Entity Matching Across Multiple Heterogeneous Data Sources

Chao Kong[1], Ming Gao[1(✉)], Chen Xu[2], Weining Qian[1], and Aoying Zhou[1]

[1] Institute for Data Science and Engineering, ECNU-PINGAN Innovative Research Center for Big Data, East China Normal University, Shanghai, China
kongchao315@163.com, {mgao,wnqian,ayzhou}@sei.ecnu.edu.cn
[2] Technische Universität Berlin, Berlin, Germany
chen.xu@tu-berlin.de

Abstract. Entity matching is the problem of identifying which entities in a data source refer to the same real-world entity in the others. Identifying entities across heterogeneous data sources is paramount to entity profiling, product recommendation, etc. The matching process is not only overwhelmingly expensive for large data sources since it involves all tuples from two or more data sources, but also need to handle heterogeneous entity attributes. In this paper, we design an unsupervised approach, called EMAN, to match entities across two or more heterogeneous data sources. The algorithm utilizes the locality sensitive hashing schema to reduce the candidate tuples and speed up the matching process. To handle the heterogeneous entity attributes, we employ the exponential family to model the similarities between the different attributes. EMAN is highly accurate and efficient even without any ground-truth tuples. We illustrate the performance of EMAN on re-identifying entities from the same data source, as well as matching entities across three real data sources. Our experimental results manifest that our proposed approach outperforms the comparable baseline.

Keywords: Entity matching · Exponential family · Locality sensitive hashing

1 Introduction

Entity matching is the problem of identifying which entities in a data source link to the same entities in the other data sources. It is a well known and paramount problem that arises in many research fields, including data cleaning and integration, information retrieval and machine learning. There are many applications which can benefit from the entity matching task. In the first place, users in a social network may be the same individuals in the other platforms, but each user profile and user behavior may be slightly different, e.g., containing different abbreviations, and missing some information. We can improve the product recommendations after determining the more complete user behaviors. There is one more point, we should touch on that two websites of second-hand housing

S.B. Navathe et al. (Eds.): DASFAA 2016, Part I, LNCS 9642, pp. 133–146, 2016.
DOI: 10.1007/978-3-319-32025-0_9

may share many house information. After we derive the matched entities across the different websites, we can insight into the more complete house information. Therefore, we have studied the problem of matching entities across two or more heterogeneous data sources in this paper.

However, the entity matching is often a challenging task due to following reasons: (1) it is extremely expensive for the large data sets since the process considers all the tuples as candidates; (2) the process is arduous to compare many heterogeneous attributes for tuple of entities from different data sources; (3) there may be some missing attributes for an entity in some Web applications.

In this paper, we provide an approach, called EMAN, to match entities across two or more heterogeneous data sources. We formulate entity matching task as an unsupervised learning problem. Our proposed approach can result in promising accuracy without ground-truth which are usually arduous and costly to collect in Web applications. For the provided approach, we would like to address the three challenges highlighted earlier. It utilizes the exponential family to combine heterogeneous entity attributes, handle missing data in the unsupervised framework, employ locality sensitive hashing (LSH) to speed up the computation. In summary, our major contributions are as follow.

- We propose an unsupervised method to match entities across two or more heterogeneous data sources. The approach is a unified one which integrates the heterogeneous entity attributes, and employs the distributions from the exponential family to model the similarities of different attributes.
- We employ the locality sensitive hashing (LSH) to block entities. With LSH, EMAN can perform very efficiently but still maintain the promising matching results.
- We illustrate the performance of our algorithm against a comparable baseline on three real data sources. Empirical study results manifest that EMAN outperforms baseline in matching entities across two or more data sources.

The rest of paper is organized as follows. We shortly discuss the related work in Sect. 2. We formally define the problem and describe the overview of our algorithm in Sect. 3. We present the entity matching method in Sect. 4 and report our empirical study in Sect. 5. Finally, we conclude this paper in Sect. 6.

2 Related Work

Entity matching aims at detecting several entities which describe the same entity from given datasets. The study of entity matching problem has become a hot topic in recent years, and some earlier studies can go back to 1950s [1]. However, entity matching is also a wide research problem studied in multiple research communities. In the traditional database community, the problem is described as data matching [2], data deduplication [3], instance identification [4], Merge/Purge [5] and record linkage [6]; In the information retrieval community, the same problem is described as entity resolution [7–11]. The object linkage,

object identification [12], and duplicate detection [13–15] are also commonly referred to the same task.

In general, the existing studies of entity matching can be mainly divided into two categories: classification-based and rule-based [17]. For classification-based approaches, they assign labels to a pair after learning the patterns from the training data. Mikhail and Raymond train a classifier by SVM with high accuracy [18]. Dong [19] implements the entity matching algorithm with three crucial features among records based on machine learning technique. The algorithm improves the accuracy of entity matching by merging the attribute values to enrich record information gradually. However, it is not easy to find the exact matched pairs for learning a classifier.

