Object Identification with Attribute-Mediated Dependences

Parag Singla and Pedro Domingos

Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350, U.S.A.
{parag, pedrod}@cs.washington.edu

Abstract. Object identification is the problem of determining whether different observations correspond to the same object. It occurs in a wide variety of fields, including vision, natural language, citation matching, and information integration. Traditionally, the problem is solved separately for each pair of observations, followed by transitive closure. We propose solving it collectively, performing simultaneous inference for all candidate match pairs, and allowing information to propagate from one candidate match to another via the attributes they have in common. Our formulation is based on conditional random fields, and allows an optimal solution to be found in polynomial time using a graph cut algorithm. Parameters are learned using a voted perceptron algorithm. Experiments on real and synthetic datasets show that this approach outperforms the standard one.

1 Introduction

In many domains, the objects of interest are not uniquely identified, and the problem arises of determining which observations correspond to the same object. For example, in vision we may need to determine whether two similar shapes appearing at different times in a video stream are in fact the same object. In natural language processing and information extraction, a key task is determining which noun phrases are co-referent (i.e., refer to the same entity). When creating a bibliographic database from reference lists in papers, we need to determine which citations refer to the same papers in order to avoid duplication. When merging multiple databases, a problem of keen interest to many large scientific projects, businesses, and government agencies, we need to determine which records represent the same entity and should therefore be merged. This problem, originally defined by Newcombe et al. [14] and placed on a firm statistical footing by Fellegi and Sunter [7], is known by the name of object identification, record linkage, de-duplication, merge/purge, identity uncertainty, hardening soft information sources, co-reference resolution, and others. There is a large literature on it, including Winkler [21], Hernandez and Stolfo [9], Cohen et al. [4], Monge and Elkan [13], Cohen and Richman [5], Sarawagi and Bhamidipaty [17], Tejada et al. [20], Bilenko and Mooney [3], etc. Most approaches are variants of the original Fellegi-Sunter model, in which object identification is viewed as a classification problem: given a vector of similarity scores between the attributes of two observations, classify it as “Match” or “Non-match.” A separate match decision is made for each candidate pair, followed by transitive closure to eliminate inconsistencies. Typically, a logistic regression model is used [1].

Making match decisions separately ignores that information gleaned from one match decision may be useful in others. For example, if we find that a paper appearing in *Proc. PKDD-04* is the same as a paper appearing in *Proc. 8th PKDD*, this implies that these two strings refer to the same venue, which in turn can help match other pairs of PKDD papers. In this paper, we propose an approach that accomplishes this propagation of information. It is based on conditional random fields, which are discriminatively trained, undirected graphical models [10]. Our formulation allows us to find the globally optimal match in polynomial time using a graph cut algorithm. The parameters of the model are learned using a voted perceptron [6].

Recently, Pasula et al. [15] proposed an approach to the citation matching problem that has collective inference features. This approach is based on directed graphical models, uses a different representation of the matching problem, also includes parsing of the references into fields, and is quite complex. It is a generative rather than discriminative approach, requiring modeling of all dependences among all variables, and the learning and inference tasks are correspondingly more difficult. A collective discriminative approach has been proposed by McCallum and Wellner [12], but the only inference it performs across candidate pairs is the transitive closure that is traditionally done as a post-processing step. Bhattacharya and Getoor [2] proposed an *ad hoc* approach to matching authors taking into account the citations they appear in. Our model can be viewed as a form of relational Markov network [18], except that it involves the creation of new nodes for match pairs, and consequently cannot be directly created by queries to the databases of interest. Max-margin Markov networks [19] can also be viewed as collective discriminative models, and applying their type of margin-maximizing training to our model is an interesting direction for future research.

We first describe in detail our approach, which we call the collective model. We then report experimental results on real and semi-artificial datasets, which illustrate the advantages of our model relative to the standard Fellegi-Sunter one.

2 Collective Model

Using the original database-oriented nomenclature, the input to the problem is a database of records (set of observations), with each record being a tuple of fields (attributes). We now describe the graphical structure of our model, its parameterization, and inference and learning algorithms for it.

