From Diversity-based Prediction to Better Schema Matching

Avigdor Gal
Technion - Israel Institute of Technology
Technion City
Haifa 32000, Israel
avigal@ie.technion.ac.il

Haggai Roitman
IBM Research - Haifa
Haifa 31905, Israel
haggai@il.ibm.com

Tomer Sagi
Technion - Israel Institute of Technology
Technion City
Haifa 32000, Israel
stomers7@tx.technion.ac.il

ABSTRACT

Schema matching predictors assess the quality of schema matchers in the absence of an exact match. We propose MCD (Match Competitor Deviation), a new diversity-based predictor that compares the strength of a matcher confidence in an attribute pair correspondence with respect to other correspondences that involve either attribute. We also propose to use MCD as a regulator to optimally control a balance between Precision and Recall and use it towards $1:1$ match (combining it with a similarity measure that is based on solving a maximum weight bipartite graph matching (MWBM)) and $1:n$ match (combining it with a matcher called Max-Delta). While optimizing the latter combination is straightforward, optimizing the former combined measure is known to be an NP-Hard problem. Therefore, we propose an approximation to an optimal match by efficiently scanning multiple possible matches. Using a thorough empirical study over several benchmark real-world datasets, we show that MCD outperforms other state-of-the-art predictors. We also show that the proposed schema matching algorithms significantly outperform existing schema matchers.

1. INTRODUCTION

The research in the area of schema matching [3] and related areas (*e.g.*, ontology alignment [10]) has focused for many years on the identification of high quality matchers, automatic tools for identifying correspondences among database attributes. Initial heuristic attempts (*e.g.*, COMA [6]) were followed by theoretical grounding (*e.g.*, see [12, 3]). Recently, the use of predictors to assess the quality of schema matchers in the absence of an exact match was proposed [33] and implemented in tools for dynamic ensemble weight setting and process matching [38].

Prediction is performed on a similarity matrix, in which for each pair of attributes, one of each schema, an automatic matcher provides a measure of similarity. In this work we propose MCD, a new predictor that is based on comparing the strength of a matcher confidence in a pair correspondence (a_i, b_j) with respect to other attribute correspondences that involve either a_i or b_j. Such a predictor measures the diversity in similarity among attribute pairs, interpreting high diversity as a better differentiator among true and false attribute correspondences. Our empirical evaluation indicates that MCD outperforms any other schema matching predictor in the literature so far.

The practical implication of such a finding is that pairs with high MCD values are likely to be part of a correct schema match. We show, both formally and empirically, that MCD serves as a regulator to control optimally a balance between Precision and Recall. Therefore, we propose a method for combining MCD with two known matchers. The first is based on solving a maximum weight bipartite graph matching (MWBM), aiming at $1:1$ match tasks. The second is Max-Delta [6], which returns a top-value list of correspondences for each matrix row and is particularly useful in $1:n$ matching tasks.

In this work we also tackle the challenge of identifying a match that maximizes similarity. While optimizing the latter combination is straightforward, optimizing the combined measure of MCD and MWBM is known to be an NP-Hard problem. Therefore, we propose building an optimal match by efficiently scanning multiple possible matches, using the Cross-Entropy (CE) Method [31].

Using a thorough empirical study over real-world dataset benchmarks, we show that MCD outperforms state-of-the-art predictors and MCD-based matching selection algorithms significantly outperform baseline methods.

To summarize, our contribution is threefold. We propose:

(1) a new predictor, MCD, and show its dominance over the state-of-the-art;
(2) two MCD-based methods, improving on both $1:1$ and $1:n$ selection methods; and
(3) a new method for efficiently scanning a match space.

The rest of the paper is organized as follows. We start with preliminaries, where we introduce the similarity matrix as a basic data model for schema matching (Section 2). Section 3 introduces MCD and discusses its properties. Two MCD-based methods are presented in Section 4, followed by an empirical evaluation (Section 5). We conclude with related work (Section 6) and concluding remarks (Section 7).

2. PRELIMINARIES: SCHEMA MATCHING

We now present a model for schema matching, based on [12]. Matching problems match two members of the problem domain (*schemata*) by aligning their components (*attributes*). Therefore, let S, S' be two schemata with attributes $\{a_1, a_2, \ldots, a_n\}$ and $\{b_1, b_2, \ldots, b_m\}$, respectively.

During the matching process, attribute features are utilized to deduce similarity. For example, attribute labels are used to perform string-based comparison. A matching algorithm is expected to eventually output a list of correspondences between attributes. This list is often conceptualized as a similarity matrix.

Definition 1. *let $\mathcal{S} = S \times S'$ be the set of all possible correspondences between attributes of S and S', then $M(S, S')$ is an $n \times m$* similarity matrix *over $\mathcal{S}$. $M_{i,j}$ (typically a real number in $[0, 1]$) represents a degree of similarity between the i-th and j-th attributes of S and S', respectively.*

$M(S, S')$ is a *binary* similarity matrix if for all $1 \leq i \leq n$ and $1 \leq j \leq m$, $M_{i,j} \in \{0, 1\}$. A (possibly binary) similarity matrix is the output of the matching process. For any matched schema pair (S, S'), the power-set $\Sigma = 2^{\mathcal{S}}$ is the set of all possible matches between this pair. We denote a match by $\sigma \in \Sigma$ and its cardinality by $|\sigma|$.

As an example, consider Table 1, which presents two similarity matrices for two simplified database schemata, with four and three attributes, respectively. We interpret binary similarity matrices as representing a match, where a value of 1 signifies attribute pairs that are part of a match. Therefore, the match that is represented by Table 1(bottom) is $\sigma = \langle$(cardNum, clientNum), (city, city), (checkInTime, checkInDate)$\rangle$, and its cardinality is $|\sigma| = 3$.

Matching is often a stepped process in which different algorithms, rules, and constraints are applied. Several classifications of matching steps have been proposed over the years. Following Gal and Sagi [14], we separate matchers into those that are applied directly to the problem (*first-line matchers – 1LMs*) and those that are applied to the outcome of other matchers (*second-line matchers – 2LMs*). 1LMs receive two schemata and return a similarity matrix. 2LMs receive a similarity matrix and return a similarity matrix. Among the 2LMs, we term *decision makers* those that return a binary matrix as an output. Using Table 1 once more, Table 1(top) may be the outcome of a 1LM, while Table 1(bottom) is the outcome of a 2LM decision maker, which enforces a 1 : 1 matching.

For the sake of illustration, three of the 1LMs we use in our empirical evaluation are discussed next. The Term algorithm [12] compares attribute names using, *e.g.*, edit distance and soundex, to identify syntactically similar attributes. To achieve better performance, names are preprocessed using several techniques originating in IR research. A WordNet-based algorithm [30, 16] uses abbreviation expansion and tokenization methods to generate a set of words for each attribute from its name. The resultant sets are compared with the average of their *Jiang-Conrath* similarity [18] used as the attribute similarity value. Finally, we also use the Token Path algorithm from Auto-Mapping Core (AMC) [26], which integrates node-wise similarity with structural information by comparing the syntactic similarity of the full paths from the root to a node.

Two state-of-the-art 2LMs, namely MWBM and Max-Delta are now shortly described for illustration purposes.

$S_1 \longrightarrow$ $\downarrow S_2$	cardNum	city	arrival Day	checkIn Time
clientNum	0.84	0.32	0.32	0.30
city	0.29	1.00	0.33	0.30
checkInDate	0.34	0.33	0.35	0.64

$S_1 \longrightarrow$ $\downarrow S_2$	cardNum	city	arrival Day	checkIn Time
clientNum	1	0	0	0
city	0	1	0	0
checkInDate	0	0	0	1

Table 1: Top: a similarity matrix example. Bottom: a binary similarity matrix example, representing a possible match

MWBM generates a match of a size $min(n, m)$ by solving a maximum weight bipartite graph matching problem. In the bipartite graph, nodes in each side of the graph represent attributes of one of the schemata, and the weighted edges represent the similarity measures between attributes. MWBM aims at maximizing the overall match confidence and its objective is given by:

$$\mathcal{Q}_{\mathsf{MWBM}}(\sigma, M) = \sum_{(i,j)\in\sigma} M_{i,j} \tag{1}$$

Known algorithms, *e.g.*, [15] provide the output of the MWBM matcher. Parallel implementations of MWBM exist, which provide an optimal match in $O(min(n, m)^{2.5})$.

Max-Delta is a simple heuristic for selecting matrix entries (i, j) to consist of a match σ given 1LM similarity matrix M. Max-Delta is employed by many matching systems, *e.g.*, COMA 3.0 [23] and AMC [26]. Given a Delta value (*e.g.*, 0.1), Max-Delta calculates a value δ_i for each row $i = 1, \ldots, n$ of the similarity matrix. The match result σ is constructed by selecting those entries (i, j) from each matrix row with confidence values $M_{i,j}$ that are at most δ_i below the maximum value in that row (max_i). Formally, an entry (i, j) is selected for the match σ iff the following condition holds: $M_{i,j} \geq max_i - \delta_i$, where $max_i = max_{j=1\ldots,m} M_{i,j}$ and δ_i is a proportion of row "variance," given by:

$$\delta_i = \mathsf{Delta} \times \left(max_i - \frac{1}{m} \sum_{j=1}^{m} M_{i,j} \right) \tag{2}$$

The Max-Delta heuristic assumes 1LM results are well correlated with their quality. Thus, high valued entries are assumed to be better match candidates than low valued entries. Applying the selection rule row by row ensures good coverage (recall) of the target schema. On the other hand, limiting the results to the top-valued entries per row, aims at better precision.

3. MATCH COMPETITOR DEVIATION

Schema matching predictors assess the quality of the matching outcome without any knowledge of the exact match [33]. Such prediction can be based on either internal properties of the similarity matrix or by a distance measure from some "ideal" form of solution. Predictors should be applied to tasks with different requirements of granularity, from predicting match quality for a single attribute pair, to match quality of a schema pair. Predictors should be able to predict different qualities, putting more emphasis, for example, on Precision or on Recall. Quality of predictors is measured by its correlation with match Precision or

Recall, and a good correlation should be statistically significant when tested over a substantial number of schema pairs and stable over varying datasets and schema matchers.

Match Competitor Deviation (MCD) is a new predictor, which measures the diversity of a match $\sigma \in \Sigma$ that was determined by some 2LM, given a similarity matrix M. Informally, match diversity is captured by measuring how much each matrix entry $(i, j) \in \sigma$, selected by a 2LM, deviates (in terms of match confidence) from other competing entries $(i, l); l \neq j$ or $(l, j); l \neq i$ in the similarity matrix M.

