Mobile Access Record Resolution on Large-Scale Identifier-Linkage Graphs

SHEN Xin*
Zhejiang University
Hangzhou, Zhejiang, China
sxstar@zju.edu.cn

Hongxia Yang
Alibaba Group
Hangzhou, Zhejiang, China
yang.yhx@alibaba-inc.com

Weizhao Xian
Zhejiang University
Hangzhou, Zhejiang, China
3130000312@zju.edu.cn

Martin Ester
Simon Fraser University
Burnaby, B.C., Canada
ester@sfu.ca

Jiajun Bu
Zhejiang University
Hangzhou, Zhejiang, China
bjj@zju.edu.cn

Zhongyao Wang
Alibaba Group
Hangzhou, Zhejiang, China
zhongyao.wangzy@alibaba-inc.com

Can Wang†
Zhejiang University
Hangzhou, Zhejiang, China
wcan@zju.edu.cn

ABSTRACT

The e-commerce era is witnessing a rapid increase of mobile Internet users. Major e-commerce companies nowadays see billions of mobile accesses every day. Hidden in these records are valuable user behavioral characteristics such as their shopping preferences and browsing patterns. And, to extract these knowledge from the huge dataset, we need to first link records to the corresponding mobile devices. This **M**obile **A**ccess **R**ecords **R**esolution (MARR) problem is confronted with two major challenges: (1) device identifiers and other attributes in access records might be missing or unreliable; (2) the dataset contains billions of access records from millions of devices. To the best of our knowledge, as a novel challenge industrial problem of mobile Internet, no existing method has been developed to resolve entities using mobile device identifiers in such a massive scale. To address these issues, we propose a **SP**arse **I**dentifier-linkage Graph (SPI-Graph) accompanied with the abundant mobile device profiling data to accurately match mobile access records to devices. Furthermore, two versions (unsupervised and semi-supervised) of **P**arallel **G**raph-based **R**ecord **R**esolution (PGRR) algorithm are developed to effectively exploit the advantages of the large-scale server clusters comprising of more than 1,000 computing nodes. We empirically show superior performances of PGRR algorithms in a very challenging and sparse real data set containing 5.28 million nodes and 31.06 million edges from 2.15 billion access records compared to other state-of-the-arts methodologies.

CCS CONCEPTS

• **Mathematics of computing → Graph algorithms**; • **Information systems → Entity resolution**; *Clustering*;

KEYWORDS

Mobile access record resolution; Scalable algorithms; Big data; Graph algorithms

ACM Reference Format:
SHEN Xin, Hongxia Yang, Weizhao Xian, Martin Ester, Jiajun Bu, Zhongyao Wang, and Can Wang. 2018. Mobile Access Record Resolution on Large-Scale Identifier-Linkage Graphs. In *KDD 2018: 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3219819.3219916

1 INTRODUCTION

The past few years have witnessed a great surge of mobile internet users. According to the newly released report by International Telecommunication Union (ITU), the number of subscribers has reached 7.74 billion by 2017, which has already exceeded the world population.[1] The report by CINIC (China Internet Network Information Center) also shows that by the end of June, 2017, there are 724 million Chinese users accessing the Web via smart phones, accounting for 96.3% of the national Internet population.[2] As mobile phones have overtaken desktop computers as the most widely used digital platform, characterizing mobile user preference and behavioral patterns from their access records becomes an important research topic. Compared with traditional weblogs, which mostly depend on cookies to track user behavior, mobile access records provide a clearer picture of internet users with various IDs in the access records. These IDs include International Mobile Equipment

*SHEN Xin would like to capitalize all letters of his family name.
†Can Wang is the corresponding author.

KDD 2018, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219916

[1] https://www.itu.int/en/ITU-D/Statistics/Pages/publications/mis2017.aspx
[2] http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201708/t20170803_69444.htm

Table 1: Statistics of *ID Shift* problems

	1-to-1	1-to-Many	1-to-2
$\{IMEI, IMSI\} \rightarrow \{UTDID\}$	5,159,128	29,189	-
$\{IMEI, UTDID\} \rightarrow \{IMSI\}$	2,111,394	440,800	134,266
$\{IMSI, UTDID\} \rightarrow \{IMEI\}$	5,121,022	26,150	-
$\{IMEI\} \rightarrow \{IMSI, UTDID\}$	5,714,894	458,234	-
$\{UTDID\} \rightarrow \{IMEI, IMSI\}$	5,401,795	-	173,709

Identity (IMEI), International Mobile Subscriber Identity (IMSI), and UserTrack Device IDentity (UTDID). IMEI is a unique identifier designed to identify a device. IMSI is designed to identify a user in a cellular network, which is stored in a SIM card. We can roughly regard IMEI and IMSI as identifiers for one's smartphone and mobile number respectively. UTDID, on the other hand, is quite different from these two hardware-based identifiers in that it is an identifier generated and used by Alibaba for device identification. With these IDs, access records are expected to be mapped to the corresponding mobile phones or apps, which in turn will generate high quality user profiles.

At the first glance, mapping an access record to the mobile phone or the app seems to be a trivial issue since IDs such as IMEI, IMSI and UTDID can be used to uniquely identify the device and app. However, data collected from practical applications is far from perfect, like missing attribute values, noisy IDs, and *ID shift* problems. There are various *ID shift* problems in which this one-to-one mapping between IDs and devices is no longer maintained, for instance:

(1) A Dual SIM Dual Standby phone will have 2 IMEIs and 2 IMSIs. Different IMEIs and IMSIs will be collected based on the preferred SIM card the user selected at the access time. The fifth row in Table. 1 shows possible cases for this situation.
(2) A device can get a new IMSI when a new SIM card is installed. The second row in Table. 1 shows that in the dataset, there exist more than 440 thousand cases of multiple IMSIs, for which new SIM cards are the main reason.
(3) A new UTDID will be issued after a system reinstallation. The first row in Table. 1 shows that in the dataset, there exist more than 29,000 cases of multiple UTDID, most of which are the results of system reinstallations.
(4) There are access records from emulators where all IDs are randomly or manually generated. The third row in Table. 1 shows some possible cases.
(5) A large number of knockoff mobile phones may share the same IMEI. The fourth row in Table. 1 shows this situation.