2.1 Model Structure

Consider a database relation $R = \{r_1, r_2, \ldots, r_n\}$, where r_i is the i^{th} record in the relation. Let $F = \{F^1, F^2, \ldots, F^m\}$ denote the set of fields in the relation. For each field F^k, we have a set FV^k of corresponding field values appearing in the relation, $FV^k = \{f_1^k, f_2^k, \ldots, f_{l_k}^k\}$. We will use the notation $r_i.F^k$ to refer to the value of k^{th} field of record r_i. The goal is to determine, for each pair of records (r_i, r_j), whether they refer to the same underlying entity. Our graphical model contains three types of nodes:

Record-match nodes. The model contains a Boolean node R_{ij} for each pairwise question of the form: "Is record r_i the same as record r_j?"

Field-match nodes. The model contains a Boolean node F_{xy}^k for each pairwise question of the form: "Do field values f_x^k and f_y^k represent the same underlying property?" For example, for the venue field in a bibliography database, the model might contain a node for the question: "Do the strings 'Proc. PKDD-04' and 'Proc. 8th PKDD' represent the same venue?"

Field-similarity nodes. For pair of field values $f_x^k, f_y^k \in FV^k$, the model contains a node S_{xy}^k whose domain is the [0, 1] interval. This node encodes how similar the two field values are, according to a pre-defined similarity measure. For example, for textual fields this could be the TF/IDF score [16]. Since their values are computed directly from the data, we will also call these nodes *evidence nodes.*

Because of the symmetric nature of their semantics, R_{ij}, F_{xy}^k and S_{xy}^k represent the same nodes as R_{ji}, F_{yx}^k and S_{yx}^k, respectively.

The structure of the model is as follows. Each record-match node R_{ij} is connected by an edge to each corresponding field-match node $F_{xy}^k, 1 \leq k \leq m$. Formally, R_{ij} is connected to F_{xy}^k iff $r_i.F^k = f_x^k$ and $r_j.F^k = f_y^k$. Each field-match node F_{xy}^k is in turn connected to the corresponding field-similarity node S_{xy}^k. Each record-match node R_{ij} is also directly connected to the corresponding field-similarity node S_{xy}^k. In general, a field-match node will be linked to many record-match nodes, as the same pair of field values can be shared by many record pairs. This sharing lies at the heart of our model. The field-match nodes allow information to propagate from one candidate record pair to another. Notice that merging the evidence nodes corresponding to the same field value pairs, without introducing field-match nodes, would not work. This is because evidence nodes have known values at inference time, rendering the record-match nodes independent and reducing our approach to the standard one. Figure 1(a) shows a four-record bibliography database, and 1(b) shows the corresponding graphical representation for the candidate pairs (b_1, b_2) and (b_3, b_4). Note how dependences flow through the shared field-match node corresponding to the venue field. Inferring that b_1 and b_2 refer to the same underlying paper will lead to the inference that the corresponding venue strings "Proc. PKDD-04" and "Proc. 8th PKDD" refer to the same underlying venue, which in turn might provide sufficient evidence to merge b_3 and b_4. In general, our model can capture complex interactions between candidate pair decisions, potentially leading to better object identification.

One limitation of the model is that it makes a global decision on whether two fields are the same, which may not always be appropriate. For example, "J. Doe" may sometimes be the same as "Jane Doe," and sometimes the same as "Julia Doe." In this case the model will tend to choose whichever match is most prevalent. This simplifies inference and learning, and in many domains will not sigificantly affect overall performance. Nevertheless, relaxing it is an item for future work.