For rule-based approaches, they are deterministic linkage approach [20,21] and judge whether a record pair is matched or not based on rules. Jiannan Wang trains to obtain the most appropriate set of rules on the premise of given rules which is difficult to obtain [17]. Moreover, if the given rules are not sufficient, the deviation of classification results is large and the classification result may not be acceptable. Whang [9] proposed an entity resolution method with evolving rules based on relationship between dynamic semantics and resolution rules to solve the problem of interaction among results. Whang [7] indicates the entity resolution is not an one-time process but incremental process changing constantly. Wang et al. first proposes *how similar is similar problem* in entity matching, and they address the rule-based method to identify the most appropriate similarity functions and thresholds to find entities effectively [17]. Rastogi et al. focus on the scale generic entity matching problem which is implemented with a parallel framework on Hadoop [22]. Lee et al. also attempt to address the scalability problem of entity matching [23]. In that work, they exploit a materialization structure inspired by top-k query processing and develop a scalable entity matching algorithm for evolving rules. However, the rules for linking entities are arduous to learn from data sets.

Fellegi-Sunter's approach is also a rule-based method, and solves the record linkage problem via using an unsupervised and probabilistic linkage approach [16,24]. It works well only when the linkage problem is simple and exact one-to-one matching of username and other user attributes. Sadinle et al. extend Fellegi-Sunter's model to present a probabilistic method for linking multiple data files [25]. Currently, many recent applications generate many data associated with poor quality, including heterogeneous in attributes, error, incomplete and missing values, etc. However, existing entity matching approaches cannot integrate heterogeneous entity attributes and handle missing values. Gao et al. also extend Fellegi-Sunter's approach to link users across different social networks. Their approach can handle heterogeneous user attributes and missing values in user profiles [26]. In addition, Fellegi-Sunter's approach can only link records from two data sources. For linking more than two data sources, the false positive tuples may be large if we link them pair-to-pair by using Fellegi-Sunter's approach. These are the focuses of this work.

3 Entity Matching Approach

In this section, before we overview our proposed approach, we describe a formal definition of the entity matching problem.

3.1 The Problem Definition

We assume that there are N data sources ($N > 1$). Let E_i, $1 \leq i \leq N$, be the set of entities from the $i-$th source and $\alpha_i(e_{ij})$ represent the observed features of $e_{ij} \in E_i$, i.e., $\alpha_i(e_{ij})$ represents the observed feature vector of e_{ij} from source E_i. Let $\alpha_i(E_i)$ be the set of attribute feature vectors of entities from source E_i. The set of all candidates tuples T can be represented as $\prod_{i=1}^{k} \alpha_i(E_i)$. The entity matching problem is to determine the matched tuples M and unmatched tuples U in T, i.e.,

$$M = \{(\alpha_1(e_{1j_1}), \ldots, \alpha_N(e_{Nj_N}))|e_{1j_1} = \ldots = e_{Nj_N}, e_{ij_i} \in E_i\}, \quad (1)$$

$$U = \{(\alpha_1(e_{1j_1}), \ldots, \alpha_N(e_{Nj_N}))|\exists N_1, N_2, e_{N_1 j_{N_1}} \neq e_{N_2 j_{N_2}}, e_{ij_i} \in E_i\}. \quad (2)$$

When $(\alpha_1(e_{1j_1}), \alpha_2(e_{2j_2}), \ldots, \alpha_N(e_{Nj_N})) \in M$ means that entities e_{ij_i}, $1 \leq i \leq N$, are the same, while $(\alpha_1(e_{1j_1}), \alpha_2(e_{2j_2}), \ldots, \alpha_N(e_{Nj_N})) \in U$ means that at least an entity e_{ij_i} is different from the others. Suppose that each entity can at most match an entity from E_i, M can therefore have at most $min(|E_1|, |E_2|, \cdots, |E_N|)$ matched entity tuples. We may ideally want $T = M \cup U$ but T is usually an extremely large set in real. We therefore consider a smaller T that includes M utilizing some blocking techniques.

3.2 Overview of EMAN

Our proposed entity matching approach consists of four components as following.

Step 1: Candidate tuple generation. The major computational cost is significantly impacted by generating candidate tuples. Blocking methods, such as n-gram indexing and sorted neighborhood, may be the feasible techniques to reduce the number of candidate tuples [27]. However, the number of candidate tuples with N sources of n entities containing in b blocks is $O(\frac{n}{b^{N-1}}^N)$. The efficiency and accuracy are significantly impacted by the number of blocks. We therefore utilize the LSH (Locality Sensitive Hashing) schema to speed up the candidate tuple generation since the number of blocks can be arbitrary large (Based on n-gram model, entities can be blocked by utilizing LSH when some string attributes, such as name, address etc., can be represented as a binary vector after shingling).

Step 2: Entity vectorization and similarity computations. We determine a similarity function s^j to evaluate the similarity between the $j-$th feature of two entities from a candidate tuple. For tuple

$$t_i = (\alpha_1(e_{1i_1}), \alpha_2(e_{2i_2}), \cdots, \alpha_N(e_{Ni_N})),$$

we can compute a similarity vector for m attributes of any two entities of tuple t_i using similarity functions $\{s^j\}_{j=1}^m$. There are $\binom{N}{2}$ similarity vectors between different entity pairs. We take the minimum value of each entry over $\binom{N}{2}$ similarity vectors to model the similarity of tuple t_i, denoted as a m-dimensional vector γ_i. An entity may have multiple attributes. Some of them are individual demographic attributes, e.g., name, location, date, URL and etc. These can be of different data types (e.g., numeric, text, string, categorical, etc.). These attributes can be represented in set and distribution types. Our proposed approach models the similarities between entities to accommodate heterogeneous features using different probability distributions in exponential family.

