6.26)\]

Schema matching with pdf transformation learning is especially complex because it is likely that a transformation between two columns that do not match can also be found. For example, Fig. 6.1 shows two continuous-valued attributes from a hypothetical schema \( S_1 \). The first attribute, \( X_1 \) with observations \( X_1 \), (Fig. 6.1.1) is modeled by a mixture model with parameters \( \mu_{X_1} = \{0, 5\}, \sigma^2_{X_1} = \{0.5, 3\} \) and \( \alpha_{X_1} = \{0.5, 0.5\} \). The second attribute, \( X_2 \) with observations \( X_2 \), (Fig. 6.1.2) is
6.1.1: The parameters of the mixture model describing attribute $X_1$ in $S_1$ is $\mu = \{0, 5\}$, $\sigma = \{0.5, 3\}$ and $\alpha = \{0.5, 0.5\}$.

6.1.2: The parameters of the mixture model describing attribute $X_2$ in $S_1$ is $\mu = \{4, 10\}$, $\sigma = \{3, 0.5\}$ and $\alpha = \{0.5, 0.5\}$.

Figure 6.1: Two hypothetical continuous-valued attributes from a schema.

modeled by a mixture model with parameters $\mu_{X_2} = \{4, 10\}$, $\sigma^2_{X_2} = \{3, 0.5\}$ and $\alpha_{X_2} = \{0.5, 0.5\}$. Now, consider an attribute, $Y$ with observations $Y$, in the second schema $S_2$, such that $Y = 2X_1 + 5$. This attribute is, therefore, modeled by a mixture model with parameters $\mu_Y = \{5, 15\}$, $\sigma^2_Y = \{2, 12\}$ and $\alpha_Y = \{0.5, 0.5\}$. (Note that, $\mu_Y = 2\mu_{X_1} + 5$ and $\sigma^2_Y = 4\sigma^2_{X_1}$). Since, the transformation $Y \approx -2X_2 + 25$ also exists, one could match $X_2$ with $Y$. Thus, schema matching with pdf transformation learning introduces an extra dimension of confounding, since transformations between two attributes that do not match could actually have higher log-likelihood, i.e., these two non-matching attributes are matched. To reduce this confounding, we constrain that attribute $X$ in $S_1$ and attribute $Y$ in $S_2$ have pdf transformations $Y = aX + b$, such that $a > 0$. Furthermore, candidate pdf transformations that are commonly encountered when merging schemas include English to metric transformations and related transformations, e.g., two matching “weight” columns across schemas where the measurement scale in one schema is pounds while the measurement scale in the second schema is kilograms. Another common example includes an attribute “Price” in the first schema and a matching “Price + Tax” attribute in the second schema. Thus, the transformation parameter constraint $a > 0$ is a valid and reasonable assumption for real-world schema matching.

Therefore under this transformation parameter constraint $a > 0$, the estimate for $a^{(t+1)}$ is chosen as follows:

- If both roots in Eq. (6.23) are negative, we reject the estimates for $a^{(t+1)}$ and $b^{(t+1)}$ and assume no transformation exists, i.e., $a^{(t+1)} = 1$ and $b^{(t+1)} = 0$. 
• If one root is positive and one is negative (Eq. (6.23)), we choose positive root as the estimate for \( a^{(t+1)} \).

• If both roots in Eq. (6.23) are positive, the log-likelihood for both roots is evaluated (Eq. (6.3)). Both roots are propagated as possible solutions, however, once convergence is achieved, we choose \( a^{(t+1)} \) equal to the root that gives the maximum log-likelihood.

### 6.2.3 EM initialization

Since, EM performs local optimization, it is highly dependent upon initialization. We use the following two initialization strategies, which we will utilize in our experiments presented in Section 6.4.4:

- **Standard initialization**: choose \( a = 1, b = 0 \).

- **Model-dependent initialization**: Consider matching two continuous-valued attributes \( X \) and \( Y \), modeled by mixture models with parameters \( \theta^{(x)} = \{\alpha_i^{(x)}, \mu_i^{(x)}, \sigma_i^{(x)}\}, i = 1, \ldots, K_x \) and \( \theta^{(y)} = \{\beta_j^{(y)}, \mu_j^{(y)}, \sigma_j^{(y)}\}, j = 1, \ldots, K_y \), respectively. Choose, initial estimates \( a, b \) that satisfy:

\[
\begin{align*}
\mu_1^{(y)} &= a \mu_1^{(x)} + b, \\
\sigma_1^{(y)} &= a \sigma_1^{(x)},
\end{align*}
\]

where, \( \mu_1^{(y)} \leq \mu_j^{(y)} \forall j = 2, \ldots, K_y \) and \( \mu_1^{(x)} \leq \mu_i^{(x)} \forall i = 2, \ldots, K_x \).

### 6.2.4 EM algorithm

The EM algorithm with transformation learning, therefore, produces a sequence of estimates, \( \{a^{(t)}, b^{(t)}, t=0,1,2,\ldots\} \) by alternatingly applying the Expectation and Maximization steps. The EM algorithm is illustrated in Algorithm 3.

### 6.3 The Schema Matching Framework

In this section, we describe our uninterpreted schema matching with embedded value mapping algorithms for continuous-valued attributes and with pdf transfor-
Algorithm 3  EM Algorithm with transformation learning

Input: Mixture model \( \theta^{(x)} \), Observations \( \mathcal{Y} \), Tolerance \( tols \), Max Iterations \( max\_iter \)
Output: Transformation estimates \( a, b \)

\[
a \leftarrow 1 \{ \text{Initialize } a, \text{ see Section 6.2.3} \}
b \leftarrow 0 \{ \text{Initialize } b, \text{ see Section 6.2.3} \}
tmp_{\alpha} \leftarrow 1 \{ \text{temporary variables} \}
tmp_{\beta} \leftarrow 0 \{ \text{temporary variables} \}
tmp_{\loglik} \leftarrow 0 \{ \text{temporary variables} \}
\]

\begin{align*}
\text{for } i = 1 \text{ to } max\_iter \text{ do} \\
p_z \leftarrow \text{estep}(\mathcal{Y}, \theta^{(x)}, tmp_{\alpha}, tmp_{\beta}); \{ \text{Expectation-step, Eq. (6.9)} \} \\
tmp_{\beta} \leftarrow \text{mstep}(p_z, \theta^{(x)}, tmp_{\alpha}); \{ \text{Update b for fixed a, Eq. (6.15)} \} \\
p_z \leftarrow \text{estep}(\mathcal{Y}, \theta^{(x)}, tmp_{\alpha}, tmp_{\beta}); \{ \text{Expectation-step, Eq. (6.9)} \} \\
tmp_{\alpha} \leftarrow \text{mstep}(p_z, \theta^{(x)}, tmp_{\beta}); \{ \text{Update a for fixed b, Eq. (6.23)} \} \\
\loglik \leftarrow \text{computeLogLikelihood}(\mathcal{Y}, \theta^{(x)}, tmp_{\alpha}, tmp_{\beta}); \{ \text{log-likelihood for new estimates} \} \\
tmp_{\loglik} \leftarrow \text{computeLogLikelihood}(\mathcal{Y}, \theta^{(x)}, a, b); \{ \text{log-likelihood for old estimates} \} \\
\text{if abs}(\loglik - tmp_{\loglik}) < tols \text{ then } \{ \text{If converged break} \} \\
\quad \text{break;} \\
\text{else } \{ \text{continue} \} \\
\quad a \leftarrow tmp_{\alpha} \\
\quad b \leftarrow tmp_{\beta} \\
\text{end if} \\
\text{end for}
\end{align*}

The inputs to our algorithms are two tables while the output is a schema match and transformation mappings for matching attributes. Our algorithms perform iterative hill climbing on a global objective function based on a dissimilarity metric, as elaborated in this section. Suitable dissimilarity metrics (log-likelihood or Euclidean distance) for continuous-valued attributes with embedded pdf transformations are discussed below.

6.3.1 First-order dissimilarity objectives for continuous-valued attributes with transformation learning

Consider matching two continuous-valued attributes \( X \) and \( Y \), from different schemas. Let attribute \( X \), with a set of observations \( \mathcal{X} = \{ x_1, \ldots, x_{N_x} \} \), be modeled by
a mixture model given by \( f_X(x; \theta^{(x)}) = \sum_{k=1}^{K_x} \alpha_k^{(x)} f_{X|k}(x, \theta_k^{(x)}) \), where \( K_x \) is the number of components, \( \theta_k^{(x)} = (\mu_k^{(x)}, \sigma_k^{(x)}) \) are the parameters, \( \mu_k^{(x)} \) is the mean and \( \sigma_k^{(x)} \) is the standard deviation of the \( k^{th} \) component respectively, \( \alpha_k^{(x)} \geq 0 \forall k \) and \( \sum_k \alpha_k^{(x)} = 1 \). Furthermore, let attribute \( Y \), with a set of observations \( Y = \{y_1, \ldots, y_{Ny}\} \), be modeled by a mixture model given by \( f_Y(y; \theta^{(y)}) = \sum_{k=1}^{K_y} \beta_k^{(y)} f_{Y|k}(y, \theta_k^{(y)}) \), where \( K_y \) is the number of components, \( \theta_k^{(y)} = (\mu_k^{(y)}, \sigma_k^{(y)}) \) are the parameters, \( \mu_k^{(y)} \) is the mean and \( \sigma_k^{(y)} \) is the standard deviation of the \( k^{th} \) component respectively, \( \beta_k^{(y)} \geq 0 \forall k \) and \( \sum_k \beta_k^{(y)} = 1 \).

