Front. Comput. Sci., 2021, 15(4): 154608
https://doi.org/10.1007/s11704-020-9195-9

RESEARCH ARTICLE

# Information networks fusion based on multi-task coordination

**Dong LI**[1,2], **Derong SHEN** (✉)[1], **Yue KOU**[1], **Tiezheng NIE**[1]

1 School of Computer Science & Engineering, Northeastern University, Shenyang 110004, China
2 School of Information, Liaoning University, Shenyang 110036, China



**Abstract** Information networks provide a powerful representation of entities and the relationships between them. Information networks fusion is a technique for information fusion that jointly reasons about entities, links and relations in the presence of various sources. However, existing methods for information networks fusion tend to rely on a single task which might not get enough evidence for reasoning. In order to solve this issue, in this paper, we present a novel model called MC-INFM (information networks fusion model based on multi-task coordination). Different from traditional models, MC-INFM casts the fusion problem as a probabilistic inference problem, and collectively performs multiple tasks (including entity resolution, link prediction and relation matching) to infer the final result of fusion. First, we define the intra-features and the inter-features respectively and model them as factor graphs, which can provide abundant evidence to infer. Then, we use conditional random field (CRF) to learn the weight of each feature and infer the results of these tasks simultaneously by performing the maximum probabilistic inference. Experiments demonstrate the effectiveness of our proposed model.

**Keywords** information networks fusion, multi-task coordination, conditional random field, inference

## 1 Introduction

Nowadays, a large number of information networks (e.g., IMDB, Rotten Tomatoes, DBLP, LiveJournal etc.) have appeared, which can provide users with a powerful representation of entities and the relationships between them. In the real world, the same entities are often included by multiple information networks simultaneously. For example, there are the same movies in both IMDB and Rotten Tomatoes, also there are the same papers in both DBLP and LiveJournal. These information sources are usually separated in different places. Information networks fusion is a technique for information fusion that jointly reasons about entities, links and relations in the presence of various sources [1]. Via information networks fusion, people can understand the entities and the relationships among them more comprehensively. The technique of information networks fusion can be applied in several areas. For example, it can facilitate the discovery of the hidden knowledge graph in the area of knowledge graph identification [2–6]. It can also help users identify the same entities and predict the formation of social links across multiple social networks [7–17].

### 1.1 Motivating scenario

The goal of information networks fusion is to find the hidden information network underlying multiple observed information networks. So we must infer the nodes and the edges of the hidden network based on the evidence provided by the observed networks. The problem can be casted as performing a series of tasks including entity resolution, link prediction, relation matching, community detection, information diffusion and network embeddding etc. [1] However, existing methods for information networks fusion tend to rely on a single task which might not get enough evidence for reasoning.

Let us consider the following motivating scenario.

Suppose there are two information networks (shown in Fig. 1). In information network 1, it includes four entities with the relationships such that Wenfei Fan writes a paper pressed in ACM Sigmod2016. Also there is an independent entity, San Fran, with the type of location. In information network 2, it includes four entities with the relationships such that Fan W. publishes a paper which is accepted by Sigmod'16 held in San Francisco.

In order to infer the hidden network underlying the observed information networks, some work has been done by performing a task such as entity resolution, link prediction or relation matching.

- The task of entity resolution is to infer the potential anchor links among the shared entities across multiple information networks. For example, Wenfei Fan and Fan W. are inferred to refer to the same person in the real world. Via entity resolution, they will be merged into one node in the hidden information network.
- The task of link prediction is to infer the unknown links between nodes based on the observed information networks. For example, via link prediction, we can infer whether there is a link between ACM Sigmod2016 and San Fran.
- The task of relation matching is to find the sets of equivalent relations. For example, via relation matching, we can infer that the relation "Write" and the relation "Publish" match (i.e., are equivalent), which will be represented as a uniform label in the hidden information network.