2.2 Conditional Random Fields

Conditional random fields, introduced by Lafferty et al. [10], define the conditional probability of a set of output variables $\mathbf{Y}$ given a set of input or evidence variables $\mathbf{X}$. Formally,

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \sum_{c \in C} \exp \sum_{l} \lambda_{lc} f_{lc}(y_c, x_c) \tag{1}$$

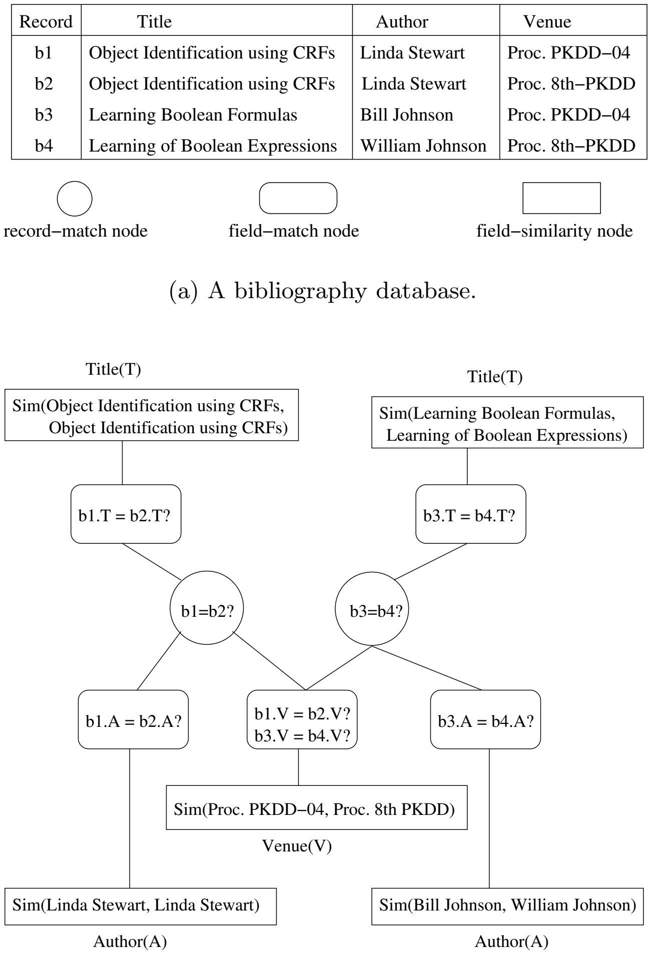
where C is the set of cliques in the graph, x_c and y_c denote the subset of variables participating in clique c, and $Z_{\mathbf{x}}$ is a normalization factor. f_{lc}, known as a feature function, is a function of variables involved in clique c, and λ_{lc} is the corresponding weight. In many domains, rather than having different parameters (feature weights) for each clique in the graph, the parameters of a conditional random field are tied across repeating clique patterns in the graph, called clique templates [18]. The probability distribution can then be specified as

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \sum_{t \in T} \sum_{c \in C_t} \exp \sum_{l} \lambda_{lt} f_{lt}(y_c, x_c) \tag{2}$$

where T is the set of all the templates, C_t is the set of cliques which satisfy template t, and f_{lt} and λ_{lt} are respectively a feature function and a feature weight, pertaining to template t.

2.3 Model Parameters

Our model has a singleton clique for each record-match node and one for each field-match node, a two-way clique for each edge linking a record-match node to a field-match node, a two-way clique for each edge linking a record-match node to a field-similarity node, and a two-way clique between each field-match node and the corresponding field-similarity node. The parameters for all cliques of the same type are tied; there is a template for the singleton record-match cliques, one for each type of singleton field-match clique (e.g., in a bibliography database, one for author fields, one for title fields, one for venue fields, etc.),



Record	Title	Author	Venue
b1	Object Identification using CRFs	Linda Stewart	Proc. PKDD−04
b2	Object Identification using CRFs	Linda Stewart	Proc. 8th−PKDD
b3	Learning Boolean Formulas	Bill Johnson	Proc. PKDD−04
b4	Learning of Boolean Expressions	William Johnson	Proc. 8th−PKDD

○ record−match node
▢ field−match node
□ field−similarity node

(a) A bibliography database.

Title(T)
Sim(Object Identification using CRFs, Object Identification using CRFs)
Title(T)
Sim(Learning Boolean Formulas, Learning of Boolean Expressions)
b1.T = b2.T?
b3.T = b4.T?
b1=b2?
b3=b4?
b1.A = b2.A?
b1.V = b2.V?
b3.V = b4.V?
b3.A = b4.A?
Sim(Proc. PKDD−04, Proc. 8th PKDD)
Venue(V)
Sim(Linda Stewart, Linda Stewart)
Author(A)
Sim(Bill Johnson, William Johnson)
Author(A)

(b) Collective model (fragment).