Suppose, that the transformation estimates \( a^{(xy)} \), \( b^{(xy)} \) and \( c^{(yx)} \), \( d^{(yx)} \) were found using the EM algorithm with embedded transformation learning (Section 6.2) such that:

\[
Y = a^{(xy)} X + b^{(xy)}, \quad a > 0 \tag{6.29}
\]

\[
X = c^{(yx)} Y + d^{(yx)}, \quad c > 0 \tag{6.30}
\]

Therefore, the log-likelihood for \( Y \) given \( X \)'s model and transformation estimates \( a, b \) can be written as:

\[
\log L_X(Y; \theta^{(x)}, a^{(xy)}, b^{(xy)}) = \sum_{i=1}^{Ny} \log \sum_{j=1}^{K_x} \frac{\alpha_j^{(x)}}{\sqrt{2\pi a^{(xy)^2} \sigma^2_{X|j}}} e^{-\frac{(y_i - (a^{(xy)} \mu_{X|j}^{(x)} + b^{(xy)}))^2}{2a^{(xy)^2} \sigma^2_{X|j}}}.
\tag{6.31}
\]

Likewise, the log-likelihood for \( X \) given \( Y \)'s model and transformation estimates \( c, d \) can be written as:

\[
\log L_Y(X; \theta^{(y)}, c^{(yx)}, d^{(yx)}) = \sum_{i=1}^{Nx} \log \sum_{j=1}^{K_y} \frac{\beta_j^{(y)}}{\sqrt{2\pi c^{(yx)^2} \sigma^2_{Y|j}}} e^{-\frac{(x_i - (c^{(yx)} \mu_{Y|j}^{(y)} + d^{(yx)}))^2}{2c^{(yx)^2} \sigma^2_{Y|j}}}.
\tag{6.32}
\]
Thus, given a schema match matrix $M$ defined as,

$$M = [M_{ij}], \text{ where } M_{ij} = \begin{cases} 1, & \text{if } i^{th} \text{ attribute matches to } j^{th} \text{ attribute across Schemas 1 and 2} \\ 0, & \text{otherwise} \end{cases}$$

(6.33)

we can define a dissimilarity metric based on negative log-likelihood for continuous-valued attributes which embeds transformation learning as:

$$D_{LL}(M, T_o) = -\sum_{i=1}^{N_{C_1}} \sum_{j=1}^{N_{C_2}} M_{ij} \left[ \log L_i(Y^j; \theta^{(i)}, a^{(ij)}, b^{(ij)}) + \log L_j(X^i; \theta^{(j)}, c^{(ji)}, d^{(ji)}) \right]$$

(6.34)

Where,

$T_o$ = the set of learnt transformation mappings for a fixed schema match. Note, that here we have constraints that $a > 0$ and $c > 0$.

6.3.2 First-order dissimilarity objectives for continuous-valued attributes with fixed pdf transformations

In this section, we extend the our dissimilarity metrics discussed in Section 4.2.1.1 to model the dissimilarity between between two continuous attributes, when there exists a known finite set of fixed pdf transformations that can be applied between attributes. Consider matching two continuous-valued attributes $X$ and $Y$, from different schemas. Let attribute $X$, with a set of observations $X = \{x_1, \ldots, x_{N_x}\}$, be modeled by a mixture model given by $f_X(x; \theta^{(x)}) = \sum_{k=1}^{K_x} \alpha_k^{(x)} f_{X|k}(x; \theta_k^{(x)})$, where $K_x$ is the number of components, $\theta_k^{(x)} = (\mu_k^{(x)}, \sigma_k^{(x)})$ are the parameters, $\mu_k^{(x)}$ is the mean and $\sigma_k^{(x)}$ is the standard deviation of the $k^{th}$ component respectively, $\alpha_k^{(x)} \geq 0 \ \forall k$ and $\sum_k \alpha_k^{(x)} = 1$. We can then write the a posteriori probability of each mixture component, given a data sample $x_i$ as:

$$P[C_x = k|x_i; \theta^{(x)}] = \frac{\alpha_k^{(x)} f_{X|k}(x_i; \theta_k^{(x)})}{\sum_j^{K_x} \alpha_j^{(x)} f_{X|j}(x_i; \theta_j^{(x)})}, \ K = 1, \ldots, K_x.$$  \hspace{1cm} (6.35)

Let attribute $Y$, with a set of observations $Y = \{y_1, \ldots, y_{N_y}\}$, be modeled
by a mixture model given by $f_Y(y; \theta^{(y)}) = \sum_{k=1}^{K_y} \beta_k^{(y)} f_{Y|k}(y; \theta_k^{(y)})$, where $K_y$ is the number of components, $\theta_k^{(y)} = (\mu_k^{(y)}, \sigma_k^{(y)})$ are the parameters, $\mu_k^{(y)}$ is the mean and $\sigma_k^{(y)}$ is the standard deviation of the $k^{th}$ component respectively, $\beta_k^{(y)} \geq 0$ \forall k and $\sum_k \beta_k^{(y)} = 1$. Now, suppose it is known prior to schema matching that there exists a set of possible pdf transformations $T$, where

$$T = \{T_1, \ldots, T_{N_T}\}.$$  

(6.36)

There exists an affine transformation $T_z \in T$ between the two attributes such that $T_z : \mathcal{X} \mapsto \mathcal{Y}$ and $T_z^{-1} : \mathcal{Y} \mapsto \mathcal{X}$. We can then estimate the component masses for $\mathcal{Y}$ under $\mathcal{X}$’s density model and given transformation operator $T_z$, as,

$$\hat{\alpha}_{k}^{(T_z^{-1}y)} = \frac{1}{N_y} \sum_{n=1}^{N_y} P[C_x = k[T_z^{-1}[y_n]; \theta^{(x)}], \ k = 1, \ldots, K_x. \tag{6.37}$$

Likewise, we can estimate the component masses for $\mathcal{X}$ under $\mathcal{Y}$’s density model and given transformation operator $T_z$, as,

$$\hat{\beta}_{k}^{(T_zx)} = \frac{1}{N_x} \sum_{n=1}^{N_x} P[C_y = k[T_z[x_n]; \theta^{(y)}], \ k = 1, \ldots, K_y. \tag{6.38}$$

Thus, given a schema match matrix $M$ (Eq. (6.33)), we can define a dissimilarity metric utilizing Euclidean distance based on (mixture model) derived pmfs for continuous-valued attributes under a set of fixed pdf transformation $T$ as,

$$D^{EP}(M, T_o) = \sum_{i=1}^{N_{C_1}} \sum_{j=1}^{N_{C_2}} M_{ij} \sum_{k=1}^{K_i} (\alpha_k^{(i)} - \hat{\alpha}_{k}^{(T_z^{-1}(i))j})^2 + \sum_{k=1}^{K_j} (\beta_k^{(j)} - \hat{\beta}_{k}^{(T_z(j))i})^2, \tag{6.39}$$

where “EP” stands for “Estimated PMF” and where,

$N_{C_1}, N_{C_2} = \text{Number of continuous-valued attributes in Schema 1 and 2, respectively}$

$T_o = \text{the set of transformation mappings for a fixed schema match.}$

A second dissimilarity metric between two continuous-valued attributes given a pdf transformation $T_z$ can be measured based on the negative data log-likelihood. Intuitively, a model best fit to observed data and a pdf transformation will have
least negative data log-likelihood. Assuming the observations $Y = \{y_i, i = 1, \ldots, N_y\}$ are generated independent and identically distributed, the negative log-likelihood of $Y$ under $X$’s density model given a pdf transformation $T_z$, which we will use to define the negative log-likelihood based dissimilarity objective, is:

$$- \log L(T_z^{-1}[Y]; \theta^{(x)}) = - \sum_{n=1}^{N_y} \log \sum_{k=1}^{K_x} \alpha_k^{(x)} f_{X|k}(T_z^{-1}[y_n]; \theta_k^{(x)})$$

(6.40)

Thus, given a schema match matrix $M$ (as defined in Eq. (6.33)), a dissimilarity metric based on negative log-likelihood for continuous-valued attributes can be defined as,

$$D_{LL}(M, T_o) = - \sum_{i=1}^{N_{C1}} \sum_{j=1}^{N_{C2}} M_{ij} \left[ \sum_{n=1}^{N_x} \log \sum_{k=1}^{K_j} \beta_k^{(j)} f_{Y|k}(T_z^{(ij)}[x_n]; \theta_k^{(j)}) + \sum_{n=1}^{N_y} \log \sum_{k=1}^{K_i} \alpha_k^{(i)} f_{X|k}(T_z^{-1(ij)}[y_n]; \theta_k^{(i)}) \right]$$

(6.41)

where,

$N_{C1}, N_{C2} =$ Number of continuous-valued attributes in Schema 1 and 2, respectively

$N_x, N_y =$ Number of rows in Schema 1 and 2, respectively

$T_o =$ the set of transformation mappings for a fixed schema match.