Received May 30, 2019; accepted May 7, 2020

E-mail: shenderong@cse.neu.edu.cn

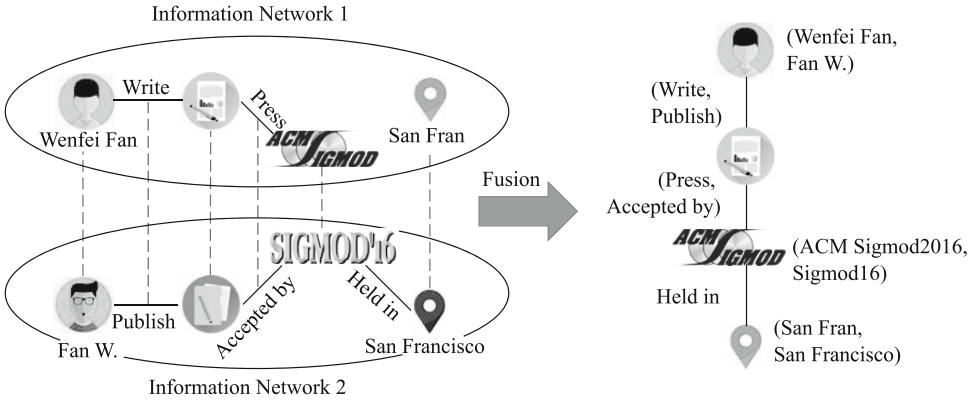


**Fig. 1** An illustration of information networks fusion

While previous work has addressed each of these tasks separately, we can only acquire partial inference results. On the contrary, if these tasks are considered together as a coherent task, we can obtain better results. For example, determining whether Wenfei Fan and Fan W. refer to the same entity and determining whether the relation "Write" and "Publish" match can make us more certain whether these two papers are the same entity. Also, determining whether these two conferences refer to the same entity and predicting whether a link between ACM Sigmod2016 and San Fran exists can ascertain whether San Fran and San Francisco are the same entity. Finally, via fusion we can infer a more complete hidden network shown in Fig. 1. Therefore, based on multi-task coordination, the information can be propagated among them, which can give us more evidence to infer.

### 1.2 Challenges

However, information networks fusion based on multi-task coordination is a highly challenging problem. The major challenges are as follows:

(1) For multiple tasks, how to capture the interaction between their predicted results? Each predicted result means the result of one task. Traditional work defines the features only to capture the dependencies between a single predicted result and the evidence within one task. The interaction among multiple tasks are often ignored. Therefore, we need to extract more features to disseminate information among the predicted results of different tasks.

(2) How to reason jointly to generate the hidden information network? Some probabilistic models (e.g., Markov Logic Networks (MLN) [18] and Conditional Random Field (CRF) [19]) have been proposed to perform probabilistic inference. Therefore, we need to cast the fusion problem as a probabilistic inference problem, and use these models to learn the weight of each feature and infer the results of these tasks simultaneously.

### 1.3 Contributions and organization

In this paper, we present a novel model called MC-INFM (Information networks fusion model based on multi-task coordination). Different from traditional models, MC-INFM casts the fusion problem as a probabilistic inference problem, and collectively processes multiple tasks (in this paper, we focus on three tasks: entity resolution, link prediction and relation matching) to infer the final result of fusion. More specifically, we make the following contributions:

(1) We define both the intra-features for each task and the inter-features based on multi-task coordination. We also model them as a series of factor graphs, which can provide abundant evidence to infer.

(2) Based on CRF model, we propose a weight learning algorithm and an iterative inference algorithm respectively. On the one hand, we use CRF to learn the weight of each feature. On the other hand, we infer the results of these tasks simultaneously by performing the maximum probabilistic inference.

(3) We conduct experimental studies based upon two pairs of information networks. Experiments demonstrate the effectiveness of our proposed model.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 formulates the main problem and gives an overview of our model. Section 4 defines the features we need. Section 5 and Section 6 propose a weight learning algorithm and an iterative inference algorithm respectively. Section 7 shows the experimental result and Section 8 concludes.

## 2 Related work

Various approaches for information networks fusion have been studied over the years, which mainly include the tasks of entity resolution, link prediction, relation matching, community detection, information diffusion and network embedding etc. In this paper, we only focus on the first three tasks. First, we briefly review techniques for them one by one. Then we introduce the related work about information networks fusion by jointly performing multiple tasks.

