Resolving Homonymy with Correlation Clustering in Scholarly Digital Libraries

Jeongin Ju
Department of Computer Science, KAIST
Daejeon, Korea
jiju@an.kaist.ac.kr

Hosung Park
Department of Computer Science, KAIST
Daejeon, Korea
hosung@an.kaist.ac.kr

Sue Moon
Department of Computer Science, KAIST
Daejeon, Korea
sbmoon@kaist.edu

ABSTRACT

As scholarly data increases rapidly, scholarly digital libraries, supplying publication data through convenient online interfaces, become popular and important tools for researchers. Researchers use SDLs for various purposes, including searching the publications of an author, assessing one's impact by the citations, and identifying one's research topics. However, common names among authors cause difficulties in correctly identifying one's works among a large number of scholarly publications. Abbreviated first and middle names make it even harder to identify and distinguish authors with the same representation (i.e. spelling) of names. Several disambiguation methods have solved the problem under their own assumptions. The assumptions are usually that inputs such as the number of same-named authors, training sets, or rich and clear information about papers are given. Considering the size of scholarship records today and their inconsistent formats, we expect their assumptions be very hard to be met. We use common assumption that coauthors are likely to write more than one paper together and propose an unsupervised approach to group papers from the same author only using the most common information, author lists. We represent each paper as a point in an author name space, take dimension reduction to find author names shown frequently together in papers, and cluster papers with vector similarity measure well fitted for name disambiguation task. The main advantage of our approach is to use only coauthor information as input. We evaluate our method using publication records collected from DBLP, and show that our approach results in better disambiguation compared to other five clustering methods in terms of cluster purity and fragmentation.

Categories and Subject Descriptors

I.5.3 [**Computing Methodologies**]: Clustering - Algorithms, Similarity measures; H.2.8 [**Information Systems**]: Database Applications - Data mining

Keywords

Name disambiguation; Correlation clustering; Scholarly digital libraries

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

1. INTRODUCTION

Disambiguation of named entities is a common task in our daily life. When a user inputs a keyword to Wikipedia and multiple meanings exist under the keyword, Wikipedia lists them all and makes the user choose. People also have this problem of disambiguation when more than one person in a community share the same name. Disambiguating people with the same name can be done with nicknames in daily life, but becomes an acute issue when searching for individual information online. More specifically, author disambiguation in scholarly digital libraries (SDLs) is getting worse as such databases never shrink in size, but only grows.

SDLs are an integral tool in extracting the publications of an author, assessing one's impact by the citations, and identifying one's research topics, just to name a few key tasks. With SDLs potential employers review candidates' academic achievements and colleagues find related work easily. As SDLs only grow over time, more publications written by people with the same name are added and harder it gets to distinguish one author from another homonymous author.

Name disambiguation is not just for the convenience of SDLs users. With electronically archived data, we can predict the number of citations of a paper [15] and study the impact of close-knit social networks among coauthors [16]. Without name disambiguation these approaches are fundamentally flawed with the issue of data credibility. As Fegley *et al.* point out [6], homonymy and synonymy problems, which are two subproblems of name disambiguation, may lead to completely different results of scholar and citation networks in terms of clustering coefficients, interlocker ratios, and other statistics of networks.

Existing methods addressing the homonymy problem require a priori inputs such as labeled training sets, the number of homonyms or topics, or rich and clear information about papers. Considering the size of scholarship records today and their inconsistent formats, we expect their requirements be very hard to be met. Then, what is the minimum information that every publication has in SDLs, despite inconsistent format? Our answer is that author names and paper title are the most common information.

We assume that for any paper only authors' last name and the initials of the first names are given, along with the paper title. Based on coauthor information of papers, our method builds a document-author matrix for the papers, applies dimension reduction to identify sets of names frequently appearing together, and does correlation clustering to group papers by its authorship. We employ SVD (Singular Vector Decomposition)-based dimension reduction to

represent data in low dimension effectively, and PCA (Principal Component Analysis) to discover latent correlations among the author names. We also devise a new distance measure, the relative correlation distance, to group papers showing similar coauthor patterns effectively. We evaluate our methods using the data from [7] which is collected from DBLP and consists of papers about 11 most ambiguous names. Our evaluation shows that our methods can resolve the homonymy problem in SDLs with high purity using the base minimum information of the paper, the author names.

The structure of this paper is as follows. Section 2 covers a survey of the related work and density-based clustering. Section 3 presents the formal definition of the problem and designs of major building blocks of our method. Concepts of SVD-based dimension reduction, relative correlation distance measure, and other things devised to relieve the homonymy problem in SDLs are introduced here. Section 4 describes the data set used in evaluation and performance comparison of our method and other four density-based clustering approaches. Finally, in section 5 we conclude.

