Identity Uncertainty

Stuart Russell and Hanna Pasula
Computer Science Division
University of California, Berkeley, CA 94720, USA
russell@cs.berkeley.edu

September 3, 2002

Abstract

We are often uncertain about the identity of objects. This phenomenon appears in theories of object persistence in early childhood; in the well-known Morning Star/Evening Star example; in tracking and data association systems for radar; in security systems based on personal identification; in almost all database systems and text-based information systems; and in many aspects of our everyday lives. The paper presents a formal probabilistic approach to reasoning about identity under uncertainty in the framework of first-order logics of probability, with application to wide-area freeway traffic monitoring and citation matching.

1 Introduction

Object identification—the task of deciding that two observed objects are in fact one and the same object—is a fundamental requirement for any situated agent that reasons about individuals. The aim of this paper is to explore uncertain reasoning about object identity, with the ultimate aim of understanding how internal representations of individual objects should be created and manipulated within a complete intelligent agent.

The existence of individuals is central to our conceptualization of the world. While *object recognition* deals with assigning objects to categories, such as 1988 Toyota Celicas or adult humans, *object identification* deals with recognizing specific individuals, such as one's car or one's spouse. One can have specific relations to individuals, such as ownership or marriage. Hence, it is often important to be fairly certain about the identity of the particular objects one encounters.

Formally speaking, identity is expressed by the equality operator of first-order logic. Having detected an object C in a parking lot, one might be interested in whether $C = MyCar$. Because mistaken identity is always a possibility, this becomes a question of the *probability of identity*: $P(C = MyCar \mid \text{all available evidence})$.

For many years, philosophers have pondered this question (or rather, more general versions thereof). The famous example of the Evening Star and Morning Star, long thought to be distinct but in fact the same object (Venus), shows that one can be mistaken about identity and that it is really a matter of inference from accumulated evidence. The doctrine of the Indiscernibility of

Identicals states, uncontroversially, that if $a = b$, $F(a) \Leftrightarrow F(b)$ for any predication F. The Identity of Indiscernibles states the converse, thereby pointing out that identity, and hence the choice of what one considers to be individuals, is strongly related to the choice of predications one wishes to make.

There has been little analytical work on object identity in AI or, it seems, in probability and statistics. The one area where the issue has been addressed is that of tracking and data association, as we discuss in Section 5. This paper sketches a more general framework, in which the data association model can be accommodated and extended. It draws extensively on the previous work of the author and his colleagues and students (Huang & Russell, 1997; Pasula, Russell, Ostland, & Ritov, 1999; Russell, 1999a, 1999b; Pasula & Russell, 2001).

We begin in Section 2 with what is perhaps the simplest possible scenario involving identity uncertainty, and we apply straightforward mathematical techniques to solve it. Section 3 developes a formal language, combining probability and first-order logic, that allows identity questions to be posed in general settings with many objects, properties, and relations. Section 4 shows how inference for this formal language may be achieved using Markov chain Monte Carlo (MCMC) algorithms, where the Markov chain is defined on a state space consisting of the relational models of the first-order language. Section 5 shows how MCMC may be used for reasoning about vehicle identities in the context of a large-scale traffic surveillance system. Section 6 describes a more complex application to the problem of *citation matching*—deciding whether two citations refer to the same paper.

2 Balls and Urns

The authors of papers on the foundations of probability are fond of drawing balls from urns. Here, we consider the following scenario. The experimenter is given an urn containing an unknown collection of balls, and k are performed. In each, a ball is removed from the urn, its colour is examined, and it is returned to the urn, which is then shaken vigorously (see Figure 1). The standard question in the literature is, "What is the probability that the next ball drawn is red?" Instead, we will ask, "How many balls are in the urn?"

Intuitively, it is clear that reasoning about identity is required in order to answer this question. Consider the case where the experimenter can determine that the first five balls drawn are identical—that is, the *same* ball is drawn five times. Then it seems reasonable to infer that the urn probably contains just one ball.

A formal probability model for this problem is defined on the following random variables:

- N is the number of balls, with prior $P(N)$.
- $\mathbf{C} = C_1, \ldots, C_N$ are the true colours of the N balls in the urn, each with an identical prior $P(C_i)$. Notice that the number of these random variables is unknown.
- $\iota = \iota_1, \ldots, \iota_k$ are the true identities of the balls drawn; the value of each is an integer in $\{1, \ldots, N\}$. By assumption, these variables are independent and uniformly distributed.
- $\mathbf{Y} = Y_1, \ldots, Y_k$ are the observed colours of the k balls drawn, each distributed according to an identical sensor model $P(Y_j | C_{\iota_j})$. Notice that ι_j is unknown, hence the identity of the

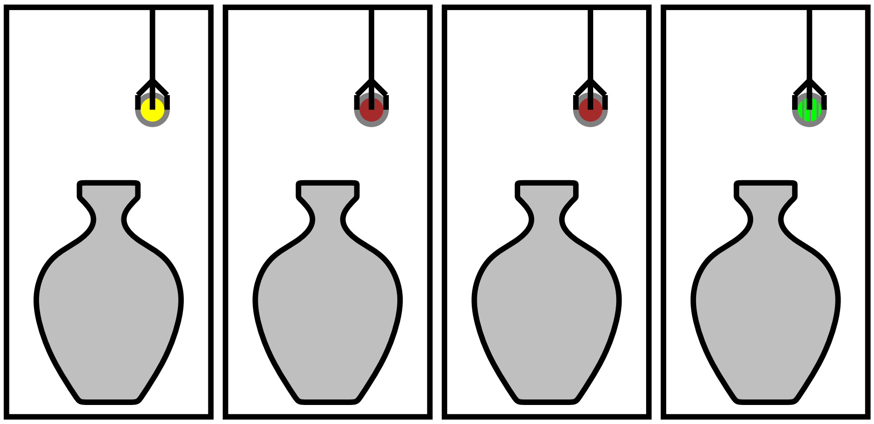


Figure 1: An urn-and-ball experiment: on each trial, a ball is removed from the urn, its colour is examined, and it is returned to the urn, which is then shaken vigorously.

variable C_{ι_j} is unknown.

