Finding the Missing with Integer Programming

Karl Zhu, Andrew Mason, Anthony Downward
Department of Engineering Science
University of Auckland
New Zealand
karl.zhu@auckland.ac.nz

Abstract

Entity resolution (ER) — the task of determining whether multiple document records refer to the same person — is a ubiquitous problem in data analytics. Traditional ER approaches treat all matching decisions independently and only compare attribute similarities between reference pairs. However, this independence assumption omits valuable relational information in situations where the references describe a rich network of relationships between people. Collective ER — where entities are resolved jointly — incorporates this previously ignored information and can predict matches with greater accuracy. This paper describes a real-world problem faced by Parininihi ki Waitōtara (PKW) — a Māori organisation with thousands of missing shareholders — and demonstrates how collective ER can be used to identify their connections. We use Markov logic networks (MLN) to convert our domain knowledge and evidence data into a Markov network. Prediction is made by performing the most probable explanation (MPE) inference on the network, which is equivalent to a Weighted Partial MaxSAT problem in an MLN. MaxSAT problems can be formulated as an integer program (IP) and solved by a mixed IP solver. We perform experiments to empirically demonstrate how the collective ER approach is able to capture the extra relational information in an example PKW problem.

Key words: Collective entity resolution, Markov logic networks, MPE inference, MaxSAT, integer programming

1 Introduction

We are working with the Māori organisation Parininihi ki Waitōtara (PKW) to help them find their missing shareholders. Established in 1976, it is responsible for managing over 20,000 hectares of land in the interests of its shareholders, who had their land returned by the New Zealand government. Currently, over 60% of their shareholders are missing — meaning PKW cannot locate or contact them due to incomplete records. These are some of the most productive lands in the world (PKW 2022), and PKW has over $5M in unclaimed dividends owing to these shareholders. PKW would like to find their shareholders to pay out their dividends and reconnect them to their ancestral land.

One way to find a missing shareholder is to find a non-missing, contactable shareholder that is related to the missing shareholder. One key piece of information that helps us predict whether two shareholders are related is the PKW share transfer data, given in confidence to us by the organisation. Other publicly available data can also serve as evidence; these include Māori land ownership, Māori Land Court applications, and obituaries records. Previous work done by our research group has obtained these data and stored them in a graph database. To demonstrate how they inform our predictions, we introduce an example problem here (and will be used throughout this paper). We replaced real names with suitable substitutes for confidentiality. The graph structure of the example problem is shown in Figure 1.

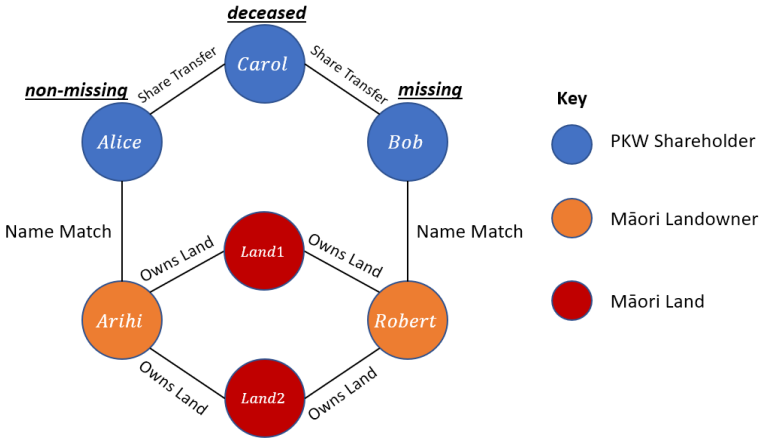

Figure 1: Example problem. The blue nodes show three PKW shareholders: `Alice` (non-missing), `Bob` (missing), and `Carol` (deceased). The two orange nodes are Māori landowners: `Arihi` and `Robert`; each owning shares on the same two pieces of land — which we label `Land1` and `Land2`. There are two name-matches: one between `Alice` and `Arihi`, and one between `Bob` and `Robert`.

In addition to Figure 1, we also know that `Carol` transferred PKW shares to `Alice` and `Bob` upon passing. We can use this evidence to infer that `Alice` and `Bob` are likely referring to relatives. However, is it possible for us to gather more evidence before contacting `Alice` about `Bob`? From Māori Land Online (Māori Land Court 2022), we obtained Māori landowner (MLO) names and their land ownerships. Note that Māori land is collectively owned by a land shares system (Ministry of Māori Development 2022). We can use the collective land ownerships as evidence to infer that `Arihi` and `Robert` may refer to relatives. If `Alice` and `Arihi` are referring to a single person, and `Bob` and `Robert` are referring to another single person, then we successfully gathered additional evidence that the two people are related. But how do we know they are indeed referring to the same person?

Determining whether two records are referring to the same real-world entity (e.g. person) is an *entity resolution* (ER) problem. Traditionally, ER is performed pairwise, where entity matching decisions are made independently for each pair (Fellegi and Sunter 1969). The decision is made by similarity comparisons between record attributes, such as names. Standard fuzzy name-matching algorithms compare how similar the spelling and pronunciations of the name pairs are (e.g. Damerau–Levenshtein distance and Metaphone), and include dictionaries for nicknames. Our research group's previous work has extended this by including dictionaries for English-Māori name borrowings and performed set partitioning on compounded words in Māori names. For the example problem (Figure 1), our

algorithm detected `Arihi` as a Māori name borrowing of `Alice`, and `Bob` as a nickname of `Robert`; thus matching the two respective shareholder and landowner reference pairs.

