Probabilistic Estimates of Attribute Statistics and Match Likelihood for People Entity Resolution

Xin Wang, Ang Sun, Hakan Kardes, Siddharth Agrawal, Lin Chen, Andrew Borthwick
Data Research
Intelius Inc
Bellevue, WA
Email: {xwang, asun, hkardes, sagrawal, lchen, aborthwitck}@intelius.com

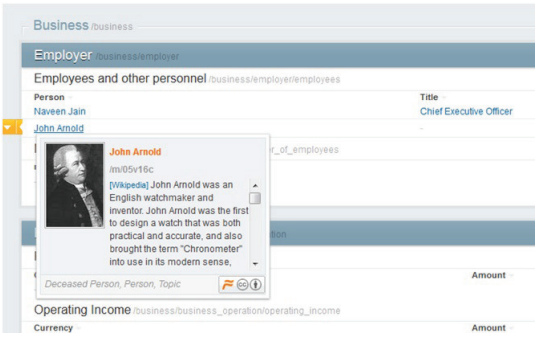

Figure 1. Freebase links John Arnold, an ex-employee at inome, to an 18th century English watch maker.

***Abstract*—For big data practitioners, data integration/entity resolution/record linkage is one of the key challenges we face from day to day. Entity resolution/record linkage with high precision and recall on a large graph with billions of nodes, and hundreds of times more edges poses significant scalability challenges. Similarity based graph partition is still the most scalable method avaiable. This paper presents a probabilistic method to approximate the match likelihood of a pair of records by incorporating values of different attributes and their aggregates/statistics. The quality of the approximates depend on the accuracy of the estimates of the aggregated values. The paper adapts the GTM model described in [1] to obtain the estimates. We present experimental results based on real world commercial data sources to show that the estimates obtained via GTM model is better than the baseline. Our experimental results also showed that the appoximate match likelihood can improve the recall of the similarity function.**

***Keywords*-Big Data Demographic Information; Approximate Probabilistic Estimates; Record Linkage; Data Integration; Entity Resolution; Data Fusion**

I. Introduction

For practitioners in big data [2], one of the challenges we face is to identify information from multiple data sources about the same entity. There are many research efforts in the area from various communities: *record linkage* [3]–[5], *duplicate detection* [6], [7], *reference reconciliation* [8], *entity resolution* [9]–[11], *entity linking* [12], or *merge-purge* [13].

For example, in Freebase (see Figure 1), for the entry, *inome Inc*[1], a company, *John Arnold* is listed as an employee[2], but the link points to an 18th century English watchmaker and inventor. *John Arnold* is a common name. One commercial data source (credit headers) claims that there are 3648 people in US with the name. Linkedin search returns 882 results. After refining the query with the location of *inome Inc*, *the Greater Seattle Area*, Linkedin still returns 10 results. It is not an easy task to link the name *John Arnold* on the *Freebase* page of *inome Inc* to the correct entity in a knowledge base.

Freebase has around 3 million people topics, most of which are on famous people or their relatives. At our company, we need to link 7 billion records from various sources (phone records, property records, criminal records, email registeries and etc) to create a single profile for every individual in US with high precision. False positives, which often result in bogus profiles with multiple people, are very detrimental to our business. Certain type of false positives (*e.g*, errors with criminal records) can even lead to law suits.

One way to formulate the problem is to treat the 7 billion records as nodes in a graph and partition them into roughly 313.9 million[3] clusters. Most graph partitioning algorithms can't be stretched to such a scale in an efficient manner with the limited resources we have (an 88 node hadoop cluster for multiple monthly builds). So we adopt an approximate solution that dynamically divides the graph into subgraphs based on various heuristics (*name*, *phone number*, *location* etc.), and then cluster each individual subgraphs into people profiles [14], [15].

For example, instead of considering every possible edge in the 7 billion node graph, we can focus on clustering subgraphs of records with the same or similar names. Efficient clustering algorthms often presume that the number of clusters is known. But how can we know how many people in *the Greater Seattle Area* have the name *John Arnold* before we actually do the clustering?

[1]http://www.freebase.com/m/02qkz7s.
[2]a co-founder who left the company in 2010.
[3]The population of US according to 2012 Census data.

Many graph partitioning algorithms use a pair-wise similarity function between pairs of records. But if we look only at pairs of records, many of them are ambiguous by the nature. For example, for the relatively common name *Patricia Johnson*, if one record has only an address in a big city and the other has job information in the same city, it is not likely the two are about the same person (Figure 2). If the two records are about a small town *Index, WA* with a population of 184 (Figure 3), or they have a common date of birth (DOB, Figure 5), then it is more likely for the two to be about the same person. Even without an exact DOB match, multiple shared locations and age similiarity between two records still hint at a higher likelihood that they match (Figure 4).

