Probability and Equality: A Probabilistic Model of Identity Uncertainty

Rita Sharma and David Poole

Department of Computer Science, University of British Columbia
Vancouver, BC V6T 1Z4, Canada
rsharma,poole@cs.ubc.ca

Abstract. Identity uncertainty is the task of deciding whether two descriptions correspond to the same object. It is a difficult and important problem in real world data analysis. It occurs whenever objects are not assigned with unique identifiers or when those identifiers may not be observed perfectly. Traditional approaches to identity uncertainty assume that the attributes in the descriptions are independent of each other given whether or not the descriptions refer to the same object. However, this assumption is often faulty. For example, in the person identity uncertainty problem – the problem of deciding whether two descriptions refer to the same person, the attributes "date of birth" and "last name" have the same values for *twins*. In this paper we discuss the identity uncertainty problem in the context of person identity uncertainty. We model the inter-dependence of the attributes and the probabilistic relations between the observed value of attributes and their actual values using a similarity network representation. Our approach allows queries such as, "what is the distribution over the actual names of a person given the names that appear in the description of the person", or, "what is the probability that two descriptions refer to the same person". We present results that show that our method outperforms the traditional approach for person identity uncertainty which considers the attributes as independent of each other.

1 Introduction

Identity uncertainty is a significant problem in many fields. The key task of this problem is to determine whether two descriptions refer to the same object. This problem has been studied independently under various names by different user communities. Within the statistics community, this problem has been studied as record linkage since at least 1969 [5]. Record linkage is used for matching records in one or more data files. The Fellegi-Sunter method [5] is the standard probabilistic method for solving this problem. In this method, for each pair of records, agreement and disagreement probabilities for each attribute are computed using frequency counts and error rates. The values of these match weights are then used to decide whether a pair of records is to be considered as a *match*, a *possible match*, or a *nomatch*. In the computer science literature the same problem has been studied under various names, duplicate detection [11], merge/purge problem [10], hardening soft information [3], or identity uncertainty [12]. Here we discuss some of the main work, but the review is not exhaustive. For summary reports on identity uncertainty or record linkage see [17, 8]. In [11, 10] this problem was considered as an extension of the string matching problem and the string matching algorithms are used to determine whether two values of attributes or records are similar enough to be duplicates. In [12] the authors proposed a relational approach for identity uncertainty for citation matching using the Relational Probabilistic model that captures the dependence between multiple records but not between the attributes. In [15] Ravikumar *et al.* consider the record-linkage problem as an unsupervised classification problem. They describe a hierarchical graphical model framework for the record linkage problem in an unsupervised setting.

With the exception of [15], in all of the above approaches, an independence assumption is made: i.e., matching of one attribute doesn't depend on other attributes. However, this assumption is often faulty. For example, people living in the same household have the same address, phone number and often the same last name. In this situation, if we assume that the last name, address and phone number are independent of each other, it becomes more likely that we have a "false positive match". As an another example, when a person moves to a different city, his address, phone number, and postal code all change together. In this situation, the independence assumption can cause a "false negative match".

In this paper we discuss the identity uncertainty problem in the context of person identity uncertainty (or person identification) using the person's demographic attributes. We model the dependence/independence between attributes using a similarity network representation [9]. In a similarity network of person identification some variables have very large domains. For example, the attribute *first name* has as domain all possible first names, which we may never know to the full extent because people can make up names. For efficient inference we represent the large CPTs using both *extensional* and *intensional* representation [16].

In the person identity uncertainty problem we need to compare a test person's description with each person's description in the database. Since we usually have large databases, instead of comparing a test record against every other record in the database, a pool of potentially matching records is created using a myopically constructed query. To deal with data entry errors, we use different error models. To test the proposed approach, as real databases are confidential, we model a reasonably realistic distribution of attribute values by modelling the people in a set of households and model, for example, how twins are born.