The joint distribution is then

$$P(N, \mathbf{C}, \iota, \mathbf{Y}) = \\ P(N) \left(\prod_{i=1}^{N} P(C_i)\right) \left(\frac{1}{N}\right)^k \prod_{j=1}^{k} P(Y_j|C_{\iota_j}) \tag{1}$$

Notice how the identity of the balls enters into this expression, by selecting the true colours of the observed balls and hence correlating the observed colours.

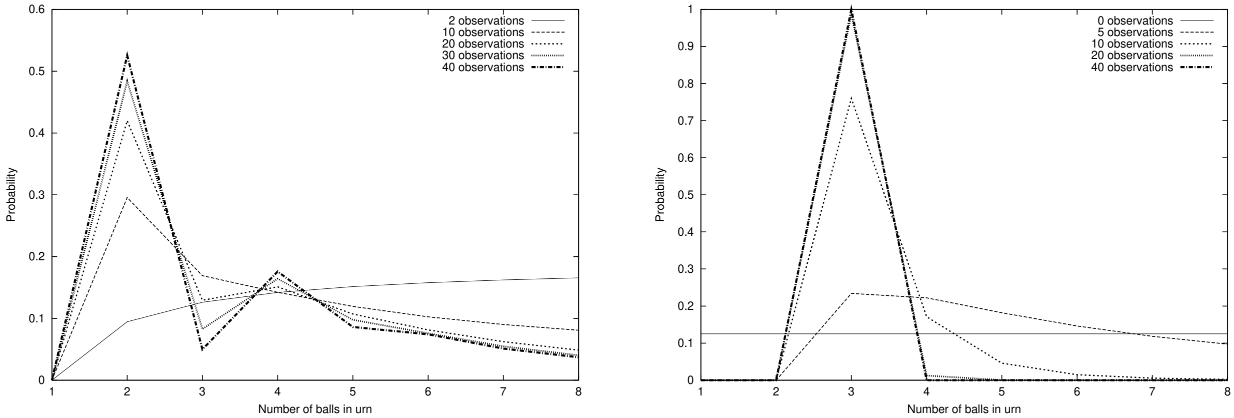
The question, "How many balls are there?" is answered by the posterior distribution $P(N|\mathbf{Y}=\mathbf{y})$, given by

$$\alpha P(N)\left(\frac{1}{N}\right)^k \sum_{\mathbf{c}} \left(\prod_{i=1}^{N} P(C_i)\right) \sum_{\iota} \prod_{j=1}^{k} P(Y_j|C_{\iota_j}) \tag{2}$$

(In this expression, the sum over $\mathbf{c}$ may be replaced by an integral if the colour distribution is continuous.) When $k < N$, this expression can be simplified somewhat, but in general the summation over all N^k identity assignments must be calculated. This is characteristic of all the identity uncertainty problems we have studied.

One question of particular interest is whether the number of balls can be identified exactly in the limit of many experiments, i.e., as $k \to \infty$. The answer to this question depends on the nature of the colour distribution and sensor model. If there can be indistinguishable but nonidentical balls—for example, if the colour distribution is discrete and the sensor model is exact—then it is impossible for any set of observations to distinguish among the situations with N, $2N$, $3N$, ... balls. In that case, the posterior will tend to a collection of delta functions placed at values of N equal to multiples of the lowest common multiple of the denominators of the reduced-form fractions of each colour. For example, if the observed fractions of two colours in the limit are 1/2 and 1/2, then there could be 2, 4, 6, ... balls. (Figure 2(a)). On the other hand, if the colour distribution is continuous, indistinguishable balls occur with probability zero and the posterior converges to the true number as $k \to \infty$ (Figure 2(b)).

In the remainder of the paper, we will outline an approach to reasoning about identity uncertainty in which problems of this kind can be posed in a declarative language and solved by general-purpose inference methods such as Markov chain Monte Carlo. The results of this approach for the case of indistinguishable balls are shown in Figure 3.



(a) (b)

Figure 2: Exact posteriors over the number of balls, for a case in which $P(N)$ is uniform on $\{1, \ldots, 8\}$ and observation is exact, for various numbers of observations k. (a) When balls may be indistinguishable, the posterior converges to a multimodal distribution (here, the observed balls are 1/2 red and 1/2 blue). Notice that even though the prior is uniform and situations with 2, 4, 6, and 8 balls are observationally indistinguishable, the posteriors still decrease for larger numbers of balls. (b) When all balls are distinguishable, the posterior converges to true number of balls (here, 3).

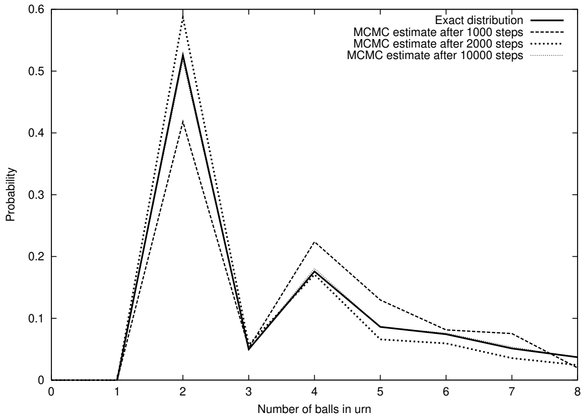


Figure 3: Approximate posterior computed by Markov chain Monte Carlo for the situation described in Figure 2(a).