So if we have a way to estimate the likelihood of two records being about the same entity, it can help us to build a pair-wise similarity function with higher precision and recall. In Section II, we describe an approach to estimate the likelihood of two records referring to the same entity. To approximate the likelihood, the model needs accurate estimates of name frequencies, population of geological regions of different sizes, number of people sharing a phone number, an address and etc.

Estimating the name frequencies and statistics is similar to the truth finding problem [16]–[22] in the field of data integration [2], [23]. Most of these algorithms focus on reconciling categorical values. The model in [1] (for estimating real-valued truth from conflicting sources) is the most suitable for our application, so we adapt it to estimate the expected number of clusters in each subgraph for our graph partitioning problem and the demographic statistics for the similarity function.

In the following sections, we describe how to approximate the Match Likelihood in a probabilistic manner in Section II. We describe the Gaussian Truth Model from [1], and how to estimate the aggregated attribute statistics in the Match Likelihood computation in III. In Section IV, we present the experimental results. In Section V, we discuss the limitations of the approach and directions for future work. Finally, we conclude the paper in Section VI.

II. Approximate the Match Likelihood

In this section, we describe an approach to approximate the likelihood of two records referring to the same entity, the *Match Likelihood*, as we call it. We will use people entities to showcase it, but the approach is general and can be applied to other types of entities, *e.g.*, organization, location, and etc.

A. A Simple Case

First, let us look at the example in Figure 2, the two records have a common name μ (name match) and share the same location l (location match). Assume that for any given name μ, the distribution of the name is uniform across the

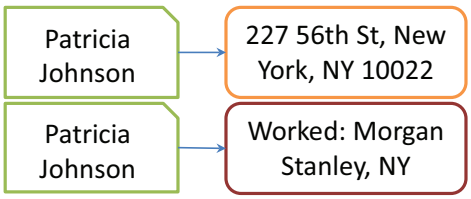

Figure 2. Low probability for two records with a common name in a big city to be about the same person

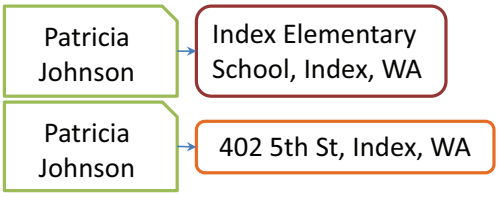

Figure 3. Two records with a common name in a small town are more likely to be about the same person

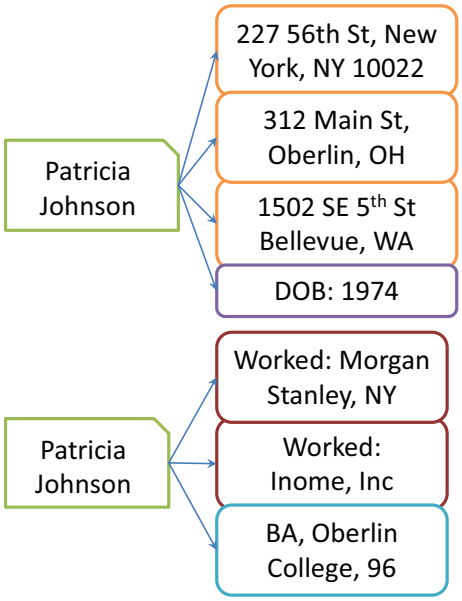

Figure 4. Combining evidence from multiple shared locations increases the likelihood that two records are about the same person

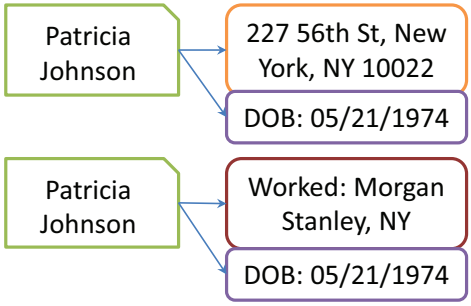

Figure 5. Two records with a common name in a big city are more likely to be about the same person if we have a DOB match.

country. That is, it is equally likely for any of the m people with the name μ to live in any city or location in US, so that the probability of a person living in anywhere in US is independent of his or her name.[4]

Assume the population of the location l is $p_r(l)$, then the probability of a person living in the location l, is $\frac{p_r(l)}{p_r(US)}$. The probability of no other people with the name μ living at location l can be computed as,

$$\left(1 - \frac{p_r(l)}{p_r(US)}\right)^{m-1} \tag{1}$$

The *Match Likelihood* as given by Eq (1) is much higher for the two records in Figure 3 than in Figure 2.

In Eq (1), we consider only *name* and *location* information, and assume that names are uniformly distributed over the country. It is a super simplification of the real world. The next step, let us look at *date of birth* (DOB) information as well (as in Figure 5).