More formally, deviation is captured by measuring the difference between entry $(i, j) \in \sigma$ confidence $M_{i,j}$ and that of a *mean* entry, $\mu_{i,j}$, defined as the average confidence among entries that share the same matrix row i or column j (including entry (i, j)), as follows:

$$\Delta_{i,j} = (M_{i,j} - \mu_{i,j})^2 , \tag{3}$$

where:

$$\mu_{i,j} = \frac{1}{n+m-1} \left(\sum_{l=1}^{n} M_{l,j} + \sum_{l=1}^{m} M_{i,l} - M_{i,j} \right) \tag{4}$$

For a given similarity matrix M (generated by some 1LM) and a match $\sigma \in \Sigma$ (generated by some 2LM), the MCD predictor evaluates the quality of the match according to the average (scaled) deviation, as follows:

$$\mathcal{Q}_{\mathsf{MCD}}(\sigma, M) = \sqrt{\frac{1}{|\sigma|} \sum_{(i,j) \in \sigma} \Delta_{i,j}} \tag{5}$$

Therefore, the main principle of the MCD predictor is to evaluate the ability of a 2LM to pick entries for the match that deviate as much as possible from their competitor entries. Such deviation may be attributed to the ability of a 2LM to choose diverse entries, rather than just consider each entry's confidence independently.

As we shall demonstrate in Section 5.3, the MCD predicted value is highly correlated with the actual match quality, as would be judged by a human assessor.

3.1 The importance of match diversity

A notable drawback of the MWBM matcher is that matrix entries (i, j) are selected **independently** of the similarity of other entries that compete on the same match selection spot. To illustrate how detrimental such a drawback may be, consider the following example similarity matrix that may be produced by some 1LM:

$$M = \begin{pmatrix} 0.9 & 0.1 & 0.9 \\ 0.1 & 0.1 & 0.1 \\ 0.9 & 0.1 & 0.9 \end{pmatrix}$$

Seeking a $1:1$ match, σ^* is the match with the optimal $\mathcal{Q}_{\mathsf{MWBM}}(\sigma^*, M) = 1.9$ value, as would be returned by the MWBM matcher. When maximizing $\mathcal{Q}_{\mathsf{MWBM}}(\sigma, M)$, entry $(2, 2)$ is included in σ^*, which may turn out to be risky, having a reasonable chance of being incorrect. The reason for that is two-fold. First, the selected entry itself is of low confidence. Second, the alternatives have the same confidence level, which may hint that the 1LM could not distinguish well between entry $(2, 2)$ and its competitors.

Such a problem in entry selection can be detected by measuring the diversity of the match according to the MCD predictor. In this example, $\Delta_{2,2} = 0$ and when maximizing $\mathcal{Q}_{\mathsf{MCD}}(\sigma, M)$ according to Eq. 5 the inclusion of entry $(2, 2)$ in σ^* is avoided, which may result in a match with a better precision. While there is some possible chance for loss in match recall, such possibility is minimized, since the entry that was eliminated had a very low confidence.

Algorithm 1 MCD

```
 1: input: $M(n, m)$
 2: for $(i, j) \in M$ do
 3:     $\Delta_{i,j} := (M_{i,j} - \mu_{i,j})^2$
 4: end for
 5: $k := \min(n, m)$
 6: $\sigma^* := \emptyset$
 7: for $p = 1, \ldots, k$ do
 8:     $\sigma := \mathsf{MWBM}(\Delta, p)$
 9:     if $\mathcal{Q}_{\mathsf{MCD}}(\sigma, M) > \mathcal{Q}_{\mathsf{MCD}}(\sigma^*, M)$ then
10:         $\sigma^* := \sigma$
11:     end if
12: end for
13: return $\sigma^*$
```

3.2 MCD optimization

Algorithm 1 describes an efficient solution for finding a match $\sigma \in \Sigma$ (either $1:1$ or $1:n$) with an optimal MCD value given any 1LM's similarity matrix M as an input. The algorithm makes use of an auxiliary algorithm $\mathsf{MWBM}(M, p)$ that returns the maximum weight match given M, where the required match size $|\sigma| = p$ is fixed [28].

The algorithm starts by creating the Δ matrix using Eq. 3 (lines 2-4). Then, it iteratively finds an optimal solution for $\mathsf{MWBM}(\Delta, p)$ [28] for $p = 1, ..., min(n, m)$, keeping the match with maximum $\mathcal{Q}_{\mathsf{MCD}}(\cdot, M)$ (lines 7-12).

THEOREM 1. *For any similarity matrix M, Algorithm 1 correctly finds a match $\sigma \in \Sigma$ that maximizes $\mathcal{Q}_{\mathsf{MCD}}(\sigma, M)$.*

PROOF. Let σ be the match returned at Step 8. $\mathsf{MWBM}(\Delta, p)$ returns $\sigma = argmax_{\sigma \in \Sigma||\sigma|=l} \mathcal{Q}_{\mathsf{MWBM}}(\sigma, \Delta)$, by definition. Let $\sigma' \in \Sigma$ such that $|\sigma'| = p$. We have that:

$$\begin{aligned} \mathcal{Q}_{\mathsf{MCD}}(\sigma, M) &= \sqrt{\frac{1}{p} \sum_{(i,j) \in \sigma} \Delta_{i,j}} \text{ (Eq. 5)} \\ &= \sqrt{\frac{1}{p} \mathcal{Q}_{\mathsf{MWBM}}(\sigma, \Delta)} \text{ (Line 8 and Eq. 1)} \\ &\geq \sqrt{\frac{1}{p} \mathcal{Q}_{\mathsf{MWBM}}(\sigma', \Delta)} \text{ By MWBM optimality} \\ &= \sqrt{\frac{1}{p} \sum_{(i,j) \in \sigma'} \Delta_{i,j}} \text{ (Eq. 1)} \\ &= \mathcal{Q}_{\mathsf{MCD}}(\sigma', M) \text{ (Eq. 5)} \end{aligned}$$

Since the algorithm maintains the optimal match for every possible fixed match size p (lines 9-10), we are guaranteed that the returned matching σ^* (Line 13) is optimal. □

Predictors provide a unique value for each similarity matrix entry and as such, can serve as schema matchers by themselves. Unfortunately, as will be illustrated in the next section, their role as schema matchers may not be better than any other schema matcher in the literature. However, the ability to assess the quality of a match is useful in deciding which correspondences to include in and which to exclude from a match. Therefore, we introduce in this work a novel way of using predictors. In addition to using them for prediction, we use them as regulators, tuning the abilities of schema matchers towards better decision making.

4. SCHEMA MATCHING USING MCD AS A REGULATOR

In this section, we show how schema matching can be done with MCD as a regulator, tuning the task towards Precision or Recall. We focus on two schema matchers as representatives of $1:1$ and $1:n$ matching tasks. The bulk of the discussion is devoted to the regulation of MWBM, which is shown to be a hard problem. We start by showing the optimality tradeoff of MCD and MWBM (Section 4.1). Sections 4.2 and 4.3 are devoted to presenting and optimizing MCD as a regulator for MWBM. Finally, Section 4.4 presents the enhancement of Max-Delta with MCD.

4.1 MCD vs. MWBM Optimality Tradeoff

While an algorithm for finding an optimal match exists for each of the two match objectives, namely MWBM (Eq. 1) and MCD (Eq. 5), the optimization of each objective *separately* may violate the optimality of the other.

Proposition 1. *Let M be a similarity matrix and $\sigma \in \Sigma$: $\mathcal{Q}_{\mathsf{MWBM}}(\sigma, M) \geq \mathcal{Q}_{\mathsf{MCD}}(\sigma, M)$*

Proof.

$$\begin{aligned}\sum_{(i,j)\in\sigma} M_{i,j} &\geq \sqrt{\sum_{(i,j)\in\sigma} M_{i,j}^2}\\ &\geq \sqrt{\sum_{(i,j)\in\sigma} \left(M_{i,j}-\mu_{i,j}\right)^2}\\ &\geq \sqrt{\frac{1}{|\sigma|}\sum_{(i,j)\in\sigma} \left(M_{i,j}-\mu_{i,j}\right)^2}\end{aligned}$$

□

Using Proposition 1 we conclude that, given a similarity matrix M, the maximization of MCD objective may basically violate the maximization of MWBM objective. Given $\sigma' \in \Sigma$ and $\sigma'' \in \Sigma$ the optimal match for the MWBM and the MCD objectives, respectively. Then, the following holds:

$$\begin{aligned}\mathcal{Q}_{\mathsf{MWBM}}(\sigma', M) &\geq \mathcal{Q}_{\mathsf{MWBM}}(\sigma'', M) \text{ (MWBM optimality)}\\ &\geq \mathcal{Q}_{\mathsf{MCD}}(\sigma'', M) \quad \text{(Proposition 1)}\\ &\geq \mathcal{Q}_{\mathsf{MCD}}(\sigma', M) \quad \text{(MCD optimality)}\end{aligned}$$

Given a matrix M, let $\sigma, \sigma' \in \Sigma$ be matches that maximize $\mathcal{Q}_{\mathsf{MCD}}(\cdot, M)$ and $\mathcal{Q}_{\mathsf{MWBM}}(\cdot, M)$, respectively. We define an MWBM *ratio of optimality* as the ratio between the scores MWBM assigns to the optimal MCD and MWBM matches ($\frac{\mathcal{Q}_{\mathsf{MWBM}}(\sigma,M)}{\mathcal{Q}_{\mathsf{MWBM}}(\sigma',M)}$). Lower ratios indicate worse performance of MCD in terms of MWBM. Proposition 2 provides an upper-bound on the MWBM ratio of optimality, demonstrating that the maximization of MCD objective may yield bad (problem size factor) performance with respect to the MWBM objective.

Proposition 2.

$$\frac{\mathcal{Q}_{\mathsf{MWBM}}(\sigma, M)}{\mathcal{Q}_{\mathsf{MWBM}}(\sigma', M)} \leq \frac{1}{min(n,m)}$$

Proof. Consider the following (symmetric) similarity matrix instance:

$$M' = \begin{pmatrix} 1 & \epsilon & \cdots & \epsilon \\ \epsilon & \epsilon & \cdots & \epsilon \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon & \epsilon & \cdots & \epsilon \end{pmatrix}$$

The optimal score of MCD match σ over M' is $\mathcal{Q}_{\mathsf{MWBM}}(\sigma, M) = 1$, while the optimal score of MWBM match σ' over M' is $\mathcal{Q}_{\mathsf{MWBM}}(\sigma', M) = 1 + (min(n,m) - 1)\epsilon$. Since $\epsilon \in [0,1]$, the worst MWBM ratio of optimality is gained when $\epsilon = 1$, having

$$\mathcal{Q}_{\mathsf{MWBM}}(\sigma, M') = \frac{1}{min(n,m)}\mathcal{Q}_{\mathsf{MWBM}}(\sigma', M')$$

and hence

$$\frac{\mathcal{Q}_{\mathsf{MWBM}}(\sigma, M)}{\mathcal{Q}_{\mathsf{MWBM}}(\sigma', M)} \leq \frac{1}{min(n,m)}$$

□

4.2 MCD-based match regularization

The selection of a match $\sigma \in \Sigma$ that exhibits both high match confidence and selection diversity is formally captured by the following bi-objective problem:

$$\max_{\sigma\in\Sigma}\{\mathcal{Q}_{\mathsf{MWBM}}(\sigma, M), \mathcal{Q}_{\mathsf{MCD}}(\sigma, M)\} \qquad (6)$$

As was demonstrated above, the optimality of each one of the objectives may violate the optimality of the other. Therefore, any optimal solution to this problem may be defined in terms of *Pareto* optimality [8], which formally captures the tradeoff among the two objectives.