The remainder of this paper is organised as follows. Section 2 describes the identity uncertainty problem. Section 3 describes the probabilistic modelling of person identity uncertainty. In Section 4 we discuss how the large CPTs can be represented compactly using the *intensional* predicates and functions. In Section 5 we describe the probabilistic inference. Section 6 describes how an optimum query can be constructed using the bits of information provided by the attributes. In Section 7 we evaluate our approach followed by the conclusion in Section 8.

2 Identity Uncertainty

In the identity uncertainty problem, any two descriptions X and Y may or may not refer to the same object. Suppose $Desc_X$ and $Desc_Y$ denote the attributes values for descriptions X and Y. Let P_{same} be the posterior probability that descriptions X and

Y refer to the same object ($X = Y$) given their attribute values and $P_{notsame}$ be the posterior probability that descriptions X and Y refer to different objects ($X \neq Y$) given their attribute values. The odds, $Odds$, for hypotheses $X = Y$ and $X \neq Y$

$$Odds = \frac{P_{same}}{P_{notsame}} = \frac{P\,(X = Y)\,P\,(Desc_X \wedge Desc_Y | X = Y)}{P\,(X \neq Y)\,P\,(Desc_X \wedge Desc_Y | X \neq Y)}$$

We would expect that description Y's attributes value are independent of $X = Y$ or $X \neq Y$ given no information about the other description's attributes value. That is,

$$P\,(Desc_Y | X \neq Y) = P\,(Desc_Y | X = Y)$$

Thus,

$$Odds = \frac{P\,(X = Y)}{P\,(X \neq Y)} \times \frac{P\,(Desc_X | Desc_Y \wedge X = Y)}{P\,(Desc_X | Desc_Y \wedge X \neq Y)}$$

The ratio $\frac{P(X=Y)}{P(X \neq Y)}$ is a *prior odds* and the ratio $\frac{P(Desc_X | Desc_Y \wedge X=Y)}{P(Desc_X | Desc_Y \wedge X \neq Y)}$ is a likelihood ratio.

There are three possible actions (decisions) that can be taken for records X and Y: *match* – decide that X and Y refer to the same object, *possible match* – hold for a clerical review, *nomatch* – decide that X and Y refer to different objects.

The decision can be made using decision theory [4], given the likelihood ratio and the cost of false positive and negative matches. Suppose we have a cost function $E\,(\alpha|\omega)$ that describes the cost of action α when ω is true in world. If the action is *match*, the expected cost E_{match} is:

$$E_{match} = E\,(match|same) * P_{same} + E\,(match|diff) * P_{notsame}$$

Similarly, we can compute the expected cost for other two actions. We select that action for which the cost is minimum. The conditions for action *match*, *possible match*, and *nomatch* are the following:

- Action *match* if $\frac{P_{same}}{P_{notsame}} > max\,(C1, C2)$
- Action *possible match* if $min\,(C2, C3) < \frac{P_{same}}{P_{notsame}} < max\,(C1, C2)$
- Action *nomatch* if $\frac{P_{same}}{P_{notsame}} < min\,(C2, C3)$

where,

$$C1 = \frac{E\,(match|notsame) - E\,(posmatch|notsame)}{E\,(posmatch|same) - E\,(match|same)}$$

$$C2 = \frac{E\,(match|notsame) - E\,(nomatch|notsame)}{E\,(nomatch|same) - E\,(match|same)}$$

$$C3 = \frac{E\,(posmatch|notsame) - E\,(nomatch|notsame)}{E\,(nomatch|same) - E\,(posmatch|same)}$$

Note that we assume here that $E\,(match|same) < E\,(posmatch|same) < E\,(nomatch|same)$ and $E\,(nomatch|notsame) < E\,(posmatch|notsame) < E\,(match|notsame)$. The constant prior odds can be merged with constants $C1$, $C2$, and $C3$; then all we need is the likelihood ratio for making the decision.