Most of the DOB information in our data are either noisy or incomplete due to clerical errors or privacy issues. Incomplete DOBs, as long as they are compatible with each other, can provide valuable information for the computation of the *Match Likelihood*.

Let $\vec{b}_x = \{b_0, b_1, \ldots, b_{n_{b_x}}\}$, and $\vec{b}_y = \{b_0, b_1, \ldots, b_{n_{b_y}}\}$ be the set of DOBs for profiles x and y, respectively. The birthday difference $\Delta_b(x, y)$ can be computed:

$$\Delta_b(x, y) = \min_{i,j}\{\mathcal{D}_B(b_i, b_j) \mid b_i \in \vec{b}_x, b_j \in \vec{b}_y\}$$

Let R_b be the range of all possible DOBs. There are m people with the name μ, and the probability of a person with a birthday in the range Δ_b is $\frac{\Delta_b}{R_b}$. So the expected number of people with the name μ and DOBs in the range Δ_b can be computed as the mean of the Poisson process: $m\frac{\Delta_b}{R_b}$

With DOBs, the *Match Likelihood* can be approximated as:

$$P_{nbp} = \left(1 - \min_l \frac{p_r(l)}{p_r(US)}\right)^{m\frac{\Delta_b}{R_b}-1} \tag{2}$$

The Likelihood estimation in Eq (2) works reasonably well in practice for our record linkage task (see Section IV).

Eq (2) can be extended to handle multiple *name matches*, $e.g.$ nickname and formal name matches or maiden name and married name matches, by replacing m with the mininum frequency of all matched names.

[4]The assumption is not always true, especially for immigrant countries like US, where ethnics groups tend to concentrate in certain geographic regions. In addition, in western cultures, it is a common practice to name a child after a parent, a grand parent or other relatives, so when there is a person with a name μ, epsecially if it is a male name, it is considerably more likely for another person with exactly the same name living in the same household (up to 10% as in our preliminary research).

B. Name Frequency not Uniformly Distributed

First, let us relax the uniform distribution assumption of name frequencies, and assume that the frequency of a name can vary from region to region, but inside a region, the distribution is still uniform.

Let the first, middle, and last of the name μ be μ_f, μ_m, μ_l. Under the new assumption, the name frequency function $\phi_n(\mu, r)$ is a function of μ and r. $\phi_n(\mu, r) = \phi_n(\mu_f, \mu_m, \mu_l)$.

Second, let us consider the case of multiple *location matches*. To simplify the formulation, we compute the likelihood for each *location match* and take the maximum. Let $\Phi_n(r)$ be the minimum frequency of all name matches that associated with a location match.

$$P_{rnbp} = \max_r P(r) \tag{3}$$

where

$$P(r) = \begin{cases} \left(1 - \frac{\min_{l\in r}\, p_r(l)}{p_r(r)}\right)^{\Phi_n(r)\frac{\Delta_b}{R_b}-1}, & \frac{\min_{l\in r}\, p_r(l)}{p_r(r)} < 1 \\ \left(1 - \frac{\Delta_b}{R_b}\right)^{\Phi_n(r)-1}, & \frac{\min_{l\in r}\, p_r(l)}{p_r(r)} = 1 \end{cases} \tag{4}$$

C. Combine Multiple Location Matches

In Figure 4, we have three location matches, each in a different metropolitan area. It makes the records a lot more likely to be about the same person. So instead of choosing the most likely match among all three as in Eq (3), combine multiple region matches into a more comprehensive likelihood.

On the other hand, people often move from one neighborhood to another, or one town to the next in the same region. So intuitively, multiple location matches in the same region should not be combined, and we should use the match with the smallest common population.

With the $e^x = 1 + x$ while $x \ll 1$, Equation (4) can be approximated by the following

$$\phi(r) = \begin{cases} e^{-\frac{\min_{l\in r}\, p_r(l)}{p_r(r)}\left(\Phi_n(r)\frac{\Delta_b}{R_b}-1\right)}, & \frac{\min_{l\in r}\, p_r(l)}{p_r(r)} < 1 \\ e^{-\frac{\Delta_b}{R_b}(\Phi_n(r)-1)}, & \frac{\min_{l\in r}\, p_r(l)}{p_r(r)} = 1 \end{cases}$$

So $1-\phi(r)$ can be seen as the cumulative density function of as an exponential random variable with λ_r as on $\frac{\Delta_b}{R_b}$,

$$\lambda_r = \begin{cases} \frac{\min_{l\in r}\, p_r(l)}{p_r(r)}\Phi_n(r), & \frac{\min_{l\in r}\, p_r(l)}{p_r(r)} < 1 \\ \Phi_n(r) - 1, & \frac{\min_{l\in r}\, p_r(l)}{p_r(r)} = 1 \end{cases} \tag{5}$$