Definition 2 (Pareto Optimal Match). *Given a similarity matrix M, match $\sigma \in \Sigma$ is a Pareto optimal solution to the bi-objective optimization problem (Eq. 6) if for any other match $\sigma' \in \Sigma$ one of the following holds:*

$$\mathcal{Q}_{\mathsf{MWBM}}(\sigma, M) \leq \mathcal{Q}_{\mathsf{MWBM}}(\sigma', M) \Rightarrow \mathcal{Q}_{\mathsf{MCD}}(\sigma, M) > \mathcal{Q}_{\mathsf{MCD}}(\sigma', M),$$

or

$$\mathcal{Q}_{\mathsf{MCD}}(\sigma, M) \leq \mathcal{Q}_{\mathsf{MCD}}(\sigma', M) \Rightarrow \mathcal{Q}_{\mathsf{MWBM}}(\sigma, M) > \mathcal{Q}_{\mathsf{MWBM}}(\sigma', M).$$

Instead of solving the bi-objective problem we combine both objectives using their weighted power mean:

$$\mathcal{Q}(\sigma, M) = \mathcal{Q}_{\mathsf{MCD}}(\sigma, M)^{\beta}\mathcal{Q}_{\mathsf{MWBM}}(\sigma, M)^{1-\beta}, \qquad (7)$$

where $\beta \in [0,1]$ is a regularization parameter that controls the relative importance of each of the two objectives. Higher β indicates a higher preference towards diverse entry selection over match confidence, and visa versa.

Proposition 3. *Given a similarity matrix M and some $\beta \in [0,1]$, let $\sigma \in \Sigma$ be a match that maximizes $\mathcal{Q}(\sigma, M)$ in Eq. 7, then σ provides a Pareto optimal solution to the bi-objective problem defined in Eq. 6.*

Proof. Let M be a given similarity matrix and $\sigma \in \Sigma$ be a match that maximizes $\mathcal{Q}(\sigma, M)$. Let $\sigma' \in \Sigma$ be some other match. Without loss of generality, assume that $\mathcal{Q}_{\mathsf{MWBM}}(\sigma, M) < \mathcal{Q}_{\mathsf{MWBM}}(\sigma', M)$ for some $\beta \in [0,1]$, from the monotonicity of the power function it follows that:

$$\mathcal{Q}_{\mathsf{MWBM}}(\sigma, M)^{1-\beta} < \mathcal{Q}_{\mathsf{MWBM}}(\sigma', M)^{1-\beta} \quad (8)$$

From the fact that $\mathcal{Q}(\sigma, M) \geq \mathcal{Q}(\sigma', M)$, combined with Eq. 8, we get that:

$$\mathcal{Q}_{\mathsf{MCD}}(\sigma, M)^{\beta} > \mathcal{Q}_{\mathsf{MCD}}(\sigma', M)^{\beta}.$$

Again from the monotonicity of the power function we get that: $\mathcal{Q}_{\mathsf{MCD}}(\sigma, M) > \mathcal{Q}_{\mathsf{MCD}}(\sigma', M)$. Therefore, σ is a Pareto optimal match. □

By solving Eq. 7 rather than Eq. 6, we still maintain Pareto optimality (Proposition 3), using β to decide on where on the Pareto curve we prefer to be. In Section 5 we show that $\mathcal{Q}(\sigma, M)$ is highly correlated with matching quality (in terms of Precision, Recall, and F1). Therefore, maximizing $\mathcal{Q}(\sigma, M)$ increases matching quality as well.

4.3 Cross-Entropy Based Optimization

The maximization of the combined bi-objective $\mathcal{Q}(\sigma, M)$, as well as its original bi-objective version in Eq. 6 is NP-Hard [1, 8]. Therefore, we next propose an efficient solution that can produce an approximate Pareto-optimal match, which captures the tradeoff encoded in the bi-objective optimization problem and effectively explores the match space Σ. We now describe the details of a novel 2LM, termed Cross-Entropy Schema Matcher (CESM for short). CESM is an **unsupervised** schema matcher that utilizes $\mathcal{Q}(\sigma, M)$ (Eq. 7) as a proxy for match quality prediction. Therefore, the CESM matcher's goal is to find a match $\sigma \in \Sigma$ that (approximately) maximizes $\mathcal{Q}(\sigma, M)$. We utilize a randomized optimization approach, namely the Cross-Entropy (CE) Method, a Monte Carlo (randomized) combinatorial optimization technique for solving hard problems. We start by providing some motivation behind the usage of the CE Method, where we focus on its novel application to schema matching (Section 4.3.1). We then introduce the CESM matcher (Section 4.3.2).

4.3.1 From match quality optimization to rare event estimation

The basic idea behind the CE Method, which we make use of in this work, is that finding an optimal solution to a (deterministic) hard problem may be casted into an equivalent **rare-event** (stochastic) estimation problem as follows.

Given similarity matrix M, assume that γ^* is the best match quality (according to $\mathcal{Q}(\sigma, M)$) that may be obtained by some optimal match (solution) $\sigma^* \in \Sigma$, that is:

$$\gamma^* = \mathcal{Q}(\sigma^*, M) = \max_{\sigma \in \Sigma} \mathcal{Q}(\sigma, M) \quad (9)$$

As a starting point, we associate with the optimization problem in Eq. 9 a meaningful estimation problem [31]. To this end, let $\mathbf{\Sigma} \sim f(v)$ denote a random match over Σ that is distributed according to some pdf $f(v)$ with parameter v. For a given parameter v we now associate with Eq. 9 the problem of estimating

$$l(\gamma) = \mathbb{P}_v(\mathcal{Q}(\mathbf{\Sigma}, M) \geq \gamma) = \mathbb{E}_v(\delta_{[\mathcal{Q}(\mathbf{\Sigma}, M) \geq \gamma]}), \quad (10)$$

where $\mathbb{P}_v$ is the probability measure under $f(v)$, and $\mathbb{E}_v$ denotes the corresponding expectation operator. $\delta_{[\theta]}$ denotes the Kronecker-delta (indicator) function, receiving the value of 1 if the condition expressed by θ is satisfied, else 0. The estimation problem in Eq. 10 is termed the *associated stochastic problem* (ASP) of Eq. 9 [31].

Unfortunately, a direct calculation of $l(\gamma)$ in Eq. 10 would require a full enumeration of Σ, which is commonly unpractical due to its size. One possible (and naïve) way to estimate the event likelihood captured by $l(\gamma)$ is to use a simple Crude Monte Carlo (CMC) estimator [32] as follows:

$$\hat{l}_{CMC}(\gamma) = \frac{1}{N} \sum_{i=1}^{N} \delta_{[\mathcal{Q}(\sigma_i, M) \geq \gamma]}, \quad (11)$$

where $\sigma_i \in \Sigma$ are i.i.d random matches drawn from $f(v)$.

We wish to find an estimator such that $l(\gamma)$=$l(\gamma^*)$. However, the original optimization problem in Eq. 9 is NP-Hard and we actually need to estimate the likelihood of the occurrence of a rare-event, *i.e.*, the probability that we have obtained at least one of the matches $\sigma^* \in \Sigma$ that have an optimal match quality γ^*. Therefore, in most cases, random match samples σ_i yields $\delta_{[\mathcal{Q}(\sigma_i, M) \geq \gamma^*]} = 0$, requiring a very large sample size N to obtain a reliable estimate [32].

The CE Method, to be presented herein, provides a more computationally efficient way to estimate $l(\gamma^*)$. We briefly explain the main idea behind the CE Method, setting basic intuition about the approach we take. Full details of the CE Method solution are provided in [31].

The CE Method is based on an *importance sampling* approach [32]. Using this approach, the optimal reference parameter $v^* \in \mathcal{V}$ may be learned and the event given by $\{\mathcal{Q}(\mathbf{\Sigma}, M) \geq \gamma^*\}$ may be efficiently estimated. Using v^*, a single match $\sigma^* \in \Sigma$ may be then sampled from the corresponding $f(v^*)$ to provide an (approximate) optimal solution with maximum match quality $\mathcal{Q}(\sigma^*, M)$.

The CE Method uses an iterative two-step approach. First, observe that for a given quality performance level $\gamma \ll \gamma^*$ (*e.g.*, $\gamma = 0$) we can find a reference parameter $v_\gamma \in \mathcal{V}$ under which the event $\{\mathcal{Q}(\mathbf{\Sigma}, M) \geq \gamma\}$ is no longer rare. That is,

$$l(\gamma) = \mathbb{P}_{v_\gamma}(\mathcal{Q}(\mathbf{\Sigma}, M) \geq \gamma) \geq \rho \quad (12)$$

for some large enough ρ (*e.g.*, $\rho = 0.01$).

Starting from some initial reference parameter v^0 (*e.g.*, one with maximum entropy), in each iteration t the CE Method learns a new pair $\langle v^t, \gamma_t \rangle$ using the previously learned reference parameter v^{t-1} for which the event $\{\mathcal{Q}(\mathbf{\Sigma}, M) \geq \gamma_t\}$ is not rare anymore and its probability is at least ρ. To this end, in each iteration the CE Method first samples random matches $\sigma_i \in \Sigma$ according to $f(v^{t-1})$ and finds a new performance level γ_t in which at least ρ of the samples have performance higher or equal to γ_t. Such γ_t can be easily estimated by first sorting the performances $\mathcal{Q}(\sigma_i, M)$ in ascending order and taking γ_t to be the $(1-\rho)$-quantile of the list. The learning of the new reference parameter v^t, therefore, is based on the ρ-best performing samples (termed the "*elite sample*" [31]), each has at least γ_t match quality.

The CE Method shall, therefore, iteratively attempt to improve the learned reference parameter v^t such that γ_t gradually increases towards the unknown optimal performance γ^*. It halts once γ_t can no longer improve.

Finally, the derivation of the next reference parameter v^t is based on a solution (using importance sampling) to a *Cross Entropy minimization* problem, where the "distance" between $f(v^t)$ of the unknown ("better") v^t parameter and the one with the previously learned parameter v^{t-1}, $f(v^{t-1})$ is being minimized.