3 A formal theory

The preceding section gave an example and developed a specific formula for that example. In this section, we describe a general representation language that combines probability theory with a restricted fragment of first-order logic, sufficient to represent many cases of identity uncertainty. Given this language, a single algorithm can be used to answer questions about all such cases.

The basic principles of first-order probabilistic logic (FOPL) were given by (Halpern, 1990). Each model structure of a FOPL knowledge base should be viewed as a probability measure over the possible worlds (logical models) defined by the constant, function, and predicate symbols of the knowledge base. Entailment between probabilistic assertions is then defined identically to ordinary logical entailment, i.e., via truth in all model stuctures. Given a complete knowledge base—one that fixes a unique model structure—the probability of a sentence is defined to be the sum of probabilities assigned to all the possible worlds in which that sentence is true.

We extend the relational probability models of (Koller & Pfeffer, 2000) as follows:

- A set $\mathcal{C}$ of *classes* denoting sets of objects, related by subclass/superclass relations. Each class has an associated prior distribution over the cardinality of the class.

- A set $\mathcal{I}$ of *named instances* denoting objects, each an instance of one class.

- A set $\mathcal{B}$ of *simple attributes* denoting functions. Each simple attribute B has a domain type $Dom[B] \in \mathcal{C}$ and a range that is a finite, enumerated set of values $Val[B]$.

- A set $\mathcal{A}$ of *complex attributes* denoting relations. Each complex attribute A has a domain type $Dom[A] \in \mathcal{C}$ and a range type $Range[A] \in \mathcal{C}$. We allow for *reference uncertainty*, i.e., the values of each complex attribute may be unknown.

- With each complex attribute A, we associate a simple *reference attribute* $ref[A]$, such that $Val[ref[A]]$ is a finite, enumerated set of named instances. (Where no confusion arises, we may drop the $ref[\cdot]$ and use the attribute name itself.)

- A special relational attribute "=", standing for identity in the standard logical sense.

- A set of conditional probability models $P(B|Pa[B])$ for the simple attributes. A *parent chain* for a simple attribute is a nonempty chain of (appropriately typed) attributes $\sigma = A_1.\cdots.A_n.B'$, where B' is a simple attribute. $Pa[B]$ is the set of B's *parents*, namely the set of all instance variables reached by any parent chain for all possible combinations of values of all the uncertain complex attributes, as well as all the reference variables for those attributes.

- Dependencies are expressed as before by a conditional distribution $P(ref[A]|Pa[ref[A]])$. The parents of $ref[A]$ are those attributes or attribute chains that influence the choice of an instance as the value of attribute A.

We also use slight variants of two additional concepts from (Pfeffer, 2000). The first is *number uncertainty*, which allows for multi-valued complex attributes of uncertain cardinality. We define each such attribute A as a relation rather than a function, and we associate with it a simple attribute $\#[A]$ (i.e., the number of values of A) with a domain type $Dom[A]$ and a range

$\{0, 1, \ldots, \max \#[A]\}$. The second concept is the *same-as* statement, which, when applied two chains of complex attributes, constrains those chains to be equivalent.

Most applications of FOPL assume that there is no reference uncertainty, and that every instance is distinct (unique names assumption, hence no identity uncertainty). Thus, a possible world is defined by the values of the *instance variables*—the simple attributes for all named instances—and the knowledge base specifies a complete distribution over all possible worlds. With reference uncertainty, possible worlds also vary according to the relations that hold among objects, which determine in turn the probabilistic influences among instance variables. With identity uncertainty, we also must specify which instances in the representation map to which objects in the possible world, and how many objects the world contains—thus, the possible world is a complete structure and interpretation as in first-order predicate calculus with equality. This means that the probability of any given sentence is given by summing over all possible identity relations and numbers of objects, all possible relational structures among objects, and all possible values for the instance variables.

To represent the balls-and-urn example, we need two classes, $Ball$ and $Observation$. A prior is specified for the cardinality of the $Ball$ class. There is one complex attribute, $generated\_by$, which maps an $Observation$ to the $Vehicle$ that generated it. Each $Observation$ of a vehicle reports a $colour$, which depends (through the sensor model) on the $colour$ of the corresponding $Ball$. For each observation, a new $Observation$ instance is added to the KB with its colour set to the measured value, and with a new $Ball$ instance attached. A query can be issued to the inference algorithm asking for the posterior cardinality of the $Ball$ class.

4 MCMC on logical models

Exact inference for full FOPL languages is undecidable; for decidable fragments with reference uncertainty, the complexity is very high. Initial experiments suggest, however, that approximate inference using MCMC is a promising approach (Pasula & Russell, 2001).

MCMC (Gilks, Richardson, & Spiegelhalter, 1996) generates samples from a posterior distribution $\pi(x)$ over possible worlds x by defining a Markov chain whose states are the worlds x and whose *stationary distribution* is $\pi(x)$. In the *Metropolis–Hastings* method (henceforth M-H), transitions in the Markov chain are constructed in two steps:

- Given the current state x, a candidate next state is generated from the *proposal distribution* $q(x'|x)$, which may be (more or less) arbitrary.
- The transition to x' is not automatic, but occurs with an *acceptance probability* defined by

$$\alpha(x'|x) = min\left(1, \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}\right)$$

It is not necessary that all the variables of state x be updated simultaneously, in a single transition function. *Single-component* M-H alters each variable in turn. It is also possible to factor q into separate transition functions for various subsets of variables. Provided that q is defined in such a way that the chain is ergodic, this transition mechanism defines a Markov chain whose stationary distribution is $\pi(x)$.