To combine multiple location matches, we use the convolution density function of these independent exponential random variables but with different scale parameters:

$$P(\sum_{i=1}^{k} \mathbf{r}_i <= s) = \sum_{i=1}^{k} \frac{\prod_{j=1, j\neq i}^{k} \lambda_j}{\prod_{j=1, j\neq i}^{k} (\lambda_j - \lambda_i)} e^{-\lambda_i s}$$

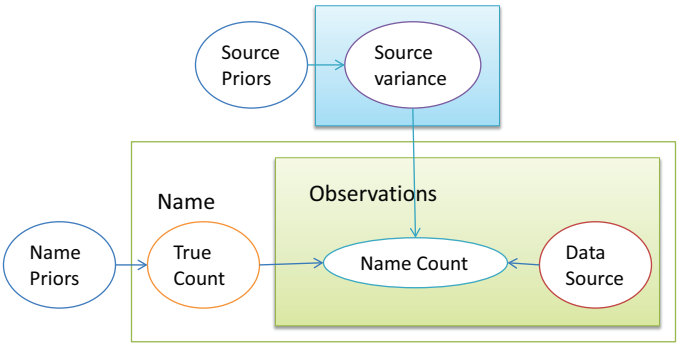

Figure 6. Gaussian Truth Model for Name Frequecy Estimation

We can come up with the probability:

$$P_{lrnbp} = \sum_{i=1}^{k} \frac{\prod_{j=1,j\neq i}^{k} \lambda_j}{\prod_{j=1,j\neq i}^{k} (\lambda_j - \lambda_i)} e^{-\lambda_i \frac{\Delta_b}{R_b}}$$

where the λs are defined as in Equation (5).

III. Estimate Attribute Statistics

The approximation to the *Match Likelihood* between two records presented in Section II depends heavily on global/region statistics/aggregates, *e.g.*, name frequency, regional population, or the population of a location. To accurately estimate these statistics/aggregates, we adapt the Gaussian Truth Model (GTM) presented in [1].

A. The Gaussian Truth Model

To estimate real-valued truth, GTM models the following three random variables and their dependencies:

1) **Quality of Sources**, for each $s \in \mathcal{S}$, its quality σ_s^2 is generated from an inverse Gamma distribution with hyper-parameter (α, β), where α is the shape parameter and β is the scale parameter:

$$\sigma_s \sim \left(\sigma_s^2\right)^{-\alpha-1} \exp\left(-\frac{\beta}{\sigma_s^2}\right)$$

2) **Truth of Entities**, for each entity $e \in \mathcal{E}$, the truth μ_e is generated from a Gaussian distribution with mean μ_0 and variance σ_0^2:

$$\mu_e \sim \exp\left(-\frac{(\mu_e - \mu_0)^2}{2\sigma_0^2}\right).$$

3) **Observations of Claims**, for each observation o_c for the entity e from data source s_c, o_c is generated by a Gaussian distribution with the truth μ_e as mean, the variance of the source s_c, $\sigma_{s_c}^2$ as variance:

$$o_c \sim \sigma_{s_c}^{-1} \exp\left(-\frac{(o_c - \mu_e)^2}{2\sigma_{s_c}^2}\right).$$

The complete likelihood of the observed data and unknown parameters given the hyper-parameters can be written as:

$$\begin{aligned} p(\mathbf{o}, \mu, \boldsymbol{\sigma^2} | \mu_0, \sigma_0^2, \alpha, \beta) = \\ \prod_{s\in\mathcal{S}} p(\sigma_s^2|\alpha,\beta) \\ \prod_{e\in\mathcal{E}} \left(p(\mu_e|\mu_0,\sigma_0^2) \prod_{c\in\mathcal{C}_1} p(o_c|\mu_e,\sigma_{s_c}^2) \right) \end{aligned}$$

Estimating the truth values is equivalent to get the *maximum a posterior* (MAP) estimates of μ.

$$\hat{\mu}_{MAP} = \arg\max_{\mu} \int p(\mathbf{o}, \mu, \boldsymbol{\sigma^2} | \mu_0, \sigma_0^2, \alpha, \beta) \mathrm{d}\sigma^2$$

The MAP estimates can be computed via an EM algorithm on the log form of the likelihood function:

$$\begin{aligned} \log p(\mathbf{o}, \mu, \boldsymbol{\sigma^2} | \mu_0, \sigma_0^2, \alpha, \beta) = \\ - \textstyle\sum_{s\in\mathcal{S}} \left(2(\alpha+1)\log\sigma_s^2 + \frac{\beta}{\sigma_s^2} \right) \\ - \textstyle\sum_{e\in\mathcal{E}} \frac{(\mu_e-\mu_0)^2}{2\sigma_0^2} \\ - \textstyle\sum_{e\in\mathcal{E}} \sum_{c\in\mathcal{C}_e} \left(\log\sigma_{s_c} + \frac{(o_c-\mu_e)^2}{2\sigma_{s_c}^2} \right) \end{aligned}$$

