Object Identification with Constraints

Steffen Rendle, Lars Schmidt-Thieme
Department of Computer Science
University of Freiburg, Germany
steffen@rendle.de, lst@informatik.uni-freiburg.de

Abstract

Object identification aims at identifying different representations of the same object based on noisy attributes such as descriptions of the same product in different online shops or references to the same paper in different publications. Numerous solutions have been proposed for solving this task, almost all of them based on similarity functions of a pair of objects. Although today the similarity functions are learned from a set of labeled training data, the structural information given by the labeled data is not used. By formulating a generic model for object identification we show how almost any proposed identification model can easily be extended for satisfying structural constraints. Therefore we propose a model that uses structural information given as pairwise constraints to guide collective decisions about object identification in addition to a learned similarity measure. We show with empirical experiments on public and on real-life data that combining both structural information and attribute-based similarity enormously increases the overall performance for object identification tasks.

1 Introduction

Object identification is the task to identify groups of equivalent objects in a database. This problem often arises when merging information from multiple sources. Object identification [11, 10] is also known among others as record linkage [12] and duplicate detection [4, 9].

Typical adaptive object identification learns a model on a training set and utilizes it on a separate set that belongs to a different structure. Particularly the two sets do not share any link. In real world applications there is usually no such separation, but there often is a single dataset of which some parts are known.

An example is an e-commerce scenario [2] where offers from different shops should be merged. The task is to identify offers that reference the same product. Some of the offers are labeled by a unique product identifier like an EAN (European Article Number), so their identification is trivial. But others do not have this label and thus have to be identified based on their attributes such as manufacturer, name of product, price, etc. A classical object identification approach would use the labeled part of the data only for learning a pairwise decision model. Here, we propose an compound model that can make use of any such pairwise decision model as a component, but additionally uses advanced methods for constraint satisfaction that are applied on top. By embedding existing object identification models into our compound model, they can easily be extended to solve constrained problems without having to modify their core.

Overall, the contributions of our paper are as follows. (i) We formulate three problem classes for object identification, including the new class of pairwise constrained problems. (ii) We propose a generic model for object identification that subsumes almost all common models. (iii) We provide a new method for the collective decision stage that utilizes constraints. (iv) We show in experiments that additionally using constraints in clustering outperforms today's methods which use training data only for learning a pairwise decision model.

2 Related Work

This work focuses on how existing object identification solutions can be extended for handling additional structural constraints, that are known in the field of semi-supervised clustering.

2.1 Object Identification

Almost all models for object identification rely on predicting the equivalence of a pair of objects. Today often an adaptive method is used where multiple heuristic similarity measures over multiple attributes are combined to a single learned similarity measure [5, 9] over pairs of objects. Here, probabilistic classifiers or conditional random fields [10] can be used for predicting the equivalence of two objects. The overall consistency is guaranteed by taking the transitive closure [10] or by more sophisticated methods such as clustering [5, 2] – usually hierarchical agglomerative clustering.

For learning the models one uses a separate training set that is assumed to be similar to the test set for which object identities should be predicted. When Bilenko et al. [2] introduce the application of online-shopping, they propose an online learning algorithm for string similarity, because the product database grows over time. Although they realized the iterative nature of the problem, they did not utilize the structural informations provided by known parts for predicting new data.

2.2 Semi-Supervised Clustering

On the other hand there is the community of semi-supervised clustering. In their work, structural constraints are used to identify groups as we do. Recently there have been proposals [1] to use both structural constraints and learned metrics for clustering tasks. Other approaches [7] bring together constrained clustering for graph and vector-based data. In contrast to our work, these proposals use more sophisticated clustering algorithms – for example in terms of conjunction of K-Means with Hidden Markov Random Fields [1]. But it is important to note that these approaches cannot be directly used on top of existing object identification models as in our method. The first reason is that these clustering algorithms focus on problems where the number of classes is known in advance and usually is small, while estimating the number of clusters often is an afterthought, e.g., solved by an expensive exhaustive search. This causes major problems in object identification as here we typically deal with many small classes and estimating their number is an essential part of the problem. The second reason is that the learned metrics in semi-supervised clustering are not as rich as in object identification where classifiers or conditional random fields are used in conjunction with expensive feature extraction methods. Furthermore algorithms in semi-supervised clustering are not designed for using candidate generators like the blockers that are essential in object identification for handling large data sets. Finally, most approaches in semi-supervised clustering require the pairwise similarity to comply with some conditions – e.g. for Bregman divergences [1] – that similarity measures estimated by a classifier do not meet.