The Gibbs sampling algorithm for Bayesian networks (Pearl, 1988) is a special case of Metropolis–Hastings in which the proposal distribution samples a single variable X_i using the distribution $P(X_i|\mathbf{mb}(X_i))$, where $\mathbf{mb}(X_i)$ denotes the current values of the variables in the Markov blanket of X_i (its parents $Pa[X_i]$, children Y_j, and children's other parents). In this case, the acceptance probability is always 1. One can show easily that

$$P(X_i|\mathbf{mb}(X_i)) = \alpha P(X_i|Pa[X_i])) \prod_j P(Y_j|Pa[Y_j]) \tag{3}$$

Gibbs sampling is very simple and also *local*: transitions are generated referring only to parts of the model directly connected to the variable in question. Hence, *the cost per transition is typically independent of model size*. M-H sampling is also typically local because all the parts of the model that are not changed by the transition cancel in the ratio $\pi(x')/\pi(x)$. In particular, if the proposal concerns a single variable X_i, this ratio reduces to $P(x_i'|\mathbf{mb}(X_i))/P(x_i|\mathbf{mb}(X_i))$, where x_i' is the proposed value of X_i and x_i is its current value. The M-H algorithm, unlike Gibbs, has the added advantage that the transition may often be computed without referring to the other values of X_i at all, as we will see.

MCMC can be applied to FOPL inference by definng the state space to be the set of possible worlds, as described above. For the balls-and-urn example, transitions may modify the true colours of the balls, the identities of the sampled balls, or the number of balls. For cases where the true posterior converges to a single delta function as $k \to \infty$, MCMC mixes well and the aprpoximation is very accurate. (It should be possible to give good convergence bounds for this case.) For cases with ambiguous posteriors, MCMC mixing with a simple "random walk" proposal on N will not converge very quickly because, for large k, the chain will become trapped in one of the peaks in teh posterior distribution. At present it is not known if the approximation problem is intractable or whether a better proposal can be designed.

5 Traffic surveillance

We have developed one large-scale application of MCMC for identity uncertainty—a traffic surveillance system that uses multiple cameras to asses the state of a large freeway network (Pasula et al., 1999). The sensors used in this project are video cameras placed on poles beside the freeway (Figure 4). Observations made at multiple cameras must be combined to track vehicles through the freeway network. Object identification is required for two purposes: first, to measure *link travel time*—the actual time taken for vehicles to travel between two fixed points on the freeway network; and second, to provide *origin/destination* (O/D) counts—the total number of vehicles traveling between any two points on the network in a given time interval.

Combining multiple observations into tracks is a task addressed by the *data association* field—see (Bar-Shalom & Fortmann, 1988; Bar-Shalom, 1992). The intractability of exact data association inference (Cox, 1993) has led to many approximate methods. In our approach, MCMC is used to sample from possible track histories for all vehicles. MCMC transitions simply switch two tracks at some point in their histories. For two cameras, this gives a polynomial-time approximation (Jerrum & Sinclair, 1997)—the first approximation result for data association. Our experiments show that MCMC converges quickly—in 100 samples or so—as each new vehicle is added, even in state spaces with thousands of vehicles.

(a) (b)

Figure 4: Images from two surveillance cameras roughly two miles apart on Highway 99 in Sacramento, California. (a) From the upstream camera. (b) From the downstream camera. Are the two boxed vehicles the same?

Figure 5 shows a simulated freeway network. The aim is to estimate the origin-destination counts between the two entry points and the three exit points. This requires tracking each object across the entire network. In this task, we compare two algorithms: Huang–Russell (Huang & Russell, 1997) and MCMC. The Huang–Russell algorithm gives a good approximation to pairwise identity probabilities, and then concatenates these identities to track vehicles over several cameras. Because MCMC does joint estimation over the whole network, we expect it to perform much better than Huang–Russell as the measurement noise increases. The results in Figure 6 bear this out. The second graph in the figure also shows that both methods are unable to find exact matches accurately for high levels of noise. The ability of the MCMC algorithm to recover reasonable counts despite the failure of individual matches suggests that its samples contain a reasonable amount of information about the ensemble behaviour of the vehicles.

6 Citation matching

Citation matching is the problem currently handled by systems such as Citeseer (Lawrence, Bollacker, & Giles, 1999).[1] Such systems process a large number of scientific publications to extract their citation lists. By grouping together all co-referring citations (and, if possible, linking to the actual cited paper), the system constructs a database of "paper" entities linked by the "$cites(p_1, p_2)$" relation. This is an example of the general problem of determining the existence of a set of objects, and their properties and relations, given a collection of "raw" text data (or other perceptual data); this problem is faced by intelligence analysts and intelligent agents as well as by citation systems.

Figure 7 shows two citations that probably refer to the same paper, despite many superficial differences. Citations appear in many formats and are rife with errors of all kinds. As a result, Citeseer—which is specifically designed to overcome such problems—currently lists more than

[1]See `citeseer.nj.nec.com`. Citeseer is now known as ResearchIndex.