Step 3: Parameter learning. Given each similarity vector γ_i, EMAN models the similarity values of tuples using two different probability distribution functions, one for matched tuples and another for unmatched ones. The parameter learning step is to infer parameters of the two distributions. More details will be covered in Sect. 4.

Step 4: Tuple scoring and label assignment. For a tuple $t_i \in T$, its score can be computed by $\log \frac{P(t_i \in M|\gamma_i, \hat{\Theta})}{P(t_i \in U|\gamma_i, \hat{\Theta})}$ (for ease of computation). Tuple t_i is more likely to be matched tuple if its score is greater than 0, and otherwise unmatched tuple. Given a threshold, the matched scores of entity tuples are used to judge if they belong to the matched or unmatched tuple sets, i.e., M or U. A tuple is judged as matched tuple if its matched score is larger than the threshold, and otherwise unmatched tuple.

Unlike the earlier Fellegi-Sunter's method, EMAN considers both discrete and continuous similarities as a wider range of probability distributions from the exponential family to model the similarity values of matched and unmatched entity tuples (in Step 1). This is an important extension to handle the heterogenous attribute types, including string, numeric, set, distribution, etc., these exist in the entity matching task.

3.3 EMAN Algorithm

We now present the full EMAN algorithm in Algorithm 1. In this algorithm, PM maintains the set of matched entity tuples. At Lines 2–4, the algorithm employs LSH to block entities from N data sources. At Lines 5–9, it generates all the candidate tuples. A candidate tuple consists of N entities from N data sources, where all entities in a tuple are from the same bucket. In this step, it also computes the similarity vector. At Lines 10–13, it infers the parameters by utilizing the EM-algorithm. Finally, from Lines 14 to 17, it judges whether a tuple belongs to the matched or unmatched one based on the computed matching scores at Line 15.

Algorithm 1. EMAN: Entity matching algorithm

Input: N data sources E_i;
Output: Matched Entities(PM);
1: $T \leftarrow \emptyset$; $j \leftarrow 0$;
//step 1: Candidate tuple generation
2: **for** $i = 1, \cdots, N$ **do**
3: employ LSH to block entities from the $i-$th source;
4: **end for**
//step 2: Entity vectorization and similarity computation
5: **for** each bucket in the LSH **do**
6: generate $t_i = (\alpha_1(e_{1i_1}), \alpha_2(e_{2i_2}), \cdots, \alpha_N(e_{Ni_N}))$;
7: compute the minimum similarity vector γ_i between entity pairs of t_i;
8: $T \leftarrow T \cup$ candidate tuples from the bucket;
9: **end for**
//step 3: Parameter Learning
10: **while** parameter set Θ has not converged **do**
11: E-Step; //handle missing data;
12: M-Step; //estimate parameters by maximizing the log-likelihood;
13: **end while**
//step 4: Tuple scoring and label assignment
14: **for** $t_i \in T$ **do**
15: $w_i \leftarrow \log \frac{P(t_i \in M|\gamma_i, \hat{\Theta})}{P(t_i \in U|\gamma_i, \hat{\Theta})}$;
16: according to the score of t_i, keep PM to the top-K candidate entities with the largest scores;
17: **end for**
18: **return** PM

4 Parameters Inference and Prediction

We employ a generative model to solve the problem defined in previous section. Given the similarity vectors of all candidate tuples, EMAN learns the parameters of similarity distributions for *matched* and *unmatched* tuples based on *exponential family distributions*. In terms of these learned parameters of similarity distributions, EMAN infers whether candidate tuple $t_i \in T$ is *matched* or *unmatched* by estimating probabilities $P(t_i \in M|\gamma_i, \Theta)$ and $P(t_i \in U|\gamma_i, \Theta)$.

4.1 Likelihood

Assume that $P(t_i \in M|\Theta) = p$, i.e., $P(t_i \in U|\Theta) = 1 - p$. Employing Bayes' rule to $P(t_i \in M|\gamma_i, \Theta)$, we can obtain:

$$P(t_i \in M|\gamma_i, \Theta) = \frac{p \times P(\gamma_i|t_i \in M, \Theta)}{P(\gamma_i|\Theta)} \tag{3}$$

$$P(\gamma_i|\Theta) = p \times P(\gamma_i|t_i \in M, \Theta) + (1 - p) \times P(\gamma_i|t_i \in U, \Theta) \tag{4}$$

Please note that we do not know the label of a candidate tuple t_i. To represent the joint probability of the observed data, we define a latent variable l_i for

candidate tuple t_i. Its value is 1 if tuple t_i is a matched tuple, and otherwise 0. And, we defined $c_i = (l_i, \gamma_i)$ as the *complete data* vector for T. The probability of observation c_j under parameter Θ can be defined as:

$$\begin{aligned} P(c_i|\Theta) &= [P(\gamma_i, t_i \in M|\Theta)]^{l_i}[P(\gamma_i, t_i \in U|\Theta)]^{(1-l_i)} \\ &= [p \times P(\gamma_i|t_i \in M, \Theta)]^{l_i}[(1-p) \times P(\gamma_i|t_i \in U, \Theta)]^{(1-l_i)} \end{aligned} \tag{5}$$