2. RELATED WORK

Among challenges that Smalheiser and Torvik raise in author name disambiguation in the metadata development of digital libraries [18], two challenges are relevant to our work. One is the situation that many different people with the same name, as in our work. Perreria, *et al.* call this problem *polysemes* [11]. The other is incomplete data, which include not the full name but the initials of the first name.

Name disambiguation is a long-studied problem and various methods have been proposed to solve the homonymy problems [2, 7, 8, 9, 12, 14, 17, 20, 21]. Roughly speaking, existing name disambiguation methods can be divided into three categories. First, general graph partitioning and clustering techniques, such as k-means clustering and spectral clustering, are applied to the scholar networks [9, 17]. Next, supervised learning [8] classifies the papers by their authorship using labeled data, called training sets. Recently, topic modeling techniques [9, 20] are also applied to solve the homonymy problems. Topic modeling techniques like Latent Dirichlet Allocation (LDA) are applied to the papers' full texts, abstracts, and author names and cluster papers having similar distributions with high accuracy. All those approaches solve the problem under different assumptions. Some assume that we have prior information such as the number of homonyms (the number of people with ambiguous names) or a training set for each author. Other approaches require rich and clear information, such as e-mail addresses, affiliations, references, and publication venues.

Recently, an unsupervised method to generate training sets is proposed [7]. They group papers with similar author names and venues, and use those groups of papers as training sets in a rule extraction method. This study has the most similar goal to ours: resolving the homonyms without labeled data, parameters unknown prior, and human intervention. However, they also request publication venue as one of features and require heavy computation owing to complexity of rule extraction method used.

Our unsupervised method adopts density-based clusterings to group papers with similar coauthorship. Density-based clustering is an efficient approach in grouping data, if data can be represented as coordinates in a vector space. Among the various density-based clustering techniques, DBSCAN [5] is an well-known, easy and efficient approach that does not require the number of clusters or the cluster distribution as input. Following the fame of DBSCAN, many variations have appeared. Among them, one kind focuses on combining the correlation analysis into density-based clustering to increase the clustering accuracy. Their intuition is that although we do not know the distributions clusters follow a priori, we can guess the shape of a cluster to which a point belongs to by inspecting the locations of the points' neighbors. From this idea, 4C [3] improves DBSCAN by changing weights of directions when computing the distance between two points. It computes a covariance matrix of each point's neighbors and analyze it to guess the shape of the cluster it belongs to. It tunes the weights of directions so that less useful ones to explain the shape of the cluster are penalized more heavily. However, traditional density-based clustering or correlation clustering is not designed for dimension-reduced vector space. On reduced dimensions, some points are lumped near the origin even though they are not similar in terms of features. Thus, our distance measure is designed to take both the correlation analysis and the issue caused by dimension reduction into consideration. Details of correlation distance, covariance matrix analysis, and our distance measure are explained in subsection 3.5.

On a different view other than SDLs, we find another area of name entity disambiguation from Wikipedia. There are many approaches solving the name disambiguation problems in Wikipedia [4, 10]. They typically use various and rich information, such as main text, category, or link structure, than the case of online scholar database whose information is incomplete and error-prone.

3. ALGORITHM

In this section, we propose our approach to resolve the homonymy problem. Before explaining details of the approach, let us introduce notations used in this paper.

3.1 Notations and Problem Definitio

From this point on, author names appearing in the papers are represented as $u = (l, f)$, the combination of the last name and the initial of the first name. We represent an individual paper as p and define authors of p as:

$$names(p) = \{u \mid u \text{ is an author name of } p\} \quad (1)$$

For each target name, u_t, we are given a set of n papers, $\mathbb{P}$, where every paper has an author with the name, u_t. Let us define $names(\mathbb{P})$ as the list of unique names appearing in $\mathbb{P}$ except u_t, where $|names(\mathbb{P})| = m$. Let u_j denote the jth name in $names(\mathbb{P})$.

Based on these notions, we define our problem formally as below:

Given a target name, u_t and a set of n papers, $\mathbb{P}$, our method outputs $\mathbb{P}_1, \ldots, \mathbb{P}_r$, where there are r presumedly unique authors with the same name u_t, and $\mathbb{P}_1, \ldots, \mathbb{P}_r$ are the papers written by each of the r authors. The goal is to achieve accurate clusters with regard to ground truth and accuracy of the algorithm is evaluated based on the level of purity and the degree of fragmentation.

3.2 Algorithm Design

Our approach is based on the assumption that coauthors are likely to write more than one paper together. By dis-

covering and using sets of author names which appear frequently together in papers, we divide papers by their authorship.

Our method represents each paper as a point in the vector space and distances between points are determined by their coauthor similarity. Based on this coordinates of papers, we cluster papers and each cluster means publications written by the same person.