### 6.3.3 Schema matching strategy

Our global objective can now be formulated as a two-dimensional minimization problem as shown in Eq. (6.42) or (6.43) where the resulting minimum score would correspond to the best (declared) schema matching.

$$D_{EP}(M, T_o) = \min_{M \in \mathcal{M}} \left\{ \min_{T_o \in \mathcal{T}} [D_{EP}(M, T_o)] \right\}$$

(6.42)

, or,

$$D_{LL}(M, T_o) = \min_{M \in \mathcal{M}} \left\{ \min_{T_o \in \mathcal{T}} [D_{LL}(M, T_o)] \right\}$$

(6.43)
Where:
$D^{EP} =$ estimated pmf based dissimilarity metric utilized (Eq. (6.39))
$D^{LL} =$ log-likelihood based dissimilarity metric utilized (Eq. (6.41))
$M =$ a fixed schema match
$\mathcal{M} =$ the search space of different schema matches
$T_a =$ a transformation mapping for a schema match
$\mathcal{T} =$ space of transformation mappings

6.4 Experimental validation

In this section, we present the results of our algorithms for schema matching for continuous-valued attributes with pdf transformations compared with several alternative methods on real-world Census Bureau datasets.

6.4.1 Transformed Continuous-valued dataset

We used the Census Bureau Public Usage Micro data Sample (PUMS) 5% datasets for the states of California (CA), New York (NY) and Texas (TX) for the year 1990. However, these datasets did not have any attributes across schemas where a pdf transformation needed to be applied. Therefore, we generated an artificial datasets from the continuous-valued attributes existing in the Census datasets. To do so, we first selected all continuous-valued attributes (26 attributes) for the states of California (CA), New York (NY) and Texas (TX). We, then, applied a set of transformations to some (10) attributes in the NY and TX datasets. The input tables to the schema matching algorithms was either a) CA and NY datasets, or, b) CA and TX datasets. We then chose $K = 20$ and averaged matching accuracy results over 50 trials where, in each trial the columns to be matched were randomly chosen ($K$ continuous-valued out of a total of 26 continuous-valued attributes). These 50 trial sets comprised our transformed continuous-valued attribute data sets and were used for the evaluation of the methods developed in Section 6.3.
6.4.2 Experimental setup

The schema matching with embedded transformation operator learning algorithm is called to perform schema matching on continuous-valued attributes. The initial schema match input to the matching algorithm is a random schema match. We compared our schema matching algorithms against four methods that, respectively, do two-opt switching of attributes to minimize squared Euclidean distance between i) pmfs, for a pair of matching attributes (The objective function in Eq. (3.3)) ii) mutual informations, for matching attribute pairs (The objective function in Kang and Naughton [39, 38]) (MI), iii) conditional entropies [34], for matching attribute pairs, (CE) and iv) entropies, for a pair of matching attributes (EN). The last three methods are consistent with the Kang and Naughton [39, 38] approach in that these three cost function do not exploit (and require) embedded value mappings, in matching a pair of discrete-valued attributes. Since these methods assume categorical-valued attributes, derived pmfs for all continuous-valued attributes (using mixture component a posteriori probabilities) were input to these methods. All matching algorithms are initialized from the same schema match.

We implemented our schema matching with embedded value mapping algorithms using R [61] 2.11 and MCLUST 3.4 [23, 21, 22]. Since the density estimation and the log-likelihood algorithm are computationally intensive, they were executed on a 224 node (Dual Intel® Xeon @ 2.67GHz) cluster. Up to 10% of the cluster CPU resources were allocated by the cluster scheduler to execute the algorithms.

We ran our experiments over a randomly chosen subset of \( n \) \( (2 \leq n \leq 20) \) continuous-valued attributes. A pdf transformation was applied to some attributes in the second schema. We randomly permuted the columns in the second schema to ensure that the algorithms cannot trivially stumble upon the correct schema matching result.

6.4.3 Evaluation Metrics

We use \( \text{Precision} = \frac{m}{n} \) to measure the accuracy of matching, where \( n \) is the total number of columns to match and \( m \) is the total number of correct matches.
Figure 6.2: Negative log-likelihood as a function of iterations for EM algorithm for transformation learning

6.3.1: Estimates for \(b\) as a function of iterations

6.3.2: Estimates for \(a\) as a function of iterations

Figure 6.3: Pdf transformation parameter estimates as a function of iterations.

6.4.4 Transformation operator learning using EM algorithm

Fig. [6.2] illustrates the negative log-likelihood as function of EM with transformation operator learning (Section [6.2]) iterations. Here, 50K random samples \((X)\) were generated from a mixture model with \(\mu = \{0, 5, 30\}\), \(\sigma^2 = \{0.5, 3, 15\}\) and \(\alpha = \{0.25, 0.25, 0.50\}\). The observations were then transformed to generate random samples \(Y = a'X + b'\), with \(a' = 2\) and \(b' = 5\). The EM algorithm for transformation learning was initialized with \(a = 1\) and \(b = 0\) and then executed \(^1\). The algorithm terminated after 27 iterations with final updates \(a = 1.999535\) and \(b = 4.978245\).

\(^1\)The EM algorithm terminated if either a) maximum iterations were reached, where max iterations = 100, or, b) log-likelihood increase is was less than tolerance, where tolerance = \(1E^{-10}\).
6.4.1: Average Precision for CA vs. NY dataset
6.4.2: Computation Time for CA vs. NY dataset

6.4.3: Average Precision for CA vs. TX dataset
6.4.4: Computation Time for CA vs. TX dataset

Figure 6.4: First-order schema matching results for continuous-valued attributes with transformation learning

Figure 6.3 illustrates the pdf transformation parameters (a, b) estimated by the EM algorithm as a function of iteration.

6.4.5 First-order schema matching on transformed continuous-valued dataset

6.4.5.1 With Transformation operator learning

Fig. 6.4 presents schema matching with transformation learning results for transformed continuous attributes datasets where the initial schema match was randomly selected. All algorithms were initialized from the same initial schema match. We increased the number of attributes in the two input tables from 2 to 10. The attributes for the continuous-valued datasets were randomly chosen from the set of 26 columns, as discussed in Section 6.4.1. For each pair of schemas, we first chose
50K rows (samples). For each of these sub-datasets, we repeated the experiment 50 times, each time randomly choosing attributes from the set of 26 (continuous-valued) attributes. Figs. 6.4.1 and 6.4.3 shows the average precision (over 50 trials) for

- log-likelihood matching with transformation learning using standard initialization (labeled EM LL SI),

- log-likelihood matching with no transformation learning using model-dependent initialization as pdf transformation (labeled LL MD), and,

- log-likelihood matching with transformation learning using model-dependent initialization (labeled EM LL MD).

Standard and model-dependent initializations are topics discussed in Section 6.2.3. For comparison, Euclidean PMF matching (FO EP) (Eq. (4.6)), the log-likelihood (FO LL) (Eq. (4.8)), first-order discrete pmf (FO DM) (Eq. (3.3)), Entropy only (EN), Kang-Naughton Mutual Information (MI) [39] and modified Kang-Naughton (labeled CE) algorithm results are also presented. Note that only EM LL SI, LL MD and EM LL MD methods integrate pdf transformation learning within schema matching.

Clearly, the FO LL and FO DM methods performs best. The FO LL method achieved 56% and 57% accuracy while the FO DM method achieved 63% and 55% accuracy for the CA vs. NY and CA vs. TX datasets, respectively, when 10 attributes were matched. Since transformations only cause mean shifts and variance scaling, i.e., the component masses are unaffected, the FO DM method was expected to perform well. The EM LL MD and LL MD methods performed second best. The EM LL MD method achieved 54% and 53% accuracy while the LL MD method achieved 47% and 53% accuracy for the CA vs. NY and CA vs. TX datasets, respectively, when 10 attributes were matched. Pdf transformation learning introduces an extra dimension of confounding since a pdf transformation between non-matching attributes could possibly reduce dissimilarity (as discussed in Section 6.2.2). Due to this effect, schema matching methods with embedded transformation learning (EM LL MD, LL MD) were unable to achieve better accuracy than schema matching methods without embedded transformation
learning (FO LL). Nevertheless, we will investigate schema matching for transformed continuous-valued attributes in greater detail in future work. The EM LL SI method performed poorly, however, this can be attributed to the fact EM algorithm was unable to estimate the pdf transformation correctly when initialized from standard initialization, i.e., $a = 1, b = 0$. All methods that do not incorporate embedded value mappings (e.g., MI, EN, CE) perform poorly in comparison to methods that incorporate embedded value mappings (FO EP, FO LL, FO DM, LL MD, EM LL MD and EM LL SI).

Figs. 6.4.2 and 6.4.4 illustrate the computation time for executing the schema matching algorithms for the three state files when tackling transformed continuous-valued attributes datasets with transformation learning. As expected, the schema matching methods which incorporate transformation learning (EM LL SI and EM LL MD) have significantly higher computation times than the versions that do not incorporate pdf transformations (FO LL, FO EP, FO DM) due to the computation complexity of the EM algorithm for transformation transformation learning.