However, the pairwise ER approach is insufficient as the name match evidence alone is not strong enough to conclude that the names refer to the same person, especially on common names such as `Bob` and `Robert`. We have *relational* data, and there is valuable information to be extracted by incorporating interdependency between entity matching decisions. A key insight is understanding that the 'related' and 'same-person' decisions are interdependent — while the 'related' decision depends on the 'same-person' decision, as previously mentioned, the reverse is also true. This idea is elaborated in Section 2.3. It means that for our example, we should classify four matches simultaneously: 1. whether `Alice` and `Bob` are referring to relatives, 2. whether `Arihi` and `Robert` are referring to relatives, 3. whether `Alice` and `Arihi` are referring to the same person, and 4. whether `Bob` and `Robert` are referring to the same person. This *collective* ER approach removes the independence assumption, captures the extra relational information and improves prediction accuracy (Bhattacharya and Getoor 2007).

The rest of the paper formulates the collective ER task into an optimisation problem. The paper is structured as follows. In Section 2, we first describe Markov logic networks (MLN), a template that combines first-order logic and Markov netowrk — a type of probabilistic graphical model. We then use MLN to construct our example problem into a Markov network in Section 3. In Section 4 we perform the most probable explanation (MPE) inference on the network to find the most probable joint matching decisions. We show that MPE is equivalent to a MaxSAT optimisation problem that can be formulated as an integer program (IP). In Section 5, we perform experiments on our example problem to empirically demonstrate how the collective approach is making better predictions by incorporating the relational information. Future work is described in Section 6.

2 Entity Resolution with Markov Logic

Singla and Domingos (2006) described how collective entity resolution (ER) problems can be modelled by Markov logic networks (MLN). This section summarises their original paper to provide a preliminary understanding of the tools we will use for our PKW problem. We also introduce a modified version of MLN (Section 2.2) for our application.

2.1 Markov Networks

A Markov network is an undirected probabilistic graphical model for the joint distribution of a set of random variables $X = (X_1, X_2, ..., X_n) \in \mathcal{X}$ having the Markov property. It is represented by an undirected graph that has a node for each variable, and an edge between each dependency pair. It is similar to a Bayesian network in its representation of dependencies, with the differences being its undirectedness and its ability to have cycles. The model has a set of potential functions ϕ_k for each maximal clique k in the graph. It is an arbitrary real-valued positive function of the state of its clique. The joint distribution represented by the Markov network is given by

$$\Pr(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}}) \tag{1}$$

where $x_{\{k\}}$ is the set of realisations in clique k. Z is the normalisation constant found by $Z = \sum_{x \in \mathcal{X}} \prod_k \phi_k(x_{\{k\}})$.

We can conveniently represent Markov networks as log-linear models. To do this, we define the weight $\hat{w}_{k,s} \in \mathbb{R}$ and the feature function $\hat{f}_{k,s}(x_{\{k\}}) = 1$ if $x_{\{k\}}$ corresponds to state s in clique k, and 0 otherwise, and let $\phi_k(x_{\{k\}}) = \exp\left(\sum_{s \in S_k} \hat{w}_{k,s} \hat{f}_{k,s}(x_{\{k\}})\right)$. Many $x_{\{k\}}$ realisations can correspond to the same clique state. The distribution is now a normalised exponentiated weighted sum of features of the state:

$$\Pr(X = x) = \frac{1}{Z} \exp\left(\sum_k \sum_{s \in S_k} \hat{w}_{k,s} \hat{f}_{k,s}(x_{\{k\}})\right) \tag{2}$$

where S_k is set of states in clique k.

2.2 Markov Logic

MLN is a probabilistic logic framework that constructs a Markov network with first-order logic. First-order logic (or predicate logic) is a formal language that can be used to express our knowledge of the system. A set of logical formulas is known as a knowledge base (KB). For example, we can express our knowledge that *"records with matching names are referring to the same person"* as the following formula:

$$\forall \mathtt{x_1,x_2}\ \mathtt{NameMatch(x_1,x_2)} \implies \mathtt{SamePerson(x_1,x_2)}$$

where `SamePerson` and `NameMatch` are Boolean predicates (see Section 2.3 and 3.3 for full definitions). If we replace $\mathtt{x_1,x_2}$ with concrete instances such as $\mathtt{Bob_{PKW}}$ and $\mathtt{Robert_{MLO}}$, then the predicate and and the formula are said to be *grounded*.

However, this example also illustrates the problem of expressing our knowledge in first-order logic: inductive reasoning is inherently probable and so the stated formula above is not generally true. If we consider $\mathtt{SamePerson(Bob_{PKW},Robert_{MLO})}$ as a decision variable in an optimisation problem that finds the most probable explanation given the evidence (see Section 4), a first-order KB is a set of hard constraints that define our feasible region. If a solution violates even one formula, it is infeasible and is considered impossible.

Markov logic addresses this issue by softening the constraints: when a solution violates a formula, it becomes less probable but not impossible. Each formula has an associated weight that reflects the strength of the soft constraint. Larger weights correspond to harder constraints. The original Markov logic network (MLN) definition can be found in (Richardson and Domingos 2006). It is a set of pairs (F_i, w_i) where F_i is a formula in the KB and $w_i \in \mathbb{R}$. However, for our application, we have found having variable weights $w_{i,j}$ for each *ground* formula $F_{i,j}$ to be more appropriate. The following is the definition of the modified MLN.

