HOKKAIDO UNIVERSITY

Title	Learning an accurate entity resolution model from crowdsourced labels
Author(s)	Wang, Jingjing; Oyama, Satoshi; Kurihara, Masahito et al.
Relation	ICUIMC 2014 : January 9-11, Siem Reap, Cambodia ; proceedings, ISBN: 978-1-4503-2644-5
Issue Date	2014-01-09
Doc URL	https://hdl.handle.net/2115/65191
Rights	©2014 ACM. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in PUBLICATION, ICUIMC '14 Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication, http://doi.acm.org/10.1145/2557977.2558060
Type	conference paper
File Information	IMCOM2014-WANG.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

Learning an Accurate Entity Resolution Model from Crowdsourced Labels

Jingjing Wang
Graduate School of Information Science and Technology
Hokkaido University
Sapporo 060-0814, Japan
jingjing.wang@live.cn

Satoshi Oyama
Graduate School of Information Science and Technology
Hokkaido University
Sapporo 060-0814, Japan
oyama@ist.hokudai.ac.jp

Masahito Kurihara
Graduate School of Information Science and Technology
Hokkaido University
Sapporo 060-0814, Japan
kurihara@ist.hokudai.ac.jp

Hisashi Kashima
Department of Mathematical Informatics
The University of Tokyo
Tokyo, Japan / JST PRESTO
kashima@mist.i.u-tokyo.ac.jp

ABSTRACT

We investigated the use of supervised learning methods that use labels from crowd workers to resolve entities. Although obtaining labeled data by crowdsourcing can reduce time and cost, it also brings challenges (e.g., coping with the variable quality of crowd-generated data). First, we evaluated the quality of crowd-generated labels for actual entity resolution data sets. Then, we evaluated the prediction accuracy of two machine learning methods that use labels from crowd workers: a conventional LPP method using consensus labels obtained by majority voting and our proposed method that combines multiple Laplacians directly by using crowdsourced data. We discussed the relationship between the accuracy of workers' labels and the prediction accuracy of the two methods.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*clustering*; I.5.3 [**Pattern Recognition**]: Clustering—*algorithms, similarity measures*

General Terms

Algorithms, Experimentation

Keywords

Entity resolution, Crowdsourcing, Link prediction, Dimensionality reduction

1. INTRODUCTION

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IMCOM (ICUIMC)'14 January 9-11, 2014, Siem Reap, Cambodia.
Copyright 2014 ACM 978-1-4503-2644-5 ...$15.00.

Entity resolution (sometimes referred to as duplicate detection, entity reconciliation, or record linkage) is the task of determining whether different data objects refer to the same real world entity. Entity resolution plays an important role in data integration and data cleaning, especially when the same name refer to different entities. For example, consider the search results by Google in Figure 1. The task is determining whether the same person name ("Abby Watkins"in this case) on two Web pages refer to the same person in real life. Entity resolution can also be considered to be a link prediction problem by regarding the entity identity as a link. In the link prediction problem, data objects and the relationships among them are considered to be nodes and edges in a graph. Thus, whether the "Abby Watkins"on the two Web pages refer to the same person can be considered the same as determining whether there is a link between "Abby Watkins"on the two pages.

Many groups over the past several years have worked on automating entity resolution by using labeled training data created using machine learning techniques [4; 3; 16; 20]. They all have estimated the classifier directly from ground truth labels. Although they improved entity resolution, it is still far from perfect. In fact, for some supervised learning tasks, it may be infeasible (or very expensive) to obtain objective and reliable labels from domain experts. Thus, it is important to consider effective ways to gather a large number of labels for use as training data.

To obtain objective and reliable labels, increasing attention has been paid to crowdsourcing. Crowdsourcing services that are able to ask many workers to complete human intelligence tasks (HITs) via the Internet, such as the Amazon Mechanical Turk (AMT), can collect a large amount of labeled data in a short period and at low cost. Among the crowd workers, some are highly skilled, and some are not. The highly skilled ones generally provide valid labels, while the lesser skilled ones generally provide variable-quality labels, thereby creating a quality control problem.

Since crowdsourcing provides labels without an absolute gold standard, there are two main ways to process labels provided by crowd workers: one is to estimate the ground truth labels [6; 17], and the other is to derive a classifier directly from crowd-generated data [7; 23; 11]. One common approach to estimating consensus la-

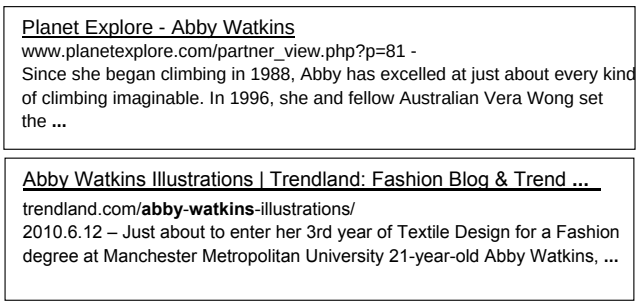

Figure 1: Entity Resolution Problem.

bels from individual worker labels is to use simple majority voting (MV) [17; 18; 25], which can often achieve relatively good results depending on the accuracy of the workers involved. In MV, the label receiving the most votes is selected as the final aggregated label.