**6.4.5.2 With fixed pdf transformations**

Fig. 6.5 presents schema matching results for transformed continuous attributes datasets where the initial schema match was randomly selected. All algorithms were initialized from the same initial schema match. A set of 6 possible pdf transformations was also input to the algorithm. We increased the number of attributes in the two input tables from 2 to 20. The attributes for the continuous-valued datasets were randomly chosen from the set of 26 columns. For each pair of schemas, we first randomly chose 50K rows (samples). For each of these sub-datasets, we repeated the experiment 50 times, each time randomly choosing attributes from the set of 26 (continuous-valued) attributes. Figs. 6.5.1 and 6.5.3 shows the average precision (over 50 trials) for Euclidean PMF matching (FO EP) (Eq. (6.39)) and the log-likelihood (FO LL) (Eq. (6.41)) metrics for the CA vs. NY and CA vs. TX datasets. For comparison, the Euclidean PMF matching without transformations (FO EP NO) (Eq. (4.6)), log-likelihood matching without transformations (FO LL NO) (Eq. (4.8)), first-order discrete Euclidean metric (FO DM) (Eq. (3.3)) [34], the Entropy only (EN), Kang-Naughton Mutual Information (MI) [39] and modified Kang-Naughton algorithm (labeled CE) results are
Figure 6.5: First-order schema matching results for transformed continuous-valued attributes with fixed pdf transformations

also presented. These methods do not include pdf transformations within schema matching.

Clearly, the FO LL dissimilarity algorithm performs best (≈ 5% better accuracy over the next best method when ≥ 10 columns are matched). The FO EP algorithm performs second best. Both these methods perform much better in comparison to the other methods with typical precision gains being ≥ 20% when ≥ 5 columns are matched for both datasets. As expected, the log-likelihood (FO LL NO) and Euclidean distance (FO EP NO) algorithms that do not incorporate pdf transformations perform relatively poorly in comparison with the versions that incorporate pdf transformations (FO EP, FO LL). The first-order pmf-based dissimilarity (FO DM, Eq. (3.3)) [34] performs poorly, with schema matching precision ranging between 30% and 50% when ≥ 10 columns are matched. Mutual information, conditional entropy and entropy methods all perform poorly even though they are supplied with soft-discretized versions of the dataset. These methods have
less than 20% matching accuracy when > 7 attributes are matched. All methods that do not incorporate pdf transformations (e.g., FO EP NO, FO LL NO, FO DM, MI, EN, CE) perform poorly in comparison to methods that incorporate pdf transformations (FO EP and FO LL). This observation demonstrates the benefit of embedding pdf transformations in determining each attribute match (Note that FO LL treats attributes as continuous-valued and does not require embedded value mappings).

Figs. 6.5.2 and 6.5.4 illustrate the computation time for executing the schema matching algorithms for the three state files when tackling transformed continuous-valued attributes datasets. As expected, the first-order Euclidean distance (FO EP) and log-likelihood methods (FO LL) have significantly higher computation times than the versions that do not incorporate pdf transformations (FO EP NO, FO LL NO) due to the need to recompute the dissimilarity for the six pdf transformations. The methods that do not incorporate pdf transformations (FO EP NO, FO DM, EN, MI, CE) have least computation times for transformed continuous-valued attributes. In addition, a significant component of the computation time of all the schema matching algorithms is spent in learning the probability density functions for all the continuous-valued attributes in both schemas in a preprocessing step.
Chapter 7

Top-K Confident Schema Matching

7.1 Introduction

Schema matching techniques developed in Chapters 3, 4, 5, and 6 utilize a locally optimal iterative strategy to optimize a global objective function to determine the schema matching. Several objective functions handling continuous-valued, categorical, mixed categorical and continuous-valued and transformed continuous-valued attributes were developed. Such schema matching techniques, therefore, return a single best (or top-1) match per attribute for a set of attributes being matched across two schemas, i.e., each attribute in the first schema is best matched to another attribute in the second schema. However, as discussed previously, certain business applications require that the schema matching algorithm return a top-K [15, 25, 24, 60] set of candidate matches per attribute to a user thereby allowing users to make the final schema match decision. We, therefore, use dis-similarity metrics, developed in previous chapters, to capture similarity between two attributes. A top-K uninterpreted schema matching strategy is then developed that significantly reduces the number of top-K schema matches returned, by performing a statistical test that accepts only the most significant (confident) matches. Experimental evaluation in Section 7.3 shows that such a strategy is effective in reducing the total number of schema matches returned to a user without compromising schema matching recall and quality. We illustrate this strategy in the sequel.
7.2 Top-K schema matching framework

In this section, we describe in detail our uninterpreted Top-K schema matching with embedded value mapping algorithms. The inputs to our algorithms are two instances of table, i.e., two schemas \( S_1 \) and \( S_2 \) with \( N_{C_1} \) and \( N_{C_2} \) attributes, respectively. The algorithm execution outputs Top-K most significant schema matches per attribute in schema \( S_1 \). In Sections 3.2.1.1 and 4.2.1.1 we described first-order dissimilarity metric model for categorical and continuous-valued attribute, respectively, for a fixed schema match. However, for a top-K schema matching strategy we require definition of the dissimilarity between two attributes, which we first, briefly, describe.

7.2.1 Dissimilarity objective between two categorical attributes

As discussed in Section 3.2.1.1 the dissimilarity metric for two categorical attributes measures how well the probability mass functions (pmfs) of matching attributes align. Let the \( i^{th} \) attribute in the first schema match to the \( j^{th} \) attribute in the second schema. Therefore, given a value mapping matrix \( V(i,j) \) between the two attributes defined as,

\[
V(i,j) = [v_{kl}^{(ij)}], \quad \text{where} \quad v_{kl}^{(ij)} = \begin{cases} 
1, & \text{if the } k^{th} \text{ value in the } i^{th} \text{ attribute in Schema 1 matches to } l^{th} \text{ value in the } j^{th} \text{ attribute in Schema 2} \\
0, & \text{otherwise}
\end{cases}
\]  

(7.1)

The first-order pmf dissimilarity between \( i^{th} \) and \( j^{th} \) categorical attributes in the two schemas can be defined as:

\[
D^{EU}(i,j) = \min_{V \in \mathcal{V}} \sum_{k=1}^{N_{V_i}} \left[ \sum_{l=1}^{N_{V_j}} v_{kl}^{(ij)} p_l^{(j)} - \left\{ \sum_{l=1}^{N_{V_j}} v_{kl}^{(ij)} p_l^{(j)} \right\} \right]^2
\]

(7.2)

where,

\( V = \) value mapping \( \mathcal{V} = \) value mapping search space

\( N_{V_i}, N_{V_j} = \) Number of values in \( i^{th} \) and \( j^{th} \) attributes in Schemas 1 and 2, respec-
7.2.2 Dissimilarity objective between two continuous-valued attributes

The first-order dissimilarity metric for two continuous-valued attributes measures how well the probability density function (pdfs) of matching attributes align. As discussed in Section 4.2.1.1, we can measure such dissimilarity using the Euclidean distance or log-likelihood metric. Let the $i^{th}$ attribute in the first schema match to the $j^{th}$ attribute in the second schema. A dissimilarity metric utilizing Euclidean distance based on (mixture model) derived pmfs for these two continuous-valued attributes is defined as:

$$D^{EP}(i,j) = \sum_{k=1}^{K_i} (\alpha_k^{(i)} - \hat{\alpha}_k^{(j)})^2 + \sum_{k=1}^{K_j} (\beta_k^{(j)} - \hat{\beta}_k^{(i)})^2. \quad (7.3)$$

Similarly, a dissimilarity metric based on negative log-likelihood between the $i^{th}$ and $j^{th}$ attributes in the two schemas can be defined as,

$$D^{LL}(i,j) = -\sum_{n=1}^{N_x} \log \sum_{k=1}^{K_i} \beta_k^{(j)} f_{Y|k}(x_n; \theta_k^{(j)}) + \sum_{n=1}^{N_y} \log \sum_{k=1}^{K_j} \alpha_k^{(i)} f_{X|k}(y_n; \theta_k^{(i)}) \quad (7.4)$$

where,

$N_x, N_y$ = Number of rows in Schema 1 and 2, respectively

7.2.3 Top-K schema matching criterion

Let $D_{ij}$ be the dissimilarity score obtained when the $i^{th}$ attribute in schema $S_1$ is matched to the $j^{th}$ attribute in Schema $S_2$ (as discussed in Sections 7.2.1 and 7.2.2). Top-K schema matching for the $i^{th}$ attribute in Schema $S_1$ can be then defined recursively as follows. For $K = 1$, the K-best schema match $M^*_i$ for the $i^{th}$
attribute in Schema $S_1$ is defined as:

$$M_i^1 = \{j | D_{ij} < D_{ij'}, \forall j' \in N_{C_2} : j \neq j'\} \quad (7.5)$$

Then, given the best $(K - 1)$ schema matches $\{M_i^1, M_i^2, \ldots, M_{K-1}\}$ for the $i^{th}$ attribute in Schema $S_1$, the $K^{th}$ best schema match $M^K_i$ is defined as a schema match of minimum dissimilarity over the attributes of schema $S_2$ that differs from each of the schema matches $\{M_i^1, M_i^2, \ldots, M_{K-1}\}$. Therefore, given the top-$K$ schema matches for the $i^{th}$ attribute of schema $S_1$, any schema match $M^i \notin \{M_i^1, M_i^2, \ldots, M_K\}$ satisfies:

$$\max_{1 \leq n \leq K} D_{iM'^i_n} = D_{iM'^i_K} < D_{iM'^i}, \quad (7.6)$$

i.e., $M^i$ is not a top-$K$ schema match. Therefore, for the $i^{th}$ attribute, we can define $k'$ rejected or non matches $M^{i}_{k'}$ such that:

$$M^{i}_{k'} \notin \{M^i_1, M^i_2, \ldots, M^i_K\}, k' = K + 1, \ldots, N_{C_2}, \quad (7.7)$$

where $k' = N_{C_2} - K$.