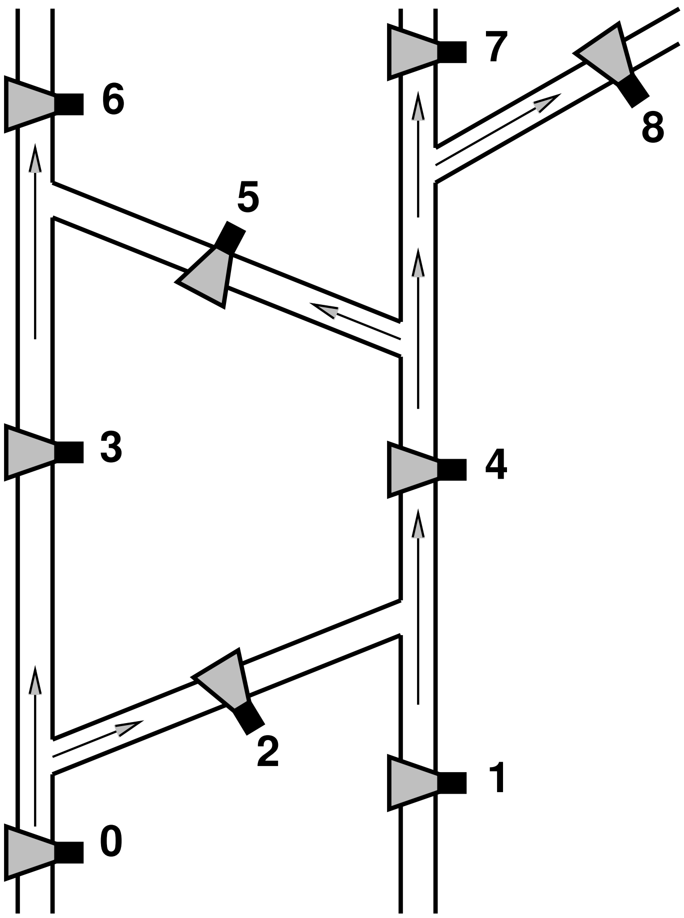
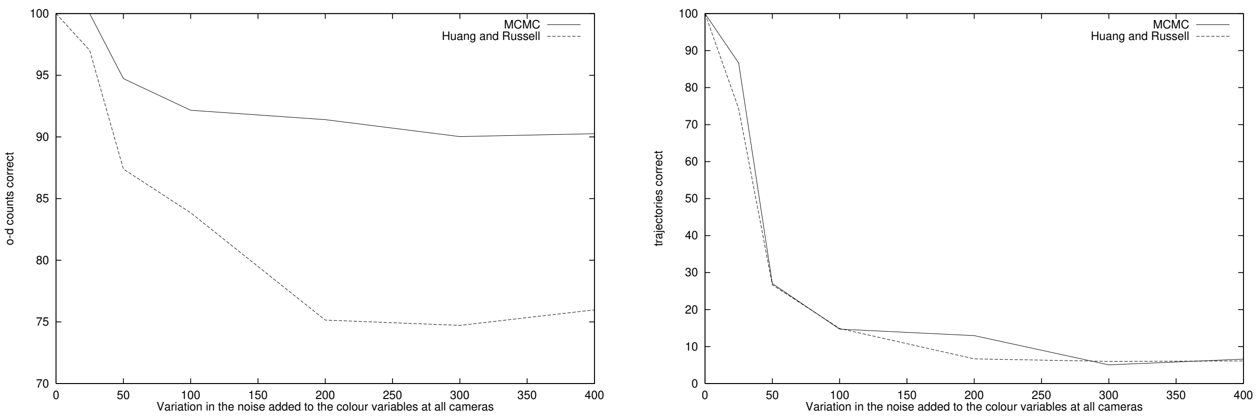


Figure 5: Schematic diagram of simulated freeway network with nine cameras.



(a) (b)

Figure 6: (a) The average accuracy, measured over entrances and exits, of the origin-destination counts for the MCMC and Huang–Russell algorithms, as a function of the colour noise variance. (b) The same graphs showing accuracy of vehicle matches.

[Lashkari et al 94] Collaborative Interface Agents, Yezdi Lashkari, Max Metral, and Pattie Maes, Proceedings of the Twelfth National Conference on Articial Intelligence, MIT Press, Cambridge, MA, 1994.

Metral M. Lashkari, Y. and P. Maes. Collaborative interface agents. In Conference of the American Association for Artificial Intelligence, Seattle, WA, August 1994.

Figure 7: Two citations that probably refer to the same paper.

100 distinct AI textbooks published by Russell and Norvig on or around 1995, from roughly 1000 citations. This particular outcome is a minor annoyance, but if the question were "How many terrorist cells operate in Germany?" or "How many nuclear missiles are heading for the US?" it would not do to have answers in error by a factor of 100.

Identity uncertainty of this kind has been studied independently in several fields. *Record linkage* (Fellegi & Sunter, 1969) is a method for matching up the records in two files, as might be required when merging two databases. For each pair of records, a *comparison vector* is computed that encodes the ways in which the records do and do not match up. EM is used to learn a naive-Bayes distribution over this vector for both matched and unmatched record pairs, so that the pairwise match probability can then be calculated using Bayes' rule. Linkage decisions are typically made in a greedy fashion based on closest match and/or a probability threshold, so the overall process is order-dependent and may be inconsistent. The model does not provide for a principled way to combine matched records. A richer probability model is developed by Cohen *et al* (Cohen, Kautz, & McAllester, 2000), who model the database as a combination of some "original" records that are correct and some number of erroneous versions. They give an efficient greedy algorithm for finding a single locally optimal assignment of records into groups. More general patterns of identity occur in natural language text, where the problem of *anaphora resolution* involves determining whether phrases (especially pronouns) co-refer; some recent work (Charniak & Goldman, 1993) has used an early form of relational probability model, although with a somewhat counterintuitive semantics. Citeseer is the best-known example of work on citation matching (Lawrence et al., 1999). The system groups citations using a form of greedy agglomerative clustering based on a text similarity metric (see Section 6.3).

With the exception of (Charniak & Goldman, 1993), all of the preceding systems have used domain-specific algorithms and data structures; the probabilistic approaches are based on a fixed probability model. Our approach uses a declarative language to specify the model, extending standard RPMs as described earlier. The system also incorporates an EM algorithm for learning the local probability models, such as the model of how author names are abbreviated, reordered, and misspelt in citations. The overall approach results in two to three times fewer errors than were made by CiteSeer on four test databases.