Definition 1. MLN with variable weights

1. MLN has a set of N formulas $F_i,\ i = 1, 2, ..., N$. For each set of realisations x, let $M_i(x)$ be the number of ground formulas for F_i. Let $F_{i,j}$ be the ground formula, and $f_{i,j}$ be the binary feature variable of $F_{i,j}$, for $j = 1, 2, ..., M_i(x)$. $f_{i,j}(x) = 1$ if $F_{i,j}$ is satisfied (interpreted to be `True`), and 0 otherwise. Each $F_{i,j}$ has an associated weight $w_{i,j} \in \mathbb{R}$.

2. The network contains one node for each grounding of each predicate appearing in the formulas. The node's value is the Boolean value of the ground predicate. Edges exist between the nodes if and only if the corresponding ground predicates appear together in at least one of the ground formulas. Thus the maximal cliques of the network will correspond to a ground formula.

MLN is a template to construct a *ground* Markov network from a set of ground formulas. Recall Eq. (2). The MLN ground network has two states for all cliques: `True` and `False` — from the interpretations of the ground formula. For our MLN with variable weights, we know that $\hat{w}_{k,\texttt{True}} = w_{i,j}, \hat{w}_{k,\texttt{False}} = 0$ and $\hat{f}_{k,\texttt{True}} = f_{i,j}\ \forall k$. The resulting distribution for the MLN ground network with variable weights is

$$\Pr(X = x) = \frac{1}{Z} \exp\left(\sum_{i=1}^{N} \sum_{j=1}^{M_i(x)} w_{i,j} f_{i,j}(x)\right) \tag{3}$$

To elucidate Eq. 3: Each ground formula has an associated weight that is rewarded when it is satisfied (interpreted to be `True`) by the set of realisations x and is not rewarded otherwise. A larger sum of rewarded weights corresponds to a higher probability of realising x. The weights can be set manually or trained by standard learning algorithms such as gradient descent (Singla and Domingos 2005).

2.3 Entity Resolution

Entity resolution (also known as *record linkage*) is about determining whether two records are referring to the same real-world entity. We define the $\texttt{SamePerson}(\texttt{x}_1,\texttt{x}_2)$ predicate to be `True` if the record constants $\texttt{x}_1,\texttt{x}_2$ are referring to the same person, and `False` otherwise. Intuitively, `SamePerson` is an equivalence relation and will therefore have the following properties:

Reflexivity: $\forall$`x` `SamePerson(x,x) = True`

Symmetry: $\forall$`x,y` `SamePerson(x,y)` $\iff$ `SamePerson(y,x)`

Transitivity: $\forall$`x,y,z` `SamePerson(x,y) AND SamePerson(y,z)` $\implies$ `SamePerson(x,z)`

These formulas are added to the MLN as hard constraints since they are always true by definition.

We also define the `Related(x,y)` predicate to be `True` if the record constants `x,y` are referring to a pair of people that are relatives to each other, and `False` otherwise. `Related` is a reflexive and symmetric relation, and these relations are added to the MLN as hard constraints. Transitivity also applies to `Related` (a pair with common relatives are related), but should be added as a soft constraint with finite weight to prevent everyone from being classified as related to each other.

`SamePerson` and `Related` matching decisions are interdependent, and their relationship is expressed with the following two formulas:

Predicate equivalence: $\forall\texttt{x}_1,\texttt{x}_2,\texttt{y}_1,\texttt{y}_2$

$\texttt{SamePerson}(\texttt{x}_1,\texttt{x}_2)\ \texttt{AND}\ \texttt{SamePerson}(\texttt{y}_1,\texttt{y}_2) \implies \texttt{Related}(\texttt{x}_1,\texttt{y}_1) \Leftrightarrow \texttt{Related}(\texttt{x}_2,\texttt{y}_2)$

This formula is always true, and so is added to the MLN as a hard constraint.

Reverse predicate equivalence: $\forall\texttt{x}_1,\texttt{x}_2,\texttt{y}_1,\texttt{y}_2$

$\texttt{Related}(\texttt{x}_1,\texttt{y}_1)\ \texttt{AND}\ \texttt{Related}(\texttt{x}_2,\texttt{y}_2) \implies \texttt{SamePerson}(\texttt{x}_1,\texttt{x}_2) \Leftrightarrow \texttt{SamePerson}(\texttt{y}_1,\texttt{y}_2)$

Equivalently, in an easier to understand form:

$\forall\texttt{x}_1,\texttt{x}_2,\texttt{y}_1,\texttt{y}_2\ \texttt{Related}(\texttt{x}_1,\texttt{y}_1)\ \texttt{AND}\ \texttt{Related}(\texttt{x}_2,\texttt{y}_2)\ \texttt{AND}\ \texttt{SamePerson}(\texttt{x}_1,\texttt{x}_2) \implies \texttt{SamePerson}(\texttt{y}_1,\texttt{y}_2)$

$\forall\texttt{x}_1,\texttt{x}_2,\texttt{y}_1,\texttt{y}_2\ \texttt{Related}(\texttt{x}_1,\texttt{y}_1)\ \texttt{AND}\ \texttt{Related}(\texttt{x}_2,\texttt{y}_2)\ \texttt{AND}\ \texttt{SamePerson}(\texttt{y}_1,\texttt{y}_2) \implies \texttt{SamePerson}(\texttt{x}_1,\texttt{x}_2)$

Note that the reverse predicate equivalence formula is not generally true. However, when added to the MLN as a soft constraint, it captures an important pattern in relational data: if two references are referring to a pair of people that each has the same relation to a person, then the pair *might* be the same person. This key observation enables collective ER and outperforms traditional pairwise ER models such as Fellegi-Sunter (1969), which treats all matching decisions as i.i.d..