In the **E** Step, set $\frac{\partial L}{\partial \mu_e^2} = 0$, to get the maximum value for μ_e:

$$\hat{\mu}_e = \frac{\frac{\mu_0}{\sigma_0^2} + \sum_{c\in\mathcal{C}_e} \frac{o_c}{\sigma_{s_c}^2}}{\frac{1}{\sigma_0^2} + \sum_{c\in\mathcal{C}_e} \frac{1}{\sigma_{s_c}^2}}$$

In the **M** Step, set $\frac{\partial L}{\partial \sigma_s^2} = 0$, to get the maximum value for σ_s^2:

$$\hat{\sigma}_s^2 = \frac{2\beta + \sum_{c\in\mathcal{C}_s} (o_c - \mu_e)^2}{2(\alpha+1) + |\mathcal{C}_s|}$$

B. Normalization and Outlier Detection

In [1], the authors emphasize the importance of two preprocessing steps: normalizing the observed values to be zero meaned, and removing outlier values. The second does not work for our domain. The number of data sources available is limited (18) so it is difficult to get an iterative outlier detection algorithm as presented in [1] to work robustly with our data.

Instead, we take inspirations from the principles presented in [21], [22], and experiment with different combinations of sources to get the best estimates.

IV. Experiments

To compute the *Match Likelihood* in Eq (3), we collect name frequencies for all the metroplitan areas (MSA) and states in US (402 all together). Limited by dictionary sizes, we can not collect frequencies for full names, so we assume that the distribution of middle name is independent with respect to the distribution of observed *first, last* names:

$$\phi_n(\mu, r) = \phi_{n_{f,l}}(n_f, n_l)\, \phi_{n_m}(n_m).$$

With the simplication, there are more than $100,000,000$ unique combinations of *first, last* names, which is equivalent to about 40 billion μ_e to estimate. So we Hadoopify the

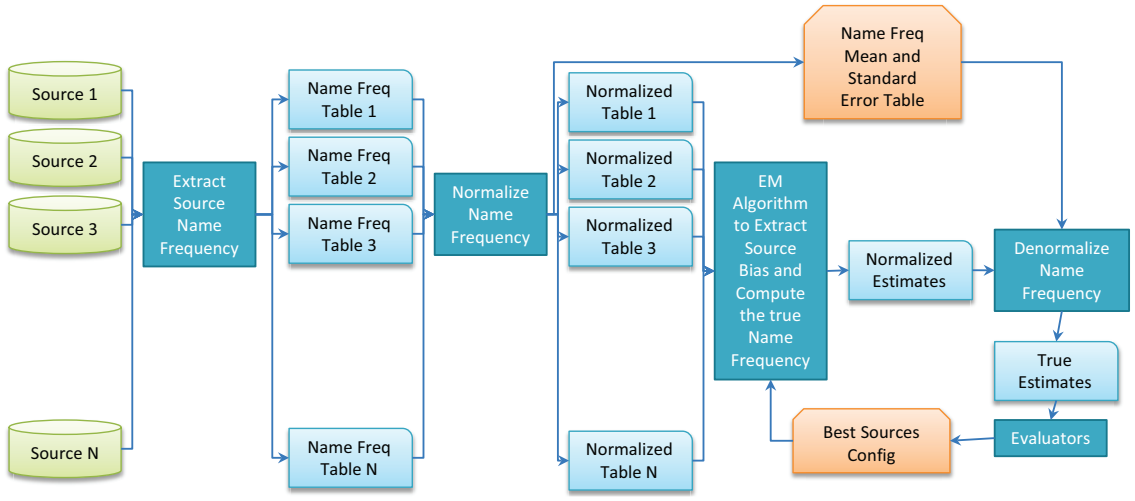

Figure 7. Implementation of GTM for Name Frequency Truth Estimates

Table I
DATA SOURCES AND THEIR RECORD COUNTS

Source	Count	Source	Count
A	75,848,150	C	100,601,282
D	75,826,071	F	86,646,948
L	85,982,872	G	88,455,393
H	219,054,407	N	107,238,935
Q	215,271,940	T	262,904,192
X	515,176,119	E	228,545,777
M	423,808,772	Y	258,533,490
Z	255,112,376	V	304,804,288
I	909,702,398	B	7,677,583

Table II
COMBINATIONS OF DATA SOURCE FOR THE EXPERIMENTS

Config	Data Sources
Full Set	A C D F L G H N Q T X E M Y Z V I B
Config 1	A C L M Q T X Y Z V I E
Config 2	A C L G H Q T X E M Y Z V I B
Config 3	A M Q T X Y Z V I E
Config 4	A M Q T X Y Z V E
Config 5	A H M Q T X Y Z V E
Config 6	A C D F L G M N Q T X E Y Z V I
Config 7	A C D F L G H N Q T X E M Y Z V I

algorithm in Section III to run multiple experiments at the same time efficiently.