As a common practice, instead of updating the parameter v^{t-1} to v^t directly, similarly to many other learning methods, we use a *smoothed* updating procedure in which:

$$v^t = \lambda v^t + (1-\lambda)v^{t-1}, \tag{13}$$

where $\lambda \in [0,1]$ is the *smoother* [31].

The details of the formal derivation of the CE optimal reference parameter are described in details in the Appendix B. The CE Method has been shown to converge to the optimal solution with probability 1 within finite number of iterations [21]. Practically, as shall be demonstrated in our evaluation in Section 5, the CE Method only explores a relatively tiny fraction of the match space Σ, hence, proving to be an effective optimization tool for our need.

4.3.2 Cross Entropy Schema Matcher

Having introduced the intuition behind the CE Method, we now describe its application to schema matching. Recall that our aim is to find a match $\sigma \in \Sigma$ that maximizes the overall predicted match quality $\mathcal{Q}(\sigma, M)$. Using the CE Method, the optimization problem has been reduced to the problem of finding an optimal reference parameter, under which the likelihood of finding some match $\sigma^* \in \Sigma$ with an optimal performance $\mathcal{Q}(\sigma^*, M)$ may be efficiently estimated.

Algorithm 2 Cross Entropy Schema Matcher

```
1: input: similarity matrix $M, N, \rho, \lambda$
2: initialize:
3: for $i = 1, \ldots, m; j = 1, \ldots, n$ do
4:     $v^0_{i,j} = \frac{1}{2}$
5: end for
6: $t = 1$
7: loop
8:     Randomly draw $N$ matches $\sigma \in \Sigma$ using $v^{t-1}$
9:     $\overrightarrow{\Sigma_l} = \mathtt{sort}_{l=1,\ldots,N}(\mathcal{Q}(\sigma_l, M))$
10:    $\gamma_t = \mathtt{quantile}_{1-\rho}(\overrightarrow{\Sigma_l})$
11:    for $i = 1, \ldots, n; j = 1, \ldots, m$ do
12:        $v^t_{i,j} := \frac{\sum_{l=1}^{N} \delta[\mathcal{Q}(\sigma_l, M) \geq \gamma_t] \delta[(i,j) \in \sigma_l]}{\sum_{l=1}^{N} \delta[\mathcal{Q}(\sigma_l, M) \geq \gamma_t]}$
13:        $v^t_{i,j} := \lambda v^{t-1}_{i,j} + (1-\lambda) v^t_{i,j}$
14:    end for
15:    if $\gamma_t$ converged then
16:        stop and return random match $\sigma^*$ sampled from $f(v^t)$
17:    else
18:        $t := t + 1$
19:    end if
20: end loop
```

Algorithm 2 describes the implementation of the CE Method for schema matching (denoted CESM hereinafter). The algorithm relies on a utility algorithm (Algorithm 3)or sampling random matches in Σ, whose details are given in Appendix A.

Algorithm 2 gets as an input the similarity matrix M, and three configuration parameters, namely, the match sample size N to be drawn on each iteration t, ρ the minimum event $\{\mathcal{Q}(\mathbf{\Sigma}, M) \geq \gamma^t\}$ occurrence likelihood, and λ the smoother. The three configuration parameters control the learning rate of the algorithm. For example, as will be shown in Section 5, smaller λ values typically result in a slower convergence.

The algorithm starts with a maximum entropy setting, where each matrix entry (i,j) has the same likelihood in the initial parameter vector v^0 to be selected (or rejected) as part of a match $\sigma \in \Sigma$ (lines 3-4).

In each iteration $t = 1, 2, \ldots$ (lines 7-20) the algorithm draws N random matches, σ_l, according to the previously learned reference parameter vector v^{t-1} (line 8). Next, sampled matches σ_l are sorted in ascending order in $\overrightarrow{\Sigma_l}$ according to their relative performance level $\mathcal{Q}(\sigma_l, M)$ (line 9). Next, γ_t, the minimum performance level in which the likelihood of the event $\{\mathcal{Q}(\mathbf{\Sigma}, M) \geq \gamma_t\}$ is at least ρ, is estimated by taking γ_t to be the $(1-\rho)$-quantile of the (ordered) performances in $\overrightarrow{\Sigma_l}$ (line 10). Lines 11-13 update the likelihood of choosing each matrix entry (i,j) for a match, $v^t_{i,j}$, based on the relative number of matches in the current iteration t sample with $\mathcal{Q}(\sigma_l, M) \geq \gamma_t$ that consists of entry (i,j). The details of the exact derivation of v^t are provided in Appendix B. This value is smoothed with the parameter vector that was learned in the previous iteration v^{t-1}.

The algorithm runs until some convergence criterion is satisfied. In this work, the algorithm halts if γ_t has not changed for several consecutive iterations [31]. Finally, the algorithm returns a single match $\sigma \in \Sigma$, sampled from the distribution having the final reference parameter v^t.

4.4 Max-Delta with MCD Regulation

We conclude this section by demonstrating how the correlation of MCD with high quality entries (Section 5.3), can be utilized to improve $1:n$ schema matching. To examine this premise, we modify the Max-Delta 2LM utilizing MCD prediction for $1:n$ match regulation.

The underlying assumption of Max-Delta is that higher valued matrix entries are more likely to be correct than lower ones. Thus, a similarity matrix entry's confidence value is used to predict its actual quality. As Sagi and Gal pointed out [33], using better correlated entry predictors may lead to improved results. Therefore, we substitute $M_{i,j}$ in Eq. 2 with $\Delta_{i,j}$ (Eq. 3), resulting in the following equation:

$$\delta_i = \mathsf{Delta} \times \left(max_i - \frac{1}{m} \sum_{j=1}^{m} \Delta_{i,j} \right) \tag{14}$$

where $max_i = max_{j=1\ldots,m} \Delta_{i,j}$. Those entries with higher predicted values (up to δ_i from the maximum predicted value) are selected. Therefore, an entry (i,j) is now selected for the match σ iff the following condition holds: $\Delta_{i,j} \geq max_i - \delta_i$. Max-Delta is suitable for $1:n$ matching as it allows more than one match per matrix row.

Delta, in some sense, is the equivalent of β, where the last regulates the decisions made by MWBM. A higher Delta value potentially allows more entries to enter the match and thus caters to Recall. A lower value provides a more strict entry selection rule, and thus, caters more to Precision.

5. EMPIRICAL EVALUATION

The empirical evaluation of MCD, CESM, and Max-Delta is now described. We first outline our evaluation setup (Section 5.1) and methodology (Section 5.2). In Section 5.3, MCD is being evaluated as a predictor, examining its ability to predict the quality of both a single similarity matrix entry and a full matrix. For a single entry, we evaluate whether the predictor can differentiate between true and false matches by consistently assigning a higher score to the former. For a full matrix, predictor values are expected to correlate well with standard quality measures (Precision and Recall). Next, the potential of using MCD for enhancing the decisions of MWBM and Max-Delta is being demonstrated. To this end, in Section 5.4, the quality of the proposed CE-Method based Schema Matcher CESM is evaluated. We also

Matcher	System	Type
Term	Ontobuilder [25]	Syntactic
Token Path	AMC [26]	Syntactic
WordNet [30, 16, 29]	ORE	Semantic

Table 2: 1LMs used in the evaluation

analyze the impact of different tuning of CESM parameters on its performance. Finally, in Section 5.5, we evaluate MCD enhancement of the decisions of the Max-Delta 2LM that produces a $1 : n$ match.

5.1 Setup

Evaluations were performed using a Dell Inc. PowerEdge R720 server. with a 20 true cores (40 virtual cores) Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz CPU, 128GB RAM (8x16GB), and a CentOS 6.4 operating system with x86_64 Kernel:2.6.32-358.6.2.el6.x86_64. In terms of software, we use Java(TM) SE Runtime Environment (build 1.8.0_45-b14) and MySQL 5.5.32.

Three schema matching tools were used for our experiments, two research tools (ORE and COMA) and one industrial (AMC), as detailed next. The Ontobuilder Research Environment[1] (ORE) allows researchers to run matching experiments using various matchers on a collection of datasets and evaluate the outcome using various quality measures. Table 2 details the ORE's 1LMs (whose details are given in Section 2) used for the evaluation. CESM is evaluated against *maximum weighted bipartite graph match (*MWBM*)* [15] and *stable marriage (*SM*)* of Marie and Gal [22], using a known algorithm for solving a problem of finding a matching between two sets of elements given an ordering of preferences for each element. We also evaluate against Dominants [12] (first used by [37] and also dubbed later as *harmony* [20]), which selects correspondences that dominate all other entries in their row and column. Two additional match selection rules were used, prevalent in many matching systems (see, *e.g.* [6]): Threshold(ν), which selects entries (i, j) such that $M_{i,j} \geq \nu$ and Max-Delta, which was described in detail in Section 4.4.

Auto-Mapping Core (AMC) [26] is a tool, developed by SAP Research, which provides an infrastructure and a set of algorithms to establish correspondences between two business schemata. We use one of the algorithms of AMC (Token Path), embedded in ORE, in our experiments.

We also compare against a state-of-the-art schema matching research tool, COMA 3.0, which has three 2LM decision makers, namely threshold(t), maxDelta(d), and maxN(n).[2] These can be combined by setting more than one of the parameters t,d,n to a non-negative value.

A parallel version of CESM was implemented in Java (JRE 8) by parallelizing its match sampling step [11] (Line 8 of Algorithm 2), sorting (Line 9), and the loop of updating $v^t_{i,j}$ (lines 11-14). Following previous recommendations for the CE Method [31, 21], CESM free parameters defaults were fixed as follows: $N = 10,000$, $\rho = 0.01$ and $\lambda = 0.3$. The bi-objective regularization parameter was varied with $\beta \in \{0.1, 0.2, \ldots, 0.9\}$.

Dataset	#Schemas	#Attr	#Pairs	Match type
Web-forms	147	10-30	247	1 : 1
Thalia	44	6-17	18	1 : 1
Purchase Order	10	50-400	44	$1 : n$
University Applications	16	50-150	182	$1 : n$

Table 3: Datasets

Table 3 details the datasets we used in the experiments. For each dataset, we detail the number of schemas it contains, its size (in terms of attributes), and the number of schema pairs. The exact match of two of the datasets is $1 : 1$ while for the other two it is $1 : n$.

The Web-forms [13] dataset contains schemas that were automatically extracted from Web forms using the OntoBuilder extractor. Exact matches for each schema pair was manually crafted by several judges. The Thalia dataset[3] is a publicly available dataset of relational database tables representing University course catalogs from computer science departments around the world. The Purchase Order dataset [19] contains XML documents describing purchase orders extracted from various systems and matched into pairs. Finally, the University Applications dataset [33] contains university application forms from various US universities, collected as part of the NisB project[4] and converted into XML Schema Definition (XSD) format.