3 Model Formulation

We demonstrate how our example problem (Figure 1) can be modelled using MLN. We define the predicates, describe the knowledge base, ground them with our example instance, and show its Markov network representation.

3.1 Predicate Definitions

We can classify the predicates into two groups: *evidence* as the knowns and *query* as the unknowns to be determined by inference. Table 1 defines the predicates and the variable types of the argument pairs the predicate takes. Note the `Related` predicate is overloaded to ensure comparison within `Shareholder` and `Landowner` types only. This is suitable as we do not have direct evidence suggesting that a PKW shareholder is related to a Māori landowner.

Table 1: Predicate definitions

	Predicates

Evidence	`ShareTransfer(Shareholder[Transferor], Shareholder[Transferree])`
	`HasLand(Land,Landowner)`
	`NameMatch(Shareholder,Landowner)`

Query	`Related(Shareholder, Shareholder)`
	`Related(Landowner, Landowner)`
	`SamePerson(Shareholder, Landowner)`

Implicit with the evidence and query classification is the *closed-world assumption*, which implies that if an event is not recorded in our database, then it did not happen (Reiter 1981). For example, we do not see a share transfer between $\texttt{Alice}_{\texttt{PKW}}$ and $\texttt{Bob}_{\texttt{PKW}}$ in our database, so we assume there are no share transfers between them. We set $\texttt{ShareTransfer(Alice}_{\texttt{PKW}}\texttt{,Bob}_{\texttt{PKW}}\texttt{) = False}$ and classify it as an evidence term rather than leaving it as unknown and treating it as a query term.

3.2 Knowledge Base (KB)

In addition to the equality axioms stated in Section 2.3, the following formulas make up our KB. We omit the related transitivity soft constraint as it adds unnecessary complexity to our example problem.

Transferees with a common transferor of PKW shares are related

$\forall$`z,x,y` `ShareTransfer(z,x) AND ShareTransfer(z,y)` $\implies$ `Related(x,y)`

Māori landowners with common land ownership are related

$\forall$`l,x,y` `HasLand(l,x) AND HasLand(l,y)` $\implies$ `Related(x,y)`

Records with matching names are referring to the same person

$\forall x_1, x_2$ NameMatch$(x_1, x_2) \implies$ SamePerson(x_1, x_2)

Predicate equivalence (see Section 2.3)

$\forall x_1, x_2, y_1, y_2$ SamePerson(x_1, x_2) AND SamePerson$(y_1, y_2) \implies$
Related$(x_1, y_1) \Leftrightarrow$ Related(x_2, y_2)

Reverse predicate equivalence (see Section 2.3)

$\forall x_1, x_2, y_1, y_2$ Related(x_1, y_1) AND Related$(x_2, y_2) \implies$
SamePerson$(x_1, x_2) \Leftrightarrow$ SamePerson(y_1, y_2)

Query predicate thresholds: We add threshold weights that default the query terms to False since most pairs are not related nor the same person. The evidence must accumulate above this threshold for the queries to be considered True. We use negated terms and set their weights to be positive.

$\forall x, y$ NOT Related(x, y)
$\forall x_1, x_2$ NOT SamePerson(x_1, x_2)

3.3 Groundings

The formulas are grounded with constants (e.g. $\text{Alice}_{\text{PKW}}$) that appear in our problem instance. Combinations that are vacuously true (e.g. NameMatch$(\text{Alice}_{\text{PKW}}, \text{Robert}_{\text{PKW}}) \implies$ SamePerson$(\text{Alice}_{\text{PKW}}, \text{Robert}_{\text{MLO}})$ is True since NameMatch$(\text{Alice}_{\text{PKW}}, \text{Robert}_{\text{MLO}})$ = False by our closed-world assumption) are omitted since they do not affect our query predicates. Table 2 lists our example problem's groundings and their associated weights. Figure 2 shows our groundings represented by a Markov network.

Table 2: Ground formulas and their associated weights for example problem. The weight is rewarded if the formula is satisfied, and not rewarded otherwise. No weights are needed for hard constraints.