As for entity resolution across multiple information networks, it is also called network alignment [7]. There are lots of work which has been proposed to solve this problem. For example, in [8] a fast alignment algorithm is proposed to align two bipartite graphs. In [9] based on various node attributes (e.g., username, typing patterns and language patterns) a matching method is proposed to match entities across social networks. In [7], by using heterogeneous information in the networks, a two-step supervised method is proposed to infer potential anchor links across networks. In [10] a partial network alignment method is proposed to solve the problem of lacking anchor users. In [11], based on a small amount of positive set and a

large unlabeled set, a model is built to train the anchor link data. In [12] an unsupervised network alignment model is designed for the case of no available training data. In [13] a concurrent alignment model is designed.

Most link prediction methods aim at solving the problem of link prediction in one single target network, which are categorized into unsupervised link prediction (e.g., [20–22]), supervised link prediction (e.g., [23]) and matrix factorization based link prediction(e.g., [24–26]). As for link prediction across multiple information networks, some research work has also been done. In [14] a method is proposed to transfer useful information across multiple information networks to help predict links for new entities. In [15] a method is proposed to predict multiple kinds of links for new networks with information transferred across partially networks. In [16] a survey about link prediction problems and methods across information networks is given. In [17] a model called MLI is proposed which includes two parts: metapath based feature extraction and iterative PU link prediction across multiple networks.

There has been less work on relation matching than on entity matching or entity resolution. Some techniques for relation matching have been proposed in the field of knowledge graph identification. For example, in [27] a system called RESOLVER uses HAC to cluster Open IE relations in TextRunner data. The probability that two relations are equivalent is computed based on counting the number of entity pairs that they had in common. In [28] matching different ways representing the same relation between entities is considered as one task of knowledge fusion. A rules-based relation matching method is proposed to identify situations where one relation implies another.

More recently, there is some work about information networks fusion by jointly performing multiple tasks. For example, in [1] the task of information networks fusion is decomposed into five subtasks, i.e., network alignment, link prediction, community detection, information diffusion and network embedding. However, these tasks are considered as inter-dependent procedures, which are executed sequentially. In [2], three tasks (i.e., entity resolution, link prediction and node labeling) are performed simultaneously to infer the hidden network. In [29], a two-stage approach is proposed, which performs entity resolution first and then discovers equivalences between synonymous relations. But they mainly rely on some rules predefined.

The differences between our work and existing work are as follows:

(1) Most methods for information networks fusion tend to rely on a single task. They define the features only to capture the dependencies between a single predicted result and an evidence. The coordination among multiple tasks are ignored. Different from the existing work, we take into account the interactions between tasks and extract more features to disseminate information among the predicted results of different tasks. So we can get enough evidence for reasoning.

(2) Although some work by jointly performing multiple tasks has been proposed, they usually execute the tasks sequentially, or rely on ontology rules or domain knowledge. While we use CRF model to learn the weight of each feature and infer the results of these tasks simultaneously by performing the maximum probabilistic inference.

## 3 Model overview

In this section, we first give some definitions, and then formulate the main problem. Finally we give an overview of our model.

### 3.1 Problem statement

The information networks considered by us are heterogeneous, that is, they contain multiple kinds of nodes and links.

**Definition 1 (Information network)** An information network is a graph $G$=($V$, $E$), where $V$ is the vertex set (i.e., the entity set), $E \subseteq V \times V$ is the edge set (i.e., the relationship among entities).

Let's define the task of information networks fusion.

**Definition 2 (Information networks fusion)** Given a set of information networks ($G_1, \ldots, G_n$), the goal of information networks fusion is to find a hidden network $G_H$=($V_H$, $E_H$) such that: (1) $V_H \subseteq V_1 \cup \cdots \cup V_n$ and $\forall v_i, v_j$ ($v_i, v_j \in V_H$) such that $v_i$ and $v_j$ might not correspond to the same entity. (2) $E_H \subseteq E_1 \cup \cdots \cup E_n$ and $\forall e_i, e_j$ ($e_i, e_j \in E_H$) such that if $e_i$ and $e_j$ are labeled as different relations, then they might not refer to the same relation in the real world.