We apply this strategy to choose top-$K$ schema matches for all $N_{C_1}$ attributes in Schema $S_1$ resulting in $N_{C_1} \times K$ schema matches and $N_{C_1} \times (N_{C_2} - K)$ non-matches.

### 7.2.4 Identifying confident matches using Hypothesis Testing

As discussed earlier, returning all $N_{C_1} \times K$ top-$K$ schema matches to a user significantly increase the human capital required for performing semi-automated schema matching. However, under the assumption that the $N_{C_1} \times K$ schema matches and $N_{C_1} \times (N_{C_2} - K)$ non-matches come from different populations or distributions, we can formulate a statistical hypothesis test which will only accept significant top-$K$ matches. The preliminary assumption, or null hypothesis $H_0$, is that a top-$K$ dissimilarity score for two attributes being matched is equivalent to the dissimilarity score for two attributes that do not match, i.e., top-$K$ dissimilarity scores are drawn from the same population as non-matches or rejected matches. The significance
Figure 7.1: The null distribution for dissimilarity scores of non-matches between two Schemas. The null distribution here is a normal distribution with \( \mu = 1 \) and \( \sigma = 0.25 \). The cutoff dissimilarity \( d = 0.588 \) is computed such that \( P(D \leq d) = 0.05 \), i.e., p-value = 0.05. Three accepted dissimilarity scores (diamonds) and three rejected dissimilarity scores (circles) are also shown.

The test will attempt to disprove this hypothesis by determining a p-value, a measurement of the probability that the observed dissimilarity score could have occurred by chance. The null distribution is the probability distribution when the null hypothesis is true, i.e., the probability distribution of dissimilarity scores of non-matches or rejected matches. Suppose that \( D = \{d_{i}^{\text{non-match}}\}, i = 1, \ldots, \binom{N_{C_1}}{K} \times (\binom{N_{C_2}}{K} - K) \) be the observed dissimilarity scores for the \( \binom{N_{C_1}}{K} \times (\binom{N_{C_2}}{K} - K) \) non-matches. Assuming that the dissimilarity scores for the non-matches are independently and identically distributed, the null distribution \( f(x) \) can be modeled by estimating the pdf of these observed dissimilarity scores. Therefore, given \( \{d_{i}^{\text{match}}\}, i = 1, \ldots, \binom{N_{C_1}}{K} \times K \) the dissimilarity scores for the top-K matches, we can determine the p-value for the top-K matches as:

\[
p - \text{value}(i) = \int_{-\infty}^{d_{i}^{\text{match}}} f(x) \, dx, \ i = 1, \ldots, \binom{N_{C_1}}{K} \times K. \tag{7.8}
\]

Prior to the test, a significance level \( \alpha \) is chosen, and if the p-value is less
than $\alpha$, we can conclude that the dissimilarity score for the two attributes being matched is different from the dissimilarity score for two attributes that do not match, i.e., the two attributes match. The p-value can be viewed as the estimate of the likelihood that the obtained dissimilarity score for two attributes implies a schema match. Fig. 7.1 illustrates a hypothetical null distribution estimated as a normal distribution with $\mu = 1$ and $\sigma = 0.25$. For accurate null distribution estimation, we fit the one of the following distributions namely:

- a Weibull distribution,
- a Log-normal distribution,
- a Normal distribution, or
- a Gaussian mixture model,

where, the best distribution is chosen as the one with the lowest Bayesian Information Criterion (BIC) scores (for more detail, refer to Appendix 13.3). Experimentally, the Weibull distribution and the Gaussian mixture model were found to most accurately estimate the null distribution.

### 7.2.5 Top-K schema matching strategy

The top-K schema matching strategy is illustrated in Algorithm 4. The inputs to the algorithms are two instances of tables. The output of the algorithm is the $n$ top most likely schema matches per attribute, such that $n \leq K$. In the pseudocode of Algorithm 4, the double “for” loop iterates and computes the top-K schema matches for each attribute in Schema 1 based on the dissimilarity score illustrated in Sections 7.2.1 and 7.2.2. The dissimilarity scores for the rejected schema matches or non-matches are used to estimate the null distribution. Once the null distribution is estimated, the matching algorithm iterates over the top-K schema matches and rejects whose p-value is less than the significance level.

### 7.2.6 Partial matching using top-K methods

Partial matching, on two schemas, implies that each attribute in the first schema matches to a unique attribute in the second schema or remains unmatched, and
vice versa. This corresponds to one of the most general and difficult settings under which schemas must be matched since it is neither known \textit{a priori} to matching which subset of attributes in one schema match to attributes in the second schema nor is it known how many attributes in one schema match to attributes in the second schema. Kang and Naughton [39] proposed using a normalized distance measure with a threshold parameter for performing partial schema matching, however, the choice of the threshold was arbitrary. However, it is reasonable to assume that:

- the dissimilarity scores for all top-K matches of an attribute in one schema which does not have a matching attribute in the other schema should have similar distribution to the null distribution, i.e., the p-value for all top-K matches of such attributes should be greater than \(\alpha\), and,

- the dissimilarity scores for a subset of the top-K matches of an attribute
in one schema which does, in fact, have a matching attribute in the other
schema are not drawn from the null distribution, i.e., the p-value for some
top-K matches of such attributes is less than \( \alpha \).

Provided there is accurate estimation of the null distribution (i.e., distribution
of non-matches), Top-K matching with hypothesis testing should naturally facili-
tate partial schema matching. We will illustrate this approach in Section 7.3.6.

7.3 Experimental Validation

In this section, we present the results of top-k schema matching for categorical
and continuous-valued attributes compared with several alternative methods on
real-world Census Bureau datasets

7.3.1 Datasets

We used the Census Bureau Public Usage Micro data Sample (PUMS) 5% datasets
for the states of California (CA), New York (NY) and Texas (TX) for the year 1990.
The common schema describing each of these tables consists of 241 attributes. Ta-
ble 3.1 summarizes these datasets. For the first set of experiments, we chose 50
attribute pairs from the two tables, selecting 50 attributes that are close in entropy
as discussed in Section 3.3.1. We, then, chose \( N = 20 \) and averaged matching ac-
curacy results over 50 trials where, in each trial the columns to be matched were
randomly chosen (N categorical-valued out of a total of 50 categorical-valued at-
tributes closest in entropy). These 50 trial sets comprised our categorical-valued
attribute data sets and were used to evaluate top-K schema matching for cat-
egorical attributes. For the second set of experiments, we chose \( N = 20 \) and
averaged matching accuracy results over 50 trials where, in each trial the columns
to be matched were randomly chosen (K continuous-valued out of a total of 26
continuous-valued attributes). These 50 trial sets comprised our continuous-valued
attribute data sets and were used for the evaluation top-K schema matching for
continuous-valued attributes.
7.2.1: Null distribution for EP scores. A lognormal distribution with $\log(\mu) = 0.57$, $\log(\sigma) = 0.65$ was fit.

7.2.2: Null distribution for LL scores. A Weibull distribution with $k = 0.66$, $\lambda = 130.23$ was fit.

Figure 7.2: Estimated and true null distributions for EP and LL dissimilarity scores.

7.3.2 Experimental setup

The top-K schema matching with embedded value mapping algorithm is called to perform schema matching on categorical or continuous-valued attributes. For categorical attributes, we compared our top-K matching algorithm with a version that measures dissimilarity based on entropy difference (EN). All matching algorithms return top-K, where $K = 5$, schema matches per attribute. We ran our experiments over 20 randomly chosen attributes.

We implemented our schema matching with embedded value mapping algorithms using R [61], MCLUST 3.4 [23, 21, 22] and MASS [67]. Fig. 7.2.1 illustrates the estimated null distribution of the EP dissimilarity objective, which was modeled as a log-normal distribution with $\log(\text{mean}) = 0.57$, $\log(\sigma) = 0.65$. The algorithms were executed on a 224 node (Dual Intel® Xeon @ 2.67GHz) cluster. Up to 2% of the cluster CPU resources were allocated by the cluster scheduler to execute the algorithms.
7.3.3 Evaluation Metrics

To measure the accuracy of the top-K schema matching, we use \( \text{Recall} = \frac{m}{n} \), where \( n \) is the total number of attributes being matched and \( m \) is the total number of correct matches within the set of top-K matches that are returned by the schema matching algorithm.

To evaluate the quality of match of top-K schema matching, we use \textit{Mean Reciprocal Rank} (MRR) defined as,

\[
\text{MRR} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\text{rank}_i},
\]

where, \( \text{rank}_i = \text{rank in top-K for attribute } i \), and \( n = \text{total attributes being matched} \).