Weight	Ground formulas
Hard	SamePerson$(\text{Alice}_{\text{PKW}}, \text{Arihi}_{\text{MLO}})$ AND SamePerson$(\text{Bob}_{\text{PKW}}, \text{Robert}_{\text{MLO}}) \implies$ Related$(\text{Alice}_{\text{PKW}}, \text{Bob}_{\text{PKW}}) \Leftrightarrow$ Related$(\text{Arihi}_{\text{MLO}}, \text{Robert}_{\text{MLO}})$
$w_{1,1}$	ShareTransfer$(\text{Carol}_{\text{PKW}}, \text{Alice}_{\text{PKW}})$ AND ShareTransfer$(\text{Carol}_{\text{PKW}}, \text{Bob}_{\text{PKW}}) \implies$ Related$(\text{Alice}_{\text{PKW}}, \text{Bob}_{\text{PKW}})$
$w_{2,1}$	HasLand(Land1,$\text{Arihi}_{\text{MLO}}$) AND HasLand(Land1,$\text{Robert}_{\text{MLO}}$) $\implies$ Related$(\text{Arihi}_{\text{MLO}}, \text{Robert}_{\text{MLO}})$
$w_{2,2}$	HasLand(Land2,$\text{Arihi}_{\text{MLO}}$) AND HasLand(Land2,$\text{Robert}_{\text{MLO}}$) $\implies$ Related$(\text{Arihi}_{\text{MLO}}, \text{Robert}_{\text{MLO}})$
$w_{3,1}$	NameMatch$(\text{Alice}_{\text{PKW}}, \text{Arihi}_{\text{MLO}}) \implies$ SamePerson$(\text{Alice}_{\text{PKW}}, \text{Arihi}_{\text{MLO}})$
$w_{4,1}$	Related$(\text{Alice}_{\text{PKW}}, \text{Bob}_{\text{PKW}})$ AND Related$(\text{Arihi}_{\text{MLO}}, \text{Robert}_{\text{MLO}}) \implies$ SamePerson$(\text{Alice}_{\text{PKW}}, \text{Arihi}_{\text{MLO}}) \Leftrightarrow$ SamePerson$(\text{Bob}_{\text{PKW}}, \text{Robert}_{\text{MLO}})$
$w_{5,1}$	NOT Related$(\text{Alice}_{\text{PKW}}, \text{Bob}_{\text{PKW}})$
$w_{5,2}$	NOT Related$(\text{Arihi}_{\text{MLO}}, \text{Robert}_{\text{MLO}})$
$w_{6,1}$	NOT SamePerson$(\text{Alice}_{\text{PKW}}, \text{Arihi}_{\text{MLO}})$
$w_{6,2}$	NOT SamePerson$(\text{Bob}_{\text{PKW}}, \text{Robert}_{\text{MLO}})$

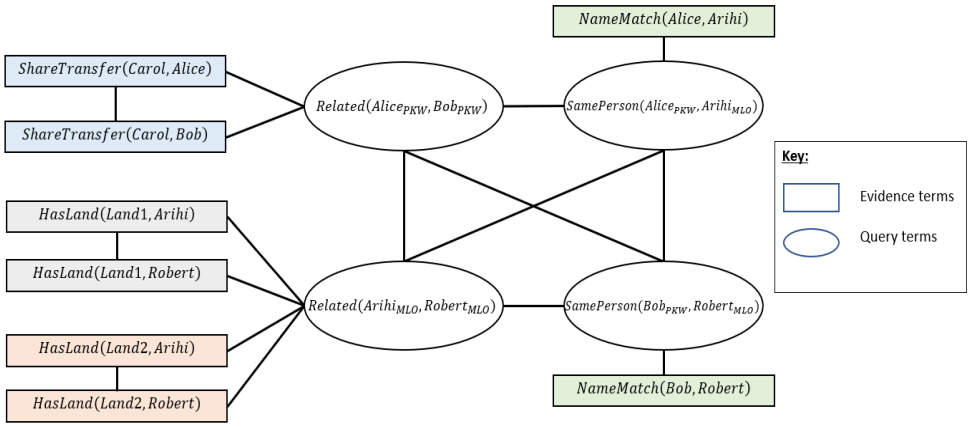

Figure 2: Ground Markov network for the example problem. Each maximal clique corresponds to a ground formula that has an associated weight.

4 Inference and Optimisation

We demonstrate how our example — a collective entity resolution problem modelled by an MLN — can be solved by optimisation. Asking for the best prediction to our `Related` and `SamePerson` terms given the evidence is an instance of most probable explanation (MPE) inference. MPE inference on an MLN is equivalent to a Weighted Partial MaxSAT problem, which can then be formulated as an integer program (IP).

4.1 Most Probable Explanation (MPE)

Recall Eq. (3). Let us partition our set of random variables X into its evidence X_e and query X_q parts. Given our knowns $X_e = x_e$ (share transfers, land ownership, and name matches), MPE seeks to find the joint truth values x_q of *all* our query terms in X_q that will maximise the probability:

$$\max_{X_q} \Pr(X_q \mid X_e = x_e) = \frac{1}{Z} \exp\left(\sum_{i=1}^{N} \sum_{j=1}^{M_i(x_q)} w_{i,j} f_{i,j}(x_q)\right) \tag{4}$$

For our example, X_q would be the set of truth values for $\texttt{Related(Alice}_{\texttt{PKW}}\texttt{,Bob}_{\texttt{PKW}}\texttt{)}$, $\texttt{Related(Arihi}_{\texttt{MLO}}\texttt{,Robert}_{\texttt{MLO}}\texttt{)}$, $\texttt{SamePerson(Alice}_{\texttt{PKW}}\texttt{,Arihi}_{\texttt{MLO}}\texttt{)}$, $\texttt{SamePerson(Bob}_{\texttt{PKW}}\texttt{,Robert}_{\texttt{MLO}}\texttt{)}$.

4.2 MPE as a Weighted Partial MaxSAT Problem

Note that since `exp` is a monotonically increasing function, to maximise $\Pr(X_q|X_e)$ we only need to maximise the `exp`'s argument in Eq. (4):

$$\max_{X_q} \sum_{i=1}^{N} \sum_{j=1}^{M_i(x_q)} w_{i,j} f_{i,j}(x_q)$$

subject to the hard constraints from the equality axioms described in Section 3.2.