Davidson and Ravi [6] also investigate constraints in traditional hierarchical agglomerative clustering. In contrast to our work, they use Euclidean distance instead of learned pairwise similarity and they also lack blockers.

3 Problem formulation

3.1 Classical Problems

In traditional object identification one assumes that there is a set of instances X that should be grouped into equivalence classes. In an adaptive setting there exists a second set Y of labeled instances, i.e., a set Y with $X \cap Y = \emptyset$ together with an equivalence relation $E_Y \subseteq Y^2$ on Y. In general instances in Y only share the same characteristics for equality as instances in X, thus the training set Y can only be used for learning the similarity measure. So these *classical problems* ($\mathcal{C}_{\text{classic}}$) have no restrictions on the equivalence relation $E_X \subseteq X^2$ on X one tries to predict.

3.2 Class of Iterative Problems

The *iterative problem class* $\mathcal{C}_{\text{iter}}$ assumes that some labels of the target dataset itself are known in advance. The labeled data can be regarded as a known partition, which should be extended by new instances. So there is structural information of parts $Y \subseteq X$ of the whole data set X in terms of an equivalence relation $E_Y \subseteq Y^2$ on Y. Therefore, the set of admissible solutions $\mathcal{E}$ contains only those equivalence relations $E \subseteq X^2$ that are consistent with E_Y, i.e., satisfy $E \cap Y^2 = E_Y$.

The iterative problem class has many applications – for example predicting object identities for collections growing over time or handling partially labeled datasets as found in online-shopping comparison systems.

3.3 Class of Constrained Problems

The *constrained problem class* $\mathcal{C}_{\text{constr}}$ assumes that sets of *must-link* R_{ml} and *cannot-link* R_{cl} pairs between problem instances X are given. Both R_{ml} and R_{cl} can be seen as binary relations over X. In this work we assume that constrained problems are consistent. Therefore one could extend the constraints specified in must-links R_{ml} to the smallest equivalence relation $E_{\text{ml}} \supseteq R_{\text{ml}}$. The cannot-links R_{cl} can be assumed to be symmetric and irreflexive. With these extensions, a consistent problem must satisfy only $E_{\text{ml}} \cap R_{\text{cl}} = \emptyset$. Each equivalence relation E in the set of admissible solutions $\mathcal{E}$ has to satisfy $E \supseteq E_{\text{ml}}$ and $E \cap R_{\text{cl}} = \emptyset$.

There are many applications for constrained problems. For example in a setting with supervision, a system might be unsure about the equivalence of two items. Therefore it presents the pair to the supervisor, whose decision is captured by a *cannot-* or *must-link*.

It can easily be shown that $\mathcal{C}_{\text{classic}} \subset \mathcal{C}_{\text{iter}} \subset \mathcal{C}_{\text{constr}}$. As the class of constrained problems subsumes all the other problems, we will focus on methods for solving this class.

4 Generic Object Identification Model

There are many proposed models for object identification. As different terminologies are in use in the literature and components of models are often not presented isolated, we suggest a separation in three components that fits almost all proposed models (see figure 1).

Pairwise feature extraction creates a real valued feature vector of two objects $f : X^2 \to \mathbb{R}^n$. For this the attributes of the two objects are compared. Pairwise feature extraction traditionally uses distance functions like *TFIDF*, *Levenshtein* or *Jaccard distance*.

The probabilistic pairwise decision model predicts the probability that two objects are equivalent $P[x \equiv y]$. Usually, either probabilistic classifier [5, 9] or recently *conditional random fields* [10] are used.