Let $L_i = (l_i, 1 - l_i)$ and thus we obtain the *log-likelihood* for sample $X = \{c_i : i = 1, 2, \cdots, |T|\}$ as:

$$\begin{aligned} \boldsymbol{L}(\Theta|X) = &\sum_{i=1}^{|T|} L_i[logP(\gamma_i|t_i \in M, \Theta), logP(\gamma_i|t_i \in U, \Theta)]^{'} \\ &+ \sum_{i=1}^{|T|} L_i[logp, log(1-p)]^{'}. \end{aligned} \tag{6}$$

4.2 Exponential Family

As mentioned above, most of the probability distributions can be represented by exponential family which is a convenient and widely used family of distributions. Distributions in the exponential family appeal to the machine learning community as some good properties of MLE which is a function of the sufficient statistic and the best unbiased estimator, etc. [22].

In probability and statistics, an *exponential family* is a set of probability distributions and represented by an exponential form which is chosen for mathematical convenience [23]. In other word, an *exponential family* is a set of probability distributions whose PDF and PMF can be expressed in the form as follows:

$$f(x;\theta) = h(x) \exp\left(\theta^{'} S(x) - z(\theta)\right) \tag{7}$$

where θ (may be a vector) is the natural parameter of a distribution. $S(x)$ is a sufficient statistic. Generally, $S(x) = x$. So when the parameter z, h, S are fixed, we will define an exponential family with parameter θ. The exponential family contains as special cases most of the standard discrete and continuous distributions that we use for practical modelling, such as Bernoulli, Multinomial, Poisson, Gamma, Dirichlet, etc.

One of our task is to calculate probabilities $P(t_i \in M|\gamma_i, \Theta)$ and $P(t_i \in U|\gamma_i, \Theta)$. In Eq. 6, we know that the critical step is to calculate $P(\gamma_i|t_i \in M, \Theta)$ and $P(\gamma_i|t_i \in U, \Theta)$. So we estimate $P(\gamma_i|t_i \in M, \Theta)$ and $P(\gamma_i|t_i \in U, \Theta)$ and assume that γ_i is drawn from a distribution of exponential family, and use the simplifying assumption that the entries of vector γ_i are conditional independent with respect to the state of indicator L_i such as:

$$\begin{aligned} P(\gamma_i^j|t_i \in M, \Theta) &\sim f_{1,j}(\gamma_i^j; \theta_{1,j}),\ for\ j = 1, \cdots, m \\ P(\gamma_i^j|t_i \in U, \Theta) &\sim f_{0,j}(\gamma_i^j; \theta_{0,j}),\ for\ j = 1, \cdots, m \end{aligned} \tag{8}$$

where $f_{.,j}(\cdot;\cdot)$ is the PDF or PMF from the exponential family in Eq. 7.

Next, the log-likelihood in Eq. 6 can be replaced with:

$$\begin{aligned}\boldsymbol{L}(\Theta|X) \propto & \sum_{i=1}^{|T|} L^i [\sum_{j=1}^{m} \theta'_{1,j} S_{1,j}(\gamma_i^j), \sum_{j=1}^{m} \theta'_{0,j} S_{0,j}(\gamma_i^j)]' \\ & - \sum_{i=1}^{|T|} L^i [\sum_{j=1}^{m} z_{1,j}(\theta_{1,j}), \sum_{j=1}^{m} z_{0,j}(\theta_{0,j})]' + \sum_{i=1}^{|T|} L^i [\log p, \log(1-p)]'.\end{aligned} \tag{9}$$

4.3 Maximum Likelihood Estimator

Since L^i is a latent vector in Eq. 9, so we estimate the parameter $\Theta = \{p, \theta_{1,j}, \theta_{0,j}, for\, j = 1, \cdots, m\}$ with maximum likelihood estimation using EM algorithm. The EM algorithm begins with initial estimator of unknown parameter Θ and repeat iterative calculation of the expectation(E) and maximization(M) steps until the convergence.

E-step. The objective in E-step is to calculate the conditional expectation of latent variables and estimate the missing data with observed data. Given γ_i and $\Theta^{(k-1)}$ in the *k-th* iteration, the conditional distribution of l_i is $l_i|\gamma_i, \Theta^{(k-1)} \sim B(1, p_i^{(k)})$ with

$$p_i^{(k)} = P(l_i = 1|\gamma_i, \Theta^{(k-1)}) \tag{10}$$

Then $p_i^{(k)}$ will be represented with Eq. 11 as:

$$\begin{aligned} p_i^{(k)} &= P(l_i = 1|\gamma_i, \Theta^{(k-1)}) = \frac{P(t_i \in M, \gamma_i|\Theta^{(k-1)})}{P(\gamma_i|\Theta^{(k-1)})} \\ &= \frac{p^{(k-1)} \cdot \prod_{j=1}^{m} f_{1,j}(\cdot;\cdot)}{p^{(k-1)} \cdot \prod_{j=1}^{m} f_{1,j}(\cdot;\cdot) + (1-p^{(k-1)}) \cdot \prod_{j=1}^{m} f_{0,j}(\cdot;\cdot)} \end{aligned} \tag{11}$$

By substituting $p_i^{(k)}$ for l_i, we obtain the expectation function.