Fig. 1 illustrates examples of access records from Dual SIM Dual Standby phone and knockoff mobile phones. Each record is identified by an IDSET, i.e. a combination of IMEI, IMSI and UTDID. Record 1 and 3 are generated by the same Dual SIM Dual Standby phone. Record 1 and 3 come from a single device, as an example of Dual SIM Dual Standby. Record 4 and 5 come from two distinct devices although they share one IMEI, as an example of knockoff phones sharing the same IMEI.

Access records collected in real world applications will inevitably be affected by the above-mentioned *ID shift* problems. Table. 2 presents some simple statistics of the number of devices, IMEIs,

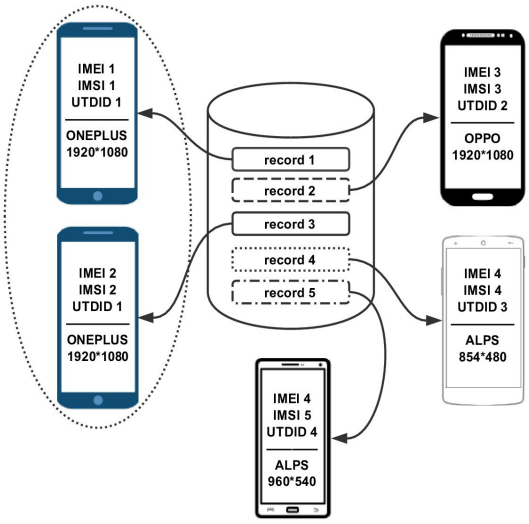


Figure 1: Examples of mobile access records

IMSIs, and UTDIDs in the Alibaba dataset. The dataset contains access records masked with MD5 hashing from about 1.76 million devices with 1.97 million distinct IMEIs, 5.01 million distinct IMSIs and 2.29 million distinct UTDIDs, as well as 5.28 million distinct IDSETs. The statistics show that no clear correspondence exists among these IDs. Consequently, a single ID in a mobile access record is generally an unreliable source for identifying the corresponding mobile phone, mainly due to the *ID shift* problems.

We observe that *ID shift* in one or two IDs in an access record might occur from time to time, but it is an extremely rare case that *ID shift* occurs in all the three IDs. Inspired by this observation, we use the combination of the three IDs (IMEI, IMSI, UTDID), which we call "IDSET", to reliably identify an access record from a specific mobile device. An example of an IDSET is given in Fig. 1, where each record is identified by the IDSET, i.e. a combination of IMEI, IMSI and UTDID. Based on the concept of IDSET, we introduce the **M**obile **A**ccess **R**ecords **R**esolution (MARR) problem as grouping the access records according to the accessing mobile devices. Two major challenges are confronted in MARR: (1) device identifiers and other attributes in access records might be missing or unreliable; (2) the dataset contains billions of access records from millions of devices. There are 5,276,424 distinct IDSETs in total in our dataset, which consists of 2,146,168,904 records. In practical scenarios, these numbers can be even larger.

One possible solution for MARR is entity resolution (ER), which relies on pairwise record comparisons to group the records. But existing ER methods do not scale to very large datasets that are common in industry. To address this issue, we consider constructing a sparse, attributed graph such that we need to compare only neighbors in that graph, and we use the node attributes to measure the similarity of node pairs (i.e. to compare node pairs). To summarize, in this paper we construct a **SP**arse **I**dentifier-linkage

Graph (SPI-Graph) for MARR by connecting the access records (identified by their IDSETs) sharing one or multiple IDs (IMEI, IMSI or UTDID). For instance, in Fig. 2 we connect all red records because of the co-occurring IMEI. To efficiently process the massive amount of the access records, we propose two versions of **P**arallel **G**raph-based **R**ecord **R**esolution (PGRR) algorithm, unsupervised and semi-supervised, on the SPI-Graph that can effectively exploit the power of the large-scale server clusters comprising of more than 1,000 computing nodes. Experimental results on a large-scale real-world dataset verify the effectiveness and efficiency of our two versions of the proposed algorithm.

We summarize the major contributions of this paper as follows:

(1) Mobile access record resolution, the problem of resolving mobile access records to their corresponding mobile device identifiers, is formally defined. MARR exists widely in the mobile Internet industry.
(2) We propose a graph partitioning-based approach based on so-called **SP**arse **I**dentifier-linkage Graphs.
(3) Two versions of **P**arallel **G**raph-based **R**ecord **R**esolution algorithm that scale to very large datasets are presented.
(4) We show results of our experimental evaluation on a very large industrial dataset which demonstrate the scalability of our method and its superiority compared to existing methods.

The rest of the paper is organized as follows. In Section 2, we give a brief review of related work. Section 3 presents our parallel algorithm. The experimental results are presented in Section 4. Finally, we summarize the paper and suggest directions for future work in Section 5.

2 RELATED WORK

Mining user patterns from web logs has drawn considerable research efforts as log files are believed to contain rich information about users. Many log mining algorithms have been developed for various purposes, including: user personalization [28, 33], predicting web page accesses [16], discovering sequential patterns [18], enhancing search results and finding interesting contents [26], etc. One of the major challenges in these web usage mining algorithms and applications is mapping web accesses records to the corresponding users. Although many useful attributes are collected in web log records such as IP address, time stamp, access request, etc., they can at best generate a rough mapping in most cases as none of these attributes can be used to uniquely identify users with reasonable confidence. This problem of identifying records in a data set is now generally known as entity resolution.

Entity resolution (ER) is the process of identifying and merging records representing the same real-world entity [4], which is a comprehensive task of extracting, matching, and resolving entity in structured and unstructured dataset [19]. Most of the existing ER models rely on pairwise comparisons between elements for resolving records and will thus incur high computational complexity. Some hash-based blocking models [6, 15, 27] improve algorithm efficiency by partitioning records into different blocks. However, hash-based models will not work in our case since there exist no definite mapping between attributes and blocks. Also, various supervised learning algorithms have been exploited to solve pairwise matching, including decision tree [14], support vector machines [7, 13], and ensembles of classifiers [11]. However, these algorithms are confronted with three major issues: (1) training data is expensive; (2) imbalanced classes problem, positive cases (pairs of records that match) are dominated by negative cases; (3) ambiguous data and missing attribute values make pairwise matching a challenging issue. To solve the training data problem, several unsupervised and semi-supervised algorithms have been proposed, for example, the latent Dirichlet model [5] and the generative model [34]. Meanwhile, crowdsourcing has been applied in ER recently [36, 37] to leverage the power of human computation in labeling the data. Other attempts to minimize the cost of training data include active learning methods, such as committee of classifiers [35] and maximal frequent itemsets [1]. Unsupervised learning methods such as hierarchical clustering based algorithms have also been proposed for large-scale entity resolution, e.g., collective relational clustering [5], pay-as-you-go approach [38]. However, clustering algorithms will usually be problematic when there exists a large number of clusters in datasets. In many ER applications, the number of entities might be huge. For instance, in our dataset there are 1.7 million devices. Existing clustering models will be incapable of dealing with such a huge number of entities.