There are a number of papers on designing fundamental algorithms for crowdsourcing [9; 14; 15] and a number of papers on developing systems for entity resolution in Web search result [13; 1; 2; 21; 12], but only a few on crowdsourcing entity resolution [22; 8]. The method proposed by Gomes *et al.* [9] uses a novel model of human clustering rather than supervised machine learning to aggregate worker annotations, but it cannot be used to predict categories of unknown data. Wang *et al.* [22] reported a hybrid human-machine workflow for dealing with crowdsourcing entity resolution: all the data are processed by machines first, and only the most likely matching data are verified manually.

We have compared two supervised machine learning methods for resolving entities: locality preserving projections (LPP) [10] using consensus labels obtained by majority voting and a combination of multiple Laplacians (multi-Lap) using crowd-sourced data directly. Since the feature vectors obtained from Web pages are high dimensional and very sparse, a dimensionality reduction approach is used in the learning process of both methods to project high-dimensional data objects to a low-dimensional latent feature space, and the data objects are identified on the basis of the distance between them. That is, the closer two data objects are to each other in the latent space, the greater the likelihood that they represent the same real world entity.

The LPP method was designed to derive an optimal entity model from actual crowdsourced data without a gold standard. Since individual crowd workers often exhibit highly variable labeling accuracy, we asked the workers to label each HIT so that we could derive a single consensus label by majority voting. If an individual worker labels a pair of Web pages more than once, the label used is the one that has the most votes among the workers. The final consensus label is the one that has the most votes among the workers.

Since majority voting treats each worker's vote equally, we use the multi-Lap method to directly make use of the variable-quality labels generated by crowd workers. To derive an accurate entity model, we added a weighting measure to the learning process: higher weights are assigned to the more accurate workers. Without such weighting, each worker's vote is treated equally. Two weighting measures are used to represent crowd worker credibility. One is based on the assumption the majority vote gives the ground truth label. Thus, we value the weight of each worker as the degree of similarity between the labels provided by the worker and the labels provided by the other workers.

However, under some circumstances, the majority opinion is incorrect. For example, if most of the crowd workers give an incorrect label, the majority opinion must be wrong. Especially with crowdsourcing, the probability is high that the majority of the workers for a particular task are unskilled. We thus use the mutual dependency principles proposed by Dai et al. [5] as our second weighting measure to evaluate crowd worker credibility. These principles include positive and negative mutual dependencies. For our problem, a positive mutual dependency means that a worker is more credible if he gives agreeing edges to pairs of Web pages referring to the same real world entity and vice versa. A negative mutual dependency means that a worker is less credible if he gives disagreeing edges to pairs of Web pages referring to the same real world entity and vice versa.

We experimentally evaluated the label quality of actual crowdsourced data by comparing the labels provided by individual workers and the consensus labels with the ground truth labels. We computed consensus labels by weighting each worker's vote equally (simple MV). Then we compared the prediction accuracy of the conventional LPP method using consensus labels with that of our proposed multi-Lap method with and without weighting.

2. SUPERVISED MACHINE LEARNING USING CROWD-GENERATED LABELS

Since the feature vectors obtained from Web pages are high dimensional and very sparse, we need to use a dimensionality reduction approach in order to project high-dimensional data objects to a low-dimensional latent feature space. This enables comparison of the data, which is difficult to do in the original high-dimensional space.

Yamanishi [24] outlined a distance metric learning method that can link heterogeneous objects such as compounds and proteins on the basis of two mappings of the heterogeneous objects to a common Euclidean space. We found in our experiments that this linear function works well for resolving entities in comparison to the other dimensionality reduction methods.

We thus used the conventional LPP method [10; 24] to resolve entities. We first describe the use of consensus labels and then describe our proposed multi-Lap method with two weighting measures.

2.1 LPP with Consensus Labels

Linear projection $\mathbf{W}$ from an original D-dimensional feature space to a d-dimensional latent feature space is derived from training data consisting of data objects known to have or not to have links between them. If data objects $\mathbf{x}$ and $\mathbf{y}$ are assumed to have a known link, the distance between them in latent space should be as small as possible.

$$\|\mathbf{W}\mathbf{x} - \mathbf{W}\mathbf{y}\|$$

The link between two data objects with unknown link status is predicted on the basis of the distance between them after they are mapped to the latent space by using $\mathbf{W}$.

Assume a set of N training data objects $\mathbf{x}_1, \ldots, \mathbf{x}_N$ in $\mathbb{R}^D$. The process commonly used for finding the optimal projection matrix $\mathbf{W}^*$ that makes linked nodes close to each other is based on the method of He and Niyogi [10].