M-step. In M-step, we maximize the likelihood after E-step. When we estimate the values of $l_i^{(k)} = p_i^{(k)}$ in E-step, we take derivatives of the log-likelihood to parameters $p, \theta_{1,j}, and\, \theta_{0,j}$ as follows:

$$\frac{\partial \boldsymbol{L}(\Theta|X)}{\partial p} = \sum_{i=1}^{|T|} \left(\frac{l_i^{(k)}}{p} - \frac{1 - l_i^{(k)}}{1-p}\right) \tag{12}$$

$$\frac{\partial \boldsymbol{L}(\Theta|X)}{\partial \theta_{1,j}} = \sum_{i=1}^{|T|} l_i^{(k)} \left(S_{1,j}(\gamma_i^j) - \frac{\partial z_{1,j}(\theta_{1,j})}{\partial \theta_{1,j}}\right) \tag{13}$$

$$\frac{\partial \boldsymbol{L}(\Theta|X)}{\partial \theta_{0,j}} = \sum_{i=1}^{|T|} (1 - l_i^{(k)}) \left(S_{0,i}(\gamma_i^j) - \frac{\partial z_{0,j}(\theta_{0,j})}{\partial \theta_{0,j}}\right) \tag{14}$$

Table 1. The MLEs of parameters for both matched and unmatched groups

Distribution	MLE	
	Matched group	Unmatched group
Bernoulli	$p_{1,i}^{(k)} = \frac{\sum_{j=1}^{\|T\|} l_j^{(k)} \gamma_i^j}{\sum_{j=1}^{\|T\|} l_j^{(k)}}$	$p_{0,i}^{(k)} = \frac{\sum_{j=1}^{\|T\|} (1-l_j^{(k)}) \gamma_i^j}{\sum_{j=1}^{\|T\|} (1-l_j^{(k)})}$
Multinomial	$p_{1,i}^{(k)} = \frac{\sum_{j=1}^{\|T\|} l_j^{(k)} I_{\gamma_i^j = h}}{\sum_{j=1}^{\|T\|} l_j^{(k)}}$	$p_{0,i}^{(k)} = \frac{\sum_{j=1}^{\|T\|} (1-l_j^{(k)}) I_{\gamma_i^j = h}}{\sum_{j=1}^{\|T\|} (1-l_j^{(k)})}$
Gaussian	$\mu_{1,i}^{(k)} = \frac{\sum_{j=1}^{\|T\|} l_j^{(k)} \gamma^j}{\sum_{j=1}^{\|T\|} l_j^{(k)}}$	$\mu_{0,i}^{(k)} = \frac{\sum_{j=1}^{\|T\|} (1-l_j^{(k)}) \gamma^j}{\sum_{j=1}^{\|T\|} (1-l_j^{(k)})}$
	$(\sigma_{1,i}^{(k)})^2 = \frac{\sum_{j=1}^{\|T\|} l_j^{(k)} (\gamma^j - \mu_{1,i}^{(k)})^2}{\sum_{j=1}^{\|T\|} l_j^{(k)}}$	$(\sigma_{0,i}^{(k)})^2 = \frac{\sum_{j=1}^{\|T\|} (1-l_j^{(k)}) (\gamma^j - \mu_{1,i}^{(k)})^2}{\sum_{j=1}^{\|T\|} (1-l_j^{(k)})}$
Exponential	$\lambda_{1,i}^{(k)} = \frac{\sum_{j=1}^{\|T\|} l_j^{(k)}}{\sum_{j=1}^{\|T\|} l_j^{(k)} \gamma_i^j}$	$\lambda_{0,i}^{(k)} = \frac{\sum_{j=1}^{\|T\|} (1-l_j^{(k)})}{\sum_{j=1}^{\|T\|} (1-l_j^{(k)}) \gamma_i^j}$

By calculating, we can infer $\frac{\partial z_{\cdot,j}(\theta_{\cdot,j})}{\partial \theta_{\cdot,j}} = E_{\theta_{\cdot,j}}(S_{\cdot,j}(\gamma^j))^2$ where $\cdot$ can be 1 or 0 and obtain the MLEs of parameters as shown in Table 1. Due to the page limitation, we omit the proofs and computations.

While the distributions of all similarity values were assigned, we can estimate the parameter in the M-step of the *k-th* iteration. The probability p can be estimated as $p^{(k)} = \frac{\sum_{i=1}^{\|T\|} l_i^{(k)}}{\|T\|}$.

4.4 Missing Data

As a result of presence of missing data, we denote the sample X as (X_o, X_m), where X_o represents the observed data and X_m represents the missing data. Let $\Theta^{(0)}$ be the initial value for parameter. The E-step of EM algorithm computes $Q(\Theta; \Theta^{(k-1)}) = E(\boldsymbol{L}(\Theta|X)|X_o, \Theta^{(k-1)})$ during *k-iteration*. Due to the missing data, $S_{1,i}(\gamma_i^j)$ and $S_{0,i}(\gamma_i^j)$ in Eq. 9 are missing. So $Q(\Theta; \Theta^{(k-1)})$ can be calculated by $E(S_{1,j}(\gamma_i^j)|\Theta^{(k-1)})$ and $E(S_{0,j}(\gamma_i^j)|\Theta^{(k-1)})$ for $S_{1,j}(\gamma_i^j)$ and $S_{0,j}(\gamma_i^j)$ in Eqs. 13 and 14 respectively.