To evaluate the rejection capability of the hypothesis testing, we use \textit{Accept Rate} defined as,

\[
\text{Accept Rate} = \frac{m}{K \times n},
\]

where,
\( n = \text{total number attributes being matched} \),
\( m = \text{number of significant matches for all attributes detected by hypothesis testing at preset significance level} \),
\( K = \text{maximum number of matches per attribute, i.e., top-K matches} \).

To evaluate “partial” schema matching, we measure the performance of the top-K matching algorithm in correctly identifying schema matches and non-matches. Therefore, we use “Specificity” and “Sensitivity” to measure the performance of the top-K matching algorithm which are defined as:

\[
\text{Sensitivity} = \frac{\text{Number of true schema matches}}{\text{Number of true schema matches} + \text{Number of false non-matches}}
\]

\[
\text{Specificity} = \frac{\text{Number of true non-matches}}{\text{Number of true non-matches} + \text{Number of false schema matches}}
\]
where,
true match = a matching attribute pair identified as a schema match
true non-match = two attributes that do not match identified as not a schema match
false schema match = two attributes that do not match identified as a schema match
false non-match = a matching attribute pair identified as not a schema match

### 7.3.4 Top-K schema matching on Categorical-valued attributes

Figs. 7.3 and 7.4 presents the results of top-K, where K=5, schema matching for the CA vs. NY and CA vs. TX datasets. For both datasets, 20 categorical attributes were matched across the two schemas. The attributes for the categorical-valued datasets were randomly chosen from a set of 50 attributes closest in entropy (as discussed in Section 3.3.1). For each pair of schemas, we first randomly chose 250K rows (samples). For each of these sub-datasets, we repeated the experiment 50 times, each time randomly choosing 20 attributes from the set of 50 categorical-valued attributes. The top-K matching methods selected the top-5 matches for each attribute in the first schema. The significance level for the hypothesis test was 0.05. Fig. 7.3 shows the average recall and mean reciprocal rank (over 50 trials) for the first-order pmf based (FO, Eq. (7.2)) top-K matching, at $\alpha = 0.05$ significance level. For comparison, schema matching results for a top-K strategy utilizing entropy difference (EN) is also presented.

Clearly, the first-order pmf top-K method performs best with $\geq 70\%$ recall within top-3 matching and $\approx 80\%$ recall at top-5 matching, i.e., on average the FO method correctly matched 14 attributes out of 20 attributes within top-3 schema matches. For, both datasets the FO top-K strategy is able to achieve higher recall at top-4 matching than the top-5 matching recall for the EN strategy. Furthermore, Figs. 7.3.2 and 7.3.4 shows that the FO top-k strategy achieves 20% higher mean reciprocal rank than the EN top-k strategy at all top-K matching. This clearly demonstrates that the FO top-K methods returns greater number of cor-
7.3.1: Average Recall for CA vs. NY dataset 7.3.2: Average Mean Reciprocal Rank for CA vs. NY dataset

7.3.3: Average Recall for CA vs. TX dataset 7.3.4: Average Mean Reciprocal Rank for CA vs. TX dataset

Figure 7.3: First-order Top K schema matching results for categorical-valued attributes at 0.05 significance level

correct schema matches at a higher rank than the EN top-K method. For example, the FO method returns the correct schema match, on average, within the top-2 matches for both datasets while the EN method returns the correct schema match, on average, within the top-3 matches. This clearly demonstrates that the FO top-K method is better suited for uninterpreted top-K schema matching. In addition, matching results for partial schema matching, discussed in Sections 7.3.6 and 7.3.7 will further demonstrate this greater discriminative capability of embedded value mapping methods for top-K schema matching.

Fig. 7.4 illustrates the average attribute accept rate for top-K schema matching for categorical attributes. Again, the FO top-K method performs best with ≈ 70% accept rate at top-5 matching, i.e., ≈ 30% of top-5 matches are rejected. Therefore, for top-5 matching on 20 attributes, a total of 30 attribute matches are rejected by the FO top-K method. The EN method has ≈ 10% higher average attribute accept rate for top-K matching. Thus, the FO top-K method is best
7.4.1: Average column accept rate for CA vs. NY dataset

7.4.2: Average column accept rate for CA vs. TX dataset

Figure 7.4: Average top-K accept rate for first-order top-K schema matching for categorical-valued attributes at 0.05 significance level

7.5.1: Effect of rows on average recall for CA vs. TX dataset

7.5.2: Effect of rows on average Mean Reciprocal Rank for CA vs. TX dataset

Figure 7.5: Effect of rows on first-order Top K schema matching results for categorical-valued attributes at 0.05 significance level

suited for underinterpreted top-K schema matching for categorical attributes since it is more discriminative (low accept rate), has better accuracy (higher recall) and best quality (higher MRR) of matching.

Fig. 7.5 illustrates the effect of rows on recall and mean reciprocal rank for top-K schema matching for categorical attributes on the CA vs. TX datasets. Recall and mean reciprocal rank for 50K tuples (labeled “FO 50K” and “EN 50K”), 100K tuples (labeled “FO 100K” and “EN 50K”) and 250K tuples (labeled “FO 250K” and “EN 100K”) are shown for the FO and EN top-K methods. As earlier, the FO top-K method performed best with $\geq 70\%$ recall within top-3 matching and $\approx 80\%$ recall at top-5 matching for all tuples, i.e., on average the FO method
correctly matched 14 attributes out of 20 attributes within top-3 schema matches. Similarly, the FO top-k strategy achieved 20% higher mean reciprocal rank than the EN top-k strategy for all tuple sizes.

7.3.5 Top-K Schema Matching on Continuous-valued attributes

Figs. 7.6 and 7.7 presents the results of top-K, where K=5, schema matching for the CA vs. NY and CA vs. TX datasets. For both datasets, 20 continuous-valued attributes were matched across the two schemas. The attributes for the continuous-valued datasets were randomly chosen from a set of 26 attributes (as discussed in Section 7.3.1). For each pair of schemas, we first randomly chose 100K rows (samples). For each of these sub-datasets, we repeated the experiment 50 times, each time randomly choosing 20 attributes from the set of 26 continuous-valued attributes. The top-K matching methods selected the top-5 matches for each attribute in the first schema. The significance level for the hypothesis test was 0.05. Fig. 7.6 shows the average recall and mean reciprocal rank (over 50 trials) for the estimated-pmf (EP, Eq. (7.3)) and log-likelihood (LL, Eq. (7.4)) based top-K matching at $\alpha = 0.05$ significance level. For comparison, schema matching results for a top-K strategy utilizing first-order pmf (FO, Eq. (7.2)) is also presented.

Clearly, the EP top-K and FO top-K perform best. The EP and FO top-K methods had $\geq 70\%$ recall within top-3 matching, i.e., on average the FO and EP methods correctly matched 14 attributes out of 20 attributes within top-3 schema matches. For, both datasets the FO top-K strategy is able to achieve higher recall at top-4 matching than the top-5 matching recall for the EN strategy. The LL top-K method had poor performance for both datasets though this is can be attributed to inaccurate pdf estimation of the null distribution, as illustrated in Fig. 7.2.2. Furthermore, Figs. 7.6.2 and 7.6.4 shows that both the FO and EP top-k methods achieve high mean reciprocal rank. For example, the EP top-K method had 80% MRR for the CA vs. TX dataset while the FO top-K method had 90% MRR for the CA vs. NY datasets at top-3 matching. Moreover, both methods returns the correct schema match, on average, within the top-2 matches for both datasets.
Moreover, the EP method on average achieved 70% recall within top-2 matching.

Fig. 7.7 illustrates the average attribute accept rate for top-K schema matching for categorical-valued attributes. The LL top-K method was most selective, however, this method achieved poor recall. Overall, the EP method has low attribute accept rate and good recall for both datasets. For example, the EP top-K method has $\approx 50\%$ accept rate at top-5 matching while achieving $\geq 70\%$ match recall, i.e., $\approx 50\%$ of top-5 matches are rejected. Therefore, for top-5 matching on 20 attributes, a total of 50 attribute matches are rejected by the EP top-K method. The FO top-K method is not, however, as selective. Thus, the EP top-K method is best suited for uninterpreted top-K schema matching for continuous-valued attributes since it is more discriminative (low accept rate), has better accuracy (higher recall) and best quality (higher MRR) of matching.