This is a *weighted partial maximum satisfiability* (Weighted Partial MaxSAT) problem. MaxSAT is a ubiquitous optimisation problem, and many exact and heuristic solvers exist. The fastest exact MaxSAT solver that won the MaxSAT Evaluations 2022 competition

is CASHWMaxSAT-CorePlus (SAT-2022 2022). It is a hybrid solver that formulates the problem into an integer program (IP) for 'small' instances and solves with the open-source solver SCIP (Bacchus et al. 2022). For our PKW problem, it is suitable to use IP when our graph size is small. Solving large graphs is discussed in Section 6.

4.3 Weighted Partial MaxSAT as an Integer Program (IP)

Weighted Partial MaxSAT problems can be formulated as binary IPs (Li and Manyà 2009). For our groundings in Table 2, the IP formulation is easy to understand we convert our ground formulas into its standard clausal form and simplify the expressions given the evidence. The groundings are in clausal form when all the terms in the formulas are connected by `AND` and `OR`s only. A clause is a statement connected only by `OR`s. A formula that breaks into two clauses will each have half its original weight (Singla and Domingos 2005). For simplification, we first substitute our evidence terms with their truth value and reduce the formula to its simplest form. For example, `True` $\Rightarrow q$ is the same as q. The evidence can be inferred from Figure 1. Next, we sum the weights with the same ground formulas and treat it as a single 'net weight'. A unit clause (clause with one term) with a negated term with positive weight is equivalent to having the term non-negated with negative weight. Table 3 shows the simplified groundings in clausal form.

Table 3: Simplified ground formulas in clausal form. Each formula is a clause, where the terms for each clause are connected by `OR`s. The clauses are connected by `AND`s.

Net weight	**Ground formulas**
Hard	NOT SamePerson(Alice$_{\text{PKW}}$,Arihi$_{\text{MLO}}$) OR NOT SamePerson(Bob$_{\text{PKW}}$,Robert$_{\text{MLO}}$) OR NOT Related(Alice$_{\text{PKW}}$,Bob$_{\text{PKW}}$) OR Related(Arihi$_{\text{MLO}}$,Robert$_{\text{MLO}}$)
Hard	NOT SamePerson(Alice$_{\text{PKW}}$,Arihi$_{\text{MLO}}$) OR NOT SamePerson(Bob$_{\text{PKW}}$,Robert$_{\text{MLO}}$) OR NOT Related(Arihi$_{\text{MLO}}$,Robert$_{\text{MLO}}$) OR Related(Alice$_{\text{PKW}}$,Bob$_{\text{PKW}}$)
$w_{1,1} - w_{5,1}$	Related(Alice$_{\text{PKW}}$,Bob$_{\text{PKW}}$)
$w_{2,1} + w_{2,2} - w_{5,2}$	Related(Arihi$_{\text{MLO}}$,Robert$_{\text{MLO}}$)
$w_{3,1} - w_{6,1}$	SamePerson(Alice$_{\text{PKW}}$,Arihi$_{\text{MLO}}$)
$w_{3,2} - w_{6,2}$	SamePerson(Bob$_{\text{PKW}}$,Robert$_{\text{MLO}}$)
$\frac{1}{2}w_{4,1}$	NOT Related(Alice$_{\text{PKW}}$,Bob$_{\text{PKW}}$) OR NOT Related(Arihi$_{\text{MLO}}$,Robert$_{\text{MLO}}$) OR NOT SamePerson(Alice$_{\text{PKW}}$,Arihi$_{\text{MLO}}$) OR SamePerson(Bob$_{\text{PKW}}$,Robert$_{\text{MLO}}$)
$\frac{1}{2}w_{4,1}$	NOT Related(Alice$_{\text{PKW}}$,Bob$_{\text{PKW}}$) OR NOT Related(Arihi$_{\text{MLO}}$,Robert$_{\text{MLO}}$) OR NOT SamePerson(Bob$_{\text{PKW}}$,Robert$_{\text{MLO}}$) OR SamePerson(Alice$_{\text{PKW}}$,Arihi$_{\text{MLO}}$)

The simplified form is desirable for the IP since only the query terms are left to be the decision variables — reducing the extra variables and constraints needed to define the evidence terms and set them to their truth value.

The following is an IP formulation of the Weighted Partial MaxSAT problem. It assumes each soft clause has a non-negative weight. If the weight is negative, we can switch the sign to positive by negating the clause. For hard clauses, hard constraints are added to ensure they are satisfied.

Sets

C, D be the set of soft and hard clauses, respectively;
I, J be the set of terms in all the soft and hard clauses, respectively;
—I_c^+, J_d^+ be the non-negated terms in clause $c \in C, d \in D$;
—I_c^-, J_d^- be the negated terms in clause $c \in C, d \in D$.

Parameters

w_c = the net weight for soft clause c. No weights are needed for hard clauses.

Decision variables

$y_i = 1$ if the term i is true, and 0 otherwise, $i \in I$;
$y_j = 1$ if the term j is true, and 0 otherwise, $j \in J$;
$z_c = 1$ if the soft clause c is satisfied, and 0 otherwise, $c \in C$.