3 Probabilistic Modelling of Person Identity Uncertainty

The key task in person identity uncertainty is to identify persons based on their demographic attributes (e.g., first name, last name, etc.). It occurs in many different person-centric applications. For example, in health care applications [6, 1] to identify patients, in social services applications to identify clients etc. The standard probabilistic approach for this problem [5] consider that the attributes are independent of each other. We relaxed this assumption and model the inter-dependence between the attributes using a similarity network representation [9]. This representation exploits the hypothesis-specific independence between variables. In particular, separate local Bayesian networks are constructed for each hypothesis. To identify a person we consider the following seven attributes: *Social insurance number (SIN)*, *first name (Fname)*, *last name (Lname)*, *date of birth (DOB)*, *gender (Gen)*, *phone number (PH)*, and *postal code (PC)*.

3.1 The Model of Attribute Dependence for Hypothesis $\mathrm{X} \neq Y$

When records X and Y refer to different people, we expect that their attributes values are independent. However, this is not always the case. We model the dependence between the attributes using the known relationships between people[1]. The statistical dependence among the attributes that we assume is shown in Figure 1. Propositions *twins*, *relative*, *samehousehold*, and *samelastname* represent that X and Y are twins, relatives, living in the same household, or have the same last name. We assume here that the gender of two different people is independent of each other[2].

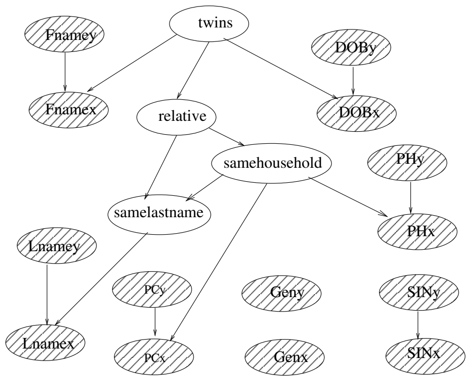

Fig. 1. Similarity network representation of attribute dependency for hypothesis $X \neq Y$ (shaded nodes are observed).

[1] The dependence may be derived from knowledge of domain experts or potentially can be learned.

[2] A more detailed model may specify that twins are more likely to be of the same gender and adults who live with children are more likely to be of different genders.

Attribute SIN doesn't depend on the other attributes. However, we cannot assume that the SIN of two different people is independent. Knowing a different person's SIN changes our belief in X's SIN, because, we expect that they shouldn't be the same. Thus,

$$P\left(SIN_x|SIN_y \wedge X \neq Y\right) = \begin{cases} r & \text{if } SIN_x = SIN_y \\ P\left(SIN_x\right) & \text{if } SIN_x \neq SIN_y \end{cases}$$

where, r denotes the probability that two different persons have the same SIN recorded, which is very, very small.

3.2 The Model of Attribute Dependence for Hypothesis $X = Y$

If records X and Y refer to the same person, we expect that the attributes values should be the same for both X and Y. However, there may be differences because of errors, for example: typing errors, nick names, and so on. We consider two cases of attribute dependence: first, the typist could have been sloppy, and second, the person could have moved to a new place of residence between the times that the records were input. We model the dependence among attributes using their actual values, the sloppiness of the data entry person (*SloppyX, SloppyY*), and the possibility of movement (*move*). Here, we consider the change in *phone number* and *postal code* because of the *move*.

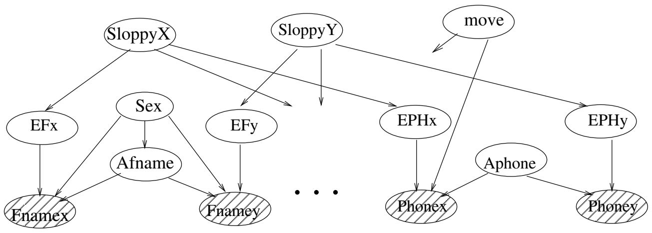

Fig. 2. Similarity network representation of attribute dependency for hypothesis $X = Y$ (shaded nodes are observed).