4.5 Matching Score Computation

Once parameters Θ are estimated, EMAN determines whether candidate tuple t_i belongs to matched or unmatched one by computing its matching score. To speed up the computation of matching scores for exponential family, we define the match score function as:

$$W_i = log(\frac{P(t_i \in M|\gamma_i, \hat{\Theta})}{P(t_i \in U|\gamma_i, \hat{\Theta})}) \propto \sum_{j=1}^{m} w_i^j \tag{15}$$

where

$$w_i^j = (\Theta'_{1,j} S_{1,j}(\gamma_i^j) - z_{1,j}(\Theta_{1,j})) - (\Theta'_{0,j} S_{0,j}(\gamma_i^j) - z_{0,j}(\Theta_{0,j})) \tag{16}$$

Table 2. Descriptive statistics of datasets

Data source	# entities	# features
NetEaseHouse[a]	2776	36
AnJuKe[b]	581	34
PingAnFang[c]	630	15

[a]http://house.163.com/.
[b]http://shanghai.anjuke.com/.
[c]http://www.pinganfang.com/.

where $P(t_i \in M|\gamma_i, \hat{\Theta}) > P(t_i \in U|\gamma_i, \hat{\Theta})$ when $W_i > 0$. Alternatively, we can assign t_i to the matched tuple set if $W_i > W_0$ where $W_0 > 0$ is a threshold.

5 Empirical Evaluation

We conduct two experiments to compare the proposed EMAN with the baseline method using three real data sources. First, we manifest the performance of the self-matching problem in which there is a clearly ground truth one-one matching between entities from the identical data source. Secondly, we study the performance of matching three real heterogeneous data sources.

5.1 Experimental Setup

Datasets. We crawled three datasets from the famous estate websites in China for our experiments. The descriptive statistics about the datasets are shown in Table 2. However, the schemes of three data sources are different from each other. We only find 10 useful attributes as shown in the first column of Table 3. In our experiment, we model the similarities of property fees and price with Gaussian distribution, purpose(shop or dwelling) with Bernoulli distribution, and remaining similarities with Exponential distribution as shown in the last column of Table 3.

Comparative Method and Evaluation Measures. We find an unsupervised approach, called Felliegi-Sunter (shorted in **FS**), to be the comparative baseline. Fellegi-Sunter's approach therefore evaluates all attributes by using binary similarity, i.e., the similarity is 1 if two attributes are the same, and otherwise 0. Currently, many data sources are low in quality. The FS approach is too simple to obtain reasonable performance. In our implementation for FS, we therefore set the similarity value to be 1 if the value of similarity of attributes is larger than a tuned threshold, and otherwise 0.

As the mentioned above, we evaluate our method using *Precision*@K, and *Recall*@K. *Precision*@K is the fraction of the matched tuples in the *top-K* result that are correctly matched. *Recall*@K is the fraction of ground truth matched entities that appear among the *top-K* results. To evaluate the scalability of our proposed approach, we also measure the elapsed time in second and the number of candidate tuples.

Table 3. The setup of experiment and parameter estimation for matched and unmatched groups

Feature	Matched	Unmatched	Similarity	Distribution
name	$\lambda_m = 8.36$	$\lambda_u = 33423.12$	LCS	Exponential
address	$\lambda_m = 5.27$	$\lambda_u = 31127.66$	Jaccard	Exponential
developer	$\lambda_m = 23.82$	$\lambda_u = 27.58$	Jaccard	Exponential
construction time	$\lambda_m = 3.92$	$\lambda_u = 4.24$	Jaccard	Exponential
PMC	$\lambda_m = 2.31$	$\lambda_u = 2.38$	Jaccard	Exponential
property fees	$\mu_m = 0.43$	$\mu_u = 0.42$	Euclidean distance	Gaussian
	$\sigma_m = 0.05$	$\sigma_u = 0.04$		
building type	$\lambda_m = 2.76$	$\lambda_u = 2.87$	Jaccard	Exponential
launch date	$\lambda_m = 2.11$	$\lambda_u = 2.20$	Jaccard	Exponential
price	$\mu_m = 0.45$	$\mu_u = 0.43$	Euclidean distance	Gaussian
	$\sigma_m = 0.02$	$\sigma_u = 0.02$		
purpose	$Prob_m = 0.23$	$Prob_u = 0.12$	1,0 for matched or not	Bernoulli