5.2 Evaluation Measures

Following the method described by Sagi and Gal [33], correlation of matrix level predictors is measured using the Pearson product-moment correlation coefficient (*Pearsons's r*). Entry level prediction evaluation is performed by calculating the Goodman and Kruskal's gamma (*GK-Gamma* for short) correlation. GK-Gamma is a rank correlation measure, used to measure the correlation between matrix values predicted by entry predictors' and the actual (binary) values of an exact match produced by human assessors. For a binary quality measure (match / no match), GK-Gamma counts the number of concordant (N_c) and discordant (N_d) pairs. In concordant pairs, the prediction values are aligned with the actual result and, thus, the true entry was predicted higher than the false entry. For discordant pairs, the situation is reversed and the predictor falsely predicted a higher score for the false entry (ties are ignored). The measure value is given by the following equation:

$$G = \frac{N_c - N_d}{N_c + N_d} \tag{15}$$

A good entry predictor can separate well true and false matches by **consistently** assigning lower values to false matches than to true matches and, thus, has a value of GK-Gamma closer to 1.0.

2LM performance is evaluated using binary *Precision* (P), *Recall* (R), and their harmonic *F1-Score*. MCD-based 2LM are compared with other 2LMs using the *Robustness Index* (RI) of the former. Robustness Index is calculated by assigning a score of 1 for each schema pair $\langle S, S' \rangle$ where the MCD-based 2LM improved over the (existing) baseline 2LM

[1] https://bitbucket.org/tomers77/ontobuilder-research-environment/wiki/Home

[2] http://sourceforge.net/p/coma-ce/mysvn/HEAD/tree/coma-project/coma-engine/src/main/java/de/wdilab/coma/matching/Selection.java

[3] http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/THALIATestbed

[4] http://www.nisb-project.eu/

and -1 to each pair where the result was worsened, and averaging over all pairs. Thus, RI spans $[-1.0, 1.0]$ where 1.0 and -1.0 indicate an improvement and a decline in performance over all pairs, respectively.

Finally, we also measured performance in terms of runtime (comparing time until convergence of CESM and COMA) and number of iterations.

5.3 MCD Prediction

In this experiment we evaluate MCD's role as a predictor for various match quality measures. We begin by evaluating MCD (Eq. 5) as a matrix level predictor. We then evaluate MCD as an entry level predictor using $\Delta_{i,j}$ (Eq. 3).

5.3.1 *MCD-based matrix quality prediction*

We evaluate MCD as a matrix predictor together with seven other predictors, six of which were previously suggested[33] and the seventh adopted from [5]. We describe these predictors briefly. BMM and LMM are obtained by first "flattening" the similarity matrix M into a vector with $n \cdot m$ entries, each vector entry uniquely corresponds to one entry value of matrix M. Then, BMM and LMM measure the cosine similarity between that vector and an "ideal" (similarity) vector that is constructed from it. LMM constructs an ideal vector that has a single 1-valued entry per matrix row and BMM constructs the "closest" binary vector [33]. Max, STDEV, and Avg all calculate the measure they are named after for each matrix row and average the values over n, the number of similarity matrix rows. For example, $\mathsf{Max}(M) = \frac{1}{n}\sum_{i=1}^{n} \max_i$, where $\max_i = \max_{j=1,\dots,m} M_{i,j}$. Dominants counts the number of matrix entries which are the largest in their respective row and column, dividing the result by the number of matrix rows. Finally, LC is an attribute-level measure, designed to use a given matrix row and a selection over it to compute the difference between the average similarity of selected and unselected attributes. We convert LC to a matrix level predictor by averaging over row scores.

Prediction was performed over 960 matrices generated by running all 1LMs of Table 2 on 90 different schema pairs randomly sampled from three datasets: Web-forms, Purchase Order and University. Using the similarity matrix produced by each 1LM, the following 2LMs were run: Max-Delta with Delta $= 0.1$, Threshold($\nu = 0.5$), MWBM, and SM.

Predictor	P Correlation	R Correlation
BMM	.379**	.206**
LMM	.246**	.338**
Max	.180**	.506**
STDEV	.124**	.630**
Avg	.565**	.077**
Dominants	.429**	.039
LC	.425**	.048
MCD	.568**	-.002

Table 4: Pearson's r correlation to Precision (P) and Recall (R) of the various matrix predictors

Table 4 presents the Pearson's r correlation between the predictors and the two quality measures, Precision (P) and Recall (R). A two-tailed t-test of significance at 95% confidence level was performed against the null hypothesis of no-correlation. Table entries marked with double star (**) denote significant results ($p\text{-}value < 0.05$). Results indicate that MCD predicted values (Eq. 5) are well correlated with Precision, yet not much with Recall. The results for the other predictors are in line with those previously presented in [33]. LC as a matrix predictor demonstrates a similar behavior to Dominants with strong (yet not as strong as MCD) correlation with Precision and low correlation with Recall. Note that the Max predictor and Recall are strongly correlated. Compared together with the MCD result, side-by-side, this result empirically confirms our assumption that MCD may regulate MWBM. By increasing the β regularization free parameter, MCD is expected to improve on Precision, yet with some sacrifice on Recall. MCD regulation is expected to direct the CE optimization process made by CESM towards a Precision oriented solution. The effect of the β regularization parameter is demonstrated in Section 5.4.

5.3.2 *MCD-based entry quality prediction*

As an entry predictor, MCD is evaluated against three entry predictors: VAL, NNV and CRV, previously proposed by Sagi and Gal [33]. Entry predictors attempt to predict the value of a specific entry. This prediction can then be used to select more promising entries. For example, in a 1 : 1 matching setting, a 2LM would aim to select the highest scoring matrix entries among those competing for a single attribute. Often, 2LMs implicitly assume the confidence value reported by the 1LM as predictive of its quality. Thus, a matched attribute (a_i, b_j) pair with $M_{i,j} = 0.9$ confidence is preferred over one with $M_{i,j} = 0.8$ confidence. Accordingly, a prediction method named VAL uses the reported 1LM confidence value $M_{i,j}$ as the predicted value. The two additional entry predictors are based upon the observation that while an entry predictor provides a prediction for a single matrix entry, surrounding entries from its similarity matrix neighborhood can assist in assessing its quality. Both methods evaluate an entry with respect to its row and column. NNV normalizes the entry value by the difference of the highest and lowest entries in this neighborhood, while CRV normalizes by the maximum rank difference between entries.

	MCD		CRV		CNV		Val	
	Γ	sig.	Γ	sig.	Γ	sig.	Γ	sig.
Term	**0.98**	0.018	0.91	0	0.95	0	0.96	0
Token Path	**0.93**	0.002	0.67	0	0.67	0	0.34	0.042
WordNet	**0.93**	0	0.51	0.01	0.59	0	0.67	0

Table 5: Goodman-Kruskal Gamma correlation (Γ) of various Entry Predictors. sig. denotes the statistical significance level (p-$value$)

In this evaluation, two randomly selected schema pairs from the Web-forms dataset were matched using (AMC) Token Path, Term, and WordNet. For each similarity matrix entry (i, j), we thus had the 1LM result (VAL), three predictions calculated on its row and column neighborhood (NNV, CRV and MCD), and its expected true result (match/no match). Overall, 5869 entries were used to calculate the GK-Gamma correlation. The evaluation results are reported in Table 5. As we can observe, MCD has a better correlation than previously suggested predictors, with GK-Gamma values above .92 for all 1LMs. A two-tailed t-test confirms the significance of the results ($p\text{-}value < 0.05$).

5.4 CESM

We now evaluate CESM, providing a 1 : 1 match and aiming to optimize the bi-objective match problem that captures the tradeoff between MWBM and MCD. As was demonstrated by the Max predictor in Table 4, MWBM objective is expected to guide the optimization towards higher Recall, while MCD is expected to guide the optimization towards higher Precision. The β regularization free parameter control the tradeoff between Precision and Recall.

Three 1LM (Term, (AMC) TokenPath, and WordNet) were applied over the schema pairs of the Web-forms and Thalia datasets (Table 3), with 1 : 1 ground truth match. Using these 1LMs produced similarity matrices, which were provided as an input to CESM and five additional 2LMs, namely MWBM, SM, Dominants, Threshold(ν=0.85) and Max-Delta(Delta=0.1). Tuning parameters (ν and Delta) were selected to optimize F1-Score.

5.4.1 CESM *Effectiveness*

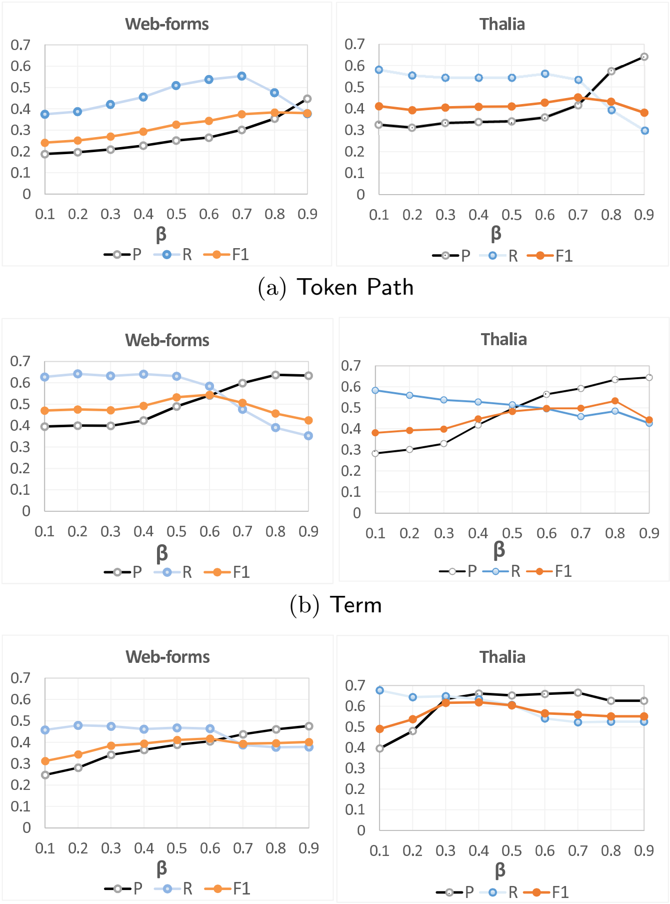
Table 6 presents the result of a standard binary Precision (P) and Recall (R) evaluation, reported per 1LM. Statistical significant differences in performance of CESM compared to the other 2LMs were measured using a one-side paired t-test (*p-value* < 0.05) and are marked with $**$.