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \sum_{i,j} \mathbf{A}_{ij} \|\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j\|_2^2,$$

where $\|\cdot\|_2$ is the Euclidean norm (2-norm), and A_{ij} represents the link status between the data objects in the training data set, which

can be defined as an adjacency matrix:

$$A_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ have a link,} \\ 0 & \text{otherwise.} \end{cases}$$

We use consensus labels obtained by MV as the adjacency matrix. The problem can be rewritten as a generalized eigenvector problem:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \mathbf{tr}\left(\mathbf{W}\mathbf{\Phi}^\mathsf{T}\mathbf{L}\mathbf{\Phi}\mathbf{W}^\mathsf{T}\right)$$
$$\text{s. t. } \mathbf{W}\mathbf{\Phi}^\mathsf{T}\mathbf{D}\mathbf{\Phi}\mathbf{W}^\mathsf{T} = \mathbf{I}_d,$$

where $\mathbf{\Phi}$ is the design matrix defined as $\mathbf{\Phi} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\mathsf{T}$, $\mathbf{D}$ is a diagonal degree matrix in which the entries are column sums of $\mathbf{A}$, $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$, and $\mathbf{L}$ is the Laplacian matrix defined by $\mathbf{L} = \mathbf{D} - \mathbf{A}$. $\mathbf{I}_d$ is the $d \times d$ identity matrix. We use the constraint to avoid the trivial solution ($\mathbf{W} = 0$) and ensure the uniqueness of the solution.

Solving this optimization problem is equivalent to solve the following generalized eigenvalue problem.

$$\mathbf{\Phi}\mathbf{L}\mathbf{\Phi}^\mathsf{T}\mathbf{w} = \lambda\mathbf{\Phi}\mathbf{D}\mathbf{\Phi}^\mathsf{T}\mathbf{w}.$$

By finding d eigenvectors with the smallest positive eigenvalues we can get the optimal linear projection matrix $\mathbf{W}^*$.

2.2 Combination of Multiple Laplacians

Though we can solve entities with aggregated crowdsourced labels, the information on the distribution of workers' judgment is missing in the learning process, e.g., the consensus label obtained by MV is truncated to 0 or 1. We introduce a combination of multiple Laplacians to entity resolution to directly derive an accurate model from crowd-generated labels. The process of finding the optimal projection matrix can be seen as

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \sum_t^T \sum_{i,j} \mathbf{A}_{ij}^{(t)} \|\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j\|_2^2,$$

where the adjacency matrix $\mathbf{A}^{(t)}$ is defined as $\mathbf{A}^{(t)} = \{\mathbf{A}_{ij}^{(t)}\}$, which is provided by worker t. T is the number of crowd workers.

The problem can be also formulated as an optimization problem:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \mathbf{tr}\left(\mathbf{W}\mathbf{\Phi}^\mathsf{T}\sum_t^T \mathbf{L}^t\mathbf{\Phi}\mathbf{W}^\mathsf{T}\right),$$

where $\mathbf{L}^t$ is the Laplacian matrix corresponding to adjacency matrix $\mathbf{A}^{(t)}$, which is defined the same as $\mathbf{A}$ in the LPP method, $\mathbf{A}_{ij}^{(t)} = 1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ have a link otherwise $\mathbf{A}_{ij}^{(t)} = 0$.

2.2.1 Weighting Workers

The model derived for a more qualified worker has a higher degree of accuracy. To compensate for the variability in the accuracy of the crowd workers and thereby get a more accurate true model, we impose an additional weighting measure on the learning process: we assign higher weights to more accurate workers.

As a result, the process of finding the optimal projection matrix can be rewritten as

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \sum_t^T \sum_{i,j} r_t\mathbf{A}_{ij}^{(t)} \|\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j\|_2^2,$$

where r_t is the weighting parameter.

The problem can also be formulated as an optimization problem:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \mathbf{tr}\left(\mathbf{W}\mathbf{\Phi}^\mathsf{T}\sum_t^T r_t\mathbf{L}^t\mathbf{\Phi}\mathbf{W}^\mathsf{T}\right).$$

This means that the problem can be reduced to a generalized eigenvalue problem:

$$\mathbf{\Phi}\sum_t^T r_t\mathbf{L}^t\mathbf{\Phi}^\mathsf{T}\mathbf{w} = \lambda\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T}\mathbf{w}.$$

The optimal projection matrix $\mathbf{W}^*$ is obtained by finding the d eigenvectors with the smallest positive eigenvalues. After the data objects are mapped to the $d-$dimensional space by using projection matrix $\mathbf{W}^*$, the links are identified on the basis of the distance between them, and the prediction accuracy can be calculated by comparing the predicted result with the gold standard.

To find the d smallest eigenvectors can be problematic, so we found the eigenvectors for the largest positive eigenvalues of the generalized eigenvalue problem by setting $\lambda = 1 \setminus \mu$. Then, the final generalized eigenvalue problem becomes:

$$\mathbf{\Phi}\sum_t^T r_t\mathbf{L}^t\mathbf{\Phi}^\mathsf{T}\mathbf{w} = \mu\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T}\mathbf{w}.$$

Three weighting measures were used to represent crowd worker credibility. One values the weight of each worker as the degree of similariy, one uses mutual dependency principles to score the weight of each worker and without weighting. Without weighting, the labels provided by the crowd workers would be treated equally; that is, $r_t = 1/T$.

Similarity.