At last the collective decision model utilizes the likelihood of pairwise decisions to create a consistent prediction, i.e., an equivalence relation, for the whole dataset. In a constraint setting the space of consistent solutions is limited to equivalence relations in $\mathcal{E}$. For classical problems without constraints, $\mathcal{E}$ is unrestricted. For this task often the transitive closure over the predicted pairs is taken [10, 3]. Recently a more sophisticated method is used by clustering the instances based on their pairwise probability [5, 2]. For this purpose usually variants of hierarchical agglomerative clustering are adapted.

For speedup, often a candidate pair generator $b : \mathcal{P}(X) \to \mathcal{P}(X^2)$ with $b(Y) \subseteq Y^2$, called *blocker*, is added. It restricts the number pairs to regard in the time-consuming parts. As the number of pairs is $O(n^2)$ in the number of instances, large datasets have to use such a blocking component.

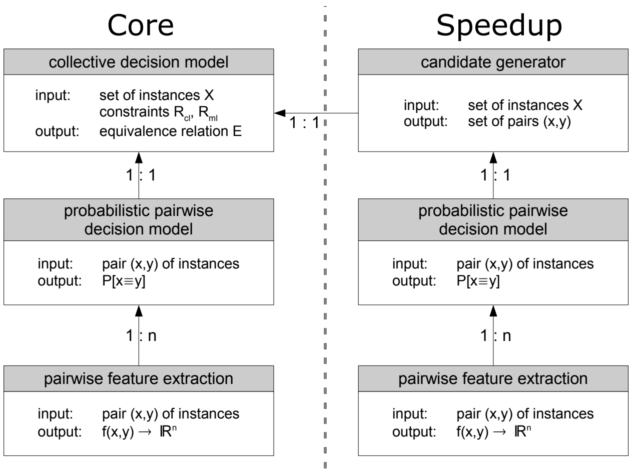
5 Handling Constraints

There are two stages where the additional knowledge of constraints might be used. First, there is the pairwise decision model that could be learned from *must-links* and *cannot-links*. Second, in the collective decision process the *must-links* and *cannot-links* can additionally guide a clustering algorithm.

As the pairwise decision models have already been investigated by several researchers [5, 9, 3, 10], we will not deal with this step in more detail. However, the second step has not been analyzed yet, so utilizing constraints in the collective decision model is the topic of the remaining section.

5.1 Collective decision model with constraints

The constrained problem easily fits into the generic model by extending the collective decision layer by constraints. As this stage might be solved by clustering algorithms in the classical problem, we propose to solve the constrained problem by a constraint based clustering algorithm as well. To enforce the constraint satisfaction we suggest a constrained hierarchical agglomerative clustering (HAC) algorithm. Because in an object identification task the number of equivalence classes is almost never known, we suggest model selection via a learned threshold to stop merging clusters.



Core
Speedup
collective decision model
input: set of instances X
constraints R_{cl}, R_{ml}
output: equivalence relation E
1 : 1
candidate generator
input: set of instances X
output: set of pairs (x,y)
1 : 1
1 : 1
probabilistic pairwise decision model
input: pair (x,y) of instances
output: P[x≡y]
probabilistic pairwise decision model
input: pair (x,y) of instances
output: P[x≡y]
1 : n
1 : n
pairwise feature extraction
input: pair (x,y) of instances
output: f(x,y) → IRⁿ
pairwise feature extraction
input: pair (x,y) of instances
output: f(x,y) → IRⁿ

Figure 1. Generic Object Identification Model

5.2 Constrained HAC Algorithm

A simplified representation of our HAC algorithm for constrained object-identification problems is shown in Algorithm 1. First, the algorithm initializes each instance in an own cluster as usual in HAC methods. Second, the *must-link* constraints are applied to form a partition of the objects given in R_{ml}. As mentioned before we assume the problem to be consistent, so the *must-links* induce an equivalence relation over a subset of all problem instances. After that the main loop merges the two closest clusters until the similarity of the closest clusters drop below the threshold θ. In contrast to standard HAC algorithms the cannot-links are taken into account when choosing the closest pair of clusters. For calculating the similarity between two clusters standard HAC variations may be used. Well known variations are:

$$\mathrm{sim}_{sl}(c_1, c_2) = \max_{x \in c_1, y \in c_2} \mathrm{sim}(x, y) \qquad \textit{single linkage}$$

$$\mathrm{sim}_{cl}(c_1, c_2) = \min_{x \in c_1, y \in c_2} \mathrm{sim}(x, y) \qquad \textit{complete linkage}$$

$$\mathrm{sim}_{al}(c_1, c_2) = \operatorname*{avg}_{x \in c_1, y \in c_2} \mathrm{sim}(x, y) \qquad \textit{average linkage}$$