6.1 RPMs for citations

Figure 8 outlines an RPM for the example citations of Figure 7. There are four classes, the self-explanatory *Author*, *Paper*, and *Citation*, as well as *AuthorAsCited*, which represents not actual authors, but author names as they appear when cited. Each citation we wish to match leads to the creation of a *Citation* instance; instances of the remaining three classes are then added as needed to fill all the complex attributes. E.g., for the first citation of Figure 7, we would create a *Citation*

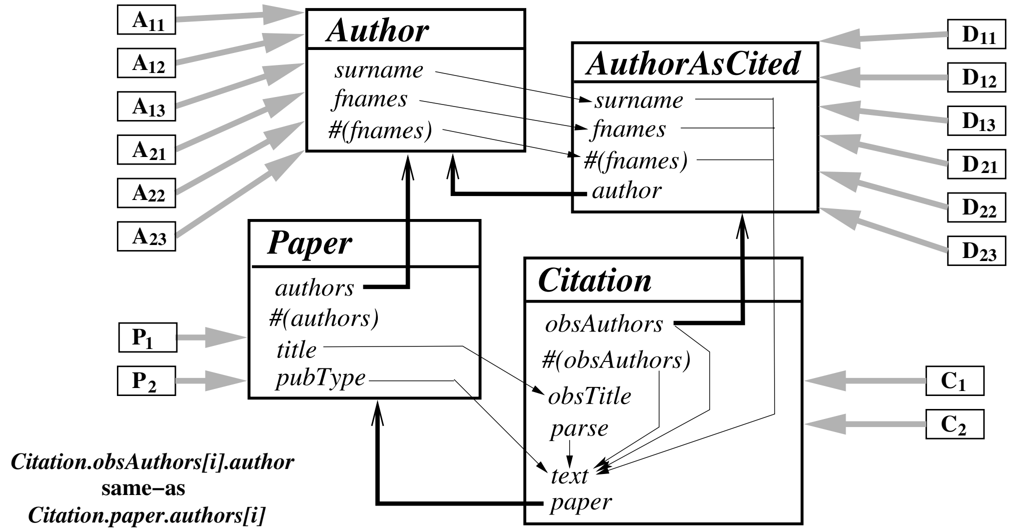


Figure 8: An RPM for our Citeseer example. The large rectangles represent classes: the dark arrows indicate the ranges of their complex attributes, and the light arrows lay out all the probabilistic dependencies of their basic attributes. The small rectangles represent instances, linked to their classes with thick grey arrows. There is a single *same-as* statement.

instance C_1, set its *text* attribute to the string "`Metral M. ...August 1994.`", and set its *paper* attribute to a newly created *Paper* instance, which we will call P_1. We would then introduce $\max(\#[author])$ (here only 3, for simplicity) *AuthorAsCited* instances (D_{11}, D_{12}, and D_{13}) to fill the P_1.*obsAuthors* (i.e., observed authors) attribute, and an equal number of *Author* instances (A_{11}, A_{12}, and A_{13}) to fill both the P_1.*authors[i]* and the D_{1i}.*author* attributes. (The authors of the actual paper and the cited authors are constrained to be equal by the *same-as* statement.) Assuming (for now) that the value of C_1.*parse* is observed, we can set the values of all the basic attributes of the *Citation* and *AuthorAsCited* instances. (E.g., given the correct parse, D_{11}.*surname* would be set to Lashkari, and D_{12}.*fnames* would be set to (Max)). The remaining basic attributes — those of the *Paper* and *Author* instances — represent the "true" attributes of those objects, and their values are unobserved.

The Bayesian network equivalent to our RPM is shown in Figure 9. We will now fill in the

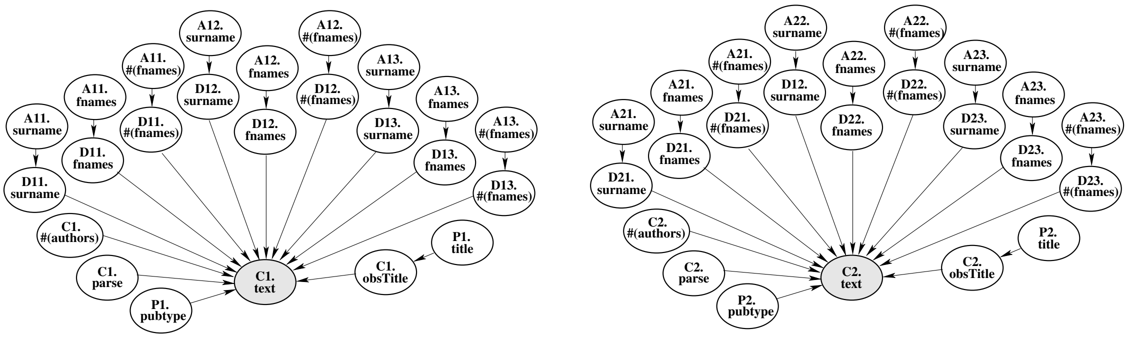


Figure 9: The Bayesian network equivalent to our RPM, assuming $C_1 \neq C_2$.

details of its conditional probability models. Our priors over the "true" attributes are constructed off-line, using the following resources: the 2000 Census data on US names, a large A.I. BibTeX bibliography, and a hand-parsed collection of 500 citations. We learn several bigram models (actually, linear combinations of a bigram model and a unigram model): letter-based models of first names, surnames, and title words, as well as higher-level models of various parts of the citation string. More specifically, the values of *Author.fnames* and *Author.surname* are modeled as having a a 0.9 chance of being drawn from the relevant US census file, and a 0.1 chance of being generated using a bigram model learned from that file. The prior over *Paper.title*s is defined using a two-tier bigram model constructed using the bibliography, while the distributions over *Author.#(fnames)*, *Paper.#(authors)*, and *Paper.pubType* [2] are derived from our hand-parsed file. The conditional distributions of the "observed" variables given their true values (i.e., the corruption models of *Citation.obsTitle*, *AuthorAsCited.surname*, and *AuthorAsCited.fnames*) are modeled as noisy channels where each letter, or word, has a small probability of being deleted, or, alternatively, changed, and there is also a small probability of insertion. *AuthorAsCited.fnames* may also be abbreviated as an initial. The parameters of the corruption models are learnt online, using stochastic EM.