If we assume that most workers give true labels, a worker is more qualified if the labels provided by the worker are more similar to the labels provided by other workers, so a higher weight should be given to that worker. The similarity between labels $\mathbf{A}_{t_i}$ provided by worker t_i and labels $\mathbf{A}_{t_j}$ provided by worker t_j is defined as

$$Sim_{t_it_j} = \frac{(\mathbf{A}_{t_i} \cdot \mathbf{A}_{t_j})}{||\mathbf{A}_{t_i}||||\mathbf{A}_{t_j}||}.$$

Thus, the weight of worker t_i is defined as the sum of the similarities:

$$r_{t_i} = \sum_{j\neq i, j\in\mathcal{T}} Sim_{t_it_j},$$

where $\mathcal{T}$ is the set of crowd workers, $\mathcal{T} = \{1, \ldots, T\}$. In practice, we use normalization to make the sum of the weights of all workers equal to 1.

Mutual Dependence.

A worker-labeled page pair can be modeled as a bipartite graph in which the label given by a worker to a pair of Web pages can be seen as a directed edge that carries the *"opinion"* of a source node towards a target node. From the perspective of the target node (a pair of Web pages), the edge can be identified as "agreeing"or "disagreeing"by determining whether the opinion carried by the edge agrees with the majority voting opinion for the target node. In our case, if most workers label a pair of Web pages p as*"0,"* worker t_1 labels it as *"1,"* and worker t_2 labels it as *"0,"* edge (t_1, p) is a disagreeing edge, and edge (t_2, p) is an agreeing edge.

We use the mutual dependency principles proposed by Dai et al. [5] to evaluate of crowd worker credibility. These principles include the positive and negative mutual dependencies. An agreeing

Table 1: Training data and test data (Web pages) for each person name.

	Trainning data	Test data
Abby Watkins	60	52
Alvin Cooper	40	47
Armando Valencia	30	29
David Lodge	40	50
Jerry Hobbs	40	48
Michael Howard	40	48
Patrick Killen	40	48
Paul Clough	30	34
Stephan Johnson	40	35
Thomas Baker	30	37

edge has a positive mutual dependency on its source and target: a worker is more credible if he or she gives a label representing an agreeing edge to a pair of Web pages that are related to the same real world entity; similarly, a worker is less credible if he or she gives a label representing an agreeing edge to a pair of Web pages that are not related to the same real world entity. A disagreeing edge acts as a negative mutual dependency: a worker is less credible if he or she gives a label representing a disagreeing edge to a pair of Web pages that are related to a real world entity; a worker is more credible if he or she gives a label representing a disagreeing edge to a pair of Web pages that are not related to a real world entity.

Given bipartite graph $G = \langle \mathcal{T} \cup \mathcal{P}, E, S \rangle$, where $\mathcal{T}$ is the set of crowd workers, $\mathcal{P} = \{p_1, ..., p_{|\mathcal{P}|}\}$ is a set of Web page pairs, $E \subset \mathcal{T} \times \mathcal{P}$ is a set of directed edges derived from the labels given by the workers to pairs of pages, and $S = \{s_{ij}\}$ is the set of labels attached to the edges. If edge (t_i, p_j) is an agreeing edge, $s_{ij} = 0$; otherwise, $s_{ij} = 1$.

Let $\mathfrak{t}_i$ be the weight assigned to worker t_i and $\mathfrak{p}_j$ be the weight assigned to Web page pair p_j. The corresponding edge is (t_i, p_j). The use of the mutual dependency principles means that, if $s_{ij} = 0$, $\mathfrak{t}_i$ mirrors $\mathfrak{p}_j$, i.e., $\mathfrak{t}_i = \mathfrak{p}_j$; if $s_{ij} = 1$, $\mathfrak{t}_i$ is the opposite of $\mathfrak{p}_j$, i.e., $\mathfrak{t}_i = 1 - \mathfrak{p}_j$. The edges are treated equally due to the absence of information on the relative credibility of the connected crowd workers.

$$\begin{cases} \mathfrak{t}_i = AVG_{p_j:(t_i,p_j)\in E}(1 - 2s_{ij})\mathfrak{p}_j + s_{ij} \\ \mathfrak{p}_j = AVG_{t_i:(t_i,p_j)\in E}(1 - 2s_{ij})\mathfrak{t}_i + s_{ij} \end{cases}$$

These formulae are converted into matrix form, and an iterative algorithm is used to compute the scores for the crowd workers. The score for each worker ranges from 0 to 1. Scores in the range $[0, 0.5)$ are considered workers in low credibility, and those in the range $[0.5, 1)$ are considered high credibility.

3. CROWDSOURCED LABELING FOR WEB ENTITY RESOLUTION

In this section, we introduce the method we used for obtaining crowdsourced labels and the *majority voting (MV)* method we used for aggregating the labels. Then, we evaluate the label quality of an actual crowdsourced data set.

3.1 Data Set

We used data obtained from the *Searching Information about Entities in the Web (WEPS)* dataset[1], which provides Web pages for given person names, especially for very common names that have high ambiguity on the Web and for names of famous or historical people. This dataset also provides a gold standard for Web pages containing each person's name, and these Web pages were well clustered by experts. The experts were asked to decide, on the basis of the contents, whether the person names on two Web pages referred to the same person in the real world. The number of Web pages used as training data and as test data for each person name are shown in Table 1 (blank pages were removed). We used tf-idf (the term frequency-inverse document frequency) as the feature.