In the above definition, $V_H$ is the subset of all the nodes in ($G_1, \ldots, G_n$). While $E_H$ contains the edges of these networks, because besides these edges, some new edges might be predicted during fusion and will be added to $G_H$. Conditions (1) and (2) are to ensure the uniqueness of nodes and relations in $G_H$ respectively. Therefore, in this paper, we decompose the task of information networks fusion into three subtasks: entity resolution, link prediction and relation matching across multiple information networks.

**Definition 3 (Entity resolution across multiple information networks)** Given a set of information networks ($G_1, \ldots, G_n$), the goal of entity resolution across them is to find the sets of anchor links ($A_{12}, A_{13}, \ldots, A_{1n}, A_{23}, \ldots, A_{(n-1)n}$) where $A_{pq}$ ($p$, $q \in \{1, \ldots, n\}$) is the set of undirected anchor links between $G_p$ and $G_q$.

Here each anchor link means an undirected link between two nodes from different networks, which represents they refer to the same entity. We use a random variable $x^{ER}(v_i, v_j)$ to represent whether the link between two nodes $v_i$ and $v_j$ is an anchor link. If so, the value of $x^{ER}(v_i, v_j)$ is 1. Otherwise, its value is 0. Therefore, the aim of entity resolution is to compute the value of a random vector $X^{ER}$ containing all such $x^{ER}$ s.

**Definition 4 (Link prediction across multiple information networks)** For each information network $G_k$ ($k \in \{1, \ldots, n\}$), the goal of link prediction across multiple information networks is to predict the existence of one link in $G_k$ by using the information disseminated from ($G_1, \ldots, G_{k-1}, G_{k+1}, \ldots, G_n$).

We use a random variable $x^{LP}(v_i, v_j)$ to represent whether there is a link between two nodes $v_i$ and $v_j$. If so, the value of $x^{LP}(v_i, v_j)$ is 1. Otherwise, its value is 0. Here the edge $(v_i, v_j)$ is from the unconnected link set $V \times V - E$. We use a random vector $X^{LP}$ containing all such $x^{LP}$ s to store the result of the task of link prediction.

**Definition 5 (Relation matching across multiple information networks)** Given a set of information networks ($G_1, \ldots, G_n$), the goal of relation matching across them is to find the sets of equivalent relations ($R_{12}, R_{13}, \ldots, R_{1n}, R_{23}, \ldots, R_{(n-1)n}$) where $R_{pq}$ ($p, q \in \{1, \ldots, n\}$) is the set of equivalent relations between $G_p$ and $G_q$.

We use a random variable $x^{RM}((v_i, v_j), (v'_i, v'_j))$ to represent whether the relation between $v_i$ and $v_j$, and the relation between $v'_i$ and $v'_j$ match with each other. If so, the value of $x^{RM}((v_i, v_j), (v'_i, v'_j))$ is 1. Otherwise, its value is 0. We use a random vector $X^{RM}$ containing all such $x^{RM}$ s to store the result of the task of relation matching.

For simplicity, in the remainder of this paper, we denote $x^{ER}(v_i, v_j)$, $x^{LP}(v_i, v_j)$ and $x^{RM}((v_i, v_j), (v'_i, v'_j))$ as $x^{ER}_{ij}$, $x^{LP}_{ij}$ and $x^{RM}_{(ij,i'j')}$ respectively.

Frequently used notations in this paper are summarized in Table 1.

### 3.2 Overview of our model

In this paper, we present a novel model called MC-INFM. It casts the problem of information networks fusion as a probabilistic inference problem, and collectively processes multiple tasks to infer the hidden network. Given a set of information networks ($G_1, \ldots, G_n$) and a group of random vectors to be predicted, MC-INFM is to predict the values of these vectors (shown in Fig. 2). It mainly includes three parts.

(1) Feature extraction (see Section 4). We classify the features into two categories: the intra-features and the inter-features, which are used to represent the local features within one task and the coordination among different tasks respectively.

(2) Weight learning (see Section 5). We cast the problem of information networks fusion as a probabilistic inference problem. The extracted features are modeled as factor graphs, which can provide abundant evidence to infer. Based on CRF model, we propose a weight learning algorithm to train the weights of both intra-features and inter-features.

(3) Iterative Inference (see Section 6). Based on the trained weights, we propose an iterative inference algorithm to infer the results of these tasks simultaneously by performing the maximum probabilistic inference.