Clustering Since labels may be difficult to obtain, more recent works in ER attempt unsupervised learning, or clustering techniques. Existing clustering algorithms include k-means [23], spectral clustering [29], Gaussian mixture model [39], DBSCAN [17], co-clustering [10], fuzzy clustering [25], etc. Some recent graph-based models are showing better promise in resolving large number of entities. For instance, Network Lasso [21] achieves efficient graph-based partition by generalizing the Group Lasso regularization to a network setting. Network Lasso can be solved by the Alternating Direction Method of Multipliers (ADMM) [9, 31]. Similar to network lasso, some convex clustering methods using ℓ_{21}-norm have been proposed as an alternative to k-means [12, 24, 32]. These methods differ from network lasso in that they do not require the graph to be fully connected.

3 THE MOBILE ACCESS RECORDS RESOLUTION PROBLEM

Each mobile access record contains an IDSET and several attributes describing the hardware and software of the accessing mobile device. These attributes include account id, device brand and model, screen resolution, etc.

The objective of the MARR problem is to identify the physical device for each access record, since each access record is generated by one specific mobile device. More precisely, we aim to group access records according to the device, which can be used to generate profiles for the device users. Considering the sheer size of the dataset and poor data quality mainly due to the *ID shift* problems, MARR is a highly challenging problem.

An intuitive approach is to group together records that share one or multiple IDs. For this purpose, we construct the SPI-Graph $G = (V, E)$, where V is the vertex set, and E is the edge set. Each IDSET is a vertex in the SPI-Graph. An edge between two IDSETs is created if they share at least one common ID. A naive solution is then to regard each connected component as a real world physical

Table 2: Statistics of five randomly picked subsets and the full Alibaba dataset

Dataset	# of Records	# of IDSETs (Vertex)	# of Edges	# of IMEIs	# of IMSIs	# of UTDIDs
Subset 1 (0.005%)	106,031	242	2,368	109	241	157
Subset 2 (0.007%)	141,047	253	2,373	118	250	164
Subset 3 (0.09%)	1,980,901	2,861	35,417	586	2,832	2,043
Subset 4 (0.2%)	3,674,313	5,094	212,568	1,094	4,847	3,904
Subset 5 (1.3%)	27,764,971	56,631	1,821,622	14,053	52,649	45,879
Entire Dataset	2,146,168,904	5,276,424	31,058,117	1,977,565	5,011,614	2,291,047

Table 3: Model notations and definitions

Notation	Definition
x_i	features of the node i
c_i	features of the cluster center of the node i
s_{ij}	similarity between node i and j
w_{ij}	similarity between 2 cluster centers of node i and j
p_i	the cluster number of the node i
f_{ij}	predict whether node i and j are in the same cluster
y_{ij}	label of f_{ij} from the input

Table 4: Statistics of the sparsity of Alibaba dataset

# of nodes ($\|V\|$)	5,276,424
# of edges ($\|E\|$)	31,058,117
# of connected components	772,536
Max # of neighbors of a node (M)	1,167
Avg # of neighbors of a node	11.77
Max # of nodes of a connected component (K)	1,625
Avg # of nodes of a connected component	6.83

device. But this solution will be plagued by the *ID shift* problems as access records from different devices might be placed in the same connected component.

To overcome this problem, we can consider graph-based clustering using the attributes contained in the access records. Many classical clustering algorithms such as kmeans++ can be used to cluster these IDSETs using their similarity on the attributes. However, most of the existing clustering algorithms require the knowledge of the number of clusters as input, which is unknown in our case. In addition, the performance of clustering algorithms will deteriorate as the number of clusters increase. In our dataset, the largest connected component in the SPI-Graph contains 2,203,326 IDSETs from 86,707 devices. This is a challenging situation for the existing clustering algorithms.

Table. 3 lists all the significant symbols and definitions used throughout this paper. We provide the following formal problem definition with the unsupervised version. We are given a graph $G = (V, E)$ and attributes of nodes $X = \{x_i | x_i \in \mathbb{R}^d, i \in V\}$, where V is the set of IDSETs, E is the set of the co-occurrences between IDSETs, and d is the number of dimensions of the feature vector of an IDSET. The task of MARR is to predict the linkage relationship $F = \{f_{ij} | f_{ij} \in \{0, 1\}, (i, j) \in E\}$, where $f_{ij} = 1$ indicates that we predict node i and j belonging to a same device, 0 otherwise.

When we observe some labels of edges, we are also given $Y = \{y_{ij} | y_{ij} \in \{0, 1\}, (i, j) \in E', E' \subset E\}$, where $y_{ij} = 1$ indicates that node i and j belong to a same device, 0 otherwise. This turns to the problem definition of the semi-supervised version.

4 A GRAPH PARTITIONING-BASED APPROACH

To motivate our graph-based approach, Fig. 2 shows a sample of SPI-Graph. Every node in the graph indicates the records labelled by one IDSET, which contains at most three identifiers, IMEI, IMSI, and UTDID. To solve missing attributes values problem, every record can contribute to the attributes of its own IDSET. Every edge in the graph indicates two nodes shared at least one common identifiers. Normally, each mobile device should have only one unique IDSET. So, most of nodes in the SPI-Graph should not be linked to too many neighbours, as the yellow nodes shown. The red nodes in the SPI-Graph construct a large strongly connected component, because they have one shared IMEI: $imei1$. This situation often caused by knockoff mobile phones. In a real-world dataset before data pre-processing, one large connected component in a SPI-Graph may have millions of nodes. The blue nodes in the SPI-Graph show a typical Dual SIM Dual Standby phone, which has two IMEIs and two IMSIs. A Dual SIM Dual Standby phone can present different pairs of one IMEI and one IMSI for different situations.