We choose eight different combinations of 18 commercially available data sources (see Table I), based on our domain knowledge of each data source and how they relate to each other (see Table II), and then compare the estimates from GTM against a simple baseline, the simple mean of the numbers for the underlying data sources.

A. Evaluation Metrics on Attribute Statistics Estimates

To evaluate the estimates generated by the algorithm in Section III, we need ground truth for name frequencies. We can not obtain even a small data set just for evaluation, so we have to improvise and use other direct evaluation metrics.

1) Error w.r.t Census Last Name Data: The closest we can get to "truth data" is the frequencies of last names from US Census 2000 [24]. it is seriously outdated and contains only frequencies for last names with counts greater than or equal to 100, but unlike other data sources[5], it provides the exact counts in addition to population percentage for a last name.

For the evaluation, we add up the name frequencies for all observed "first last" combinations for a last name, and compare the sum with the frequencies released by the US Census. Let $\mathcal{L}$ be the set of all observed last names, $\mathcal{F}$ be the set of all observed first names, $\hat{\mu}(f,l)$ be the estimate by the algorithm in Section III for a name with first name f and last name l. Let $\tilde{\mu}$ be the frequency of the name as given by [24] and let $Pop_{US\ census}$ be the population of the US.

- absolute errors

$$\sum_{l\in\mathcal{L}}\left|\sum_{f\in\mathcal{F}}\hat{\mu}(f,l)-\tilde{\mu}\right| \tag{6}$$

- relative errors

$$\sum_{l\in\mathcal{L}}\left|\frac{\sum_{f\in\mathcal{F}}\hat{\mu}(f,l)-\tilde{\mu}}{Pop_{US\ census}(l)}\right| \tag{7}$$

If the estimates by the algorithm in Section III are larger than the truth, the *Match Likelihood* in Eq (3) will be smaller than it should be, the graph partitioning algorithm will produce more clusters. If the estimates are smaller, then we have clusters that contain multiple people. For us, the

[5]Hilary Mason, Data: first and last names from the US Census, http://www.hilarymason.com/blog/data-first-and-last-names-from-the-us-census/

second category of errors are more costly than the first one. So to compare the models/estimates, we also use:

- adjusted relative errors Let c_+ be the cost of positive errors and c_- be the cost of negative errors:

$$\sum_{l \in \mathcal{L}} \text{Error}(l)$$

$$\text{Error}(l) = \begin{cases} c_+ \left(\sum_{f \in \mathcal{F}} \hat{\mu}(f,l) - \tilde{\mu} \right), & \sum_{f \in \mathcal{F}} \hat{\mu}(f,l) \geq \tilde{\mu} \\ c_- \left(\tilde{\mu} - \sum_{f \in \mathcal{F}} \hat{\mu}(f,l) \right), & \sum_{f \in \mathcal{F}} \hat{\mu}(f,l) < \tilde{\mu} \end{cases} \tag{8}$$

2) Total Counts: The metrics described above considers only frequent names. As shown in Figure 8, these names are few and only a small fraction of the entire population. The rarer names, the ones in the long tail, are the most interesting and represent the majority of our revenue.

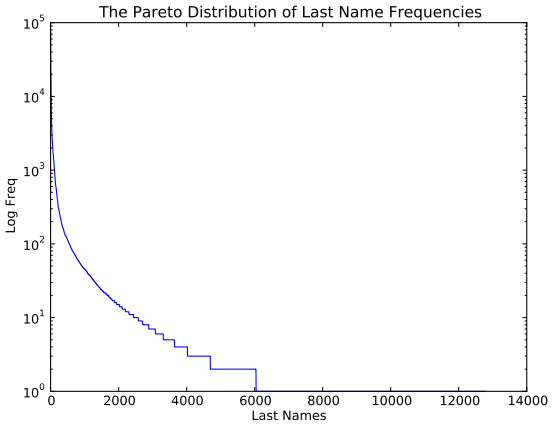

Figure 8. The Distribution of Last Name Frequencies.

To make sure the estimates by the Algorithm in Section III have high accuracies for the less common names, we total up the frequency of all observed names (to compare it with the true population of US).

B. Attribute Statistics Estimates Evaluation Results

The results of the experimentation on the 8 configurations of data sources are in Table III. c_+, c_- in Eq (8) are 10.0 and 1.0 respectively. All four evaluation metrics are consistent with each other, and they all show that *Config 4* is the best.

Table IV shows that compared with the baseline, for the combination *Full*, we have a 5.289% of improvement in relative error. For combination *Config 4*, the improvement is 9.682%.