Fig. 1. Example of collective object identification. For clarity, we have omitted the edges linking the record-match nodes to the corresponding field-similarity nodes.

and so on. The probability of a particular assignment $\mathbf{r}$ to the record-match and field-match nodes, given that the field-similarity (evidence) node values are $\mathbf{s}$, is

$$P(\mathbf{r}|\mathbf{s}) = \frac{1}{Z_{\mathbf{s}}} \exp \sum_{i,j} \left[\sum_l \lambda_l f_l(r_{ij}) + \sum_k \left(\sum_l \phi_{kl} f_l(r_{ij}.F^k) + \sum_l \gamma_{kl} g_l(r_{ij}, r_{ij}.F^k) \right.\right.$$
$$\left.\left. + \sum_l \eta_{kl} h_l(r_{ij}, r_{ij}.S^k) + \sum_l \delta_{kl} h_l(r_{ij}.F^k, r_{ij}.S^k) \right) \right] \quad (3)$$

where (i, j) ranges over all candidate pairs and k ranges over all fields. $r_{ij}.F^k$ and $r_{ij}.S^k$ refer to the k^{th} field-match node and field-similarity node, respectively, for the record pair (r_i, r_j). λ_l and ϕ_{kl} denote the feature weights for singleton cliques. γ_{kl} denotes the feature weights for a two-way clique between a record-match node and a field-match node. η_{kl} and δ_{kl} denote the feature weights for a two-way clique between a Boolean node (record-match node or field-match node, respectively) and a field-similarity node. Cliques have one feature per possible state. Singleton cliques thus have two (redundant) features: $f_0(x) = 1$ if $x = 0$, and $f_0(x) = 0$ otherwise; $f_1(x) = 1$ if $x = 1$, and $f_1(x) = 0$ otherwise. Two-way cliques involving Boolean variables have four features: $g_0(x, y) = 1$ if $(x, y) = (0, 0)$; $g_1(x, y) = 1$ if $(x, y) = (0, 1)$; $g_2(x, y) = 1$ if $(x, y) = (1, 0)$; $g_3(x, y) = 1$ if $(x, y) = (1, 1)$; each of these features is zero in all other states. Two-way cliques between a Boolean node (record-match node or field-match node) q and a field-similarity node s have two features, defined as follows: $h_0(q, s) = 1 - s$ if $q = 0$, and $h_0(q, s) = 0$ otherwise; $h_1(q, s) = s$ if $q = 1$, and $h_1(q, s) = 0$ otherwise. This captures the fact that, the more similar two field values are, the more likely they are to match.

Notice that a particular field-match node appears in Equation 3 once for each pair of records containing the corresponding field values. This reflects the fact that that node is effectively the result of merging the field-match nodes from each of the individual record-match decisions.

2.4 Inference and Learning

Inference in our model corresponds to finding the configuration $\mathbf{r}^*$ of non-evidence nodes that maximizes $P(\mathbf{r}^*|\mathbf{s})$. For random fields where maximum clique size is two and all non-evidence nodes are Boolean, this problem can be reduced to a graph min-cut problem, provided certain constraints on the parameters are satisfied [8]. Our model is of this form, and it can be shown that satisfying the following constraints suffices for the min-cut reduction to hold: $\gamma_{k0}+\gamma_{k3}-\gamma_{k1}-\gamma_{k2} \geq 0$, $\forall k, 1 \leq k \leq m$, where the $\gamma_{kl}, 0 \leq l \leq 3$, are the parameters of the clique template for edges linking record-match nodes to field-match nodes of type F^k (see Equation 3).[1] This essentially corresponds to requiring that nodes be positively correlated, which should be true in this application. Our learning algorithm ensures that the learned parameters satisfy these constraints. Since min-cut can be solved exactly in polynomial time, we have a polynomial-time exact inference algorithm for our model.