The Alibaba dataset is collected from practical applications. We delete obvious noisy IDs whose degrees exceed 10,000. Since an empirical study of the cleaned Alibaba dataset shows that all the three IDs in an IDSET are unlikely to be changed simultaneously, we have reason to believe that all the access records from one given device are placed in the same connected component of the SPI-Graph. Table. 4 reveals the high sparsity of the dataset, where the number of edges is less than 6 times the number of nodes. Note that not only the average number of neighbors, but also the maximum number of neighbors of a node and the maximum number of nodes in a connected component are very small compared to |V|. We leverage these properties of the SPI graph to scale our approach to very large datasets.

Our goal is to group IDSETs in a connected component according to their respective attribute values. Denote $c_i \in \mathbb{R}^d$ as attributes of the group center which node i belongs to. If two IDSETs have a same group center, they belong to a same physical device. We propose an unsupervised framework for graph partitioning using the following objective function:

$$\min_{c_i} \sum_{i \in V} g_1(x_i, c_i) + \lambda_1 \sum_{(i,j) \in E} g_2(c_i, c_j) + \lambda_2 \sum_{i \in V} \|c_i\|_2^2 \qquad (1)$$

where λ_1 and λ_2 are weighting parameters to balance the loss of the three components. We use $g_1()$ to penalize the distance between each vertex and the corresponding group center so as to make

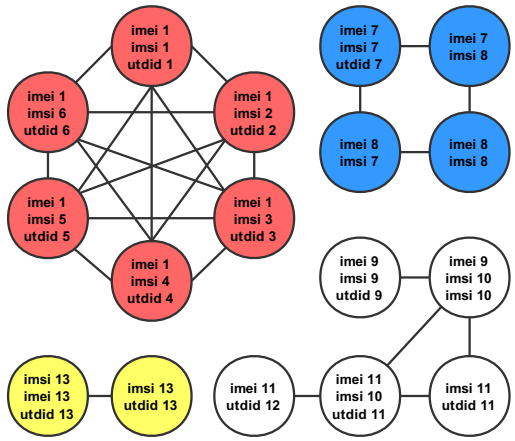


Figure 2: A sample of SPI-Graph

vertices in a group similar to each other. We use $g_2()$ to penalize the distance between two group centers so as to make two different groups dissimilar to each other. We also penalize the norm of the clusters centers to address the extremely large data dimensionality and sparseness.

In some cases, we can achieve a small part of labels for pairs of nodes in the SPI-Graph. We expand Eq. 1 and propose a semi-supervised framework with the objective function:

$$\min_{f_{ij},c_i} \sum_{(i,j)\in E'} \left\|f_{ij}-y_{ij}\right\|_2^2 + \gamma_1 \sum_{i\in V} g_1(x_i,c_i) + \gamma_2 \sum_{(i,j)\in E} g_2(c_i,c_j) + \gamma_3 \sum_{i\in V} \|c_i\|_2^2 \tag{2}$$

where f_{ij}s indicate whether node i and node j belong to one entity, y_{ij}s are the observable ground truths, and γ_is>0 are weighting parameters.

Now the MARR problem has been narrowed down to resolving entities in a connected component of the SPI-Graph. Following the above framework for graph-based partitioning, we use the ℓ_2-norm to implement $g_1()$ as the penalizing distance to keep vertices in a group close to its center. We use the ℓ_{21}-norm [30] to implement $g_2()$ as the penalizing distance to keep the two group centers far away from each other. While the rationale of using the ℓ_2-norm is clear here, the reason we use the $\ell 2,1$-norm is that it can shrink the distance of two similar nodes to exact zero, instead of closed to zero. Similar to the $\ell 2,1$-norm effect of group-based feature selection in a regression model, its effect on a graph-based model is in automatically deciding the number of groups or clusters in a connected component. We define the following objective function for unsupervised entity resolution on the SPI-Graph:

$$\min_{c_i} \sum_{i\in V} \|x_i-c_i\|_2^2 + \lambda_1 \sum_{(i,j)\in E} s_{ij} \left\|c_i-c_j\right\|_{2,1} + \lambda_2 \sum_{i\in V} \|c_i\|_2^2 \tag{3}$$

where $s_{ij}\in[0,1]$ indicates the similarity of two nodes, and can be calculated by the attributes of nodes.

From the solution to the above objective function, we can use the IDSET center c_i as a description of a real-world physical device and group all IDSETs in $D_i=\{IDSET_j : c_j=c_i\}$ to the corresponding device.

Similarly, the objective function for the semi-supervised version is:

$$\min_{f_{ij},c_i} \sum_{(i,j)\in E} \left\|f_{ij}-y_{ij}\right\|_2^2 + \gamma_1 \sum_{i\in V} \|x_i-c_i\|_2^2 + \gamma_2 \sum_{(i,j)\in E} s_{ij} \left\|c_i-c_j\right\|_{2,1} + \gamma_3 \sum_{i\in V} \|c_i\|_2^2 \tag{4}$$

To utilize given labels to improve the precision of MARR, we introduce p_i and s'_{ij}, where p_i indicates the cluster number of node i, and s'_{ij} is the alternative similarity between node i and j:

$$s'_{ij}=\begin{cases}1 & \text{if } p_i=p_j,\\ s_{ij} & \text{otherwise.}\end{cases} \tag{5}$$

IDSETs with the same cluster number are assigned to the same mobile device.

5 THE PARALLEL GRAPH-BASED RECORD RESOLUTION ALGORITHM

We expect our algorithm to efficiently process billions of records by running on a large-scale server clusters with more than 1,000 computing nodes. So, it is critical to exploit parallelism in the algorithm. Considering that the SPI-Graph is extremely sparse, $O(E)\approx O(V)$, we can investigate each edge in the SPI-Graph instead of an arbitrary pair of nodes. For this purpose, we design the PGRR algorithm based on the Alternating Direction Method of Multipliers (ADMM) [9, 31]. ADMM is a variant of the augmented Lagrangian scheme for distributed optimization on a large-scale data. It uses partial updates for the dual variables.