Let us now return to *Citation.parse*, which cannot be an observed variable, since citation parsing, or even citation subfield extraction, is an unsolved problem. It is therefore fortunate that our approach lets us handle uncertainty over parses so naturally. The state space of *Citation.parse* has two different components. First of all, it keeps track of the citation style, defined as the ordering of the author and title subfields, as well as the format in which the author names are written. The prior over styles is learned using our hand-segmented file. Secondly, it keeps track of the segmentation of *Citation.text*, which is divided into an author segment, a title segment, and three filler segments (one before, one after, and one in between.) We assume a uniform distribution over segmentations. *Citation.parse* greatly constrains *Citation.text*: the title segment of *Citation.text* must match the value of *Citation.obsTitle*, while its author segment must match the combined values of the simple attributes of *Citation.obsAuthors*. The distributions over the remaining three segments of *Citation.text* are defined using bigram models, with the model used for the final segment chosen depending on the publication type. These models were, once more, learned using our pre-segmented file.

In our application, any two citations may or may not refer to the same paper. Thus, for citations C_1 and C_2, there is uncertainty as to whether the corresponding papers P_1 and P_2 are in fact the same object. If they are the same, they will share one set of basic attributes; if they are distinct, there will be two sets. As before, each model specifies an assignment ι. For example, if P_1 and P_2 are the only terms, and they co-refer, then ι is $\{\{P_1, P_2\}\}$; if they do not co-refer, then ι is $\{\{P_1\}, \{P_2\}\}$.

We define a probability model for the space of extended possible worlds by specifying a prior $P(\iota)$ and a conditional distribution over basic attributes given ι.[3] As in standard RPMs, we assume that the class of every instance is known. Hence, we can simplify $P(\iota)$ further by rewriting it as the product of the $P(\iota_C)$s of all the classes $C \in \mathbf{C}$. We distinguish two cases:

- For some classes (such as the citations themselves), the unique names assumptions remains appropriate. Thus, we define $P(\iota_{Citation})$ to assign a probability of 1.0 to the one assignment

[2]Publication types range over {article, conference paper, book, thesis, and tech report}

[3]We could also allow ι to depend on the attribute values—e.g., the frequency of citations to a given paper might depend on the fame of the authors—provided we do not introduce cyclic dependencies.

where each citation object is unique.

- For classes such as *Paper* and *Author*, whose elements are subject to identity uncertainty, we specify a prior over the *cardinality* of the class using a high-variance log-normal distribution.[4] Then we make appropriate uniformity assumptions to construct $P(\iota_C)$. Specifically, we assume that each paper is *a priori* equally likely to be cited, and that each author is *a priori* equally likely to write a paper. Here, "*a priori*" means prior to obtaining *any* information about the object in question, so the uniformity assumption is entirely reasonable. With these assumptions, the probability of an assignment $\iota_{C,k,m}$ that maps k named instances to m distinct objects, when C contains n objects, is given by

$$P(\iota_{C,k,m}) = \frac{n!}{(n-m)!} \frac{1}{n^k}$$

When $n > m$, the world contains objects unreferenced by any of the terms. However, these filler objects are obviously irrelevant (if they affected the attributes of some named term, they would have been named as functions of that term.) Therefore, we never have to create them, or worry about their attribute values.

Our model assumes that the cardinalities of the classes are independent. It is straightforward to specify a dependency model using standard Bayesian network semantics, where the network nodes correspond to the cardinality attributes of the classes. E.g., it would be reasonable to let the total number of papers depend on the total number of authors.

6.2 Inference considerations

The state space of our MCMC algorithm is the space of all the possible worlds, where each possible world contains both an assignment ι and the values of all the basic attributes of all the objects. Since the ι is given in each world, the distribution over the attributes can be represented using a Bayesian network as described in earlier. Therefore, the probability of a state is simply $P(\iota)$ multiplied by the probability of the hidden attributes of the network.

Our algorithm uses a factored q function to propose, first, a change to ι, and then, values for all the hidden attributes of all the objects (or clusters in ι) affected by that change. The algorithm for proposing a change in ι_C works as follows:

Select two clusters $a_1, a_2 \in \iota_C$[5]
Create two empty clusters b_1 and b_2
place each instance $i \in a_1 \cup a_2$ u.a.r. into b_1 or b_2
Propose $\iota'_C = \iota_C - \{a1, a2\} \cup \{b1, b2\}$

Given a proposed ι'_C, suggesting values for the hidden attributes boils down to recovering their true values from (possibly) corrupt observations, e.g., guessing the true surname of the author currently known both as "Simth" and "Smith". Since our title and name noise models are symmetric, our basic strategy is to apply these noise models to one of the observed values. In the case of

[4]Other models are possible; for example, in situations where objects appear and disappear, $P(\iota)$ can be modeled implicitly by specifying the arrival, transition, and departure rates (Pasula, Russell, Ostland, & Ritov, 1999).

[5]Note that if the same cluster is picked twice, it will probably be split.

	Face 349 citations, 242 papers	**Reinforcement** 406 citations, 148 papers	**Reasoning** 514 citations, 296 papers	**Constraint** 295 citations, 199 papers
Phrase matching	***94%***	***79%***	***86%***	***89%***
RPM + MCMC	***97%***	***94%***	***96%***	***93%***

Table 1: Results on four Citeseer data sets, for the text matching and MCMC algorithms. The metric used is the percentage of actual citation clusters recovered perfectly; for the MCMC-based algorithm, this is an average over all the MCMC-generated samples.

surnames, we have the additional resource of a dictionary of common names, so, some of the time, we instead pick one of the set of dictionary entries that are within a few corruptions of our observed names. (One must, of course, careful to account for this hybrid approach in our acceptance probability calculations.) Parses are handled differently: we preprocess each citation, organizing its plausible segmentations into a list ordered in terms of descending probability. At runtime, we simply sample from these discrete distributions. Since we assume that boundaries occur only at punctuation marks, and discard segmentations of probability $< 10^{-6}$, the lists are usually quite short. [6] The publication type variables, meanwhile, are not sampled at all. Since their range is so small, we sum them out.