Learning involves finding maximum-likelihood parameters from data. The partial derivative of the log-likelihood L (see Equation 3) with respect to the parameter γ_{kl} is

$$\frac{\partial L}{\partial \gamma_{kl}} = \sum_{i,j} g_l(r_{ij}, r_{ij}.F^k) - \sum_{\mathbf{r}'} P_\Lambda(\mathbf{r}'|\mathbf{s}) \sum_{i,j} g_l(r'_{ij}, r'_{ij}.F^k) \qquad (4)$$

[1] The constraint mentioned in Greig et al. [8] translates to $\gamma_{k0}, \gamma_{k3} \geq 0$, $\gamma_{k1}, \gamma_{k2} \leq 0$, which is a more restrictive version of the constraint above.

where $\mathbf{r}'$ varies over all possible configurations of the non-evidence nodes in the graph, and $P_\Lambda(\mathbf{r}'|\mathbf{s})$ denotes the probability distribution according to the current set of parameters. In words, the derivative of the log-likelihood with respect to a parameter is the difference between the empirical and expected counts of the corresponding feature, with the expectation taken according to the current model. The other components of the gradient are found analogously. To satisfy the constraint $\gamma_{k0} + \gamma_{k3} - \gamma_{k1} - \gamma_{k2} \geq 0$, we perform the following re-parameterization: $\gamma_{k0} = f(\beta_1) + \beta_2$, $\gamma_{k1} = f(\beta_1) - \beta_2$, $\gamma_{k2} = -f(\beta_3) + \beta_4$, $\gamma_{k3} = -f(\beta_3) - \beta_4$, where $f(x) = log(1 + e^x)$. We then learn the β parameters using the appropriate transformation of Equation 4. The second term in this equation involves the expectation over an exponential number of configurations, and its computation is intractable. We use a voted perceptron algorithm [6], which approximates this expectation by the feature counts of the most likely configuration, which we find using our polynomial-time inference algorithm with the current parameters. The final parameters are the average of the ones learned during each iteration of the algorithm. Notice that, because parameters are learned at the template level, we are able to propagate information through field values that did not appear in the training data.

2.5 Combined Model

Combining models is often a simple way to improve accuracy. We combine the standard and collective models using logistic regression. For each record-match node in the training set, we form a data point with the outputs of the two models as predictors, and the true value of the node as the response variable. We then apply logistic regression to this dataset. Notice that this still yields a conditional random field.

3 Experiments

We performed experiments on real and semi-artificial datasets, comparing the performance of (a) the standard Fellegi-Sunter model using logistic regression, (b) the collective model, and (c) the combined model. If we consider every possible pair of records for a match, the potential number of matches is $O(n^2)$, which is a very large number even for datasets of moderate size. Therefore, we used the technique of first clustering the dataset into possibly-overlapping *canopies* using an inexpensive distance metric, as described by McCallum et al. [11], and then applying our inference and learning algorithms only to record pairs which fall in the same canopy. This reduced the number of potential matches to at most the order 1% of all possible matches. In our experiments we used this technique with all the three models being compared. The field-similarity nodes were computed using cosine similarity with TF/IDF [16].

3.1 Real-World Data

Cora The hand-labeled Cora dataset is provided by McCallum[2] and has previously been used by Bilenko and Mooney [3] and others. This dataset is a collec-

[2] www.cs.umass.edu/∼mccallum/data/cora-refs.tar.gz

Table 1. Experimental results on the Cora dataset (performance measured in %).

Citation Matching						
Model	Before transitive closure			After transitive closure		
	F-measure	Recall	Precision	F-measure	Recall	Precision
Standard	86.9	89.7	85.3	84.7	98.3	75.5
Collective	87.4	91.2	85.1	88.9	96.3	83.3
Combined	85.8	86.1	87.1	89.0	94.9	84.5

Author Matching						
Model	Before transitive closure			After transitive closure		
	F-measure	Recall	Precision	F-measure	Recall	Precision
Standard	79.2	65.8	100	89.5	81.1	100
Collective	90.4	99.8	83.1	90.1	100	82.6
Combined	88.7	99.7	80.1	88.6	99.7	80.2