The similarity $\mathrm{sim}(x, y) = P[x \equiv y]$ between two instances is given by a probabilistic pairwise model which is trained by the given *must-link* and *cannot-link* constraints. The only degree of freedom in the constrained HAC algorithm is θ. This threshold is responsible for stopping the merging of clusters. As the number of clusters is not known, θ has to be found by model selection. In our experiments we searched for optimal values of θ by repeated holdout on the training data.

For real-world object identification problems that have a huge number of instances, the proposed algorithm can easily be extended by several optimizations. In our implementation we (i) compute the cluster similarities by dynamic programming, (ii) use a blocker for reducing the number of pairs and (iii) prune cluster pairs with low similarities.

6 Evaluation methods for constrained problems

Mostly the *F-Measure* between *recall* and *precision* on all pairs $P_X := X^2 \setminus \{(x,x) | x \in X\}$ is used for evaluating the performance of a solution for a problem in $\mathcal{C}_{\text{classic}}$. In iterative and constrained problems, there are different choices for the set of pairs P_X to measure performance on:

1. **Test instances** $P_X := (X \setminus Y)^2$
   An iterative problem can be measured on the test instances only. The drawback is that the links between training and test data, which should also be predicted, are not considered by this method.

2. **All instances** $P_X := X^2$
   This method would factor in the problem of measuring links between both datasets. But as the given inner links of training data Y^2 are used for the evaluation, it generally is too optimistic.

3. **Unknown pairs** $P_X := X^2 \setminus Y^2$
   The third method is basically the same as the second one, but skips all links for evaluation that are among two data points of the training set.

We think that for evaluating constrained data the second method is the most practical one, because the user is normally interested in good overall results. When using algorithms that do not violate any given constraint, the second and third evaluation method only differ in a constant factor. As we will compare constrained methods to classical methods, which do not utilize the structural informations of constraints, all three evaluation methods are important. The first evaluation method allows only to assess if methods that factor in constraints do also perform better on pairs that are not directly connected to any labeled instance. Otherwise one might argue, that only fully or partially labeled pairs benefit from constraints.

Algorithm 1 Constrained HAC Algorithm

```
procedure ClusterHAC($X$, $R_{ml}$, $R_{cl}$)
  outputs a partition $P$ for $X$ satisfying $R_{ml}$ and $R_{cl}$

  initialize a new cluster for each instance:
    $P \leftarrow \{\{x\} | x \in X\}$

  apply must-link constraints:
    for all $(x,y) \in R_{ml}$ do
      $c_1 \leftarrow c$ where $c \in P \wedge x \in c$
      $c_2 \leftarrow c$ where $c \in P \wedge y \in c$
      $P \leftarrow (P \setminus \{c_1, c_2\}) \cup \{c_1 \cup c_2\}$
    end for

  repeat merging the most similar clusters:
    repeat
      $(c_1, c_2) \leftarrow \operatorname{argmax}_{c_1, c_2 \in P \wedge (c_1 \times c_2) \cap R_{cl} = \emptyset} sim(c_1, c_2)$
      if $sim(c_1, c_2) \geq \theta$ then
        $P \leftarrow (P \setminus \{c_1, c_2\}) \cup \{c_1 \cup c_2\}$
      end if
    until $sim(c_1, c_2) < \theta$

    return $P$
end procedure
```

7 Experiments

The evaluation section deals with the following questions: (i) Are constrained models superior to classical models for solving problems with given structure (e.g. iterative or constrained problems)? (ii) How much knowledge is necessary for constrained models to solve a task satisfactorily?