For the unsupervised model, to solve c_i, we introduce z_{ij}, where $z_{ij}=c_i$. We have the following loss function:

$$\begin{aligned} L(c,z) = &\sum_{i\in V} \|x_i-c_i\|_2^2 + \lambda_1 \sum_{(i,j)\in E} s_{ij} \left\|z_{ij}-z_{ji}\right\|_{2,1} \\ &+ \lambda_2 \sum_{i\in V} \|c_i\|_2^2 \\ &s.t. c_i - z_{ij} = 0 \end{aligned} \tag{6}$$

The augmented Lagrangian is:

$$\begin{aligned} L_\rho(c,z,y) = &\sum_{i\in V} \|x_i-c_i\|_2^2 + \lambda_1 \sum_{(i,j)\in E} s_{ij} \left\|z_{ij}-z_{ji}\right\|_{2,1} \\ &+ \lambda_2 \sum_{i\in V} \|c_i\|_2^2 + \sum_{(i,j)\in E} \left(y_{ij}^T(c_i-z_{ij}) + y_{ji}^T(c_j-z_{ji})\right) \\ &+ \sum_{(i,j)\in E} \rho/2(\|c_i-z_{ij}\|_2^2 + \|c_j-z_{ji}\|_2^2) \end{aligned} \tag{7}$$

Let $u = y/\rho$, we can rewrite it into the following scaled form:

$$\begin{aligned} L_\rho(c,z,u) = &\sum_{i\in V}\|x_i - c_i\|_2^2 + \lambda_1 \sum_{i,j\in E} s_{ij}\|z_{ij} - z_{ji}\|_{2,1} \\ &+ \lambda_2 \sum_{i\in V}\|c_i\|_2^2 \\ &+ \frac{\rho}{2}\sum_{(i,j)\in E}\left(\|c_i - z_{ij} + u_{ij}\|_2^2 + \|c_j - z_{ji} + u_{ji}\|_2^2\right) \\ &- \frac{\rho}{2}\sum_{(i,j)\in E}(\|u_{ij}\|_2^2 + \|u_{ji}\|_2^2) \end{aligned} \tag{8}$$

We can update c, z, u iteratively as follows:

$$\begin{aligned} c_i^{(k+1)} &= \arg\min_{c_i}\Big(\sum_{i\in V}\|x_i - c_i\|_2^2 + \lambda_2\sum_{i\in V}\|c_i\|_2^2 \\ &\quad + \sum_{j\in N(i)}\frac{\rho}{2}\|c_i - z_{ij}^{(k)} + u_{ij}^{(k)}\|_2^2\Big) \\ z_{ij}^{(k+1)}, z_{ji}^{(k+1)} &= \arg\min_{z_{ij},z_{ji}}\big(\lambda_1 s_{ij}\|z_{ij} - z_{ji}\|_{2,1} \\ &\quad + \frac{\rho}{2}\|c_i^{(k+1)} - z_{ij} + u_{ij}^{(k)}\|_2^2 \\ &\quad + \frac{\rho}{2}\|c_j^{(k+1)} - z_{ji} + u_{ji}^{(k)}\|_2^2\big) \\ u_{ij}^{(k+1)} &= u_{ij}^{(k)} + c_i^{(k+1)} - z_{ij}^{(k+1)} \end{aligned} \tag{9}$$

where $N(i)$ indicates all neighbourhoods of node i in the graph G.

Solving Eq. 9, we have:

$$\begin{aligned} c_i^* &= \frac{2x_i + \sum_{j\in N(i)}\rho(z_{ij}^{(k)} - u_{ij}^{(k)})}{\left(2 + 2\lambda_2 + \sum_{j\in N(i))}\rho\right)} \\ z_{ij}^* &= \theta(c_i + u_{ij}) + (1-\theta)(c_j + u_{ji}) \\ z_{ji}^* &= (1-\theta)(c_i + u_{ij}) + \theta(c_j + u_{ji}) \end{aligned} \tag{10}$$

where

$$\theta = \max\left(1 - \frac{\lambda_1 s_{ij}}{\rho\|c_i + u_{ij} - (c_j + u_{ji})\|_2}, 0.5\right)$$

To exploit parallelism in our algorithm, we implemented our algorithm following the work-depth model [8] to achieve efficient parallel computing. It can be seen from Algorithm. 1 that we can update c_i, z_{ij}, u_{ij} one by one in each iteration.

To solve the semi-supervised problem, we propose a two-step algorithm. In the objective function (Eq. 4), the first term is the supervised part, while the other three terms are unsupervised part. We can update these two parts separately.

- For each edge (i,j), update f_{ij} as

$$f_{ij} = \begin{cases} 1 & \text{if } w_{ij} > \delta, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

  Notice that $p_i = p_j$ is equivalent to $f_{ij} = 1$, and δ is the best decision boundary according to the input labels y_{ij}.
- For each i, update c_i using the similar method we proposed for the unsupervised model.

As shown in Algorithm. 2, we iterate the above two steps through the stochastic gradient descent algorithm.

Input: $G = (V, E), X$
Output: F
initialization $c_i^0 = x_i$, $z_{ij}^0 = x_i$, $u_{ij}^0 = 0$, $k = 0$;
s_{ij} = similarity of x_{ij} for each edge (i,j);
repeat
  update c_i^{k+1} by Eq. 10;
  `// parallel by each vertex i`
  update $z_{ij}^{k+1}, z_{ji}^{k+1}$ by Eq. 10;
  `// parallel by each edge (i,j)`
  update $u_{ij}^{k+1}, u_{ji}^{k+1}$ by Eq. 9;
  `// parallel by each edge (i,j)`
until *Termination*;
foreach *edge* $(i,j) \in E$ **do**
  if $c_i == c_j$ **then**
    $f_{ij} = 1$;
  else
    $f_{ij} = 0$;
  end
end

Algorithm 1: PGRR (unsupervised version)