Venue Matching						
Model	Before transitive closure			After transitive closure		
	F-measure	Recall	Precision	F-measure	Recall	Precision
Standard	48.6	36.0	75.4	59.0	70.3	51.6
Collective	67.0	62.2	77.4	74.8	90.0	66.7
Combined	86.5	85.7	88.7	82.0	96.5	72.0

tion of 1295 different citations to computer science of research papers from the Cora Computer Science Research Paper Engine. The original dataset contains only unsegmented citation strings. Bilenko and Mooney [3] segmented each citation into fields (author, venue, title, publisher, year, etc.) using an information extraction system. We used this processed version of Cora. We further cleaned it up by correcting some labels. This cleaned version contains references to 132 different research papers. We used only the three most informative fields: author, title and venue (with venue including conferences, journals, workshops, etc.). We compared the performance of the algorithms for the task of de-duplicating citations, authors and venues.[3] For training and testing purposes, we hand-labeled the field pairs. The labeled data contains references to 50 authors and 103 venues. We carried out five runs of two-fold cross-validation, and report the average F-measure, recall and precision on post-canopy record match decisions. (To avoid contamination of test data by training data, we ensured that no true set of matching records was split between folds.) Next, we took the transitive closure over the matches produced by each model as a post-processing step to remove any inconsistent decisions. Table 1 shows the results obtained before and after this step. The combined model is the best-performing one for de-duplicating citations and venues. The collective model is the best one for de-duplicating authors. Transitive closure has a variable effect on the performance, depending upon the algorithm and the de-duplication task (i.e. citations, authors, venues).

[3] For the standard model, TFIDF similarity scores were used as the match probabilities for de-duplicating the fields (i.e. authors and venues).

Table 2. Experimental results on the BibServ dataset (performance measured in %).

Citation Matching						
Model	Before transitive closure			After transitive closure		
	F-measure	Recall	Precision	F-measure	Recall	Precision
Standard	82.7	99.8	70.7	68.5	100.0	52.1
Collective	82.8	100.0	70.7	73.6	99.5	58.4
Combined	85.6	99.8	75.0	76.0	99.5	61.5

We also generated precision/recall curves on Cora for de-duplicating citations, and the collective model dominated throughout. [4]

BibServ BibServ.org is a publicly available repository of about half a million pre-segmented citations. It is the result of merging citation databases donated by its users, CiteSeer, and DBLP. We experimented on the user-donated subset of BibServ, which contains 21,805 citations. As before, we used the author, title and venue fields. After forming canopies, we obtained about 58,000 match pair decisions. We applied the three models to these pairs, using the parameters learned on Cora (Training on BibServ was not possible because of the unavailability of labeled data.). We then hand-labeled 100 random pairs on which at least one model disagreed with the others, and 100 random pairs on which they all agreed. From these, we extrapolated the (approximate) results that would be obtained by hand-labeling the entire dataset.[5] Table 2 shows the results obtained for de-duplicating citations before and after transitive closure. All the models have close to 100% recall on the BibServ data. The combined model yields the best precision, resulting in the overall best F-measure. Transitive closure hurts all models, with the standard model being the worst hit. This is attributable to the fact that BibServ is much noisier and broader than Cora; the parameters learned on Cora produce an excess of matches on BibServ, and transitive closer compounds this. Collective inference, however, makes the model more resistant to this effect.

Summary These experiments show that the collective and the combined models are able to exploit the flow of information across candidate pairs to make better predictions. The best combined model outperforms the best standard model in F-measure by 2% on de-duplicating citations in Cora, 27.5% on de-duplicating venues in Cora and 3% on de-duplicating citations in BibServ. On de-duplicating authors in Cora, the best collective model outperforms the best standard model by 0.9%.

3.2 Semi-Artificial Data

To further observe the behavior of the algorithms, we generated variants of the Cora dataset by taking distinct field values from the original dataset and

[4] For the collective model, the match probabilities needed to generate precision/recall curves were computed using Gibbs sampling starting from the graph cut solution.