7.1 Data sets and model setup

For evaluation we used three data sets. The first one is the public Cora dataset [8] containing citations. The other ones are two product groups (namely *DVD player* and *Cameras*) of the online-shopping dataset of Mentasys GmbH[1].

The Cora model uses the TFIDF cosine similarity, Levenshtein string distance and Jaccard distance between every single attribute. The Mentasys models use three composite variations of the attributes *manufacturer*, *productname* and *description* with TFIDF cosine similarity, a boolean feature for comparing the *merchant*, the relative difference for *prices* and four other comparison functions, handtuned for the domain of product deduplication. As decision model a C-SVM from libSVM [2] is used. For collective overall decisions we report results for different HAC methods. A

[1] Mentasys GmbH, Karlsruhe, Germany, http://www.mentasys.de.

[2] http://www.csie.ntu.edu.tw/ cjlin/libsvm

Data set	Cora	DVD player	Camera
F-Measure for best classic method	0.88/0.90/0.89	0.87/0.87/0.87	0.67/0.67/0.67
F-Measure for best constrained method	0.92/0.94/0.93	0.92/0.96/0.95	0.81/0.90/0.86
absolute error reduction on F-Measure	0.04/ 0.04/ 0.04	0.05/ 0.09/ 0.08	0.14/ 0.23/ 0.19
absolute error reduction with at least 97.5% confidence	0.01/ 0.01/ 0.01	0.03/ 0.06/ 0.05	0.09/ 0.19/ 0.16
relative error reduction on F-Measure	33%/ 40%/ 36%	38%/ 69%/ 62%	42%/ 70%/ 58%
number of clusters mentioned in training data	83 ±1.89	119 ±2.08	319 ±4.12
total number of clusters to find	112	147	399

Table 1. F-Measure of classic and constrained methods for each of the three evaluation methods. Reduction of error when switching from best classic method to best constrained one.

simple canopy blocker [8] reduces the candidate pairs. With this setup both models are state-of-the-art in object identification. The Mentasys model is comparable to [2], but uses a more expressive classifier. The Cora model is similar to [5], but with more sophisticated clustering methods, a more expressive classifier and partly different similarity functions.

All parameters in the decision model – including the stopping threshold θ – were tuned on holdout parts of the training data for optimal F-Measure. The thresholds for the canopy blocker were tuned for optimal weighted harmonic mean between Pair Completeness ($weight = 0.75$) and Reduction Ratio. All experiments were repeated 4 times.

7.2 Influence of constraints

For this experiment we placed randomly 50% (25% for Cora) of the instances into the training set. The rest (50% and 75% for Cora) is assumed to be unknown. First we learned the pairwise decision model on the training set. Afterwards both classic and constrained versions of the HAC variations single, complete and average linkage made an overall decision of the whole dataset. The classic versions use no constraints and only made decisions based on the trained pairwise decision model. So the classic method corresponds to state-of-the-art object identification without constraints. The constrained versions are additionally guided by the given constraints in the training set.

Table 1 shows the performance of the best classic and the best constrained linkage method. As you can see on all data sets and evaluation methods, the best constrained method outperforms the best classic one dramatically. For example the constrained single linkage result for the *Camera* dataset is 14% above the best classical method on the test set, 23% on the whole dataset and 19% on all unknown pairs. Table 1 also shows that the absolute error reduces significantly. Another result of this experiment is that the given problems cannot be seen as a typical classification task because many classes are not mentioned in the randomly drawn training set. E. g. in the 112-class problem of the Cora data set 29 classes (about 26%) are unknown in advance on average.

7.3 Amount of training data

To see how constrained models depend on the amount of training data, we varied the number of known instances from 10% to 60% on the *Camera* data set. The F-Measure for the evaluation method on all unknown pairs is shown in figure 2. As you can see, constrained average linkage always outperforms the best classic method, whereas constrained complete linkage has always the worst performance. It is interesting that the best classical method has already its peak at 20% known instances and does not profit noticeable from more labeled instances. This could indicate that the pairwise features are exhausted by the SVM at 20% and that they are not rich enough to describe the data in more detail. On the other hand, the constrained methods average and single linkage always benefit from more knowledge. Although they are using the identical, learned similarity measure as the classic method, they profit from the additional structural information that they take into account. When comparing both constrained methods to each other, one recognizes that average linkage is more effective than single linkage if few instances are known. The reason is that the single linkage strategy tend to make many mistakes of the type ‘false positive’ when the *cannot-links* are too sparse. As the knowledge increases, single linkage becomes better and at about 30% percent it outperformes all other strategies.