Input: $G = (V, E), X, E', Y$
Output: F
initialization $c_i^0 = x_i$, $z_{ij}^0 = x_i$, $u_{ij}^0 = 0$, $k = 0$;
s_{ij} = similarity of x_{ij} for each edge (i,j);
repeat
  update w_{ij};
  update δ;
  update f_{ij}, p_i by Eq. 11;
  `// parallel by each edge (i,j)`
  update s'_{ij} to replace s_{ij} by Eq. 5;
  `// parallel by each edge (i,j)`
  repeat
    update c_i^{k+1} by Eq. 10;
    `// parallel by each vertex i`
    update $z_{ij}^{k+1}, z_{ji}^{k+1}$ by Eq. 10;
    `// parallel by each edge (i,j)`
    update $u_{ij}^{k+1}, u_{ji}^{k+1}$ by Eq. 9;
    `// parallel by each edge (i,j)`
  until *Termination*;
until *Termination*;
foreach *edge* $(i,j) \in E$ **do**
  if $p_i == p_j$ **then**
    f_{ij} = TRUE;
  else
    f_{ij} = FALSE;
  end
end

Algorithm 2: PGRR (semi-supervised version)

Table 5: Statistics of attribute values

Attributes	Distinct values	Records with values(%)
account id	19,941	2.20%
brand	2,247	100.00%
device model	11,397	100.00%
resolution	601	100.00%
client IP	21,555	26.33%
city	11	99.91%
district	72	29.88%

6 EXPERIMENTS AND RESULTS

We implement and evaluate our algorithm on a large-scale dataset of Alibaba, a world-leading e-commerce company. We start with a description of the data set.

6.1 Data Set and Experimental Settings

6.1.1 Data Description. Our experiments are performed on a very large real-world dataset with billions of mobile access records. Each record contains at least two types of device identifiers and some attributes collected from the accessing device such as the mobile phone brand and model number, resolution of the screen, IP address, geographic locations, etc. Table. 5 shows the statistics of the data set. All the attributes are categorical values. Because some comparison algorithms cannot scale to such large dataset, we also generated 5 datasets by randomly sampling from this dataset for the purpose of comparison. Table. 2 shows the statistics of all the experiment datasets.

6.1.2 Ground Truth. Alibaba is a world-leading e-commerce company. Besides providing consumer-to-consumer, business-to-consumer and business-to-business sales services, Alibaba also extends its services to entertainment, navigation, online payment, express, etc. Every day, billions of mobile access records have been generated through various of mobile applications, including cooperative partners' applications. Different data sources have different confidence levels. Authorized by users, we can generate, store, and achieve some security IDs in our core applications. The security IDs are more confident than mobile devices IDs. We can create a limited amount of ground truth based on these confident security IDs, to test our PGRR and other compared algorithms.

6.1.3 Compared Algorithms. To the best of our knowledge, no existing entity resolution algorithm can deal efficiently with both graph structure and feature similarity of such large scale. Our baseline algorithms are following:

- network lasso [21]
- k-means++ [2, 3]
- spectral clustering [29]
- Gaussian mixture model [39]
- identifiers co-occurrence relationship without attributes.

However, the time complexity of k-means++, spectral clustering, and GMM are too large to solve our real-world dataset with billions of records. For example, the time complexity of k-means++ algorithm is $O(nkdi)$, where n is the number of d-dimensional vectors, k is the number of clusters and i the number of iterations. In this problem, n and k are both more than 1 million, so k-means++ obviously is incapable of handling the entire dataset. Thus, we ran k-means++ on the sampled subsets only. Another important issue for k-means++, spectral clustering, and GMM is setting the number of clusters (k). There is actually no way of knowing the number of clusters beforehand in our problem. For the purpose of algorithm comparisons, we use the number of devices returned by PGRR.

6.1.4 Similarity of Node. There are 7 attributes in the dataset, namely account id, device brand, device model, screen resolution, client IP address, city and district. The edge weights in SPI-Graph are computed based on the similarity between two vertices. We use these seven attributes in the records to measure the similarity between two nodes. Since all the attributes are categorical values, we adopt one-hot encoding [22] to convert these seven features into sparse binary features. High-dimensionalities in feature space incur high computational cost. To handle this issue, dimensionality reduction is applied in this feature space. We use Multiple correspondence analysis (MCA) [20] to reduce the feature dimensions from 55,824 to 100.

6.2 Evaluation Metric

Addressing the needs for both theoretical comparison and industrial benchmarking, we propose a novel and intuitive evaluation metric for this specific problem, called Precision of Sets of Identifier Resolution (PSIR). Denote G_i as a set of all IDs belonging to one physical device in ground truth, P_{y_i} as a set of all IDs belonging to one physical device in the results of our algorithm, we have:

$$PSIR = \frac{\sum_i |G_i \bigcap P_{y_i}|}{\sum_i |G_i \bigcup P_{y_i}|} \tag{12}$$

where

$$y_i = \underset{\substack{G_i \in Groundtruth, \\ P_{y_i} \in AlgorithmOutput}}{\arg\max} \frac{|G_i \bigcap P_{y_i}|}{|G_i \bigcup P_{y_i}|}$$

The range of PSIR is [0,1]. If every G_i finds its correct match P_{y_i} in our experimental results, $PSIR = 1$. Any mismatch will lower the value of PSIR.

The second evaluation metric used in our experiments is recall and is defined as follows:

$$recall = \frac{|Groundtruth \bigcap Prediction|}{|Groundtruth|} \tag{13}$$

The third evaluation metric, SIR-F1 strikes a balance between PSIR and Recall. SIR-F1 is computed as follows:

$$\text{SIR-F1} = 2 * \frac{PSIR * recall}{PSIR + recall} \tag{14}$$

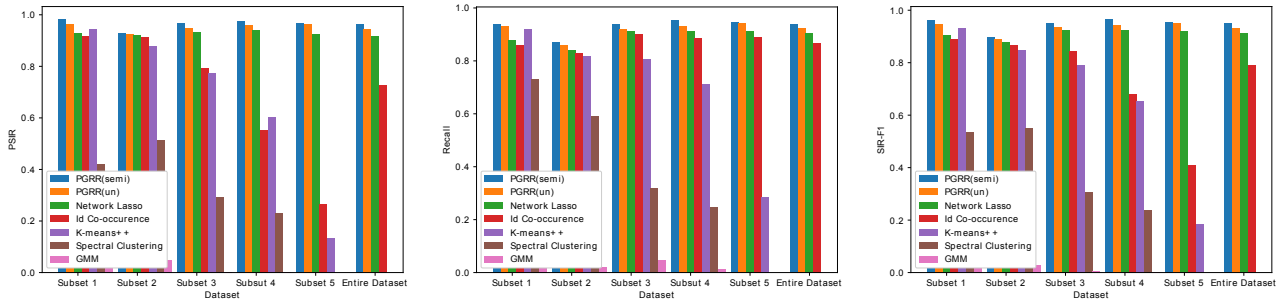
6.3 Experimental Results and Analysis

Table. 6 summarizes the experimental results in terms of PSIR, recall and SIR-F1 for PGRR and the compared algorithms in 5 data subsets and the full dataset. The results are shown in Fig. 3. The following conclusions can be drawn from the experimental results:

(1) The PGRR algorithm proposed in this paper consistently outperforms all the compared algorithms in all the PSIR,

Table 6: PSIR, recall, SIR-F1 of comparisons of different algorithms(%)

Dataset	PGRR(semi)			PGRR(un)			NL			Id Co.			K-means			Spectral Cl.			GMM		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Subset 1 (0.005%)	98.28	94.00	96.09	96.31	93.00	94.63	93.01	88.00	90.44	91.94	86.00	88.87	94.48	92.00	93.22	42.16	73.00	53.45	5.05	3.00	3.76
Subset 2 (0.007%)	92.96	87.00	89.88	92.42	86.00	89.09	92.06	84.00	87.85	91.18	83.00	86.90	87.98	82.00	84.88	51.58	59.00	55.04	4.87	2.00	2.84
Subset 3 (0.09%)	96.60	93.80	95.18	94.75	92.20	93.46	93.32	91.20	92.25	79.37	90.00	84.35	77.48	80.60	79.00	29.47	32.00	30.68	0.38	4.80	0.70
Subset 4 (0.2%)	97.55	95.60	96.57	95.88	93.10	94.47	93.92	91.10	92.49	55.29	88.70	68.12	60.14	71.30	65.25	22.96	24.90	23.89	0.21	1.40	0.37
Subset 5 (1.3%)	96.79	94.54	95.65	96.28	94.32	95.29	92.44	91.42	91.93	26.57	89.16	40.94	13.50	28.62	18.35	n/a	n/a	n/a	n/a	n/a	n/a
Entire Dataset	96.46	93.96	95.19	94.46	92.30	93.37	91.73	90.55	91.14	72.56	86.91	79.09	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a



(a) PSIR (b) Recall (c) SIR-F1

Figure 3: PSIR, recall, SIR-F1 of comparisons of different algorithms

recall and SIR-F1, in all the 5 data subsets and the full dataset. The results show that by combining both the graph-based ID co-occurrence information and attribute information, PGRR is capable of more precisely resolving access records. We can also observe from the results that as the size of dataset increases, the gap between PGRR and comparing algorithms widens in all the evaluation metrics. The results not only verify the effectiveness of PGRR, but also show its excellence in big data scenarios.

(2) Network Lasso is a good method for our MARR problem. However, we add extra regularization to the original network lasso model to accelerate the convergence and improve the results of resolution over all three metrics.

(3) ID co-occurrence proves itself as a good algorithm in all evaluation metrics and its lead over the classic clustering methods, k-means++, spectral clustering and GMM algorithms, becomes more obvious as the data size increases. The results show the contributions of the graph-based ID co-occurrence information and attribute information in access records. As can be deduced from the experimental results, ID co-occurrence represents more important information for device resolution than the attribute information.

(4) The main reason why classic clustering methods fail to obtain good results in the MARR problem is the extremely large number of clusters, k. Traditional datasets for clustering problem often have a small and fixed k. However, in our real-world dataset, the number of clusters is proportional to the number of nodes.

6.4 Runtime Performance of PGRR

There are two important metrics for runtime performance of the Work-Depth parallel algorithm: work load and worst-case running time. Work load is represented by the total number of operations, and worst-case running time by the parallel time complexity.

First, we analyze the unsupervised version of PGRR. For the initialization and similarity calculation, the work load is $O(d(|V|+|E|))$, and the time complexity is $O(d)$. When updating c^{k+1}, the work load is $O(d(|V|+|E|))$, and the time complexity is $O(dM)$, where M is the maximum number of the neighbors for a node in the SPI-graph. When updating z^{k+1} and u^{k+1}, the work load is $O(d|E|)$, and the time complexity is $O(d)$. To compute the output, the work load is $O(d|E|)$, and the time complexity is $O(d)$. Denoting I as the number of iterations, the total work load of the unsupervised version of PGRR is $O(Id(|V| + |E|))$, and the worst-case time complexity is $O(IdM)$.

For the semi-supervised version of PGRR, the initialization and similarity calculation are the same as the unsupervised version. When updating w_{ij}, the work load is $O(d|E|)$, and the time complexity is $O(d)$. When updating δ, the work load is $O(|E'|log|E'|)$, and the time complexity is $O(log^2|E'|)$. When updating f_{ij}, the work load is $O(|E|)$, and the time complexity is $O(1)$. When updating p_i, the work load is $O(|E|logK)$, and the time complexity is $O(logK)$, where K is the maximum number of nodes in every connected component in G. When updating s'_{ij}, the work load is $O(|E|)$, and the time complexity is $O(1)$. For the inner loop, all work load and time complexity are the same as the unsupervised version. Denoting J as the number of iterations of the outer loop, the total work load of the unsupervised version of PGRR is $O(J(Id(|V + E|) + |E|logK))$, and the worst-case time complexity is $O(J(IdM + log^2|E'| + logK))$.

7 CONCLUSIONS

In this paper, we introduce the MARR problem for resolving mobile access records using multiple mobile device identifiers. We propose a new entity resolution model on SPI-Graphs for resolving mobile access records and develop a parallel algorithm for efficiently implementing the model on large-scale datasets. Extensive experimental results on large-scale real-world datasets validate the effectiveness and efficiency of our algorithm.

There are several interesting problems to be investigated in our future work:

- Internet companies like Alibaba nowadays will have heterogeneous data sets from different applications and services. Entity resolution from heterogeneous sources is a more difficult issue. Heterogeneity poses particular challenges in a big data setting. But graph-based models and parallel computing are expected to address this issue well. It will be very interesting and valuable to extend our algorithm to multiple heterogeneous data sets.
- We can further group mobile access records for a specific device into access sessions and thus better characterize user profile. Personalized recommendations based on session resolution can then be generated even for anonymous sessions.