[5] Notice that the quality of this approximation does not depend on the size of the database.

randomly combining them to generate distinct papers. This allowed us to control various factors like the number of clusters, level of distortion, etc., and observe how these factors affect the performance of our algorithms. To generate the semi-artificial dataset, we created eight distorted duplicates of each field value taken from the Cora dataset. The number of distortions within each duplicate was chosen according to a binomial distribution whose "probability of success" parameter we varied in our experiments; a single Bernoulli trial corresponds to the distortion of a single word in the original string. The total number of records was kept constant at 1000 in all the experiments with semi-artificial data. To generate the records in the dataset, we first decided the number of clusters, and then created duplicate records for each cluster by randomly choosing the duplicates for each field value in the cluster. The results reported are over the task of de-duplicating citations, were obtained by performing five runs of two-fold cross-validation on this data, and are before transitive closure.[6]

The first set of experiments compared the relative performance of the models as we varied the number of clusters from 50 to 400, with the first two cluster sizes being 50 and 100 and then varying the size at an interval of 100. The binomial distortion parameter was kept at 0.4. Figures 2(a), 2(c) and 2(e) show the results. The F-measure (Figure 2(a)) drops as the number of clusters is increased, but the collective model always outperforms the standard model. The recall curve (Figure 2(c)) shows similar behavior. Precision (Figure 2(e)) appears to drop with increasing number of clusters, with collective model outperforming the standard model throughout.

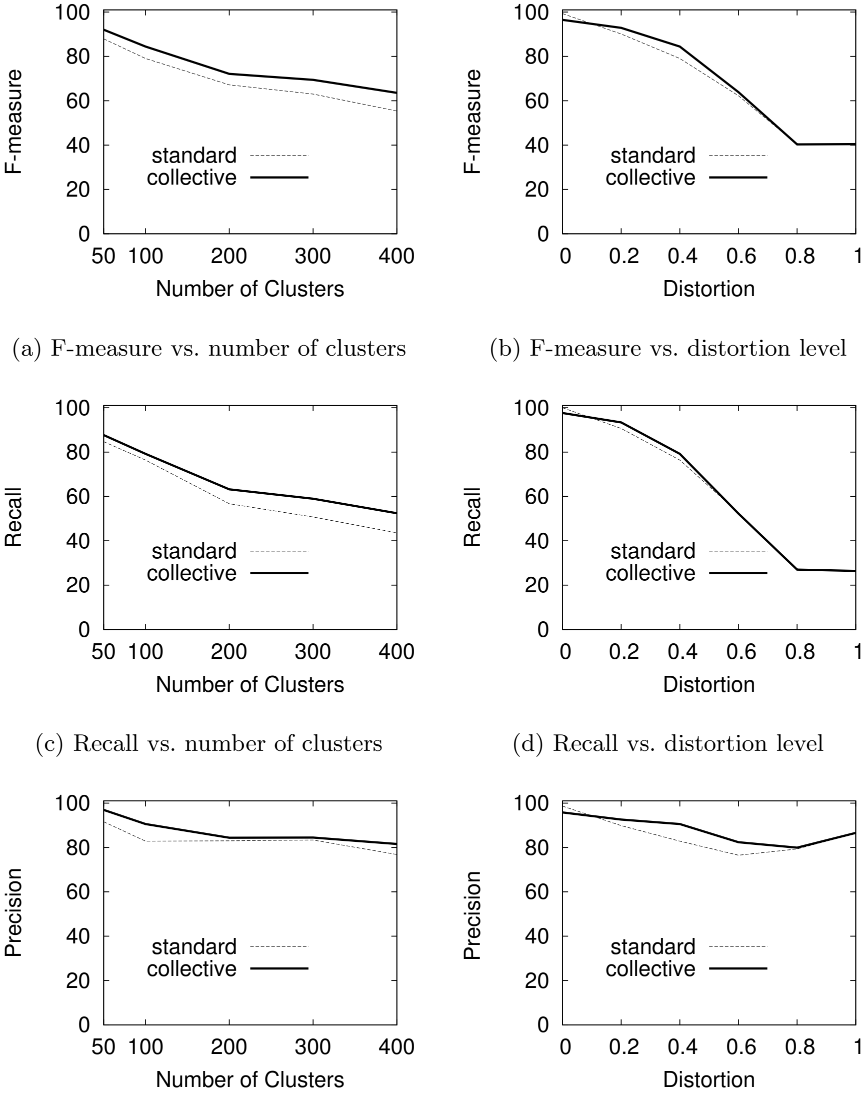
The second set of experiments compared the relative performance of the models as we varied the level of distortion from 0 to 1, at intervals of 0.2. (0 means no distortion, and 1 means that every word in the string is distorted.) The number of clusters in the dataset was kept constant at 100. Figures 2(b), 2(d) and 2(f) show the results. As expected, the F-measure (Figure 2(b)) drops as the level of distortion in the data increases, with the collective model dominating between the distortion levels of 0.2 to 0.6. The two models seem to perform equally well at other distortion levels. The recall curve (Figure 2(c)) shows similar behavior. Precision (Figure 2(e)) seems to fluctuate with increasing distortion, with the collective model dominating throughout.