The dependence between attributes is shown in Figure 2. The unshaded nodes show the hidden variables. The proposition *Afname* represents the actual first name. The proposition *move* represents the possibility that the person has moved. The proposition *EFx* represents the error in first name for record X. To make this paper more readable, we consider only the following errors[3] (values of *EFx*): *copy error* (ce), an error where a person copies a correct name, but from the wrong row of a table, *single digit/letter error* (sde), and the lack of any errors, or *no error* ($noerr$). The random variables $Fnamex$, $Fnamey$, and $Afname$ have, as domains, all possible first names.

We assume that we have a procedural way for generating the prior probabilities of the variables that have very large domains (even unbounded). For the probability

[3] Although, we consider many more errors in the experiment.

$P\left(Afname|Sex\right)$, we use name lists available from the U.S. Census Bureau[4]. There are two name lists with associated probabilities: one for male names, and the other for female names. These lists cover 90% of all first names for both males and females. We need a different mechanism for names that do not exist in these lists. A number of approaches have been proposed to solve this problem [2, 7]. In our implementation, we use a very small probability as the estimate of the probability of a new name[5].

To compute the probability $P\left(Aphone\right)$ a model for generating phone numbers can be used. We use the simple procedure $P\left(Aphone\right)$ is $1/N$, where N is the number of legal phone numbers if $Aphone$ is a legal phone number and is 0 otherwise.

The probability table $P\left(Fnamex|Afname \wedge Sex \wedge EFx\right)$ can not be represented in a tabular form as we do not know all names, and even if we did, the domains of $Afname$ and $Fnamex$ are very large. The conditional probability table $P(Afname|sex)$ is also very large. To represent these large CPTs we need a compact representation so that we can reason in an efficient manner.

4 Representation of Large CPTs

To represent the large CPTs that arise in the probabilistic model of identity uncertainty we don't assume that there are explicit tables for its values. Rather, the conditional probabilities are computed from the structure of the values involved. We can represent these big CPTs in a compact form using both *intensional* and *extensional* representation. For example, the CPT $P\left(Fnamex|Afname \wedge Sex \wedge EFx\right)$ of Figure 2 can be represented in a decision tree form by conditioning on the values of EFx as shown in Figure 3.

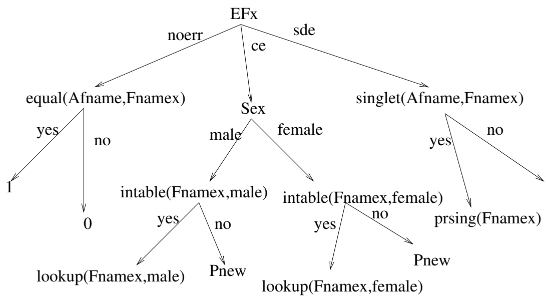

Fig. 3. A Decision Tree Representation of the CPT $P(Fnamex|Afname \wedge Sex \wedge EFx)$

The tree representation as shown in Figure 3 uses the *intensional* functions and predicates. The predicate $equal$ tests whether variables $Fnamex$ and $Afname$ have the same value or not, predicate $singlet$ tests whether the values for variables $Fnamex$

[4] http://www.census.gov/genealogy/names/

[5] The data available from U.S. Census Bureau is too noisy and incomplete to apply any of these approaches.

and $Afname$' are a single letter apart or not, and predicate $intable\ (Fnamex, male)$ tests whether the value of $Fnamex$ exist is in the male name file or not. The function $prsing$ is used to compute the probability when the data entry person makes the "single digit error" (sde). For example, if $EFx = sde$, $Fnamex = dave$ then $prsing(dave) = \frac{1}{100}$. The function $lookup(Fnamex, male)$ computes the probability of $Fnamex$ by looking in the male name file.