One of the acknowledged flaws of the MCMC algorithm is that it often fails to scale. In this application, as the number of papers increases, the simplest approach — one where the two clusters a_1 and a_2 are picked u.a.r — is likely to lead to many rejected proposals, as most pairs of clusters will have little in common. The resulting Markov chain will mix slowly. Clearly, we would prefer to focus our proposals on those pairs of clusters which are actually likely to exchange their instances. We have implemented an approach based on the efficient clustering algorithm of McCallum et al (McCallum, Nigam, & Ungar, 2000), where a cheap distance metric is used to preprocess a large dataset and fragment it into many *canopies*, or smaller, overlapping sets of elements that have a non-zero probability of matching. We do the same, using word-matching as our metric. Then, at runtime, our $q(x'|x)$ function proposes first a canopy c, and then a pair of clusters u.a.r. from c. ($q(x|x')$ is calculated by summing over all the canopies which contain any of the elements of the two clusters.)

6.3 Experimental Results

We have applied the MCMC-based algorithm to the hand-matched datasets used in (Lawrence et al., 1999). (Each of these datasets contains several hundred citations of machine learning papers, about half of them in clusters ranging in size from two to twenty-one citations.) We have also implemented their *phrase matching* algorithm, a greedy agglomerative clustering method based on a metric that measures the degrees to which the words and phrases of any two citations overlap. (They obtain their "phrases" by segmenting each citation at all punctuation marks, and then taking all the bigrams of all the segments longer than two words.) The results of our comparison are displayed in Figure 1, in terms of the Citeseer error metric. Clearly, the algorithm we have developed

[6] It would also be possible to sample directly from a model such as a hierarchical HMM

easily beats our implementation of phrase matching.

We have also applied our algorithm to a large set of citations referring to the textbook *Artificial Intelligence: A Modern Approach.* It clusters most of them correctly, but there are a couple of notable exceptions. Whenever several citations share the same set of unlikely errors, they are placed together in a separate cluster. This occurs because we do not currently model the fact that erroneous citations are often copied from reference list to reference list. This can be handled by extending the model to include a *CopiedFrom* attribute. More generally, we expect to find a variety of previously unnoticed phenomena in the patterns of data generation by examining the failures of our simple probability models.

7 Conclusions

This paper has introduced the general problem of identity uncertainty as a subject of study for AI and soft computing generally. MCMC was proposed as a promising technique that can be applied to perform approximate inference on general first-order probabilistic logics, by sampling from possible worlds. Many open problems remain. Perhaps the most important is that of finding natural ways to specify distributions over possible identity relations—i.e., how many things are there? How frequently do new things appear? Resolving this issue and combining the solution with existing methods—such as Bayes nets and relational probability models—should open up many new applications.

References

Bar-Shalom, Y. (Ed.). (1992). *Multitarget-multisensor tracking: Advanced applications*. Artech House, Norwood, Massachusetts.

Bar-Shalom, Y., & Fortmann, T. E. (1988). *Tracking and Data Association*. Academic Press, New York.

Charniak, E., & Goldman, R. P. (1993). A Bayesian model of plan recognition. *AAAI*.

Cohen, W., Kautz, H., & McAllester, D. (2000). Hardening soft information sources. In *KDD*.

Cox, I. J. (1993). A review of statistical data association techniques for motion correspondence.. *International Journal of Computer Vision*, *10*, 53–66.

Fellegi, I., & Sunter, A. (1969). A theory for record linkage. In *JASA*.

Gilks, W., Richardson, S., & Spiegelhalter, D. (Eds.). (1996). *Markov chain Monte Carlo in practice*. Chapman and Hall, London.

Halpern, J. Y. (1990). An analysis of first-order logics of probability. *Artificial Intelligence*, *46*(3), 311–350.

Huang, T., & Russell, S. (1997). Object identification in a Bayesian context. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)* Nagoya, Japan. Morgan Kaufmann.

Jerrum, M., & Sinclair, A. (1997). The Markov chain Monte Carlo method. In Hochbaum, D. S. (Ed.), *Approximation Algorithms for NP-hard Problems*. PWS Publishing, Boston.

Koller, D., & Pfeffer, A. (2000). Semantics and inference for recursive probability models. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-00)* Austin, Texas. AAAI Press.

Lawrence, S., Bollacker, K., & Giles, C. L. (1999). Autonomous citation matching. In *Agents*.

McCallum, A., Nigam, K., & Ungar, L. H. (2000). Efficient clustering of high-dimensional data sets with application to reference matching. In *KDD-00*.

Pasula, H., Russell, S., Ostland, M., & Ritov, Y. (1999). Tracking many objects with many sensors. In *IJCAI-99*.

Pasula, H., & Russell, S. (2001). Approximate inference for first-order probab ilistic languages. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)* Seattle. Morgan Kaufmann.

Pasula, H., Russell, S., Ostland, M., & Ritov, Y. (1999). Tracking many objects with many sensors. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)* Stockholm, Sweden. Morgan Kaufmann.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California.

Pfeffer, A. (2000). *Probabilistic Reasoning for Complex Systems*. Ph.D. thesis, Stanford.

Russell, S. (1999a). Expressive probability models for speech recognition and understanding. In *Proc. of the International Workshop on Automatic Speech Recognition and Understanding (ASRU)* Keystone, Colorado. IEEE.

Russell, S. (1999b). Expressive probability models in science. In *Proc. of the 2nd Int'l Conf. on Discovery Science* Tokyo, Japan. Springer Verlag.