In our algorithm, one round consists of four phases and roles of them are explained below. First phase is assigned to compose a document-author matrix of $\mathbb{P}$ and decide coordinates of the papers. Second phase reduces dimension of the matrix so that we can find groups of coauthor names frequently appearing together and improve computational efficiency while valuable information remains. Third one is clustering step based on relative correlation distance. The last is paper summarization phase. Papers belonging to the same cluster are considered as written by the same author and represented as one point in next round. Our algorithm achieves the goal by repeating rounds until there is no grouped papers at that round. Details of each round will be explained in following subsections and overall structure of the algorithm is described in *Algorithm* 1.

Algorithm 1 Pseudo Code of the Proposed Method

```
1: procedure DISAMBIGUATION(Data, ϵ, κ, ν τ)
2:     Marks ← unique index for every paper
3:     while True do
4:         D ← matrixCompose(Data)
5:         reducedD ← dimReduce(D, τ)
6:         Marks ← relCorCluster(reducedD, ϵ, κ, ν)
7:         Merge papers marked by the same index
           as one summary vector
8:         if There is no merge then
9:             break
10:        end if
11:    end while
12:    return Marks
13: end procedure
```

3.3 Matrix Composition

Our algorithm resolves homonymy problems on the vector space. Each paper is represented as one point in the space and papers having similar coauthors are located closely by each other. For this purpose, we construct a $n \times m$ matrix D that is composed of m-feature vectors of n papers based on below weighting scheme:

$$D_{ij} = \begin{cases} w_{ij} & \text{if } u_j \in names(\mathbb{P}_i) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where each row (or vector) corresponds to individual paper and each column corresponds to a unique name appearing in the paper set, $\mathbb{P}$.

When we represent each paper as m dimensional vector, we should consider an weight of each author name in the paper so that distance between points is inversely proportional to their author name similarity. If we assign the same constant weight for each author and paper, Euclidean distances between papers having no common author become larger and larger as the number of authors increases. As a result, papers written by many authors are penalized so seriously. This does not make sense. Moreover, weighting the same constant value to all name occurrences can disturb finding papers having similar coauthors. As a toy example, let us consider three papers as below:

Paper 1: A. Kim, C. Chen
Paper 2: A. Kim, M. Brown
Paper 3: A. Kim, C. Chen, A. Kumar, M. Jones, F. Silva

Even though Paper 2 has no common author name with Paper 1 except target name, A. Kim, Paper 2 is closer to Paper 1 than Paper 3 in terms of Euclidean distance.

Considering these points, we design weighting scheme defined as:

$$w_{ij} = \frac{1}{\sqrt{\mid names(\mathbb{P}_i) \mid}} \quad (3)$$

It normalizes maximum distance between any pair of papers to 2 regardless of the number of authors for the papers. As the number of author names belonging to both papers increases, distance between them is closer until it reaches 0 when two papers show exactly the same coauthors. Moreover, distance between two papers becomes smaller not only when they have many common author names but also when the number of one paper's authors are similar to that of another. This weighting scheme gets rid of the penalty imposed on papers written by many authors and lets the number of authors be one factor to define distance between papers.

3.4 Dimension Reduction

Since m, the number of unique names in a paper set, is usually large number, the original data matrix, D, itself is not suitable for matrix decomposition computation. Moreover, dimensions which are less informative to identify ownership and correlation do harm than good for clustering accuracy. For accuracy and executional performance, we need some way to reduce dimension of the data without significant information loss.

Among various techniques, we adopt SVD (Singular Vector Decomposition) as the dimension reduction method. SVD is a mathematical technique similar to eigenvector decomposition, frequently used in text mining to identify relationships or patterns among documents, terms, and concepts, given document corpus. In our problem, terms and concepts are substituted by author names and groups with author names that co-appear frequently. SVD decomposes a $n \times m$ document-author matrix, D, into three matrices as below :

$$D = USV^T \quad (4)$$

where U is a $n \times n$ matrix whose columns are orthonormal eigenvectors of DD^T, V is a $m \times m$ matrix whose columns are orthonormal eigenvectors of D^TD, and S is a $n \times m$ diagonal matrix whose diagonal elements are singular values which are equal to square rooted eigenvalues of DD^T or D^TD, sorted in a decreasing order. In our case, each of U, V, and S can be interpreted as the document-to-coauthor group similarity matrix, the author-to-coauthor group similarity matrix, and the coauthor group strength matrix.

To represent a document-to-coauthor group similarity considering group strength, we discard a matrix V and use only U and S. Reconstructed document-to-coauthor group similarity matrix is :

$$D' = US \tag{5}$$

Even though we gain a document-to-coauthor group similarity matrix, its dimension, $n \times m$, is still the same as the original matrix, D. To reduce the number of dimensions, we borrow the concept of truncation from truncated SVD. Instead of using a full matrix, truncated SVD uses first l columns of U and first l rows and columns of S and discards all other dimensions. Truncated matrix is an optimal l-rank approximation of the original matrix satisfying :

$$D_l = U_l S_l V_l^T = \min_{X:rank(X)=l} \|D - X\| \tag{6}$$

where $\|A\|$ denotes the Frobenius norm of A.