3.1.1 Crowd-generated Labels

We used the *Lancers* crowdsourcing service [2] to allocate HITs to crowd workers. Each HIT contained n Web page URLs. The workers were instructed to open these pages one by one and find all the ones that referred to the same entity in the real world. A drop-down list on the HIT page enabled a worker to assign the same id number to the Web pages considered to refer to the same real world entity. Since we structured the HITs as the clustering of N Web pages, we had to specify a sampling scheme. If we had structured the HITs as simply separate Web pages in disjoint sets, it would have been impossible to infer a clustering over the entire data set because we would not have known whether two Web pages in different HITs were in the same cluster or not. Therefore, the Web pages had to be members of multiple HITs.

We used a random sampling scheme [19; 9] in which a simple parameter V is used to control the level of sampling redundancy. We set V to be the number of HITs to which a Web page belonged. This meant that each Web page was voted on V times by the crowd workers. We set n to 10 or 20 depending on the number of Web pages available for the person name and V to 5. After removing the blank Web pages in *WEPS*, we had 30 Web pages as training data for person names *"Armando Valencia," "Paul Clough," "Thomas Baker* ($n = 10$), 60 for *"Abby Watkins "* and 40 for the other person names (n=20). The total number of data items was N. Thus, the total number of different HITs was

$$H = NV/n.$$

Because individual crowd workers often exhibit highly variable labeling accuracy, we asked multiple crowd workers to perform each HIT so that we could derive a single consensus label. Parameter R is the number of crowd workers who performed the same HIT. Therefore, the total number of HITs generated was

$$H = NVR/n.$$

In the experiment described below, each HIT was assumed to be performed five times ($R = 5$).

3.1.2 Consensus Labels

Majority voting was used to generate consensus labels for the LPP method in two ways: to derive a single consensus label for each crowd worker and to generate the final consensus labels for all crowd workers. Since Web pages were assigned to multiple HITs, a worker could label a pair of Web pages several times. The two Web pages were assumed to be linked once the same id number was assigned to them by the same person.

As mentioned, simple MV was used to compute the final consensus labels (weighting each worker's vote equally). Consider a set of noisy labels $\{y_{ij}^t\}$ for data items $pair\ (\mathbf{x}_i, \mathbf{x}_j)$, where $y_{ij}^t \in \{0, 1\}$ is a noisy label provided by the $t-$th worker ($t \in \{1, \ldots, T\}$). The set of workers who labeled this pair is defined as $\mathcal{J}_t \subseteq \{1, \ldots, T\}$.

[1] http://nlp.uned.es/weps/weps-1/weps1-data

[2] http://www.lancers.jp

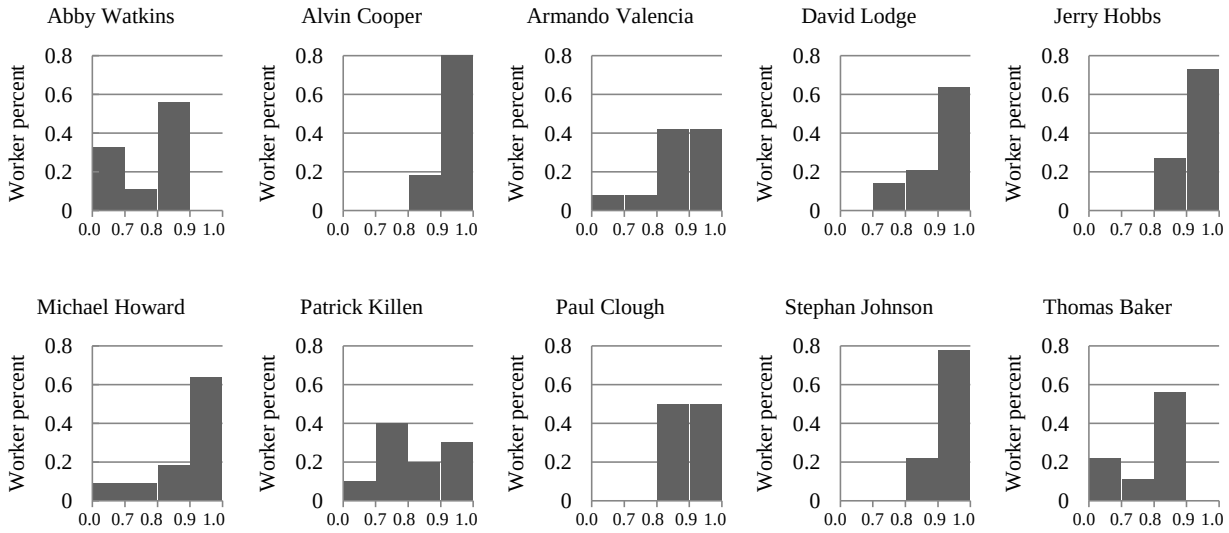

Figure 2: Accuracy of labels provided by individual workers.