Overall, independently of a given 1LM, CESM 2LM 1 : 1 decision maker has produced (on average) a better quality match, with up to +14.6% and +9.8% improvement in F1-Score over the **second best** 2LM baseline for the Web-forms and Thalia datasets, respectively. On average, CESM provides a match that has a significantly better Precision (with up to +17.5% boost over the **second best** 2LM), yet with slightly less Recall (but an overall better F1-Score). Comparing CESM with MWBM, this experiment confirms the ability of MCD to regulate the decision making of the former, with up to +31.2% and +35.1% improvement in F1-Score for the Web-forms and Thalia datasets, respectively. Furthermore, for the majority of cases, a notable improvement in Precision was measured with up to +37.9% and +55.8% improvement for the Web-froms and Thalia datasets, respectively. As can be observed, the drop in Recall by CESM was moderate compared to MWBM, and for some 1LMs, CESM even managed to improve the Recall level of MWBM. Finally, notable improvements in F1-Score were also measured in performance robustness terms, with an average RI value (across 1LMs) of 0.19 and 0.28 using CESM over MWBM for the Web-forms and Thalia datasets, respectively.

5.4.2 MCD *and the Precision vs. Recall tradeoff*

Using the Web-forms and Thalia datasets we empirically validate MCD's role in regulating the Precision vs. Recall tradeoff. The results of this validation are depicted in Figure 1 per 1LM and varied β regularization parameter value.

For all 1LMs, more emphasis is given to the MCD objective as β increases and the general trend is towards increased Precision at the expense of Recall. Such trend is most notable for the Term 1LM (with $R^2 = 0.93$ and $R^2 = 0.97$ for the Web-forms and Thalia datasets, respectively) compared to the two other 1LM (with an average of $R^2 = 0.92$ and $R^2 = 0.60$ for the Web-forms and Thalia datasets, respectively).



(a) Token Path

(b) Term

(c) WordNet

Figure 1: Effect of β using various 1LMs for the Web-forms (left) and Thalia (right) datasets

5.4.3 CESM *Efficiency*

The efficiency of CESM is measured both in terms of the number of iterations t and the absolute time in seconds it takes the algorithm to converge. For the Web-forms dataset, on average, CESM converged within 18($\pm$1) iterations or 16($\pm$1) seconds. For the Thalia dataset, the same convergence was reached, on average, within 12($\pm$1) iterations or 2($\pm$1) seconds. Recall that, on each iteration t, CESM samples $N = 10,000$ matches. Therefore, the average maximum number of matches explored during a single CESM run is about 180,000 and 120,000 for the Web-forms and Thalia datasets, respectively. A full enumeration of matches, on the other hand, has an exponential time-factor in the similarity matrix M dimensions. For example, an average similarity matrix within the Web-forms dataset has (on average) 40($\pm$2) rows and 39($\pm$2) columns with about 739($\pm$59) non-zero entries. Therefore, an enumeration of few hundred thousands of matches by CESM is actually negligible compared to the alternative of full enumeration.

The effect of problem complexity on CESM effectiveness (convergence), as determined by the input similarity matrix M size, was analyzed using the Web-forms dataset and the Term 1LM. The results of this analysis are depicted in Figure 2. Overall, CESM's number of iterations and absolute run-time increases linearly with matrix size, with $R^2 = 0.62$ and $R^2 = 0.67$, respectively.

	Threshold			Max-Delta			Dominants			SM			MWBM			CESM		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Token Path	.02	.03	.02	.20	**.67**	.30	**.48**	**.45**	.45	.27	.62	.36	.32	.58	.41	.29	.60	.38
Term	.51	.43	.41	.27	**.78**	.38	.09	.67	.15	.28	.64	.37	.41	.63	.48	**.53****	.60	**.55****
WordNet	.36	.52	.38	.15	**.67**	.24	.20	.62	.29	.20	.45	.27	.26	.46	.32	**.40****	.47	**.42****

(a) Web-forms

	Threshold			Max-Delta			Dominants			SM			MWBM			CESM		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Token Path	.00	.00	.00	.25	.53	.33	**.46**	.46	**.45**	.31	**.56**	.40	.33	.54	.41	.42	.54	**.45**
Term	.53	.48	.48	.25	.55	.33	.44	.53	.47	.32	**.58**	.40	.30	.52	.37	**.59****	.46	**.50****
WordNet	.57	.51	.51	.34	**.72**	.45	.50	.63	.53	.39	.71	.50	.43	.66	.51	**.67****	.52	**.56****

(b) Thalia

Table 6: Comparison of CESM ($N = 10,000$, $\rho = 0.01$, $\lambda = 0.3$, $\beta = 0.6$) with other 2LMs. Bold-face values mark the best performing matcher per each quality measure and 1LM. Statistical significant differences in performance of CESM compared to the other 2LMs are marked with **

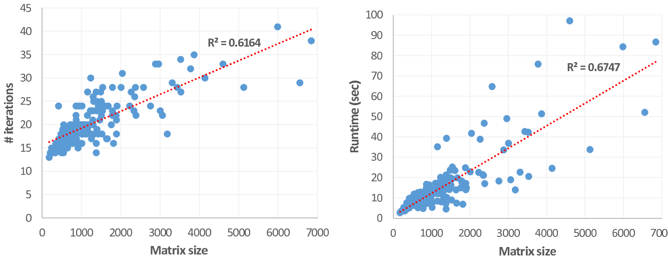


Figure 2: Effect of input similarity matrix M size on CESM effectiveness (convergence)

5.4.4 CESM *Sensitivity Analysis*

CESM sensitivity to changes in its configuration parameters, namely N (sample size), ρ (which defines the size of the elite sample on each iteration), and λ (which trades between the algorithm's model exploitation and exploration) is next examined. For that, the Web-forms dataset was used with Term as the 1LM. We fixed $\beta = 0.6$, the same parameter that was used in Table 6. Modifying each time a single parameter while fixing the two other parameters (using the default configuration of $N = 10,000$, $\rho = 0.01$, $\lambda = 0.3$ for reference), we recorded the variation in CESM effectiveness (as captured by its match Precision, Recall and F1) and efficiency (as captured by the number of iterations t and the absolute time in seconds it takes the algorithm to converge).

The sensitivity analysis results are given in Figure 3, confirming previous reports on the impact of the three parameters on the convergence of the CE Method [31, 21]. Specifically to the CESM instantiation, we observe that among the three parameters the sample size N and the λ smoothing parameters have the strongest impact on CESM efficiency.

Analyzing the effect of the sample size, we observe that with the increase in N CESM effectiveness (as measured by P,R and F1) improves until reaching a plateau around $N = 5,000$ with no significant impact on its convergence (in terms of number of iterations t), yet with an expected direct (linear) effect on its runtime due to increase in sample size.

Next, for the smoothing parameter, we observe that when λ increases, CESM effectiveness again improve up to a point. Here we observe an opposite trend, yet expected [31, 21], as smaller λ values allow CESM better model exploration (using the current derived reference parameter v^t) with less model exploitation of previously learned reference parameter v^{t-1}, leading to a slower convergence.

Finally, as we can observe, the elite sample size that is defined by the ρ parameter has a moderate effect on CESM convergence, which also coincide with previous studies on the CE Method [31, 21]. With the increase in ρ, the algorithm learns from a larger set of elite samples, corresponding with a more frequent event estimation. Hence, it takes longer to reach the goal of estimating the rare event of obtaining the optimal solution. Therefore, learning from a smaller elite sample results in a more effective match.

5.5 Improving Max-Delta using MCD

We now compare the MCD enhanced Max-Delta version (Section 4.4) against its basic version (Section 2). To this end, we use three 1LMs of ORE, namely Term, Token-Path, and WordNet, and the AllContextW recommended configuration of COMA 3.0 [23], on 30 randomly selected pairs from the University dataset and 30 pairs from the Purchase-Order dataset. Both datasets assume a $1 : n$ matching, allowing us to evaluate the impact MCD has on the Max-Delta matcher performance. We set **Delta** $= 0.1$ for all MCD experiments. The community version of COMA 3.0 was used with its recommended workflow AllContextW. The workflow applies Max-Delta with **Delta** $= 0.05$ and a threshold $t = 0.4$. We implemented MCD prediction in COMA, using $t = 0.2$.

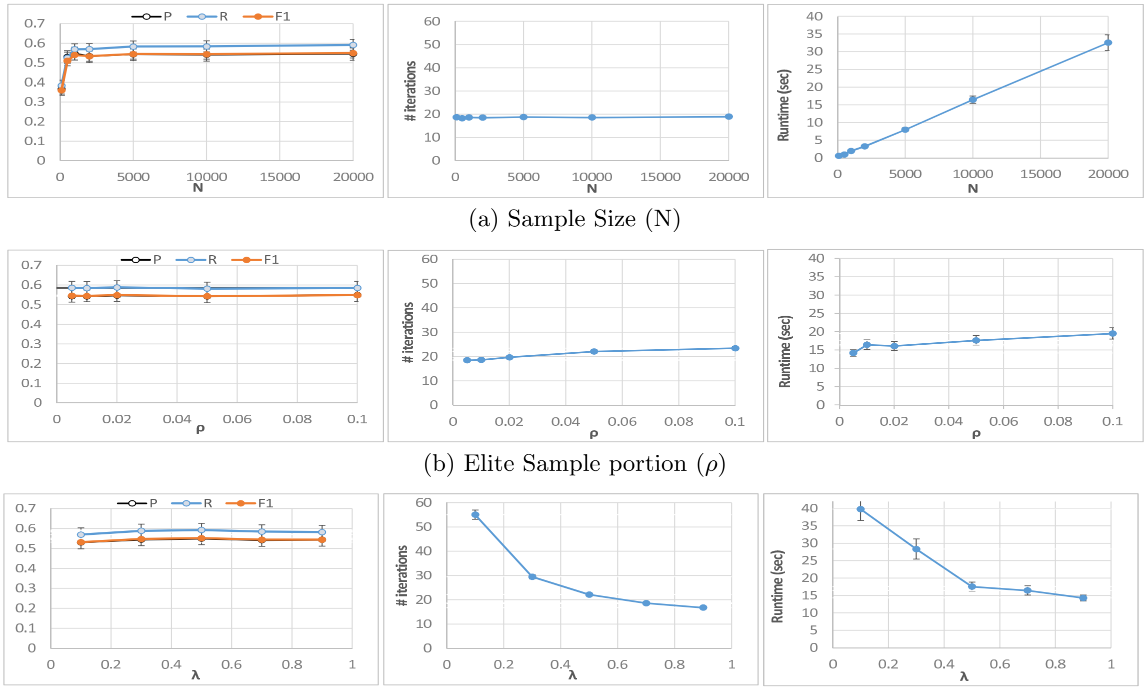
	Max-Delta + MCD			Max-Delta		
	P	R	F1	P	R	F1
Term	**0.17****	0.38	**0.21****	0.10	**0.50**	0.16
Token-Path	**0.18****	0.38	**0.21****	0.11	**0.43**	0.17
Word-Net	**0.11****	0.45	**0.17****	0.09	**0.50**	0.15
COMA 3.0	0.29	**0.29****	**0.28****	0.36**	0.15	0.19