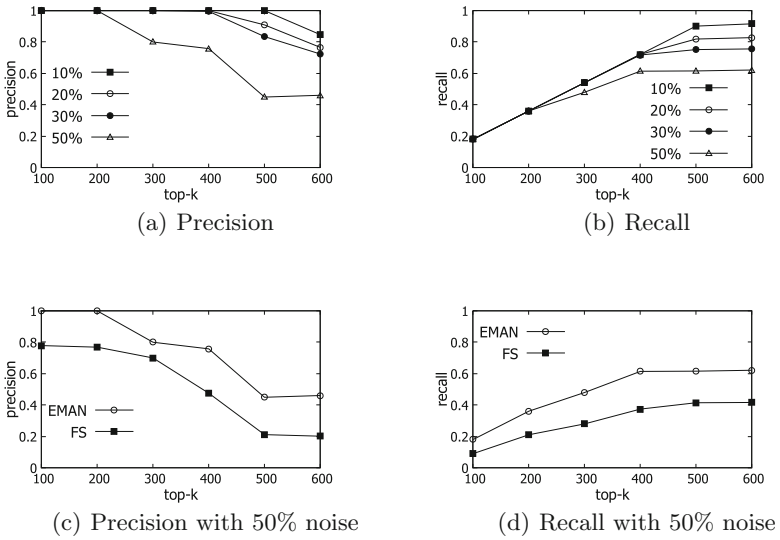
5.2 Self-matching Evaluation

Firstly, we perform our method in self-matching task which is designed such that we know the complete ground truth matched entities, i.e., we match the entities from three replicas of the identical data source. We create two new data sources which are injected some noise into the given data source. For each replica, we randomly inject some noises into string attributes. For each character, it has the same probability to be inserted, deleted or replaced. Take $PingAnFang$ as an example, $PingAnFang_1(\psi)$ is the first replica of $PingAnFang$, where ψ denotes the probability of each character being changed. In our experiment, the value of ψ varies from 10 % to 50 %.

Figure 1(a) and (b) manifest the accuracy of EMAN on $PingAnFang$ by varying ψ from 10 to 50 %. We observe that the accuracy of EMAN is promised. If 10 % noise is injected into the data, almost 90 % matched entities can be found by EMAN. Even 50 % noise is injected into the data, the precision in the top-200 is almost 100 %. Figure 1(c) and (d) illustrate that EMAN outperforms the baseline significantly for self-matching on $PingAnFang$. The result indicates that exponential family is helpful to integrate heterogeneous entity attributes.

5.3 Matching Heterogeneous Data Sources

Scalability of EMAN. In this experiment, we address whether LSH is helpful to speed up EMAN. For entity matching on heterogeneous data sources, we only change the size of $NetEaseHouse$ from 500 to 2,500. In Fig. 2(a) ($EMAN\_L$ is an approach that $EMAN$ employs LSH to block entities), we can find that only less than 1 % candidate tuples are remained after using LSH to block entities. In Fig. 2(b), EMAN associated with LSH detects entities within 2,000 s when the size of data source is almost 2,000. However, the elapsed time of EMAN without



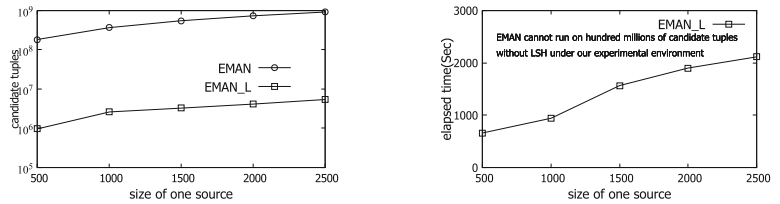
(a) Precision (b) Recall

(c) Precision with 50% noise (d) Recall with 50% noise

Fig. 1. Accuracy of EMAN and FS

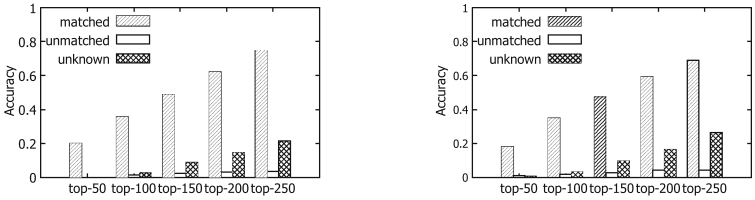
LSH is more than 12 h. In summary, LSH is helpful to reduce the number of candidate tuples and speed up the computation of EMAN.

Manually Judgement. We now turn to match three heterogeneous data sources, namely complete $NetEaseHouse$, $AnJuKe$ and $PingAnFang$. Since the maximum size of matched tuples is less than the minimal number of $|NetEaseHouse|$, $|AnJuKe|$ and $|PingAnFang|$, we manually annotated the top-250 matched entities labelled by EMAN and FS. Three entities are judged to be matched tuple when (1) the similar entity name; (2) the similar values in some attributes, such as *address*, *region* and *developer*. The remaining entity triples are assigned the undetermined label. As shown in Fig. 3(a) and (b), we find that the accuracy for the top-250 result of EMAN is more than 70 %, but it is about 60 % for FS. This illustrates that both EMAN and FS are quite good in returning the correctly matched entities for different top-K ranked tuples. EMAN also returns fewer undetermined tuples than FS.



(a) The number of candidate tuples (b) Elapsed time(Sec.)

Fig. 2. Efficiency of EMAN



(a) Accuracy for EMAN (b) Accuracy for FS

Fig. 3. Accuracy for entity matching

For this experiment, we list all parameters learned from our approach. An attribute is more important to match entities if the difference of parameters between matched and unmatched groups is larger. As shown in Table 3, we can also observe that *name* and *address* are two most important attributes to match entities for this task.