C. Experiments with Match Likelihood

To evaluate the Match Likelihood approximation of Section II, we train two ADTrees [25], one without any feature based on the *Match Likelihood*, one with three features derived from the *Match Likelihood*:

Table IV
BASELINE ESTIMATES

Config	Relative Error	Total Count	Improvement
Full	95280.7	$5.190E8$	5.289%
Config 4	91702.1	$4.576E8$	9.682%

Table V
MODEL RECALL AT PRECISION OF 0.996

Model	precision	recall
With ML Features	0.996	0.895
Without ML Feature	0.996	0.844

- the *Match Likelihood* as a feature,
- the *Match Likelihood* of all relatives combined as a feature,
- the *Match Likelihood* but with a more relaxed way to match names.

The estimates from *Config 4* in the previous experiments are used to train the model with the *Match Likelihood* features. Figure 9 shows the precision recall trade-off curves of models built via 5 fold cross-validation on a training set of $102K$ manually collected and curated pair-wise people comparison examples. The one with the *Match Likelihood* features has much higher recall at the high precision we desire, 0.996, as shown in Table V.

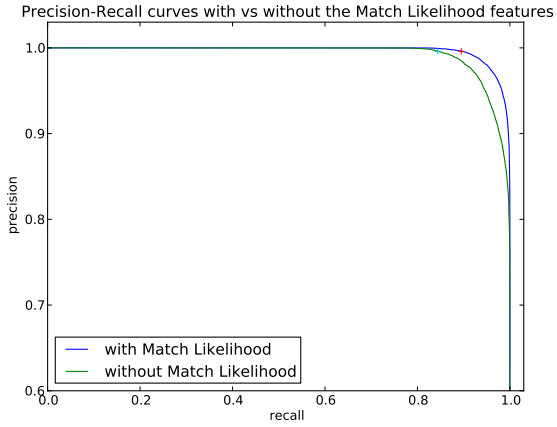

Figure 9. With the Match Likelihood features, the recall is much higher at the desired precision of 0.996.

V. LIMITATIONS AND FUTURE WORK

The GTM model used during the approximation of the match likelihood assumes that the noise from each data source can be modeled by a Gaussian distribution for the entire data source. The assumption does not hold for our problem domain. It is reasonable to assume that each data source introduces its own bias, but not so to assume that the bias is zero meaned or symmetrical.

Table III
EVALUATION OF ESTIMATES FROM 8 COMBINATIONS OF DATA SOURCES

Sources Config	Asolute Error	Relative Error	Adjusted Relative Error	Total Count
Full	$1.064E8$	90241.6	829859.8	$4.758E8$
Config 1	$1.285E8$	92344.9	850097.2	$5.074E8$
Config 2	$1.126E8$	91102.7	837579.5	$4.893E8$
Config 3	$1.606E8$	94557.5	872317.4	$5.437E8$
Config 4	$7.779E7$	82823.4	724021.7	$4.057E8$
Config 5	$8.510E7$	84853.0	755268.6	$4.238E8$
Config 6	$1.060E8$	90045.9	827916.0	$4.770E8$
Config 7	$1.066E8$	90393.4	831463.5	$4.785E8$

Most of our data sources cover only a certain part of the demographics. Credit header data cover mostly people who have established credit history, property data cover mostly people who own or lease business or residential properties, and social data source that can be crawled publicly usually have a better coverage on the younger generation who care more about their online presence than privacy. So naturally, the counts from these data sources are all underestimates to the target values. Most of our aggregate data sources have a dispersion problem so they usually give overestimates.

In the implementation presented in Section III, we did not remove outliers, and simply keep them in the data set. Since many of our data sources cover a subset of the entire population, we can use some of these underestimates as lower bounds. Similarly, for the observations from the aggregate data sources, we can use them as upper bounds. We will consider incorporate these bounds directly into the truth value finding process.

VI. CONCLUSIONS

This paper presents a probabilistic method to approximate the match likelihood of a pair of records by incorporating values of different attributes and their aggregates/statistics. The quality of the approximate depends on accuracy of the estimates of these aggregated values. The paper adapts the GTM model presented by [1] so we have a principled way to obtain the estimates. The paper presents experimental results based on 18 real world commercial data sources to show that the estimates obtained via GTM model is significantly better than the baseline (5 to 9 percent better) . The experimental results also showed that the appoximate match likelihood can significantly improve the recall of a high precision similarity functiion (5 percent of recall improvement).

ACKNOWLEDGMENTS

Special thanks to Ben Huntley at inome Inc for his hard-work on upgrading and maintaining our hadoop cluster, so we can run the experiments reported in this paper, to David McAlpin for providing with many interesting examples for record linkage/entity resolution problems he found with Freebase, including the John Arnold example we introduced at the beginning of the paper.