(a) University

	Max-Delta + MCD			Max-Delta		
	P	R	F1	P	R	F1
Term	**0.16****	0.47	**0.22****	0.12	**0.54**	0.18
Token-Path	**0.28****	0.39	0.26	0.22	**0.45**	**0.27**
Word-Net	**0.19****	0.51	**0.26****	0.17	**0.56**	0.24
COMA 3.0	**0.51****	0.48	**0.48**	0.43	**0.55****	0.47

(b) Purchase Order

Table 7: Comparison of Max-Delta enhanced with MCD vs. basic version. Bold-face values for best performing matcher per quality measure and 1LM. Statistical significant differences in performance of the former compared to the last are marked with **



(a) Sample Size (N)

(b) Elite Sample portion (ρ)

(c) Model Smoothing (λ)

Figure 3: CESM sensitivity analysis (N, ρ, λ)

Results are reported in Table 7. Similarly to $1:1$ match results with CESM, the results demonstrate a consistent improvement of an MCD based Max-Delta over the basic Max-Delta. Significance of the differences between the methods was tested using a one-side paired t-test (p-$value < 0.05$) and significantly higher performance values are marked with **. In most 1LM-dataset combinations, basing Max-Delta on the MCD matrix Δ improves Precision up to +70% and +33.3% for the University and Purchase Order datasets, respectively. This comes somewhat at the expense of Recall, yet with an overall improvement in F1-Score, up to +31.2% and +22.2% for the University and Purchase Order datasets, respectively. COMA results for University are surprising as Recall is improved at the expense of somewhat lower Precision, with an overall improvement in F1-Score for both datasets. Finally, notable improvements in F1-Score were also measured in performance robustness terms, with an average RI value (across 1LMs) of 0.53 and 0.2 using Max-Delta + MCD over Max-Delta for the University and Purchase Order datasets, respectively.

6. RELATED WORK

Schema matching research has expanded and evolved over three decades (see surveys [2, 34, 4, 35] and books [9, 12, 3]) and is widely recognized as a basic research field, contributing to efforts in data integration, semantic reasoning, and deep Web exploration (*e.g.*, [17, 7, 24]).

Early approaches of schema matching assume the raw 1LM similarity to be predictive of the match quality and attempted to maximize it. The application of a-priori evaluation of schema features to direct and influence the execution of schema matching was first suggested by Tu and Yu [36], which used schema features to select execution strategies. Similar work was done by Peukert et al. [27]. This approach was generalized and explored by Sagi and Gal [33], who introduced prediction as a method to evaluate similarity scores and predict match quality. This paper explores a new predictor, MCD and uses it to regularize matchers by using the prediction it provides.

Cruz et al. [5] proposed a local confidence measure LC that computes the difference between the average similarity of selected matches for a given concept and the average of all other similarity measures of the same row in a similarity matrix. LC is similar in spirit to MCD, but its use was entirely different. It was used (and assumed to be) a measure of quality in the absence of an exact match. MCD, on the other hand, is shown to serve as a good predictor and a regulator for tuning the matching task. In Section 5 we showed LC, as a matrix predictor, to be inferior to MCD.

CESM is based on the Cross Entropy (CE) Method [31], a Monte Carlo framework for rare event estimation and combinatorial optimization. This Method has been applied in domains such as machine learning, simulation, and networks [31]. To the best of our knowledge, our work is the first to use the CE Method in the schema matching domain.

7. CONCLUSIONS AND FUTURE WORK

In this work we presented a new schema matching predictor, MCD, discussed its properties, and used it to enhance the performance of two existing state-of-the-art schema matchers. Our empirical evaluation shows MCD to be more predictive than any known schema matching predictor in the literature by far. We also demonstrated empirically its usefulness for schema matching in general.

Our work can be extended in several ways. First, we intend to test the impact of MCD on additional schema matchers. Second, an important observation from this work is that, diversification in schema matching plays an important role. Hence, we would like to explore additional methods for schema matching diversification and analyze their impact on quality using the evaluation methodology proposed in this work. Finally, while diversification was mainly utilized in this work for evaluating and improving the performance of 2LM decision makers, we believe that such diversification considerations may be used to develop new baseline 1LMs whose decisions are encoded in the similarity matrix.

Acknowledgments

We thank Roee Shraga and Igal Shprincis for their assistance in performing the empirical evaluation.

8. REFERENCES

[1] S. Anand. The multi-criteria bipartite matching problem. 2006.

[2] C. Batini, M. Lenzerini, and S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys (CSUR)*, 18(4):323–364, 1986.

[3] Z. Bellahsene. *Schema Matching and Mapping*. Springer-Verlag New York Inc, 2011.

[4] P. A. Bernstein, J. Madhavan, and E. Rahm. Generic schema matching, ten years later. *Proceedings of the VLDB Endowment*, 4(11):695–701, 2011.

[5] I. F. Cruz, F. P. Antonelli, and C. Stroe. Efficient selection of mappings and automatic quality-driven combination of matching methods. In *Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009) Chantilly, USA, October 25, 2009*, 2009.

[6] H. H. Do and E. Rahm. Coma: a system for flexible combination of schema matching approaches. In *Proceedings of VLDB*, pages 610–621. VLDB Endowment, 2002.

[7] R. dos Santos Mello, S. Castano, and C. A. Heuser. A method for the unification of XML schemata. *Information and Software Technology*, 44(4):241 – 249, 2002.

[8] M. Ehrgott. *Multicriteria optimization*, volume 2. Springer, 2005.

[9] J. Euzenat. Semantic precision and recall for ontology alignment evaluation. In *Proc. IJCAI*, pages 348–353, 2007.

[10] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag New York Inc, 2007.

[11] G. E. Evans, J. M. Keith, and D. P. Kroese. Parallel cross-entropy optimization. In *Proc. of WSC*, pages 2196–2202, 2007.

[12] A. Gal. Uncertain schema matching. *Synthesis Lectures on Data Management*, 3(1):1–97, 2011.

[13] A. Gal, A. Anaby-Tavor, A. Trombetta, and D. Montesi. A framework for modeling and evaluating automatic semantic reconciliation. *The VLDB Journal*, 14(1):50–67, 2005.

[14] A. Gal and T. Sagi. Tuning the ensemble selection process of schema matchers. *Information Systems*, 35(8):845–859, 2010.

[15] Z. Galil, S. Micali, and H. Gabow. An O(EV\logV) algorithm for finding a maximal weighted matching in general graphs. *SIAM Journal on Computing*, 15(1):120–130, 1986.

[16] M. Gawinecki. Abbreviation expansion in lexical annotation of schema. *Camogli (Genova), Italy June 25th, 2009 Co-located with SEBD*, page 61, 2009.

[17] B. He and K. C.-C. Chang. Statistical schema matching across web query interfaces. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 217–228, New York, NY, USA, 2003. ACM.

[18] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997.

[19] J. Madhavan, P. Bernstein, A. Doan, and A. Halevy. Corpus-based schema matching. In *Proc. ICDE*, pages 57 – 68, april 2005.

[20] M. Mao, Y. Peng, and M. Spring. A harmony based adaptive ontology mapping approach. In *Proc. of SWWS*, 2008.

[21] L. Margolin. On the convergence of the cross-entropy method. *Annals of Operations Research*, 134(1):201–214, 2005.

[22] A. Marie and A. Gal. On the stable marriage of maximum weight royal couples. In *Proceedings of AAAI Workshop on Information Integration on the Web*, 2007.

[23] S. Massmann, S. Raunich, D. Aumüller, P. Arnold, and E. Rahm. Evolution of the coma match system. *Ontology Matching*, 49, 2011.

[24] P. D. Meo, G. Quattrone, G. Terracina, and D. Ursino. Integration of XML schemas at various 'severity' levels. *Information Systems*, 31(6):397 – 434, 2006.

[25] G. Modica, A. Gal, and H. Jamil. The use of machine-generated ontologies in dynamic information seeking. In *CoopIS*, pages 433–447, 2001.

[26] E. Peukert, J. Eberius, and E. Rahm. AMC-a framework for modelling and comparing matching systems as matching processes. In *ICDE*, pages 1304–1307. IEEE, 2011.

[27] E. Peukert, J. Eberius, and E. Rahm. A self-configuring schema matching system. In *ICDE*, 2012.

[28] L. Ramshaw and R. E. Tarjan. On minimum-cost assignments in unbalanced bipartite graphs. Technical report, HP Labs technical report HPL-2012-40R1, www. hpl. hp. com/techreports/HPL-2012-40R1. html, 2012.

[29] L. Ratinov and E. Gudes. Abbreviation expansion in schema matching and web integration. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 485–489. IEEE Computer Society, 2004.

[30] P. Rodriguez-Gianolli and J. Mylopoulos. A semantic approach to xml-based data integration. In H. S.Kunii, S. Jajodia, and A. Slvberg, editors, *Conceptual Modeling ER 2001*, volume 2224 of *Lecture Notes in Computer Science*, pages 117–132. Springer Berlin Heidelberg, 2001.

[31] R. Y. Rubinstein and D. P. Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer, 2004.

[32] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*, volume 707. John Wiley & Sons, 2011.

[33] T. Sagi and A. Gal. Schema matching prediction with applications to data source discovery and dynamic ensembling. *The VLDB Journal*, 22(5):689–710, 2013.

[34] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys (CSUR)*, 22(3):183–236, 1990.

[35] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics IV*, pages 146–171, 2005.

[36] K. Tu and Y. Yu. CMC: Combining multiple schema-matching strategies based on credibility prediction. In L. Zhou, B. Ooi, and X. Meng, editors, *Database Systems for Advanced Applications*, volume 3453 of *LNCS*, pages 995–995. Springer Berlin / Heidelberg, 2005.

[37] J. Wang, J. Wen, F. Lochovsky, and W. Ma. Instance-based schema matching for web databases by domain-specific query probing. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 408–419. VLDB Endowment, 2004.

[38] M. Weidlich, T. Sagi, H. Leopold, A. Gal, and J. Mendling. Predicting the quality of process model matching. In *Business Process Management*, volume 8094 of *LNCS*, pages 203–210. Springer, 2013.

APPENDIX

A. RANDOM MATCH SAMPLING

To apply the CE Method, we first need to propose a pdf family $f(\cdot; v)$ defined over Σ for various reference parameter values $v \in \mathcal{V}$ from which random matches can be sampled [31]. By choosing the pdf family to come from the *natural exponential family* (NEF) [31], the optimal (learning) reference parameter v^t can be analytically derived at each iteration t [31].

We can define such choice of NEF density $f(\cdot; v)$ for the schema matching problem as follows. Let $G(V_1, V_2, E)$ be a bipartite graph with $|V_1| = n$ and $|V_2| = m$, whose edge weights $w(e)$ are assigned according to matrix M: $w(e) = M_{i,j}; \forall e = (i,j) \in E : v_i \in V_1, v_j \in V_2$.