To handle various data, we do not fix the number of reduced dimensions. Rather, we use the first l dimensions, which can explain at least τ percent of variance in original matrix D, defined as :

$$l = \min_{d \in 1,\ldots,m} \left\{ d \mid \frac{\sum_{i=1}^{d} s_i^2}{\sum_{i=1}^{m} s_i^2} \geq \tau \right\} \tag{7}$$

where s_i is the ith singular value of a matrix, S.

Finally, the $n \times l$ dimension document-to-coauthor group similarity matrix, D'_l, is defined and used in next phases.

$$D'_l = U_l S_l \tag{8}$$

3.5 Relative Correlation Distance Clustering

While the dimension reduction step makes points having large values on remaining dimensions clear in lower dimensions, other points are lumped near origin and not distinguished well. Thus, traditional density-based clustering or correlation clustering, such as DBSCAN or 4C, not considering positions of points, group all points near origin as one cluster. As papers written by different authors are mixed in the cluster, it lowers accuracy of the result seriously.

To handle this matter, we design a new distance measure based on the correlation distance measure defined in 4C. In our distance measure, we consider that how much distance can be reduced considering arrangements of neighbors and similarity of the arrangements.

To handle various shape and density of the points, we collect k-nearest neighbors of each point and compose covariance matrix of neighbors at first. k is set to $\nu \times$ number of points in current round to handle different number of points in each round.

$$\Sigma_{N_p} = \sum_{x \in kNearest(p)} (x - \bar{x})(x - \bar{x})^T \tag{9}$$

$$\Sigma_{N_p} = V_{N_p} E_{N_p} V_{N_p}^T \tag{10}$$

After, what we do is eigenvector decomposition so that major correlations can be analyzed. Among l eigenvectors, we emphasis 'strong' eigenvectors so that the distance between two points laying on the direction of strong eigenvectors becomes smaller than that of points laying on weak eigenvector directions. We divide eigenvectors into strong ones and weak ones based on their eigenvalues. Eigenvectors whose eigenvalues are greater or equal to averaged eigenvalue are defined as strong eigenvectors while the others are weak ones. Modified eigenvalue matrix, $\hat{E}_{N_p}$, is a diagonal matrix whose diagonal elements are defined as :

$$\hat{e}_i = \begin{cases} 1 & \text{if } e_i \geq \bar{e} \\ \kappa & \text{otherwise} \end{cases} \tag{11}$$

Since diagonal values of eigenvalue matrix represent weights of eigenvectors, the modification intends assigning κ times less weight on strong eigenvectors than weak eigenvectors. Based on this modified eigenvalue matrix, we reconstruct modified covariance matrix and use it to measure correlation distance between two points.

$$\hat{\Sigma}_{N_p} = V_{N_p} \hat{E}_{N_p} V_{N_p}^T \tag{12}$$

$$corDist_p(p, q) = \sqrt{(p - q) \cdot \hat{\Sigma}_{N_p} \cdot (p - q)^T} \tag{13}$$

This measure, called correlation distance, is devised by Bohm, *et al.* [3] and can quantify how two points are close each other considering arrangement of their neighbors.

However, a distance, $(p-q)$, between any two points lumped near origin on the reduced space is very small so their correlation distance is also small regardless of the covariance term. As a result, a chunk of points near origin forms a huge cluster even though they are written by different people. To fix this problem, we define one more modified covariance matrix and another measure, called relative correlation distance.

$$\hat{\Sigma}'_{N_p} = V_{N_p}(\kappa \times I) V_{N_p}^T \tag{14}$$

$$relCorDist_p(p, q) = \frac{\sqrt{(p - q) \cdot \hat{\Sigma}_{N_p} \cdot (p - q)^T}}{\sqrt{(p - q) \cdot \hat{\Sigma}'_{N_p} \cdot (p - q)^T}} \tag{15}$$

In this measure, we measure distance based on not only correlation with their neighbors but also how much distance can be reduced using correlation based measure. Even though a distance, $(p - q)$, is very small because they are lumped together, they will be assigned large distance value unless they show considerable reduction of distance when considering arrangement of neighbors. To prevent division by zero problem, the relative correlation distance between two points is set to infinity when value of the denominator is zero.

To employ DBSCAN algorithm, distance measure must be symmetric. Otherwise, results become nondeterministic and are different seriously for each trial. We define symmetric relative correlation distance between two points as :

$$relCorDist(p, q) = (1 - eigSim(p, q)) \times \max(relCorDist_p(p, q), relCorDist_q(p, q)) \tag{16}$$

$eigSim(p, q)$ is maximum cosine similarity between two sets of strong eigenvectors from two points' covariance matrices. We used this extra term to stress on similarity between two neighbors' arrangements. Overall process of relative correlation distance clustering is described in *Algorithm* 2. Except that we use relative correlation distance instead of Euclidean distance and do not use a parameter, μ, our algorithm is actually the same as DBSCAN.