Example: Suppose the data-entry person makes a copying error ($EFx = ce$). In this case we can consider that the value of $Fnamex$ is distributed according to the distribution on names. To compute the probability of $Fnamex$ we can use male or female name files depending upon the value of Sex. If $Sex = male$ then $intable(Fnamex, male)$ tests whether $Fnamex$ exists in the male name. Now, if $intable(Fnamex, male) = yes$, $lookup(Fanmex, male)$ computes the probability of $Fnamex$, otherwise, the probability of $Fnamex$ is taken as the probability of a new name ($Pnew$). Thus,

$$\begin{aligned} P\left(Fnamex|Afname \wedge Sex = male \wedge EFx = ce\right) &= lookup(Fnamex, male) \\ &\quad \text{if } intable(Fnamex, male) = yes \\ &= Pnew \\ &\quad \text{if } intable(Fnamex, male) = no \end{aligned}$$

We assume here that we have the procedures that can compute these predicates and functions in an efficient manner. To compute efficiently the predicates and functions that involve query to large files, such as $intable(Fnamex, male)$, and $lookup(Fnamex, male)$, we need some efficient data structure for storing these files.

5 Inference

To compute the likelihood ratio we need to condition on the observations and marginalize over the unobserved variables in the Bayesian networks shown in Figures 1 and 2. We can marginalize over the unobserved variables for Bayesian network shown in Figure 1 using the Variable Elimination (VE) algorithm [18]. We get the likelihood of the observed data given the hypothesis $X \neq Y$. The marginalization for the network shown in Figure 2 is complicated. Although, we can represent the large CPTs of Figure 2 in a tree structure form we cannot use the standard Bayesian network inference algorithm that uses tree structure CPTs. These algorithms don't allow the intensional representation. To overcome this, we use the *Large Domain VE* algorithm [16] that allows us to make inference with intensional representation. Like the VE algorithm, *Large Domain VE* also has two main steps: conditioning on observations, and summing out all non observed non-query variables according to some elimination ordering.

In the conditioning on observation step the observed values of the variables are incorporated in the tree structure representation of factors. The intensional representation that have observed values are computed to simplify the factors. For example, suppose that for records X and Y we observed the first names, $Fnamex = david$ and $Fnamey = davig$. After setting the observed values for $Fnamex$ and $FnameY$ in the tree representation as shown in Figure 3 the tree gets simplified as shown by the tree $T1$ in Figure 4.

Like VE algorithm, in *Large Domain VE* to sum out a variable, first we need to multiply all those factors that contain that variable, then from the resulting factor, we sum out the variable [18, 16]. Since, the factors are represented by the tree structures, in *Large Domain VE* the factors are multiplied using two tree operations [16]: **Tree pruning** and **Tree merging**. After multiplying the factors the variable is summed out from the tree representation of the new factor. To sum out a variable from a tree in *Large Domain VE* we need to do two main steps: first, we need to compute the *probability mass* for all the values of the summing variable that end up at each leaf, and second, sum the subtrees that correspond to different blocks (subsets) for a partition of the summing variable; see ([16]) for details. Note that the complexity of *Large Domain VE* is mostly governed by the computation of the probability masses that involve the computation of the predicates and functions that are particular to a problem.

The main challenges of applying the *Large Domain VE* algorithm to "person identity uncertainty" problem are in the computation of *intensional functions* and predicates that arise in this problem. In the next section we discuss how we computed these intensional function and predicates without actually enumerating the values of variables.

Example: Suppose after conditioning on $Fnamex = david$, and $Fnamey = davig$ we want to eliminate the variable $Afname$ from the Bayesian network shown in Figure 2. As shown in Figure 4 $T1$, $T2$ and $T3$ are the decision tree representations of factors corresponding to CPTs $P(Fnamex|EFx \wedge Sex \wedge Afname)$, $P(Fnamey|Afname \wedge Sex \wedge EFy)$, and $P(Afname|sex)$ that contain the variable $Afname$. After multiplying trees $T1$, $T2$, and $T3$ we get a new factor. Part of the tree representation, T, of the new factor is shown in Figure 4. After we sum out the variable $Afname$ from tree T we get a new factor. Part of the tree representation, T', of the new factor is shown in Figure 4.