Weighted Partial MaxSAT

$$\max \quad \sum_{c \in C} w_c z_c \qquad \text{(max the sum of weights of satisfied soft clauses)}$$

$$\text{s.t.} \quad \sum_{i \in I_c^+} y_i + \sum_{i \in I_c^-} (1 - y_i) \geq z_c \quad \forall c \in C \quad \text{(soft clauses being satisfied in clausal form)}$$

$$\sum_{j \in J_d^+} y_j + \sum_{j \in J_d^-} (1 - y_j) \geq 1 \quad \forall d \in D \quad \text{(hard clauses must be satisfied in clausal form)}$$

$$y_i \in \{0, 1\} \quad \forall i \in I \quad \text{(every soft clause term is either true or false)}$$

$$y_j \in \{0, 1\} \quad \forall j \in J \quad \text{(every hard clause term is either true or false)}$$

$$z_c \in \{0, 1\} \quad \forall c \in C \quad \text{(every soft clause is either satisfied or not)}$$

5 Experiments

We test a set of weights for each experiment on our example problem to demonstrate how predicate equivalence and reverse predicate equivalence constraints can incorporate relational information in their predictions. Each weight's corresponding formula can be found in Table 2.

5.1 Experiment 1: Effects of Predicate Equivalence

The following weights used for experiment 1:
$[w_{1,1}, w_{2,1}, w_{2,2}, w_{3,1}, w_{3,2}, w_{4,1}, w_{5,1}, w_{5,2}, w_{6,1}, w_{6,2}] = [0.95, 0.9, 0.2, 1.1, 0.0, 1.1, 1.0, 1.0, 1.0, 1.0]$.
We set the threshold weight for each query to be 1.0. We set $w_{4,1} = 0$ to remove the reverse predicate equivalence constraint for this experiment.

Experiment 1 consists of two setups: 1A where the predicate equivalence hard constraint is removed, and 1B where the constraint is included. Without the two types of constraints linking the `Related` and `SamePerson` decisions, experiment 1A is equivalent to a traditional, pairwise ER system. Table 4 shows the optimal solution results for both setups.

Table 4: Optimal solutions for experiment 1.

Decision variables	Experiment 1A	Experiment 1B
Related(Alice$_{\text{PKW}}$,Bob$_{\text{PKW}}$)	False	True
Related(Arihi$_{\text{MLO}}$,Robert$_{\text{MLO}}$)	True	True
SamePerson(Alice$_{\text{PKW}}$,Arihi$_{\text{MLO}}$)	True	True
SamePerson(Bob$_{\text{PKW}}$,Robert$_{\text{MLO}}$)	True	True

For both setups, the three decisions Related(Arihi$_{\text{MLO}}$,Robert$_{\text{MLO}}$), SamePerson(Alice$_{\text{PKW}}$,Arihi$_{\text{MLO}}$), SamePerson(Bob$_{\text{PKW}}$, Robert$_{\text{MLO}}$) are all True since their respective net weights (Table 3) $w_{2,1} + w_{2,2} - w_{5,2}, w_{3,1} - w_{6,1}, w_{3,2} - w_{6,2}$ are all $0.1 > 0$. However for Related(Alice$_{\text{PKW}}$,Bob$_{\text{PKW}}$), 1A's decision is False whereas 1B's is True. For 1A, the decision is due to $w_{1,1} - w_{5,1} = -0.05 < 0$. For 1B, the extra hard constraint SamePerson(Alice$_{\text{PKW}}$,Arihi$_{\text{MLO}}$) AND SamePerson(Bob$_{\text{PKW}}$,Robert$_{\text{MLO}}$) $\implies$ Related(Alice$_{\text{PKW}}$,Bob$_{\text{PKW}}$)$\Leftrightarrow$ Related(Arihi$_{\text{MLO}}$,Robert$_{\text{MLO}}$) must be satisfied. Setting all four variables to True is optimal, and the objective value only decreased by 0.05 from obtaining $w_{1,1}$ and forgoing $w_{5,1}$. Setting either one of the three variables as False instead would also satisfy the hard constraint, but that would mean an objective decrease of 0.1. The result demonstrates how predicate equivalence is necessary for coherent predictions. Without any constraints between the variables, experiment 1A is free to make the decisions independently, resulting in a logically incoherent solution.

5.2 Experiment 2: Effects of Reverse Predicate Equivalence

We include the predicate equivalence hard constraint for this experiment. We set $w_{4,1} = 0$ for experiment 2A and $w_{4,1} = 0.1$ for 2B to demonstrate the effects of the reverse predicate equivalence soft constraint. The remaining weights used for experiment 2 are the following: $[w_{1,1}, w_{2,1}, w_{2,2}, w_{3,1}, w_{3,2}, w_{5,1}, w_{5,2}, w_{6,1}, w_{6,2}] = [1.1, 0.9, 0.2, 1.1, 0.95, 1.0, 1.0, 1.0, 1.0]$. Table 5 shows the optimal solution for both setups.

Table 5: Optimal solutions for experiment 2

Decision variables	Experiment 2A	Experiment 2B
Related(Alice$_{\text{PKW}}$,Bob$_{\text{PKW}}$)	True	True
Related(Arihi$_{\text{MLO}}$,Robert$_{\text{MLO}}$)	True	True
SamePerson(Alice$_{\text{PKW}}$,Arihi$_{\text{MLO}}$)	True	True
SamePerson(Bob$_{\text{PKW}}$,Robert$_{\text{MLO}}$)	False	True