Algorithm 2 Pseudo Code of the relCorCluster

procedure $relCorCluster(reducedD,\ \kappa,\ \nu,\ \epsilon)$
  $n \leftarrow numRow(reducedD)$
  $k \leftarrow n \times \nu$
  for $i = 1$ to n **do**
    $N_i \leftarrow k - Neighbors(i, reducedD, k)$
    $\Sigma_{N_i} \leftarrow covMatrix(N_i)$
    $V_i, E_i \leftarrow eigenDecomp(\Sigma_{N_i})$
    $\hat{\Sigma}_{N_i} \leftarrow V_i \hat{E}_i V_i^T$
    $\hat{\Sigma}'_{N_i} \leftarrow V_i(\kappa \cdot I)V_i^T$
  end for
  $unclassified \leftarrow \{1, .., n\}$
  while $unclassified$ is not empty **do**
    $i \leftarrow pop(unclassified)$
    $relCorReach \leftarrow relCorNeighbors(i, \epsilon)$
    for $c \in relCorReach$ **do**
      add $relCorNeighbors(c, \epsilon)$ to $relCorReach$
    end for
    Remove $relCorReach$ from $unclassified$
    Mark $c \in relCorReach$ using the same unique cluster ID
  end while
  return Marks
end procedure

3.6 Paper Set Summarization

As stated previously, the dimension reduction methods make points with large values on reduced dimensions remarkable while others become undistinguished each other. It is exactly what we intend and good for precise clustering. However, after remarkable points are clustered, we should consider how to handle papers not distinguished on the reduced dimensions. They are usually focused on very small region so we have no clue to classify them under current l-dimensional coordinates system.

To handle this problem, we introduce the concept of summary vector. For every cluster of papers, T, we summarize all papers in T as one vector under below weight scheme.

$$w_j = \frac{\sqrt{\sum_{p \in T} \mathbb{I}(u_j \in names(p))}}{\sqrt{\sum_{p \in T} \mid names(p) \mid}} \tag{17}$$

Since there can be names appearing more frequently in T, assigning the same weight to all names appearing in T does not make sense. We set that names gain more weight as they appear more in the cluster.

By grouping all papers in T as one summary vector, amount of data in T is compressed to that of single paper. Thus, in next round, the dimension reduction phase can reveal some of relationships among points having large values on currently discarded dimensions. In other words, some papers previously lumped get chance to be distinguished at next dimension reduction phase and we can cluster some of them precisely.

4. EVALUATION

4.1 Data Set

Evaluation is done on the data set used in work of Ferreira [7]. Originally, this data set is created and used by Han [8]. Due to noise in the original data, the data set is updated by Ferreira. The data set is composed of 11 most common names' sets of papers. For each target name, u_t, there are n papers written by an author named u_t. And there are m unique names appearing in those n papers except u_t. All information is collected from DBLP, one of well-defined online scholarly digital libraries, and ground truth data are hand-labeled so that we can check performance measure. Detailed statistics about the data are described in Table 1.

TargetName(u_t)	#Paper(n)	#Name(m)	#Homonyms
A. Gupta	576	487	26
A. Kumar	243	221	14
C. Chen	798	671	60
D. Johnson	368	293	15
J. Martin	112	134	16
J. Robinson	171	175	12
J. Smith	904	868	29
K. Tanaka	280	205	10
M. Brown	153	128	13
M. Jones	260	246	13
M. Miller	405	333	12

Table 1: Statistics about the Data Set

4.2 Performance Metrics

For evaluation of our disambiguation method, we employ three performance measures, frequently used in name disambiguation problem. ACP (Average Cluster Purity), AAP (Average Author Purity), and K Metric are explained in [1, 13, 19] and evaluate how similar clusters from the method are to ground truth clusters.

- **ACP (Average Cluster Purity)**

  ACP evaluates how pure clusters from the method are with regard to ground truth. As each cluster from methods contains more papers from the same author, ACP gains bigger value. When every cluster from the method contains only papers from the same author, it will be one. Formally, ACP is defined as follow :

$$ACP = \frac{1}{n} \sum_{i=1}^{c} \sum_{j=1}^{t} \frac{n_{ij}^2}{n_i}$$

  where c is the number of clusters from the method, t is the number of clusters in ground truth, n_{ij} is the number of papers belonging to both ith cluster from the method and jth cluster of ground truth, and n_i is the number of papers belonging to ith cluster from the method.

- **AAP (Average Author Purity)**

  AAP evaluates how fragmented clusters from the method are with regard to ground truth. As more papers, written by the same author, belongs to the same cluster after applying the method to data, AAP gains bigger value. When each set of papers from the same author is contained in only one cluster, it will be one. AAP is defined as follow :

$$AAP = \frac{1}{n} \sum_{j=1}^{t} \sum_{i=1}^{c} \frac{n_{ij}^2}{n_j}$$

where n_j is the number of papers belonging to jth cluster in ground truth.