REFERENCES

[1] B. Zhao and J. Han, "A probabilistic model for estimating real-valued truth from conflicting sources," in *10th International Workshop on Quality in Databases*, August 2012.

[2] X. L. Dong and D. Srivastava, "Big data integration," *Proc. of VLDB Endowment*, vol. 6, no. 11, pp. 1188–1189, 2013.

[3] W. E. Winkler, "Overview of record linkage and current research directions," in *Technical Report Statistics #2006-2 US Census Bureau*, 2006.

[4] I. P. Fellegi and A. B. Sunter, "A theory of record linkage," *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 371–380, 1969.

[5] M. Bilenko and R. J. Mooney, "On evaluation and training-set construction for duplicate detection," in *Proc. of the 2003 KDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, 2003, pp. 7–12.

[6] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 1–16, 2007.

[7] M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," in *Proc. of the Ninth International Conference on Knowledge Discovery and Data Mining*, August 2003, pp. 39–48.

[8] X. Dong, A. Halevy, and J. Madhavan, "Reference reconciliation in complex information spaces," in *Proc. of the 2005 ACM SIGMOD International Conference on Management of Data*, 2005, pp. 85–96.

[9] G. Papadakis, E. Ioannou, C. Niederee, and P. Frankhauser, "Efficient entity resolution for large heterogeneous information spaces," in *Proc. of the Fourth ACM International Conference on Web Search and Data Mining*, 2011, pp. 535–544.

[10] I. Bhattacharya and L. Getoor, "Collective entity resolution in relational data," *IEEE Data Engineering Bulletin*, vol. 29, no. 2, pp. 4–12, 2006.

[11] H. Kardes, D. Konidena, S. Agrawal, M. Huff, and A. Sun, "Graph-based approaches for organization entity resolution in mapreduce," in *Proc. of the ACL TextGraph 8 Workshop*, 2013, pp. 70–78.

[12] A. Gattani and other, "Entity extraction, linking, classification and tagging for social media: A wikipedia-based approach," *Proc. of VLDB Endowment*, vol. 6, no. 11, pp. 1126–1137, August 2013.

[13] M. A. Hernández and S. J. Stolfo, "The merge/purge problem for large databases," in *Proc. of the 1995 ACM SIGMOD International Conference on Management of Data*, 1995, pp. 127–138.

[14] W. P. McNeill, H. Kardes, and A. Borthwick, "Dynamic record blocking: Efficient linking of massive database in mapreduce," in *9th ACM International Workshop on Quality in Databases*, 2012.

[15] H. Kardes, S. Agrawal, X. Wang, and A. Sun, "CCF: Fast and scalable connected component computation in mapreduce," in *Proc IEEE International Conference on Computing, Networking, and Communications*, 2014.

[16] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivasta, "Truth finding in the deep web: Is the problem solved?" *Proc. of VLDB Endowment*, vol. 6, no. 2, pp. 97–108, 2013.

[17] D. Wang, L. Kaplan, H. Le, and T. Abbdelzaher, "On truth discover in social sensing, a maximum likelihood estimation approach," in *Proc. of the 11th IPSN Internationl Conference on Information Processing in Sensor Networks*, 2012, pp. 233–244.

[18] B. Zhao, B. I. P. Rubinstein, J. Gemmell, and J. Han, "A bayesian approach to discovering truth from conflicting sources for data integration," *Proc. of VLDB Endowment*, vol. 5, no. 6, pp. 550–561, 2011.

[19] D. Wang, T. Abbdelzaher, L. Kaplan, and C. Aggarwal, "Recursive fact-finding: A streaming approach to truth estimation in crowdsourcing applications," in *33rd IEEE International Conference on Distributed Computing Systems*, July 2013, pp. 530–539.

[20] X. Yin and W. Tan, "Semi-supervised truth discovery," in *Proc. of WWW 2011*, March 2011, pp. 217–226.

[21] X. L. Dong and B. S. D. Srivastava, "Less is more: Selecting sources wisely for integration," *Proc. of VLDB Endowment*, vol. 6, no. 2, pp. 37–48, 2012.

[22] X. L. Dong, L. Berti-Equille, and D. Srivastava, "Integrating conflicting data: The role of source dependence," *Proc. of VLDB Endowment*, vol. 2, no. 1, pp. 550–561, August 2009.

[23] ——, "Data fusion: Resolving conflicts from multiple sources," in *Lecture Notes in Computer Science 7923, Web-Age Information Management*, 2013, pp. 64–76.

[24] D. L. Word, C. D. Coleman, R. Nunziata, and R. Kominski, "Demographic aspects of surnames from census 2000," 2003.

[25] S. Chen, A. Borthwick, and V. R. Carvalho, "The case for cost-sensitive and easy-to-interpret models in industrial record linkage," in *8th ACM International Workshop on Quality in Databases*, 2011.