5.1 Computation of Probability Masses

In this section we describe how the *probability masses* $p1'$ and $p2'$ as shown in Figure 4 can be computed efficiently. Let us first consider the computation of the probability mass, $p1'$.

$$p1' = \sum_{\forall Afname=afname \in dom(Afname)(C1 \wedge C2 \wedge C3 = true)} p1$$

where, $C1 = (singlet(Afname, david) = yes)$, $C2 = (singlet(Afname, davig) = yes)$, and $C3 = (intable(Afname, male) = yes)$

We can query to the male name file representation to get the values of $Afname$ that are a single letter apart from both $david$ and $davig$, we get $Afname = \{davis\}$. Thus,

$$p1' = \sum_{Afname=\{davis\}} p1 = \left(\frac{1}{125}\right)^2 \times Pdavis$$

where, $Pdavis$ is the probability of name $davis$ from the male name file

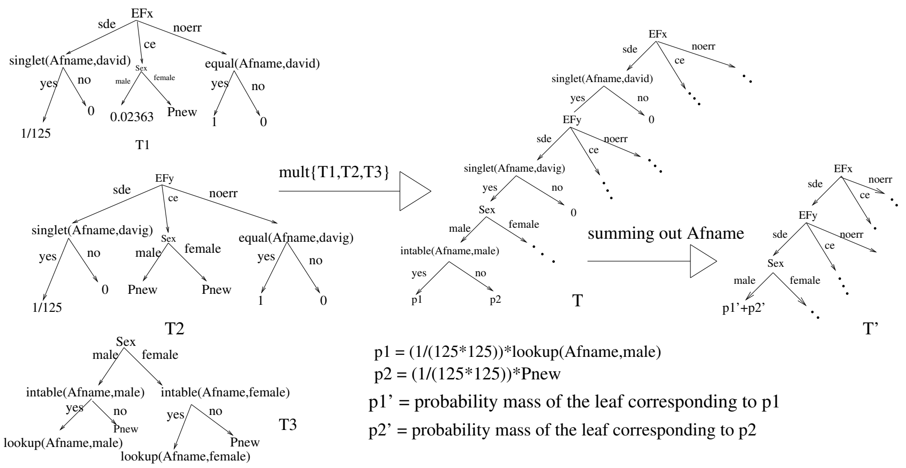

Fig. 4. Decision tree representations for trees: after multiplying trees $T1$, $T2$, and $T3$ together we get a new tree T and after summing out variable $Afname$ from T we get new tree T'($*$ represents multiplication operator).

Let us now consider the computation of *probability mass*, $p2'$.

$$p2' = \sum_{\forall Afname=afname \in dom(Afname)(C1 \wedge C2 \wedge C4=true)} p2$$

where $C4 = ((intable(Afname, male) = no)$

As shown in Figure 4, $p2$ is a not a function of $Afname$, to compute the value of $p2'$ we need the count of the values of $Afame$ that satisfy the predicates. To count efficiently the number of values of $Afname$ that are a single letter apart from both $david$ and $davig$, we first generate the patterns of names that are a single letter apart from $david$. For example, $?avid$, where $?$ is any letter except d. After generating these patterns we test which of these patterns makes the predicate $singlet(Afname, davig) = yes$. Here, the pattern $davi?$ makes the predicate yes if $? \neq d \wedge ? \neq g$. Thus, the possible number of values for $Afname$ is 24 that are a single letter apart from both $david$ and $davig$. Out of these 24 values of $Afname$ we have already found that one value "davis" exists in male name file. Thus,

$$p2' = 23 \times \left(\frac{1}{125}\right)^2 \times Pnew$$

6 Optimum Query Construction

In order to avoid comparing a test record against every record from the database, a test record should be compared with only potential matches from the database. Potential matches can be found using a query that can quickly retrieve a manageable but comprehensive set of records from a very large database. We can construct a query myopically using the number of bits of information provided by the attribute. We can compute the bits of information provided by each attribute (for matched and unmatched states) conditioned on other attributes using the Bayesian network shown in Figures 1 and 2. The query is constructed in rounds of ascending numbers of query attributes. In each round the attribute that provides the most bits of information is added to the query. The greedy procedure terminates when the bits of information provided by the current query attributes cannot be more than the lower threshold, $min(C2, C3)$ (see Section 2), by adding another attribute to the current query, or when bits of information cannot be lower than upper threshold, $max(C1, C2)$ (see Section 2).