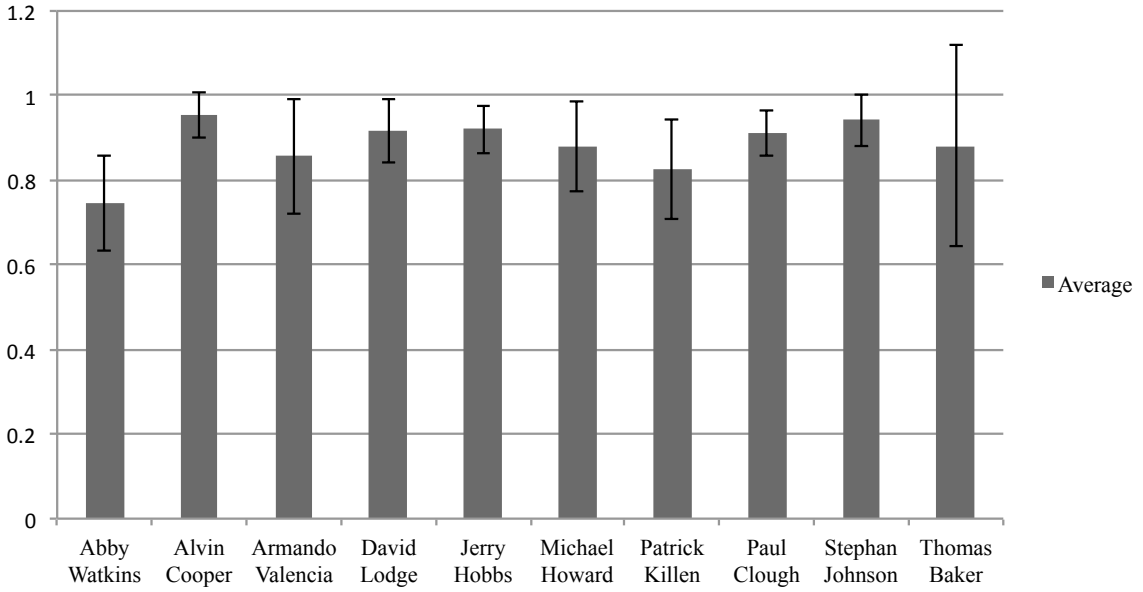

Figure 3: Average accuracy and standard deviation for each person name.

The consensus label y_{ij} for this pair derived from majority voting is given by

$$y_{ij} = \begin{cases} 1 & \text{if } \sum_{t \in \mathcal{J}_t} y_{ij}^t > \mid \mathcal{J}_t \mid /2, \\ 0 & \text{if } \sum_{t \in \mathcal{J}_t} y_{ij}^t < \mid \mathcal{J}_t \mid /2, \\ \text{random} & \text{otherwise.} \end{cases}$$

3.2 Label Quality

Using the methods described above, we collected actual crowdsourced data for ten person names *("Abby Watkins," "Alvin Cooper," "Armando Valencia," "David Lodge, " "Jerry Hobbs," "Michael Howard," "Patrick Killen," "Paul Clough," "Stephan Johnson," "Thomas Baker,")* and generated consensus labels. We estimated the label quality of the crowdsourced data by comparing both the labels provided by individual workers and the consensus labels estimated by MV with the gold standard.

The experimental results presented in Figure 2 and Figure 3 and 4 show that the accuracy of the labels provided by individual workers was satisfactory although there was some dispersion of accuracy across workers (the average accuracy was above 80% except for *Abby Watkins*, and the standard deviation was mostly around 10%). The accuracy for the consensus labels was mostly around 80% and went as high as 90%. Since entity resolution is difficult, the 80% accuracy is higher than predicted and indicates that it is

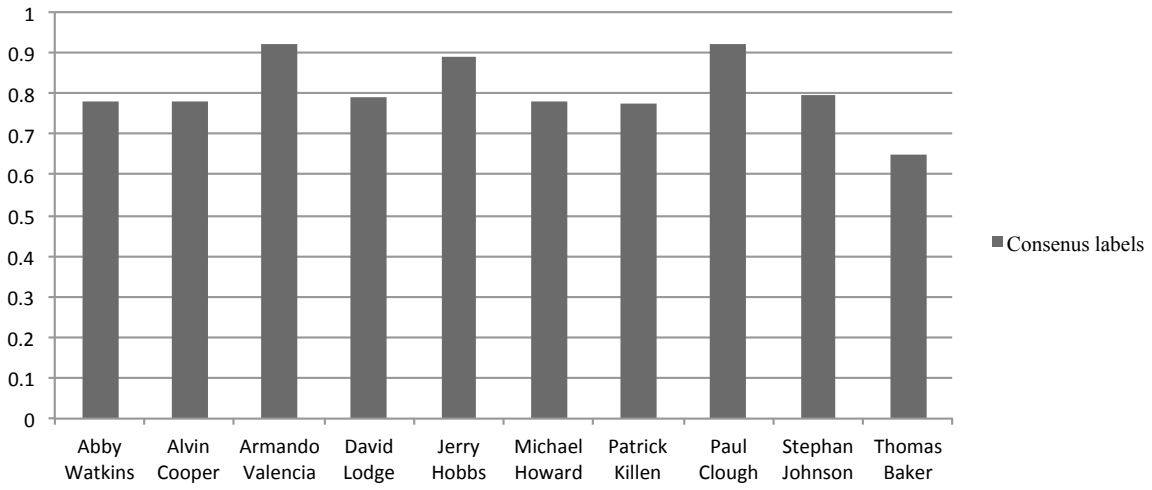

Figure 4: Accuracy of consensus labels.

possible to derive an accurate entity resolution model using these methods.