ACKNOWLEDGMENTS

This work is supported by Alibaba-Zhejiang University Joint Institute of Frontier Technologies and Zhejiang Provincial Key Research and Development Plan (Grant no. 2017C01012).

SHEN Xin would like to express the deepest appreciation to Jingyi Zhang for her continued assistance and encouragement.

REFERENCES

[1] Arvind Arasu, Michaela Götz, and Raghav Kaushik. 2010. On active learning of record matching packages. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 783–794.

[2] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1027–1035.

[3] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. 2012. Scalable k-means++. *Proceedings of the VLDB Endowment* 5, 7 (2012), 622–633.

[4] Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. 2009. Swoosh: a generic approach to entity resolution. *The VLDB JournalâĂŤThe International Journal on Very Large Data Bases* 18, 1 (2009), 255–276.

[5] Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007), 5.

[6] Mikhail Bilenko, Beena Kamath, and Raymond J Mooney. 2006. Adaptive blocking: Learning to scale up record linkage. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*. IEEE, 87–96.

[7] Mikhail Bilenko and Raymond J Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 39–48.

[8] Guy E Blelloch. 1996. Programming parallel algorithms. *Commun. ACM* 39, 3 (1996), 85–97.

[9] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 3, 1 (2011), 1–122.

[10] Jiajun Bu, Xin Shen, Bin Xu, Chun Chen, Xiaofei He, and Deng Cai. 2016. Improving collaborative recommendation via user-item subgroups. *IEEE Transactions on Knowledge and Data Engineering* 28, 9 (2016), 2363–2375.

[11] Zhaoqi Chen, Dmitri V Kalashnikov, and Sharad Mehrotra. 2009. Exploiting context analysis for combining multiple entity resolution systems. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. ACM, 207–218.

[12] Eric C Chi and Kenneth Lange. 2015. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics* 24, 4 (2015), 994–1013.

[13] Peter Christen. 2008. Automatic record linkage using seeded nearest neighbour and support vector machine classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 151–159.

[14] Munir Cochinwala, Verghese Kurien, Gail Lalk, and Dennis Shasha. 2001. Efficient data reconciliation. *Information Sciences* 137, 1 (2001), 1–15.

[15] Anish Das Sarma, Ankur Jain, Ashwin Machanavajjhala, and Philip Bohannon. 2012. An automatic blocking mechanism for large-scale de-duplication tasks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 1055–1064.

[16] Mukund Deshpande and George Karypis. 2004. Selective markov models for predicting web page accesses. *ACM Transactions on Internet Technology (TOIT)* 4, 2 (2004), 163–184.

[17] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.

[18] Christie I Ezeife and Yi Lu. 2005. Mining web log sequential patterns with position coded pre-order linked wap-tree. *Data Mining and Knowledge Discovery* 10, 1 (2005), 5–38.

[19] Lise Getoor and Ashwin Machanavajjhala. 2013. Entity resolution for big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1527–1527.

[20] Michael Greenacre. 2017. *Correspondence analysis in practice*. CRC press.

[21] David Hallac, Jure Leskovec, and Stephen Boyd. 2015. Network lasso: Clustering and optimization in large graphs. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 387–396.

[22] David Harris and Sarah Harris. 2010. *Digital design and computer architecture*. Morgan Kaufmann.

[23] John A Hartigan and JA Hartigan. 1975. *Clustering algorithms*. Vol. 209. Wiley New York.

[24] Toby Dylan Hocking, Armand Joulin, Francis Bach, and Jean-Philippe Vert. 2011. Clusterpath an algorithm for clustering using convex fusion penalties. In *28th international conference on machine learning*. 1.

[25] Frank Höppner. 1999. *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. John Wiley & Sons.

[26] Ida Mele. 2013. Web usage mining for enhancing search-result delivery and helping users to find interesting web content. In *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 765–770.

[27] Matthew Michelson and Craig A Knoblock. 2006. Learning blocking schemes for record linkage. In *AAAI*. 440–445.

[28] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. 2000. Automatic personalization based on web usage mining. *Commun. ACM* 43, 8 (2000), 142–151.

[29] Andrew Y Ng, Michael I Jordan, and Yair Weiss. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*. 849–856.

[30] Feiping Nie, Heng Huang, Xiao Cai, and Chris H Ding. 2010. Efficient and robust feature selection via joint âĎŞ2, 1-norms minimization. In *Advances in neural information processing systems*. 1813–1821.

[31] Neal Parikh, Stephen Boyd, et al. 2014. Proximal algorithms. *Foundations and Trends® in Optimization* 1, 3 (2014), 127–239.

[32] Kristiaan Pelckmans, Joseph De Brabanter, JAK Suykens, and B De Moor. 2005. Convex clustering shrinkage. In *PASCAL Workshop on Statistics and Optimization of Clustering Workshop*.

[33] Dimitrios Pierrakos, Georgios Paliouras, Christos Papatheodorou, and Constantine D Spyropoulos. 2003. Web usage mining as a tool for personalization: A survey. *User modeling and user-adapted interaction* 13, 4 (2003), 311–372.

[34] Pradeep Ravikumar and William W Cohen. 2004. A hierarchical graphical model for record linkage. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 454–461.

[35] Sheila Tejada, Craig A Knoblock, and Steven Minton. 2001. Learning object identification rules for information integration. *Information Systems* 26, 8 (2001), 607–633.

[36] Norases Vesdapunt, Kedar Bellare, and Nilesh Dalvi. 2014. Crowdsourcing algorithms for entity resolution. *Proceedings of the VLDB Endowment* 7, 12 (2014), 1071–1082.

[37] Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. 2012. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment* 5, 11 (2012), 1483–1494.

[38] Steven Euijong Whang, David Marmaros, and Hector Garcia-Molina. 2013. Pay-as-you-go entity resolution. *IEEE Transactions on Knowledge and Data Engineering* 25, 5 (2013), 1111–1124.

[39] Xinhua Zhuang, Yan Huang, Kannappan Palaniappan, and Yunxin Zhao. 1996. Gaussian mixture density modeling, decomposition, and applications. *IEEE Transactions on Image Processing* 5, 9 (1996), 1293–1302.