We observe that each match $\sigma \in \Sigma$ is formed by a selection of a subset of edges $E' \subseteq E$. Let $\delta_{i,j} = \delta_{[e=(i,j) \wedge e \in E']}$ be an indicator function, encoding the event that edge $e = (i,j)$ has been chosen to be part of E'. For a given reference parameter $v_{i,j} \in [0,1]$, a random choice of such edge selection follows the (discrete) $Bernoulli(v_{i,j})$ distribution:

$$\mathbb{P}_{v_{i,j}}(\delta_{i,j}) = v_{i,j}^{\delta_{i,j}} (1 - v_{i,j})^{1-\delta_{i,j}}. \tag{16}$$

Assuming that edges are selected independently of each other (i.e., $\delta_{i,j} \sim Bernoulli(v_{i,j})$ are i.i.d's), we now choose $f(\cdot; v)$ to be the density that captures the events of edge subsets $E' \subseteq E$ formation, with reference parameter vector $v = (v_1, v_2, \ldots, v_{|E|})$. Therefore, we have:

$$f(E'; v) = \prod_{(i,j) \in E'} v_{i,j}^{\delta_{i,j}} (1 - v_{i,j})^{1-\delta_{i,j}}. \tag{17}$$

We note that, choosing subsets of edges E' may sometimes produce incorrect matches that actually violate the 1:1 match restriction. To overcome this "hurdle" and make sure that the result of the CE Method would be a correct match, we now define an adjusted match quality measure for any selection $E' \subseteq E$, as follows:

$$\mathcal{Q}'(E', M) = \left\{ \begin{array}{ll} \mathcal{Q}(E', M), & E' \in \Sigma \\ -\infty & otherwise \end{array} \right. \tag{18}$$

Obviously, for each edge $e = (i,j)$ we have $M_{i,j} \geq 0$, hence $\max_{E' \in E} \mathcal{Q}'(E', M) \geq 0$. Therefore, from the definition of $\mathcal{Q}'(\cdot, M)$, any match that will be produced will be a correct match in Σ. Moreover, since we assume that E' maximizes $\mathcal{Q}'(E', M)$, it implies that E' maximizes $\mathcal{Q}(E', M)$.

Algorithm 3 now describes the details of the match sampling. The main idea behind the algorithm relies on the fact that, the only edge subsets E' that we need to consider are those that provide a correct 1:1 match in Σ.

Algorithm 3 Random Match Sampling

```
input: $M$,$v$
$E := \{(i,j); i = 1, \ldots, n; j = 1, \ldots, m\}$
$\sigma := \emptyset$
while $E \neq \emptyset$ do
    select next edge $(i,j) \in E$ to consider at random
    draw $u \sim U[0,1]$
    if $v_{i,j} \geq u$ then
        $\sigma := \sigma \cup \{(i,j)\}$
        $E := E \setminus \{(i,j)\}$
    end if
    for $(i', j') \in S$ do
        if $i' = i \vee j' = j$ then
            $E := E \setminus \{(i', j')\}$
        end if
    end for
end while
return $\sigma$
```

The algorithm gets as an input a similarity matrix M and a reference (Bernoulli) parameter vector v. The algorithm starts with the full set of bipartite graph edges E, and picks edges at random to be included in the match σ, each edge is picked according to a single random Bernoulli trial [32]. To maintain a correct match, the algorithm removes any adjunct edges that violate the 1:1 match correctness.

B. REFERENCE PARAMETER DERIVATION

Here we explain how the v^t reference parameter vector is learned on each iteration t. Recall that, for a given performance level $\gamma_t \in \mathbb{R}$, derived as the $(1-\rho)$ quantile of $\overrightarrow{\Sigma_{\mathcal{Q}}}$, our aim is to estimate the following likelihood:

$$l(\gamma_t) = \mathbb{P}_{v^{t-1}}(\delta_{[\mathcal{Q}(\boldsymbol{\Sigma}, M) \geq \gamma_t]}) = \mathbb{E}_{v^{t-1}}(\delta_{[\mathcal{Q}(\boldsymbol{\Sigma}, M) \geq \gamma_t]}) \tag{19}$$

The estimation is done via importance sampling as follows. First, we observe that:

$$\mathbb{E}_{v^{t-1}}(\delta_{[\mathcal{Q}(\boldsymbol{\Sigma}, M) \geq \gamma_t]}) \frac{f(\boldsymbol{\Sigma}; v^t)}{f(\boldsymbol{\Sigma}; v^t)} = \int_{\Sigma} \delta_{[\mathcal{Q}(\boldsymbol{\Sigma}, M) \geq \gamma_t]} f(\boldsymbol{\Sigma}; v^{t-1}) \frac{f(\boldsymbol{\Sigma}; v^t)}{f(\boldsymbol{\Sigma}; v^t)} d\sigma \tag{20}$$

After applying the change of measure, the estimation problem defined in Eq. 19 can be now expressed by the following Likelihood Ratio (LR) estimation problem:

$$l_{LR}(\gamma_t) = \mathbb{E}_{v^t}(\delta_{[\mathcal{Q}(\boldsymbol{\Sigma}, M) \geq \gamma_t]}) \frac{f(\boldsymbol{\Sigma}; v^{t-1})}{f(\boldsymbol{\Sigma}; v^t)} \tag{21}$$

The corresponding LR estimator is therefore:

$$\hat{l}_{LR}(\gamma_t) = \frac{1}{N} \sum_{k=1}^{N} \delta_{[\mathcal{Q}(\sigma_k, M) \geq \gamma_t]} \frac{f(\sigma_k; v^{t-1})}{f(\sigma_k; v^t)}, \tag{22}$$

where $\sigma_k \sim f(\cdot; v^t); k = 1, \ldots, N$.

An hypothetic reference distribution $f^* = f^*(\boldsymbol{\Sigma})$ under which the LR estimation is the most accurate possible (i.e., $\hat{l}_{LR}(\gamma_t) = l(\gamma_t)$), would be obtained as follows:

$$f^*(\boldsymbol{\Sigma}) = \frac{\delta_{[\mathcal{Q}(\boldsymbol{\Sigma}, M) \geq \gamma_t]} f(\boldsymbol{\Sigma}, v^{t-1})}{l(\gamma_t)} \tag{23}$$

Unfortunately, $f^*(\boldsymbol{\Sigma})$ depends on the unknown parameter $l(\gamma_t)$. Choosing a density f' from the same family of densities $\{f(\cdot; v^t), v^t \in \mathcal{V})\}$, the idea now would be to choose the reference parameter $v^t \in \mathcal{V}$ such that the "distance" between f^* and $f' = f(\cdot; v^t)$ is as minimum as possible. Such "distance" is captured in the CE Method using the **Kullback-Leibler Divergence** (KLD) between the hypothetic optimal density f^* and the reference density $f(\cdot; v^t)$ in hand, i.e.:

$$\begin{array}{l} \mathcal{D}_{KL}(f^*(\boldsymbol{\Sigma}), f(\boldsymbol{\Sigma}; v^t)) = \mathbb{E}_{f^*} \ln \frac{f^*(\boldsymbol{\Sigma})}{f(\boldsymbol{\Sigma}; v^t)} \\ = \int_{\Sigma} f^*(\boldsymbol{\Sigma}) \ln f^*(\boldsymbol{\Sigma}) d\sigma - \int_{\Sigma} f^*(\boldsymbol{\Sigma}) \ln f(\boldsymbol{\Sigma}; v^t) d\sigma \end{array} \tag{24}$$

Noting that the left term of the KLD measure is independent of v^t, all we need is to minimize the Cross Entropy (CE) "distance" between f^* and $f(\cdot; v^t)$. Minimizing the CE distance in Eq. 24 is further equivalent to solving the following maximization problem:

$$\max_{v^t} \int_{\Sigma} \frac{\delta_{[\mathcal{Q}(\boldsymbol{\Sigma}, M) \geq \gamma_t]} f(\boldsymbol{\Sigma}, v^{t-1})}{l(\gamma_t)} \ln f(\boldsymbol{\Sigma}, v^t) d\sigma, \tag{25}$$

where $f^*(\boldsymbol{\Sigma})$ was substituted according to Eq. 23.

Finally, dropping the "constant" $l(\gamma_t)$ we get the following equivalent maximization problem:

$$\max_{v^t} \mathbb{E}_{v^{t-1}}(\delta_{[\mathcal{Q}(\boldsymbol{\Sigma}, M) \geq \gamma_t]}) \ln f(\boldsymbol{\Sigma}, v^t) d\sigma \tag{26}$$

The optimal reference parameter v^* can be, therefore, estimated as follows:

$$\max_{v^t} \frac{1}{N} \sum_{k=1}^{N} \delta_{[\mathcal{Q}(\sigma_k, M) \geq \gamma_t]} \ln f(\sigma_k, v^t) d\sigma, \tag{27}$$

where $\sigma_k \sim f(\cdot; v^{t-1}); k = 1, \ldots, N$.

Next, given that $f(\cdot; v^{t-1})$ follows the distribution defined in Eq. 17, we now note that:

$$\frac{\partial}{\partial v_{i,j}^t} \ln f(\cdot, v^t) = \frac{\delta_{i,j}}{v_{i,j}^t} - \frac{1-\delta_{i,j}}{1-v_{i,j}^t} = \frac{1}{v_{i,j}^t(1-v_{i,j}^t)}(\delta_{i,j} - v_{i,j}^t) \tag{28}$$

For each parameter $v_{i,j}^t$, its optimal value is achieved by taking the partial derivative in Eq. 27 according to $v_{i,j}^t$ and equal it to zero:

$$\frac{\partial}{\partial v_{i,j}^t}\left(\frac{1}{N}\sum_{k=1}^{N}\delta_{[\mathcal{Q}'(\sigma_k,M)\geq\gamma_t]}\ln f(\sigma_k, v^t)\right) = 0 \tag{29}$$

Noting that we can push the derivative inside and using the result of Eq. 28, we get that:

$$\frac{1}{v_{i,j}^t(1-v_{i,j}^t)}\frac{1}{N}\sum_{k=1}^{N}\delta_{[\mathcal{Q}'(\sigma_k,M)\geq\gamma_t]}(\delta_{i,j} - v_{i,j}^t) = 0 \tag{30}$$

Finally, after dropping the constants we obtain the closed formula for calculating the optimal reference parameters:

$$v_{i,j}^t = \frac{\sum_{k=1}^{N}\delta_{[\mathcal{Q}'(\sigma_k,M)\geq\gamma_t]}\delta_{i,j}}{\sum_{k=1}^{N}\delta_{[\mathcal{Q}'(\sigma_k,M)\geq\gamma_t]}} \tag{31}$$