Here, we give only a general idea of how we can generate consensus labels by using crowdsourcing to resolve entities. Though there are some well-know techniques (e.g., expectation maximization) that may outperform MV, the labeling accuracy of MV is sufficient for our purposes here. The prediction accuracy of the LPP method with consensus labels was less than that of the gold standard no matter how much the labeling accuracy was improved, while experiment results given in section 4 show that the proposed multi-Lap method has accuracy comparable to or better than that of the LPP method with the gold standard.

4. EVALUATION

We experimentally evaluated the prediction accuracy of the LPP method using consensus labels and the multi-Lap method with three weighting measures by implementing it on the Matlab platform (Matlab R2011b) and using the built-in $eigs$ function to solve the generalized eigenvalue problem. The dimension of the latent feature space was set to ten.

We estimated the prediction accuracy of the conventional LPP method [10] for the ground truth labels and for the consensus labels of the crowdsourced data. We also evaluated the prediction accuracy of the multi-Lap method without weighting (Uni), with weighting based on similarity (Sim), and with weighting based on mutual dependency (Mut). We used the AUC (area under the curve) values to measure the prediction.

Generally, for our proposed multi-Lap method, the accuracies using crowd-sourced data were comparable to or better than the conventional LPP method using gold standard and were superior to the LPP method using consensus labels eight out of ten cases in Figure 5.

However, interestingly, the multi-Lap method sometimes outperformed the LPP method using the gold standard, e.g., *"Alvin Cooper"* (Mut), *"Armando Valencia "* (Mut). In entity resolution, some Web pages are ambiguous even for human experts. Especially in certain difficult cases, the labels have to be decided to be a certain value that may cause overfitting. In the multi-Lap method, such difficult cases are treated without aggregating the labels.

By using the mutual dependency principles to calculate the credibility of crowd workers, we can get the correct label even if the majority of the crowd workers give an incorrect label. The multi-Lap method with mutual dependency had better performance than the others two weighting measures five out of ten times and had performance comparable to the LPP method using the gold standard even for two exceptional cases *"Patrick Killen"* and *"Paul Cloug"*.

Comparing the accuracy distributions in Figure 2 with the prediction accuracy, when most workers had virtually the same level of labeling skill (*"Jerry Hobbs"*, *"Stephan Johnson"*), the LPP method with consensus labels had higher prediction accuracy than the multi-Lap method. However, when most workers were in low labeling skill, as they did For *"Abby Watkins"*, *"Patrick Killen"* and *"Thomas Baker"*, the multi-Lap method still provided performance comparable to the LPP method using the gold standard.

Furthermore, from the results in Figure 6, we can see that our Multi-lap method always performs as least as fast as the LPP method with the true labels and the LPP method with the consensus labels. Though our Multi-lap with mutual dependency relatively the most time-consuming, most of the time is less than 0.1 second difference.

5. CONCLUSION

We have evaluated the label quality of actual crowdsourced data for entity resolution and compared the prediction accuracy of our proposed combination of multiple Laplacians (with and without weighting) with that of the conventional locality preserving projection method using consensus labels. Analysis of label quality showed that the accuracy of labels provided by individual workers was satisfactory while there was some dispersion of accuracy across workers. The prediction accuracy of the combination of multiple Laplacians method using crowd-sourced labels was comparable to or better than the LPP method using the gold standard and was superior to the LPP method using the consensus labels.

Our combination of multiple Laplacians method with mutual dependency had better performance than the other two weighting measures. However, the multiple Laplacians methods with mutual dependency cannot win in every case, a better weighting measurement is still in need. Our future work is solving the measure used for assigning weights to workers in the learning process.

	LPP (True Label)	LPP (Consensus Label)	Multi-LAP		
			Uni	Mut	Sim
Abby Watkins	0.730	0.654	0.634	0.637	**0.715**
Alvin Cooper	0.596	0.502	0.589	**0.617**	0.577
Armando Valencia	0.532	0.556	0.517	**0.569**	0.533
David Lodge	0.543	0.506	0.556	**0.584**	0.557
Jerry Hobbs	0.523	**0.560**	0.508	0.508	0.503
Michael Howard	0.545	0.549	**0.564**	0.532	0.509
Patrick Killen	0.632	0.538	**0.675**	0.630	0.625
Paul Clough	0.528	0.512	0.510	0.524	**0.529**
Stephan Johnson	0.657	**0.658**	0.570	0.508	0.507
Thomas Baker	0.636	0.533	0.509	**0.691**	0.501

Figure 5: AUC values.