7 Experimental Evaluation

To test our approach for the person identity uncertainty (as real databases are confidential), we model a reasonably realistic distribution of attribute values by modelling the people in a set of households and model, for example, how twins are born. We model a small town of 1500 households. We generate the population of the town. The generated population was intended to be a good model of real world population. Persons living in the same household have the same address and phone number. The probability that a *single person* lives in a house is 0.4. The probability that a person is living with a *partner* is 0.6. For a *single person* there is a 30% chance of having one child. The chances for a subsequent child is 10%. For each birth there is a 3% chance that twins will be born.

The probability that partners have the same last name is 0.5. For partners there is a 70% chance of having one child. The chances for a subsequent child is 30%. When both partners have different last names then the probability that the child will have any of the parent's last name is the same. Each record of the population contains seven fields as mentioned in Section 3. Personal first names and last names are chosen according to the distribution from U.S. census file[6].

After creating the true population, we made two datasets, D_A and D_B. To create D_A we randomly took 600 records from the true population. We corrupt the records using the database generator of Hernandez and Stolfo [10], using typographical errors and movement into the true record. The typographical errors introduced by the generator occur with relative frequencies known from previous research on spelling correction algorithms [13, 14]. We place these corrupted records in dataset D_A. Similarly, we made the database D_B but we took 1500 records from the true population.

We compared each record of dataset D_A with each record of dataset D_B. In these comparisons there were 227 duplicate cases. We compute the likelihood ratio considering both attribute dependence and independence. After computing the likelihood ratio between all pairs of records, we set the deciding threshold equal to the maximum of maximum likelihood ratio from both cases. The pair of records with likelihood ratio greater than the deciding threshold were taken as duplicates. We compute the precision and recall [7].

[6] http://www.census.gov/genealogy/names/

[7] We consider only two actions *match* and *no match*

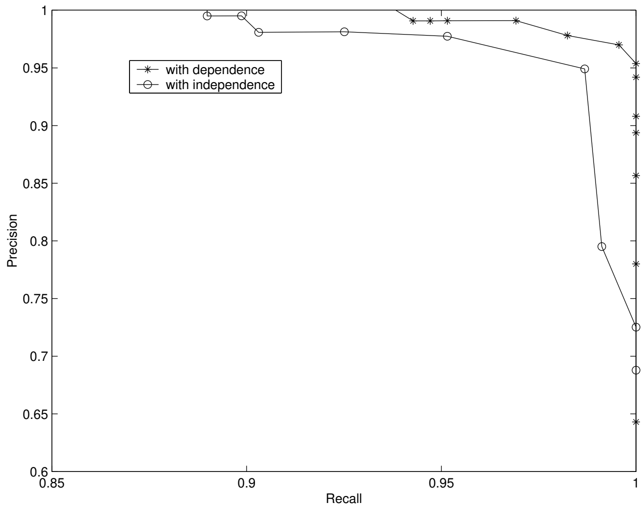

Fig. 5. Recall versus precision for both attribute dependence and attribute independence.

$$Precision = \frac{\#of\ correctly\ Identified\ Duplicate\ Pairs}{\#of\ Identified\ Duplicate\ pairs}$$

$$Recall = \frac{\#of\ correctly\ Identified\ Duplicate\ Pairs}{\#\ of\ True\ Duplicate\ pairs}$$

We reduce the deciding threshold with a step of 1 until the deciding threshold is equal to the minimum likelihood ratio from both cases. For each value of threshold we compute the precision and recall for both cases. As more pairs with lower similarity are labelled as duplicates, recall increases, while precision begins to decrease. Figure 5 shows the precision versus recall for both cases. The resulting recall/precision curve shows that with attribute dependence the precision of the prediction is 95% with 100% recall, while with attribute independence precision is 70% for 100% recall. Also, with attribute dependence 100% accuracy is achieved with more coverage than attribute independence.