For both setups, the three decisions Related(Alice$_{\text{PKW}}$,Bob$_{\text{PKW}}$), Related(Arihi$_{\text{MLO}}$,Robert$_{\text{MLO}}$), SamePerson(Alice$_{\text{PKW}}$,Arihi$_{\text{MLO}}$) are all True since their respective net weights (Table 3) $w_{1,1} - w_{5,1}, w_{2,1} + w_{2,2} - w_{5,2}, w_{3,1} - w_{6,1}$ are all $0.1 > 0$. However for SamePerson(Bob$_{\text{PKW}}$,Robert$_{\text{MLO}}$), 2A's decision is False whereas 2B's is True. For 2A, the decision is due to $w_{3,2} - w_{6,1} = -0.05 < 0$. For 2B, the extra soft constraint
Related(Alice$_{\text{PKW}}$,Bob$_{\text{PKW}}$) AND Related(Arihi$_{\text{MLO}}$,Robert$_{\text{MLO}}$) $\implies$
SamePerson(Alice$_{\text{PKW}}$,Arihi$_{\text{MLO}}$) $\Leftrightarrow$ SamePerson(Bob$_{\text{PKW}}$,Robert$_{\text{MLO}}$)
rewards SamePerson decisions to be the same if both Related decisions are True. Setting all four variables to True is optimal, and the objective value increased by 0.05 from obtaining $w_{3,2}, w_{4,1}$ and forgoing $w_{6,1}$. Setting either one of the three variables as False instead would also satisfy the soft constraint, but would leave the objective value unchanged.

The result demonstrates how reverse predicate equivalence enables the name match evidence to be supported by PKW share transfer and Māori landownership evidence. It is able to conclude that the PKW shareholder `Bob` and the Māori landowner `Robert` are the same person; whereas a pairwise ER would consider the name match evidence alone and conclude that they are not.

6 Future Work

Learning weights: Singla and Domingos (2005) outlined the steps to train an MLN. They showed learning by gradient descent is equivalent to a weighted MaxSAT problem for training MLN weights and thus can be solved by IP or any MaxSAT solvers. Finding efficient ways to label a sufficiently large training set will be necessary if we use a learning algorithm.

Solving large networks: IP is intractable for large networks. Interestingly, suppose we solve the IP model's linear progamming (LP) relaxation. In that case, it is equivalent to an instance of *probabilistic soft logic* (PSL), a closely related framework to MLN (Bach et al. 2017). Their original paper used consensus optimisation techniques to solve large PSL models. Other MaxSAT solvers that avoid the IP/LP formulation for large networks should also be investigated.

Implementing other collective ER approaches: MLN is only one method to perform collective ER. Other methods such as a PSL instance with quadratic objective, Latent Dirichlet Allocation (Bhattacharya and Getoor 2005) models and networking clustering (Bhattacharya and Getoor 2007) should also be implemented to find the best approach to our PKW problem.

7 Conclusion

We identified the PKW missing shareholder problem as a collective entity resolution (ER) task. Although ER is a ubiquitous data analytics task and fundamentally an optimisation problem, it is not well known in the operations research (OR) community. This paper formulates our PKW problem into an integer program and demonstrates its potential to accurately make ER predictions on large relational data. In describing this process, we use Markov logic networks. This framework allows the modeller to express their probabilistic reasoning clearly in first-order logic, construct a Markov network from the data, and perform the most probable explanation inference on the network. We hope this paper introduced the reader to analytics tools and approaches to problems that are new to them and would be helpful to their research and practice.

References

Bacchus, Fahiem, Jeremias Berg, Matti Järvisalo, Ruben Martins, and Andreas Niskanen, eds. 2022. *MaxSAT Evaluation 2022: Solver and Benchmark Descriptions*. Department of Computer Science Series of Publications B. Finland: Department of Computer Science, University of Helsinki.

Bach, Stephen H, Matthias Broecheler, Bert Huang, and Lise Getoor. 2017. "Hinge-Loss Markov Random Fields and Probabilistic Soft Logic." Technical Report.

Bhattacharya, Indrajit, and Lise Getoor. 2005. "A Latent Dirichlet Allocation Model for Entity Resolution." Technical Report.

———. 2007. "Collective entity resolution in relational data." *ACM Transactions on Knowledge Discovery from Data* 1, no. 1 (mar).

Fellegi, Ivan P., and Alan B. Sunter. 1969. "A Theory for Record Linkage." *Journal of the American Statistical Association* 64 (328): 1183–1210.

Li, Chu Min, and Felip Manyà. 2009. "MaxSAT, Hard and Soft Constraints." In *Handbook of Satisfiability*, edited by Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, Volume 185 of *Frontiers in Artificial Intelligence and Applications*, 613–631. IOS Press.

Māori Land Court. 2022. Māori Land Online. [Online; accessed 27-October-2022].

Ministry of Māori Development. 2022. How Māori Land Ownership Works Today. `https://www.tupu.nz/media/pf5l13tv/how-maori-land-ownership-works-today_english_print.pdf`. [Online; accessed 25-October-2022].

PKW. 2022. Parininihi ki Waitotara. `http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm`. [Online; accessed 25-October-2022].

Reiter, Raymond. 1981. "on Closed World Data Bases." *Readings in Artificial Intelligence*, jan, 119–140.

Richardson, Matthew, and Pedro Domingos. 2006. "Markov logic networks." *Machine Learning* 62 (1-2 SPEC. ISS.): 107–136 (feb).

SAT-2022. 2022. MaxSAT Evaluation 2022: Summary of Weighted Complete Track. `https://maxsat-evaluations.github.io/2022/results/complete/weighted/summary.html`. [Online; accessed 25-October-2022].

Singla, Parag, and Pedro Domingos. 2005. "Discriminative training of Markov logic networks." *Proceedings of the National Conference on Artificial Intelligence* 20:868.

———. 2006. "Entity resolution with Markov logic." *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 572–582.