Fig. 7.8 illustrates the effect of rows on recall and mean reciprocal rank for top-K schema matching for continuous-valued attributes on the CA vs. TX datasets.
7.7.1: Average column accept rate for CA vs. NY dataset

Figure 7.7: Average top-K accept rate for first-order top-K schema matching for continuous-valued attributes at 0.05 significance level

7.7.2: Average column accept rate for CA vs. TX dataset

7.8.1: Effect of rows on average recall for CA vs. TX dataset

7.8.2: Effect of rows on average Mean Reciprocal Rank for CA vs. TX dataset

Recall and mean reciprocal rank for 50K tuples (labeled “EP 50K”, “LL 50K” and “FO 50K”), 100K tuples (labeled “EP 100K”, “LL 100K” and “FO 100K”) and 250K tuples (labeled “EP 250K”, “LL 250K” and “FO 250K”) are shown for the EP, LL and FO top-K methods. As earlier, the EP and FO top-K method performed best with $\geq 70\%$ recall within top-3 matching and $\approx 80\%$ recall at top-5 matching for all tuples, i.e., on average the EP method correctly matched 14 attributes out of 20 attributes within top-3 schema matches. Similarly, the EP top-k strategy achieved 10% higher mean reciprocal rank than the FO top-k strategy for all tuple sizes.
Figure 7.9: First-order partial schema matching results for categorical attributes at 0.01 significance level

7.3.6 Partial schema matching using Top-K methods for Categorical-valued attributes

Fig. 7.9 present partial schema matching results for categorical-valued attributes using top-K schema matching. Two schemas, each having 12 attributes, were input to the algorithms. We increased the number of matching attributes within the two input tables from 2 to 10. The attributes for the categorical-valued datasets were randomly chosen from the set of 50 columns, as discussed in Section 7.3.1. For each pair of schemas, we first randomly chose 250K rows (samples). For each of these sub-datasets, we repeated the experiment 50 times, each time randomly choosing attributes from the set of 26 (continuous). The top-K matching methods select the top-3 matches for each attribute in the first schema. The significance level for the hypothesis test was 0.01. Figs. 7.9.1 and 7.9.3 shows the average specificity and sensitivity of partial matching for the first-order pmf (FO) and entropy difference
(EN) top-K methods for the CA vs NY and CA vs. TX datasets, respectively. Clearly, the FO top-K method is best suited for partial matching since it has high sensitivity and specificity, i.e., high precision of finding attribute matches and high precision of rejecting non-matching attributes. On average, the sensitivity of the FO top-K method is $\approx 15\%$ better than the EN method. Comparatively, the EN top-K method has high sensitivity but very low specificity, i.e., when only 2 attributes within the 12 attributes match in the two schemas, the EN method will not only find matches for the 2 attributes that have matches in the other schema but will also find matches for a large proportion of attributes in the first schema that do not have matches in the second schema. Figs. 7.9.2 and 7.9.4 presents the average mean reciprocal rank for the CA vs NY and CA vs TX datasets. The FO top-K method overall has best mean reciprocal rank over the set of 2 to 10 matching attributes.

7.3.7 Partial schema matching using Top-K methods for continuous-valued attributes

Fig. 7.10 present partial schema matching results for continuous-valued attributes using top-K schema matching. Two schemas each having 12 attributes were input to the algorithms. We increased the number of matching attributes within the two input tables from 2 to 10. The attributes for the continuous-valued datasets were randomly chosen from the set of 26 columns. For each pair of schemas, we first randomly chose 100K rows (samples). For each of these sub-datasets, we repeated the experiment 50 times, each time randomly choosing attributes from the set of 26 (continuous). The top-K matching methods selected the top-3 matches for each attribute in the first schema. The significance level for the hypothesis test was 0.01. Figs. 7.10.1 and 7.10.3 shows the average specificity and sensitivity of partial matching for the estimated-pmf (EP, Eq. (7.3)), log-likelihood (LL, Eq. (7.4)) and first-order pmf (FO) top-K methods for the CA vs NY and CA vs. TX datasets, respectively. Clearly, the EP top-K method is best suited for partial matching since it has high sensitivity and specificity, i.e., high precision of finding attribute matches and high precision of rejecting non-matching attributes. The FO top-K method has high sensitivity but very low specificity, i.e., when only 2 attributes
Figure 7.10: First-order partial schema matching results for continuous-valued attributes at 0.01 significance level

within the 12 attributes match in the two schemas, the EN method will not only find matches for the 2 attributes that have matches in the other schema but will also find matches for a large proportion of attributes in the first schema that do not have matches in the second schema. The LL method performs second best. Figs. 7.10.2 and 7.10.4 presents the average mean reciprocal rank for the CA vs NY and CA vs TX datasets. The EP top-K method overall has best mean reciprocal rank over the set of 2 to 10 matching attributes.
Future Work

8.1 Introduction

In this section, we look at some of the future directions that can be pursued that build upon the schema matching techniques developed thus far. These directions aim at improving automated and semi-automated schema matching techniques further, allowing such methods to be applied to tackle schema matching under a general business-oriented environment.

8.2 Second-order schema matching

In this dissertation, we have experimentally validated first-order schema matching for categorical, continuous-valued, mixed (categorical and continuous-valued) attributes in Sections 3.3.5, 4.3.5 and 5.3.3 respectively. Furthermore, we have evaluated second-order schema matching, which exploits statistical dependencies between columns, for categorical attributes as shown in Section 3.3.5. It was demonstrated that, provided there is accurate frequency-count estimation of the second-order pmfs, the second-order method should perform best. The second-order methods suitable for continuous-valued attributes, which exploit statistical dependencies between joint pdfs, are described in Section 4.2.1.2. Similarly, we can also extend Euclidean-distance metrics (Eqs. (3.5) and (4.14)) to handle second-order schema matching for mixed (continuous-valued and categorical) at-
tributes. Here, care must be taken especially when an attribute pair consists of one continuous-valued and one discrete-valued attribute. For example, to handle such pair-wise matching (a continuous-valued and discrete-valued attribute pair), we could use the mixture component a posteriori probabilities to generate a derived attribute pair. Schema matching can then be performed using the second-order pmf approach developed in Section 3.2.1.2. In Section 6.3.1, we outlined a first-order method for performing schema matching between continuous-valued attributes with automatic pdf transformation learning. Since, the Expectation-Maximization algorithm (please see Section 6.2) can learn pairwise transformations, we can develop second-order methods to perform schema matching between continuous-valued attributes while embedding transformation learning. However, the second-order methods are computationally more expensive. Nevertheless, we will investigate second-order schema matching in future work.

8.3 Top-K schema matching

In Chapter 7, we developed and experimentally validated first-order top-K schema matching for categorical and continuous-valued attributes. These approaches can easily be utilized to perform schema matching for mixed (continuous-valued and categorical) and mixed quantized (continuous-valued, quantized and categorical) attributes. Moreover, top-K methods can also be used to handle continuous-valued attributes with pdf transformations. Investigating the efficacy of top-K methods for matching these different attribute types is another area of future work. In Sections 7.3.6 and 7.3.7 “partial matching” using top-K methods was evaluated. However, due to small sizes of input datasets and since only top-3 schema matches were used, it is possible that the estimation of the null distribution is not very accurate. Investigating this approach on larger input datasets would provide a better estimate of the suitability of top-K methods for partial matching.

8.4 “Many-to-Many” schema matching

In many-to-many schema matching, each attribute in schema 1 must be matched to a set of attributes in schema 2 or remain unmatched, and vice versa. This is the
most difficult and general problem in schema matching since one does not know beforehand which attributes in one schema match with which in the other schema. Furthermore, it is also not known beforehand that a) how many attributes in one schema have matching attributes in the other schema, and b) how many attributes in one schema do not have matching attributes in the other schema. Solving such schema matching by optimizing dissimilarity metrics for fixed schema matching (e.g., methods discussed in Sections 3.2, 4.2 and 5.2) is not feasible. However, top-K schema matching methods, discussed in Section 4.2, naturally handles “one-to-many” and “many-to-many” schema matching since top-K methods return a set of matching attributes in the second schema for every attribute in the first schema and there is no constraint that the same attribute in the second schema cannot be matched to multiple attributes in the first schema. Furthermore, hypothesis testing rejects those top-K schema matches that do not occur by chance, i.e., matches that have dissimilarity scores that are more likely to come from the population of dissimilarity scores for non-matches. Ideally, attributes in one schema that do not have matching attributes in the second schema should, therefore, have all their top-K schema matches rejected. Thus, top-K uninterpreted schema matching should be suitable to handle “one-to-many” and “many-to-many” schema matching, provided the null distribution can be accurately estimated. We will investigate using top-K schema matching to handle “many-to-many” schema matching in future work.
Conclusions

We have proposed a fine-grained uninterpreted “onto” schema matching criterion, based on matching probability density and probability mass functions between attributes that requires *embedded* learning of value mappings between the attributes, similar to previous work [35, 34] which considered only discrete-valued attributes. We have developed an iterative descent algorithm for our matching objective and demonstrated that this technique achieves greater attribute matching accuracies than previous methods without substantial increase in computational cost. Moreover, we extend this approach to handle schema matching when two continuous-valued attributes requires the learning and application of an (a priori unknown) affine pdf transformation before matching can be performed. Furthermore, we have proposed a fine-grained uninterpreted “top-K” schema matching criterion that finds top-K schema matches by matching pdfs and pmfs between attributes while *embedding* learning of value mappings between attributes. We have developed an algorithm that incorporates statistical hypothesis testing to accept only statistically significant top-K matches and demonstrated that this technique achieves high attribute matching recall and quality with substantial reduction in accepted top-K matches. Lastly, we have applied top-K schema matching methods to investigate “partial matching”, which demonstrates that the uninterpreted schema matching metrics developed are fairly discriminative and useful for such matching. Our approach has been validated through extensive experiments on real-world datasets. Table 9.1 summarizes the overall “one-to-one” schema matching accuracy of the different schema matching methods for categorical, continuous-valued and mixed
(continuous-valued, categorical and quantized) datasets. Similarly, Table 9.2 summarizes the overall “top-K” schema matching accuracy of the different schema matching methods for categorical and continuous-valued datasets.