- **K Metric**

  AAP and ACP evaluate performance of disambiguation in terms of purity and fragmentation. However, using only one of them as performance measure is not good idea. In the view point of AAP, both ground truth clusters and one huge cluster, containing all papers, get the best score. On the other hand, n singleton clusters, representing each of n papers, is scored 1 in terms of ACP. Thus, we need another balanced performance measure considering both scores. K metric is defined as a geometric mean of ACP, AAP and will be 1 when both AAP and ACP scores are 1. Equation of K metric is defined as:

$$K = \sqrt{AAP \times ACP}$$

4.3 Comparison with Other Methods

In order to show validity of proposed method, we apply our algorithm and five other methods to the data set and compare performance scores.

In our algorithm, we use 4 parameters, ϵ, κ, ν, and τ. κ is set to 50 as recommended by the work of Bohm [3] where define original correlation distance measure, equation 13. For remaining three parameters, we do grid search and find that $\epsilon = 0.2$, $\tau = 0.2$, and $\nu = 0.2$ gives the best performance over our data set. Optimal parameter estimation method is currently out of scope in this paper but should be handled in future.

Parameters of other methods are also found by grid search and details of five other methods are explained below.

- **DBSCAN**

  Original DBSCAN algorithm is applied to $n \times m$ dimensional the data matrix, D. Values of parameters are $\epsilon = 0.5$, $\mu = 2$.

- **4C**

  We collect the result of correlation clustering method, 4C, applied to $n \times m$ dimensional the data matrix, D, Values of parameters are $\epsilon = 0.95$, $\mu = 2$, $\delta = 0.05$, $\lambda = 5$.

- **DBSCAN on low dimension**

  At first, we reduce the dimension of original data matrix, D, to $n \times l$ using SVD-based dimension reduction and use the reduced matrix, D'_l. We use only first l dimensions explaining 20% of variance in D. (In other words, $\tau = 0.2$) After dimension reduction step, we apply the DBSCAN algorithm and collect the performance scores. Parameter setting is that $\epsilon = 0.25$, $\mu = 2$.

- **4C on low dimension**

  Similarly as DBSCAN on low dimension, we apply 4C algorithm on reduced data matrix, D'_l, and check performance. Dimensions explaining 20% of variance in D remain and are used in algorithm. Parameters, ϵ, μ, δ, λ are set to 0.1, 2, 0.05, 5 each.

- **k-means Clustering**

  For this method, we construct a data matrix differently. Instead of using normalized weights defined previously, we use constant weights to every name occurrence. And we do not whiten data matrix on a per feature basis. It is because this configuration results in better performance. Also, we give the number of homonyms as the number of clusters. We run the clustering 10 times for each target name and present average performance score.

Scores for 6 different methods are described in Table 2. Without dimension reduction phase, papers are not grouped well unless they show extremely high similarity on their coauthor names. Thus, clusters from these methods are usually consist of very small number of papers or become singleton clusters. As a result, DBSCAN and 4C without dimension reduction result high ACP, but very low AAP score. Since papers from the same authors are too seriously fragmented, it is not a good disambiguation approach.

In contrast, DBSCAN and 4C applied after dimension reduction phase achieve high AAP score. That is because points locating near origin are clustered together and form huge mass. However, as papers from different authors are mixed in the mass, it hurts ACP score severely. Thus, it is also not a good solution.

Our proposed approach achieves better AAP score because it reduces dimension of data but do not lose ACP seriously due to our relative correlation distance measure. In terms of K metric, proposed method reaches the highest score. Though there is much room resolving fragmentation, it is a better solution than other four methods. When an user provides their name, the system will provide bunches of papers presumedly written by the same author. The user will then be able to click on some subsets of bunches and confirm the bunches are their own papers. Due to its high AAP score, clusters from our approach can also be used as training sets of other supervised algorithms.

Moreover, our method outperforms k-means clustering even though we give the number of homonyms as input for k-means clustering. Its AAP, ACP, and K metric remain near 0.5 for all paper set, which is not outstanding. This shows that papers are hard to be grouped by their authorship without proper weighting and clustering strategy.

4.4 Result Interpretation

One possible and interesting question is why performance scores, especially AAP scores, are so different for each target name data. As we can see in Table 2, our method disambiguates publications very well for A. Gupta and M. Miller while it shows less effectiveness for A. Kumar and M. Jones. After inspecting each data set, we reach the promising explanation for these differences.