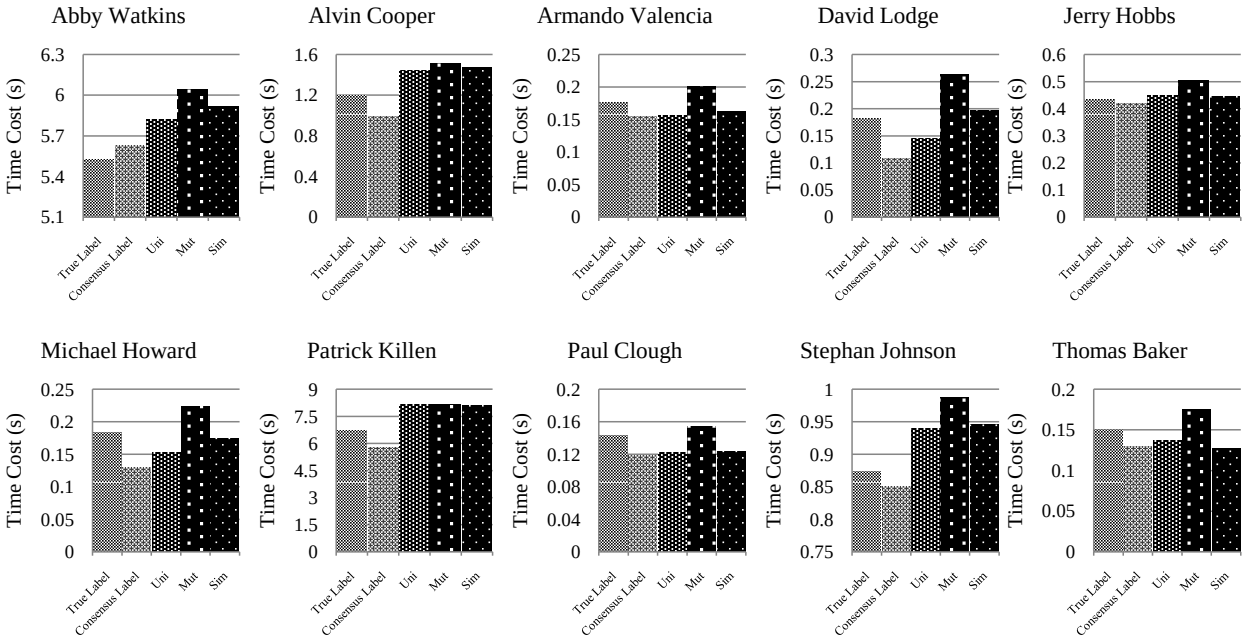

Figure 6: Time cost (CPU: Intel Core i7-3770 3.40GHz*2, memory: 8.00GB, operating system: Windows 7 Professional).

6. REFERENCES

[1] R. Al-Kamha and D. W. Embley. Grouping search-engine returned citations for person-name queries. In *Proceedings of the 6th Annual ACM international Workshop on Web information and Data management*, pages 96–103, 2004.

[2] R. Bekkerman and A. McCallum. Disambiguating Web appearances of people in a social network. In *Proceedings of the 14th international conference on World Wide Web (WWW)*, pages 463–470, 2005.

[3] M. Bilenko, W. W. Cohen, S. Fienberg, R. J. Mooney, and P. Ravikumar. Adaptive name-matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23, 2003.

[4] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 39–48, 2003.

[5] H. Dai, F. Zhu, E. Lim, and H. Pang. Detecting anomalies in bipartite graphs with mutual dependency principles. *12th IEEE International Conference on Data Mining (ICDM)*, pages 171–180, 2012.

[6] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C*, 1979.

[7] O. Dekel and O. Shamir. Vox populi: Collecting high-quality labels from a crowd. In *Proceedings of the Conference on*

Learning Theory (COLT), 2009.

[8] W. S. Euijong, M. Julian, and G. M. Hector. Compare me maybe: Crowd entity resolution interfaces. In *Technical Report, Standford InfoLab*, 2012.

[9] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. *In Advances in Neural Information Processing 24 (NIPS)*, 2011.

[10] X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing 16 (NIPS)*, 2004.

[11] H. Kajino, Y. Tsuboi, and H. Kashima. A convex formulation for learning from crowds. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

[12] R. Kimura, S. Oyama, H. Toda, and K. Tanaka. Creating personal histories from the Web using namesake disambiguation and event extraction. In *Proceedings of the 7th International Conference on Web Engineering (ICWE)*, pages 400–414, 2007.

[13] G. S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*, pages 33–40, 2003.

[14] A. Marcus, E. Wu, D. Karger, S. Madden, and R. Miller. Human-powered sorts and joins. In *Proceedings of the VLDB Endowment*, pages 13–24, 2011.

[15] A. G. Parameswaran, H. Garcia-Molina, H. Park, N. Polyzotis, A. Ramesh, and J. Widom. Crowdscreen: algorithms for filtering data with humans. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 361–372, 2012.

[16] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 269–278, 2002.

[17] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 614–622, 2008.

[18] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 254–263, 2008.

[19] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2003.

[20] S. Tejada, C. A. Knoblock, and S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *Proceedings of the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 350–359, 2002.

[21] X. Wan, J. Gao, M. Li, and B. Ding. Person resolution in person search results: Webhawk. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 163–170, 2005.

[22] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. CrowdER: crowdsourcing entity resolution. In *Proceedings of the VLDB Endowment*, pages 1483–1494, 2012.

[23] F. L. Wauthier and M. I. Jordan. Bayesian bias mitigation for crowdsourcing. In *Advances in Neural Information Processing 24 (NIPS)*, pages 1800–1808, 2011.

[24] Y. Yamanishi. Supervised bipartite graph inference. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 1841–1848, 2009.

[25] H. Yang, A. Mityaginr, K. M. Svore, and S. Markov. Collecting high quality overlapping labels at low cost. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 459–466, 2010.