6 Conclusion

In this paper, we have studied the problem of entity matching across two or more heterogeneous data sources. It is a challenging task due to the overwhelming expensive, heterogeneous attributes for each entity, and incomplete and missing data. We propose an unsupervised method to deal with the mentioned challenges. We have illustrated our proposed method on three real data sources. Experimental results indicate that EMAN not only outperforms the comparable baseline but also obtains the promising performance.

In our future work, we plan to extend our work to handle some ground-truth tuples with semi-supervised approach, and deploy a distributed algorithm to support more efficient computation.

Acknowledgements. This work is supported by the National Basic Research Program (973) of China (No. 2012CB316203) and NSFC under Grant No. U1401256, 61402177, 61402180 and 61232002. This work is also supported by CCF-Tecent Research Program of China (No. AGR20150114), NSF of Shanghai (No. 14ZR1412600), and a fund of ECNU for oversea scholars, international conference and domestic scholarly visits.

References

1. Newcombe, H.B., Kennedy, J.M., Axford, S.J., James, A.P.: Automatic linkage of vital records. Science **130**(3381), 954–959 (1959)
2. Scannapieco, M., Figotin, I., Bertino, E., Elmagarmid, A.K.: Privacy preserving schema and data matching. In: SIGMOD, pp. 653–664 (2007)
3. Sarawagi, S., Bhamidipaty, A.: Interactive deduplication using active learning. In: ACM SIGKDD, pp. 269–278 (2002)

4. Wang, Y.R., Madnick, S.E.: The inter-database instance identification problem in integrating autonomous systems. In: Data Eng, pp. 46–55. IEEE (1989)
5. Hernandez, M.A., Stolfo, S.J.: The merge/purge problem for large databases. In: SIGMOD, pp. 127–138 (1995)
6. Jin, L., Li, C., Mehrotra, S.: Supporting efficient record linkage for large data sets using mapping techniques. World Wide Web **9**(4), 557–584 (2006)
7. Whang, S.E., Garcia-Molina, H.: Incremental entity resolution on rules and data. VLDB J. **23**(1), 77–102 (2014)
8. Kolb, L., Thor, A., Rahm, E.: Block-based load balancing for entity resolution with MapReduce. In: CIKM, pp. 2397–2400 (2011)
9. Whang, S., Garcia-Molina, H.: Entity resolution with evolving rules. PVLDB **3**(1), 1326–1337 (2010)
10. Getoor, L., Machanavajjhala, A.: Entity resolution: theory practice & open challenges. PVLDB **5**(12), 2018–2019 (2012)
11. Singla, P., Domingos, P.: Entity resolution with markov logic. In: ICDM, pp. 572–582 (2006)
12. Tejada, S., Knoblock, C.A., Minton, S.: Learning object identification rules for information integration. Inf. Syst. **26**(8), 607–633 (2001)
13. Christen, P.: Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer, Heidelberg (2012)
14. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: a survey. IEEE Trans. Knowl. Data Eng. **19**(1), 1–16 (2007)
15. Winkler, W.E.: Overview of Record Linkage and Current Research Directions. U.S. Census Brueau, Washington (2006)
16. Fellegi, I.P.: A theory for record linkage. J. Am. Stat. Assoc. **64**(328), 1183–1210 (1969)
17. Wang, J., Li, G., Yu, J.X., Feng, J.: Entity matching: how similar is similar. PVLDB **4**(10), 622–633 (2011)
18. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learning string similarity measures. In: ACM SIGKDD, pp. 39–48 (2003)
19. Dong, X., Halevy, A., Madhavan, J.: Reference reconciliation in complex information spaces. In: ACM SIGMOD International Conference on Management of Data, pp. 85–96 (2005)
20. Roos, L.L., Wajda, A.: Record linkage strategies. part I: estimating information and evaluating approaches. Methods Inf. Med. **30**(2), 117–123 (1991)
21. Grannis, S.J, Overhage, J,M, McDonald, C.J: Analysis of identifier performance using a deterministic linkage algorithm. In: AMIA (2002)
22. Rastogi, V., Dalvi, N.N., Garofalakis, M.N.: Large-scale collective entity matching. PVLDB **4**(4), 208–218 (2011)
23. Lee, S., Lee, J., Hwang, S.-W.: Scalable entity matching computation with materialization. In: CIKM, pp. 2353–2356 (2011)
24. DuVall, S.L., Kerber, R.A., Thomas, A.: Extending the Fellegi-Sunter probabilistic record linkage method for approximate field comparators. J. Biomed. Inform. **43**(1), 24–30 (2010)
25. Sadinle, M., Fienberg, S.E.: A generalized Fellegi-Sunter framework for multiple record linkage with application to homicide record systems. J. Am. Stat. Assoc. **108**(502), 385–397 (2013)
26. Gao, M., Lim, E.-P., Lo, D., Zhu, F., Prasetyo, P.K., Zhou, A.: C.N.L.: Collective network linkage across heterogeneous social network. In: ICDM (2015)
27. Christen, P.: A survey of indexing techniques for scalable record linkage and deduplication. IEEE TKDE **24**(9), 1537–1555 (2011)