Overall, the collective model clearly dominates the standard model over a broad range of the number of clusters and level of distortion in the data.

4 Conclusion and Future Work

Determining which observations correspond to the same object is a key problem in information integration, citation matching, natural language, vision, and other areas. It is traditionally solved by making a separate decision for each pair of observations. In this paper, we proposed a collective approach, where information is propagated among related decisions via the attribute values they have

[6] For clarity, we have not shown the curves for the combined model, which are similar to the collective model's.



(a) F-measure vs. number of clusters (b) F-measure vs. distortion level

(c) Recall vs. number of clusters (d) Recall vs. distortion level

(e) Precision vs. number of clusters (f) Precision vs. distortion level

Fig. 2. Experimental results on semi-artificial data.

in common. In our experiments, this produced better results than the standard method. Directions for future work include enriching the model with more complex dependences (which will entail moving to approximate inference), using it to deduplicate multiple types of objects at once, etc.

5 Acknowledgments

This research was partly supported by ONR grant N00014-02-1-0408 and by a Sloan Fellowship awarded to the second author.

References

1. A. Agresti. *Categorical Data Analysis.* Wiley, 1990.
2. I. Bhattacharya and L. Getoor. Iterative record linkage for cleaning and integration. In *Proc. SIGMOD-04 DMKD Wkshp.*, 2004.
3. M. Bilenko and R. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proc. KDD-03*, pages 7–12, 2003.
4. W. Cohen, H. Kautz, and D. McAllester. Hardening soft information sources. In *Proc. KDD-00*, pages 255–259, 2000.
5. W. Cohen and J. Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *Proc. KDD-02*, pages 475–480, 2002.
6. M. Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *EMNLP-02*, pages 1–8, 2002.
7. I. Fellegi and A. Sunter. A theory for record linkage. *J. American Statistical Association*, 64:1183–1210, 1969.
8. D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *J. Royal Statistical Society B*, 51:271–279, 1989.
9. M. Hernandez and S. Stolfo. The merge/purge problem for large databases. In *Proc. SIGMOD-95*, pages 127–138, 1995.
10. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML-01*, pages 282–289, 2001.
11. A. McCallum, K. Nigam, and L. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proc. KDD-00*, pages 169–178, 2000.
12. A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Adv. NIPS 17*, pages 905–912, 2005.
13. A. Monge and C. Elkan. An efficient domain-independent algorithm for detecting approximately duplicate database records. In *Proc. SIGMOD-97 DMKD Wkshp.*, 1997.
14. H. Newcombe, J. Kennedy, S. Axford, and A. James. Automatic linkage of vital records. *Science*, 130:954–959, 1959.
15. H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *Adv. NIPS 15*, pages 1401–1408, 2003.
16. G. Salton and M. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.
17. S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *Proc. KDD-02*, pages 269–278, 2002.
18. B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proc. UAI-02*, pages 485–492, 2002.
19. B. Taskar, C. Guestrin, B. Milch, and D. Koller. Max-margin Markov networks. In *Adv. NIPS 16*, 2004.
20. S. Tejada, C. Knoblock, and S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *Proc. KDD-02*, pages 350–359, 2002.
21. W. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Census Bureau, 1999.