At first, some publications cannot be distinguished by our method as the method runs based only on coauthor information. Those are publications written by only one author, named u_t. Since we defined $names(\mathbb{P})$ as a list of unique names appearing in $\mathbb{P}$ except u_t, such single-authored papers has no evidence to distinguish themselves from the others. A. Kumar and M. Jones data set contain 52, and 72 publications written by single author for each and it is about 21 and 27% of the size of data set. On contrary, only 4 and 2% of publications in A. Gupta and M. Miller data sets

Method	Proposed			DBSCAN			4C			LowDimDBSCAN			LowDim4C			k-means		
	K	ACP	AAP	K	ACP	AAP	K	ACP	AAP	K	ACP	AAP	K	ACP	AAP	K	ACP	AAP
A. Gupta	0.641	0.956	0.430	0.284	0.997	0.081	0.391	0.991	0.154	0.459	0.250	0.842	0.500	0.398	0.628	0.481	0.541	0.428
A. Kumar	0.456	0.985	0.211	0.281	1.000	0.079	0.362	0.990	0.133	0.520	0.414	0.652	0.505	0.416	0.611	0.483	0.484	0.485
C. Chen	0.490	0.567	0.435	0.358	0.985	0.130	0.430	0.948	0.195	0.411	0.218	0.775	0.464	0.364	0.591	0.436	0.437	0.436
D. Johnson	0.531	0.996	0.283	0.258	1.000	0.067	0.306	1.000	0.094	0.513	0.462	0.570	0.497	0.470	0.527	0.480	0.496	0.465
J. Martin	0.615	1.000	0.378	0.430	1.000	0.185	0.560	1.000	0.314	0.568	0.422	0.764	0.563	0.422	0.751	0.590	0.552	0.631
J. Robinson	0.546	1.000	0.299	0.362	1.000	0.131	0.477	1.000	0.228	0.562	0.411	0.768	0.528	0.411	0.679	0.565	0.502	0.636
J. Smith	0.559	0.932	0.335	0.220	0.994	0.049	0.559	0.932	0.335	0.444	0.287	0.688	0.455	0.417	0.497	0.409	0.376	0.445
K. Tanaka	0.596	1.000	0.355	0.278	1.000	0.077	0.391	1.000	0.153	0.623	0.513	0.756	0.577	0.535	0.623	0.580	0.629	0.536
M. Brown	0.562	0.947	0.333	0.392	1.000	0.153	0.512	1.000	0.262	0.488	0.366	0.649	0.485	0.366	0.642	0.525	0.508	0.545
M. Jones	0.536	1.000	0.287	0.267	1.000	0.071	0.360	1.000	0.130	0.520	0.413	0.655	0.489	0.446	0.537	0.504	0.447	0.569
M. Miller	0.809	0.977	0.670	0.201	1.000	0.041	0.242	1.000	0.059	0.564	0.440	0.724	0.474	0.607	0.371	0.487	0.621	0.383
Average	0.577	0.942	0.365	0.303	0.998	0.097	0.417	0.987	0.187	0.516	0.381	0.713	0.503	0.441	0.587	0.504	0.508	0.505

Table 2: Performance Scores for the Methods

are written solely. The undistinguishable publications leave singleton clusters in our results and eventually harm AAP score. Second one is about amount of information. We check average number of authors per paper, except authors named u_t. Roughly, these values correspond to average amount of evidence for their identity contained in each paper. For A. Gupta and M. Miller data sets, plentiful information exists (2.359, 2.612 which are top two values among 11 data sets) so that we can group large number of papers with confidence. On the other hands, A. Kumar and M. Jones are the most two scarcest data sets in terms of coauthors information (1.452, 1.546). Lack of information is likely to cause fragmentation of one's publication records and result low AAP score.

This explanation implies that our method should be able to utilize other common information of papers to be better disambiguation method even if it shows promising results. Currently, we set merging another common information, title, to our method as one future plan. Details are explained in Section 5.

5. CONCLUSIONS

Name disambiguation problem means difficulties to identify individuals when named entities share the common name. Among various areas where name ambiguities becomes obstacles, in SDLs (Scholarly Digital Libraries), problems are much more serious due to its compressive name presentation and many people sharing common last names in Asia and Middle. Actually, this is not a new problem and one of the long-studied problems. However, existing name disambiguation methods have limitation in that they require a priori inputs such as labeled training set, the number of homonyms or topics, or rich and clear information about papers. In this paper, we present one possible way to overcome the limitation based on dimension reduction and correlation clustering. Our experimental evaluation confirms that proposed method can more effectively classify papers by their authorship only using their coauthor names than other four density-based clustering techniques and k-means clustering method. Even though there are much room for improvement, we believe that this work is one step to complete name disambiguation in future. Finally, we end up the paper with possible future plans to make better disambiguation method.