8 Future Work

In our evaluation we have drawn the training data randomly to simulate the fact that often some parts of the data are labeled in advance. In other cases no labeled data is given or the labeled data should be improved by a supervisor. In this case a rating function is necessary to propose those instances that are likely to increase the performance at most. There are different types of data the supervisor could label. For instance a pair of two objects or two whole clusters could be asked to be labeled as equivalent or not. The decisions of both methods could easily be formulated with

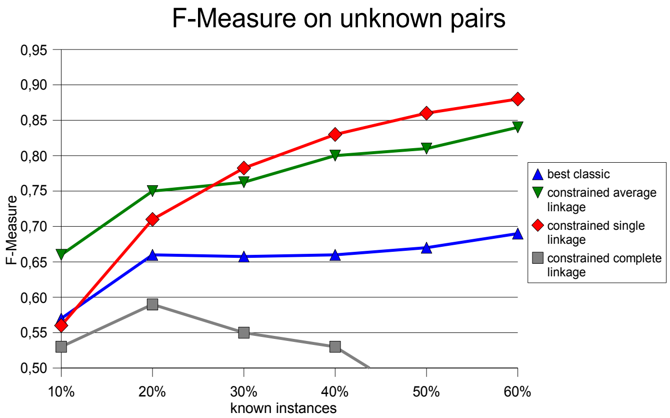


Figure 2. F-Measure on Camera dataset when evaluating on all unknown pairs.

must-link and *cannot-link* constraints, so that the algorithms proposed in this work could directly be used.

Although there has been research on active learning for classic object identification [11, 9], we hope that additionally utilizing constraints results in faster convergence and consequently in less effort for the supervisor, because our results have shown that constrained models improve much more with increasing known data than classic models.

9 Conclusions

We have presented the new problem class of constrained objective identification which is a generalization of the iterative problem. To solve this problem, we extended the traditional object identification model by collective decision models that can handle constraints. This way almost any proposed object identification model can be extended for constraint satisfaction. For this task we suggest a HAC algorithm that satisfies sets of *must-link* and *cannot-link* constraints. Altogether our overall model utilizes the structural knowledge for training a pairwise decision model as well as for guiding the collective decision process.

In our evaluation chapter we have shown that considering structural information outperforms the classical approaches on different evaluation methods. Our experiments also show that even a small proportion of randomly drawn training data is sufficient to notably improve the F-Measure.

Acknowledgements

This work was funded by the X-Media project (www.x-media-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978.

References

[1] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *KDD04*, pages 59–68, Seattle, WA, Aug. 2004.

[2] M. Bilenko, S. Basu, and M. Sahami. Adaptive product normalization: Using online learning for record linkage in comparison shopping. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM-2005)*, 2005.

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

[4] S. Chaudhuri, V. Ganti, and R. Motwani. Robust identification of fuzzy duplicates. In *Proceedings of the 21st International Conference on Data Engineering (ICDE-2005)*, Tokyo, Japan, 2005.

[5] W. W. Cohen and J. Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, pages 475–480, Edmonton, Alberta, 2002.

[6] I. Davidson and S. Ravi. Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 59–70, Porto, Portugal, 2005.

[7] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: a kernel approach. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 457–464, New York, NY, USA, 2005. ACM Press.

[8] A. K. McCallum, K. Nigam, and L. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the 6th International Conference On Knowledge Discovery and Data Mining (KDD-2000)*, pages 169–178, Boston, MA, Aug. 2000.

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

[10] P. Singla and P. Domingos. Object identification with attribute-mediated dependences. In *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PAKDD-2005)*, Porto, Portugal, 2005.

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

[12] W. E. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Census Bureau, Washington, DC, 1999.