Table 9.1: Qualitative summary of “one-to-one” schema matching accuracy for the different algorithms

<table>
<thead>
<tr>
<th>Overall Accuracy</th>
<th>Categorical Dataset</th>
<th>Continuous Dataset</th>
<th>Mixed Dataset</th>
<th>Mixed (Quantized) Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best (&gt; 75%)</td>
<td>SO DM</td>
<td>FO LL</td>
<td>FO EU, FO EU LL</td>
<td>FO EU, FO EU LL</td>
</tr>
<tr>
<td>Acceptable (60-75%)</td>
<td>FO DM</td>
<td>FO EP</td>
<td></td>
<td>FO DM</td>
</tr>
<tr>
<td>Poor (40-60%)</td>
<td>MI</td>
<td>FO DM</td>
<td>FO DM, FO LL</td>
<td></td>
</tr>
<tr>
<td>Unacceptable (&lt; 20%)</td>
<td>EN, CE</td>
<td>EN, MI, CE</td>
<td>EN, MI, CE</td>
<td>FO LL, MI, CE, EN</td>
</tr>
</tbody>
</table>

Table 9.2: Qualitative summary of “top-K” schema matching accuracy for the different algorithms

<table>
<thead>
<tr>
<th>Overall Accuracy</th>
<th>Overall Recall</th>
<th>Overall MRR</th>
<th>Categorical Dataset</th>
<th>Continuous Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>&gt; 80%</td>
<td>&gt; 50%</td>
<td>FO</td>
<td>FO, EP</td>
</tr>
<tr>
<td>Acceptable</td>
<td>60-80%</td>
<td>40-60%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poor</td>
<td>&lt; 80%</td>
<td>&lt; 40%</td>
<td>EN</td>
<td>LL</td>
</tr>
</tbody>
</table>
Appendix A

Important Definitions

We provide definitions for the likelihood function, the squared Euclidean distance and relative entropy that are utilized by our algorithms.

**Definition** The likelihood function \[ L(X; \theta) \] for a given parameter vector \( \theta \) and set of observations \( X = \{x_1, \ldots, x_n\} \) for a continuous-valued attribute, assuming that they are generated independently and identically distributed (i.i.d.) according to the attribute’s density function, \( f(x; \theta) \), is

\[
L(X; \theta) = \prod_{i=1}^{n} f(x_i; \theta) \quad (A.1)
\]

**Definition** The Euclidean distance between two pmfs, \( P = \{p_1, \ldots, p_n\} \) and \( Q = \{q_1, \ldots, q_n\} \) is

\[
D_E = \sqrt{(p_1 - q_1)^2 + \ldots + (p_n - q_n)^2} \quad (A.2)
\]

**Definition** The Relative Entropy or Kullback-Leibler Distance between two pmfs, \( P = \{p_1, \ldots, p_n\} \) and \( Q = \{q_1, \ldots, q_n\} \) is

\[
D(p \parallel q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i} \quad (A.3)
\]

where it is assumed \( q_i > 0 \) whenever \( p_i > 0 \).
Likelihood function, EM algorithm for parameter estimation, and model order selection

B.1 A posteriori probability of membership in a component

Consider an attribute X modeled by a GMM $\theta = \{\alpha_j, (\mu_j, \sigma_j), j = 1, \ldots, K\}$, with component random variable C. For a given data realization (data entry) $X = x$, the a posteriori probability that $x$ belongs to each of the components is:

$$P[C = j] = \frac{\alpha_j f_{X|j}(x; \theta_j)}{f_X(x; \theta)} = \frac{\alpha_j f_{X|j}(x; \theta_j)}{\sum_{k=1}^{K} \alpha_k f_{X|k}(x; \theta_k)}, \quad j = 1, \ldots, K. \quad (B.1)$$

This is essentially a “soft” (probabilistic) quantization of X to the set of mixture components. The a posteriori probability of membership is used to:

- define the dissimilarity metrics for continuous-valued attributes based on derived pmfs as discussed in Sections 4.2.1 and 5.2.2 and,

- discretize continuous-valued attributes before they are input to the Kang- Naughton mutual information, conditional entropy and entropy-only algorithms.
B.2 Likelihood function and EM algorithm for parameter estimation

Consider a single attribute $X$ with $N$ data entries $X = \{x_n, n = 1, \ldots, N\}$. Assuming that the entries were generated i.i.d., we have the data log-likelihood:

$$\log f_X(X; \theta) = \log \prod_{n=1}^{N} f_X(x_n; \theta) = \sum_{n=1}^{N} \log \sum_{j=1}^{K} \alpha_j f_{X|j}(x_n; \theta_j). \quad (B.2)$$

The maximum likelihood parameter estimate $\hat{\theta}_{ML}$ cannot be found analytically. However, a locally optimal model (with respect to the likelihood function) for a fixed number of mixture components, $K$, can be found via the Expectation-Maximization (EM) algorithm [14, 48, 49, 50]. This algorithm consists of alternating Expectation and Maximization steps, starting from initial parameter estimates, with each step guaranteed to increase the data log-likelihood.

The E-step for a mixture model simply computes the mixture component a posteriori probabilities, given the current parameter estimates $\theta^{(t)}$, i.e.,

$$E-step: \quad P[C = j|x_n; \theta^{(t)}] = \frac{\alpha_j^{(t)} f_{X|j}(x_n; \theta_j^{(t)})}{\sum_{k=1}^{K} \alpha_k^{(t)} f_{X|k}(x_n; \theta_k^{(t)})}, \quad n = 1, \ldots, N, j = 1, \ldots, K. \quad (B.3)$$

The M-step re-estimates the model parameters based on the E-step quantities:

$$\alpha_j^{(t+1)} = \frac{1}{N} \sum_{n=1}^{N} P[C = j|x_n; \theta^{(t)}]$$

$$\mu_j^{(t+1)} = \frac{\sum_{n=1}^{N} x_n P[C = j|x_n; \theta^{(t)}]}{\sum_{n=1}^{N} P[C = j|x_n; \theta^{(t)}]}$$

$$\sigma_j^{2(t+1)} = \frac{\sum_{n=1}^{N} (x_n - \mu_j^{(t+1)})^2 P[C = j|x_n; \theta^{(t)}]}{\sum_{n=1}^{N} P[C = j|x_n; \theta^{(t)}]} \quad (B.6)$$

B.3 Model order selection

Standard model order selection methods (e.g. Bayesian Information Criterion [63], Akaike Information Criterion [1]) can be used to estimate the number of com-
ponents. We describe the Bayesian Information Criterion [63] which we used for selecting the number of components. A set of candidate models are obtained using EM for a range of values $k = \{k_{\text{min}}, \ldots, k_{\text{max}}\}$ which is assumed to contain the optimal model order. The number of components is then selected according to:

$$\hat{k} = \arg \min_{k_{\text{min}} \leq k \leq k_{\text{max}}} BIC(k)$$

(B.7)

where,

$$BIC(k) = -2 \log f(X; \theta(k)) + k_{\text{free}}(k) \log n$$

(B.8)

and $\theta(k)$ is the EM-obtained estimate of the $k$ component mixture model, while $k_{\text{free}}(k)$ is the number of free model parameters, i.e.,

$$k_{\text{free}}(k) = \begin{cases} 2k & \text{, if all } k \text{ components have the same (tied) variance } \sigma^2 \\ 3k - 1 & \text{, if all } k \text{ components have different variances} \end{cases}$$

(B.9)

When all $k$ components have the same (tied) variance, then $k$ mean parameters, $k - 1$ mixing probabilities ($\alpha_j$) and 1 tied variance must be estimated, i.e., $k_{\text{free}}(k) = 2k$. Similarly, when all $k$ components have different variances then $k$ mean parameters, $k - 1$ mixing probabilities ($\alpha_j$) and $k$ variances must be estimated, i.e., $k_{\text{free}}(k) = 3k - 1$. 
Appendix C

Kang and Naughton Implementation

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 [66], which reduces two mutual information matrices into a cost matrix. The Hungarian Algorithm [55] was then run over this cost matrix, resulting in an initial schema match. Two-opt switching [12] (hill climbing) was then executed starting from this initial schema match while minimizing the Kang-Naughton information-theoretic metric (mutual information, entropy, conditional entropy). The final output of this approach is the optimized schema match.
Bibliography


Anuj Jaiswal was born in Mumbai, India on February 12th, 1980. He received a Bachelor in Engineering in Electronics and Communication Engineering from Govind Ballabh Pant Engineering College, Pauri Garhwal, in 2002. Thereafter, he joined the Pennsylvania State University in 2003 and later, joined the Intelligent Information Systems laboratory under the supervision of Dr. Prasenjit Mitra and Dr. David J. Miller in 2004. He received a Master of Science in Electrical Engineering in 2006 following which he worked as a research associate for the Geovista Center, Department of Geography at the Pennsylvania State University working on knowledge representation and GIS systems. He then enrolled in the Doctor of Philosophy program at the College of Information Sciences and Technology at the Pennsylvania State University in 2007 and has been a member of the Intelligent Information Systems laboratory and the Geovista Center, where he has been a research assistant since. His research interests include machine learning, database alignment, data mining, search engines, and geographic information systems.