- **Expansion to Other Common Information:** In this work, we do not use any information of papers other than author lists. And it makes sense in that most SDLs have difficulty with collecting various information of papers such as e-mail address, venue, and other things in noiseless and consistent form. However, there are other common and less-noisy information such as title. You may notice that if we can use such information as evident to decide whether two papers are written by the same author or not, it can lessen fragmentation of scholarship records and give better results. One possible option is using higher-order decomposition methods such as tensor decomposition for multi-dimensional array in place of SVD. Since tensor decomposition can detect clusters among multiple dimensions, we can include other common information such as title in disambiguation process. However, more information means higher dimensions and should be approached in a cautious way. As curse of dimensionality means, higher dimensional data can cause sparsity in the space. And the sparsity becomes problematic for clustering as most clusters become very small or singleton.

- **Optimal Parameter Estimation:** As stated earlier, we do not have any parameter estimation method at now. Even though our method works well with the data set of 11 ambiguous names using the fixed parameter setting, we need some kind of dynamic parameter selection method to cope with various nature of other data.

- **Addition of the Probabilistic Inference:** Our proposed method can determine only which cluster each paper belongs to. If we can infer the probability that papers belong to their cluster, it can improve user experience of SDLs. Users can pick out wrongly-sorted papers in a cluster easily by just seeing "This paper is written by ... with 20% of certainty".

6. ACKNOWLEDGEMENTS

We specially thank Anderson Ferreira for their standard labeled data sets and also anonymous reviewers for their help and invaluable comments. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No.2012033242)

7. REFERENCES

[1] L. I. Ajmera J, Bourlard H. Improved unknown-multiple speaker clustering using hmm. *Techinal report, IDIAP*, 2002.

[2] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. In J. Ghosh,

D. Lambert, D. B. Skillicorn, and J. Srivastava, editors, *SDM*. SIAM, 2006.

[3] C. Böhm, K. Kailing, P. Kröger, and A. Zimek. Computing clusters of correlation connected objects. In *SIGMOD Conference*, pages 455–466, 2004.

[4] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *In Proc. 2007 Joint Conference on EMNLP and CNLL*, pages 708–716, 2007.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. M. Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

[6] V. I. Fegley B. D., Torvik. On the effect of name ambiguity on measures of large-scale co-authorship networks. *International Conference on Network Science*, 2012.

[7] A. A. Ferreira, A. Veloso, M. A. Gonçalves, and A. H. F. Laender. Effective self-training author name disambiguation in scholarly digital libraries. In J. Hunter, C. Lagoze, C. L. Giles, and Y.-F. Li, editors, *JCDL*, pages 39–48. ACM, 2010.

[8] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *JCDL'04: Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries*, Mining and disambiguating names, pages 296–305, 2004.

[9] H. Han, H. Zha, and C. L. Giles. Name disambiguation in author citations using a K-way spectral clustering method. In *Proceedings of the Fifth ACM/IEEE-CS Joint Conference on Digital Libraries*, 2005.

[10] X. Han and J. Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 215–224, New York, NY, USA, 2009. ACM.

[11] X. Han and J. Zhao. Web personal name disambiguation based on reference entity tables mined from the web. In *Proceedings of the eleventh international workshop on Web information and data management*, WIDM '09, pages 75–82, New York, NY, USA, 2009. ACM.

[12] I.-S. Kang, S.-H. Na, S. Lee, H. Jung, P. Kim, W.-K. Sung, and J.-H. Lee. On co-authorship for author disambiguation. *Inf. Process. Manage*, 45(1):84–97, 2009.

[13] I. Lapidot. Self-organizing-maps with BIC for speaker clustering. Idiap-RR Idiap-RR-60-2002, IDIAP, Martigny, Switzerland, 0 2002.

[14] J. Li, J. Tang, J. Zhang, Q. Luo, Y. Liu, and M. Hong. Arnetminer: expertise oriented search using social networks. *Frontiers of Computer Science in China*, 2(1):94–105, 2008.

[15] A. Mazloumian, Y.-H. Eom, D. Helbing, S. Lozano, and S. Fortunato. How citation boosts promote scientific paradigm shifts and nobel prizes. *PLoS ONE*, 6(5):e18975, 05 2011.

[16] J. Moody. The structure of a social science collaboration network: Disciplinary cohesion from 1963 to 1999. *American Sociological Review*, 69(2):213–238, 2004.

[17] B.-W. On and D. Lee. Scalable name disambiguation using multi-level graph partition. In *SDM*. SIAM, 2007.

[18] N. R. Smalheiser and V. I. Torvik. Author name disambiguation. *Annual Review of Information Science and Technology*, 43(1):1–43, 2009.

[19] S. M. Solomonov A, Mielke A. Clustering speakers by their voices. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2:757–760, 1998.

[20] Y. Song, J. Huang, I. G. Councill, J. Li, and C. L. Giles. Efficient topic-based unsupervised name disambiguation. In E. M. Rasmussen, R. R. Larson, E. G. Toms, and S. Sugimoto, editors, *JCDL*, pages 342–351. ACM, 2007.

[21] J. Tang, D. Zhang, and L. Yao. Social network extraction of academic researchers. In *ICDM*, pages 292–301. IEEE Computer Society, 2007.