8 Conclusion

Identity uncertainty is a significant problem in many fields. In this paper, we have presented a framework for reasoning about identity uncertainty in the context of "person identity uncertainty". We model the dependence/independence between the attributes using a similarity network representation. The probabilistic modelling of identity uncertainty is difficult, since the domain of some of the variables is very large (even unbounded). For efficient inference in the Bayesian network we represent the big CPTs using the *intensional* and *extensional* representation. We show how the *intensional* functions can be computed efficiently without actually enumerating the values of variables. As Figure 5 shows, the proposed approach considering attribute dependence achieved a high level of accuracy over the standard approach considering the attribute independence.

References

1. Glenn B. Bell and Anil Sethi. Matching records in a national medical patient. In *Communication of the ACM*, volume 44, September 2001.
2. Stanley F. Chen and Joshua T. Goodman. An empirical study of smoothing techniques for language modeling. In *Technical Report TR-10-98, Computer Science Group, Harvard University*, 1998.
3. W. Cohen, H. Kauthz, and D. McAllester. Hardening soft information sources. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*, pages 255–259, August 2000.
4. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience Publication, John Willey and Sons Inc, second edition, 2000.
5. I.P. Fellegi and A.B. Sunter. A theory for record linkage. In *Journal of the American Statistical Association*, pages 1183–1210, 1969.
6. L. Gill. Ox-link: The oxford medical record linkage system. In *Record Linkage Techniques*, pages 15–33. National Academy Press, 1997.
7. I.J. Good. The population frequencies of species and the estimation of population parameters. In *Journal of the American Statistical Association*, pages 237–264, 1953.
8. L. Gu, R. Baxter, D. Vickers, and C. Rainsford. *Record Linkage: current practice and future directions*, April 2003. Technical Report 03/83, CSIRO Mathematical and Information Sciences, Canberra, Australia.
9. David Heckerman. *Probabilistic Similarity Networks*, 1990. Ph.D. thesis, Stanford University.
10. M.A. Hernandez and S.J. Stolfo. The merge/purge problem for large databases. In *Proceedings of the SIGMOD Conference, San Jose*, pages 127–138, 1995.
11. Alvaro E. Monge and Charles Elkan. An efficient domain-independent algorithm for detecting approximately duplicate database records. In *Research Issues on Data Mining and Knowledge Discovery*, 1997.
12. Hanna Pasula, Bhaskara Marthi, Brian Milch, Stuart Russell, and Ilya Shpitser. Identity uncertainty and citation matching. In *Proceedings of the Neural Information Processing Systems (NIPS-02)*, 2002.
13. James L. Peterson. A note on undetected typing errors. In *Communication of the ACM*, volume 29(7), pages 633–637, July 1986.
14. J.J. Pollock and A. Zamora. Automatic spelling correction in scientific and scholarly text. In *ACM Computing Surveys*, volume 27(4), pages 358–368, 1984.
15. Pradeep Ravikumar and William Cohen. A hierarchical graphical model for record linkage. In *Proceeding of Twentieth Conf. on Uncertainity in Artificial Intelligence (UAI-04)*, 2004.
16. Rita Sharma and David Poole. Efficient inference in large discrete domains. In *Proceeding of Nineteenth Conf. on Uncertainity in Artificial Intelligence (UAI-03)*, 2003.
17. W.E. Winkler. *The state of record linkage and current research problems*, 1999. Technical Report, Statistical Research Division, U.S. Census Bureau, Washington, DC.
18. N.L. Zhang and David Poole. A simple approach to Bayesian network computation. In *Proceeding of the 10th Candian Conference on Artificial Intelligence*, pages 171–178, 1994.