**Table 1** Notations

| Notation | Meaning |
|---|---|
| $(G_1, \ldots, G_n)$ | A set of information networks |
| $G_H$ | The hidden information network |
| $x^{ER}_{ij}$ | A random variable for entity resolution |
| $x^{LP}_{ij}$ | A random variable for link prediction |
| $x^{RM}_{(ij,i'j')}$ | A random variable for relation matching |
| $X^{ER}$, $X^{LP}$ or $X^{RM}$ | A random vector containing all $x^{ER}$ s, $x^{LP}$ s or $x^{RM}$ s, respectively |

## 4 Feature extraction

In this section, we first define the intra-features for each task , then define the inter-features by taking into account multiple tasks simultaneously. For simplicity, in the remainder of this paper, we denote the task of entity resolution, link prediction and relation matching as ER, LP and RM respectively.

### 4.1 Intra-feature extraction

The intra-features are extracted to propagate information among random variables within a certain task. For each task, we define the following intra-features. Here we use superscript form to indicate which task the feature belongs to.

Entity similarity ($f^{ER}_{sim}$ or $f^{LP}_{sim}$): Entity similarity is measured by a variety of attribute similarities between two nodes. There are lots of similarity measures having been proposed [30]. In this paper, we use Jaccard to measure the similarity between two entities (denoted as $f^{ER}_{sim}(v_i, v_j)$ or $f^{LP}_{sim}(v_i, v_j)$).

Neighbor similarity ($f^{ER}_{N-sim}$): We also consider the similarity between the neighbor sets of two nodes as one of the intra-features. We use $f^{ER}_{N-sim}(v_i, v_j)$ to denote the similarity between the neighbor sets of $v_i$ and $v_j$.

Common neighbors ($f^{LP}_{CN}$): We also consider the number of common neighbors of $v_i$ and $v_j$.

Similarity transferred ($f^{ER}_{T-sim}$, $f^{LP}_{T-sim}$ or $f^{RM}_{T-sim}$): Suppose there are three entities $v_i$, $v_j$ and $v_k$. If $v_i$ and $v_j$ are identified as the same entity, and $v_j$ and $v_k$ are identified as the same entity, then there is a higher probability that $v_i$ and $v_k$ refer to the same entity as well. That is, if the values of $x^{ER}$ for all three pairs of them are 1, they are mutually reinforcing. If the values of $x^{ER}$ for only two pairs of them are 1, they contradict each other. Otherwise, the feature $f^{ER}_{T-sim}$ has no effect on the final result. We use $f^{ER}_{T-sim}(x^{ER}_{ij}, x^{ER}_{jk}, x^{ER}_{ik})$ to denote this kind of transitivity (Eq. (1)). Similarly, we can define $f^{LP}_{T-sim}$ and $f^{RM}_{T-sim}$.

$$f^{ER}_{T-sim}(x^{ER}_{ij}, x^{ER}_{jk}, x^{ER}_{ik}) = \begin{cases} 1, & \text{if } x^{ER}_{ij} + x^{ER}_{jk} + x^{ER}_{ik} = 3, \\ -1, & \text{if } x^{ER}_{ij} + x^{ER}_{jk} + x^{ER}_{ik} = 2, \\ 0, & \text{Otherwise.} \end{cases} \quad (1)$$

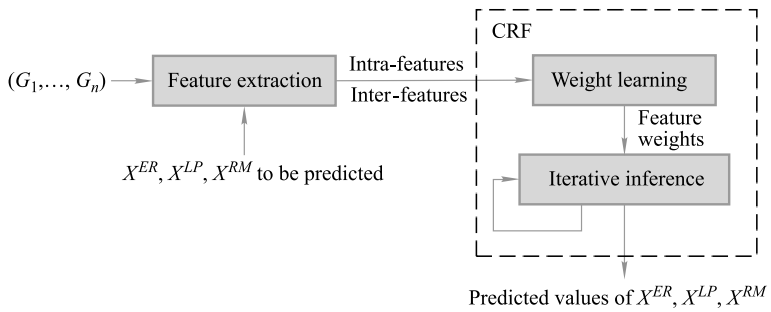


**Fig. 2** Overview of MC-INFM model

Outgoing degree ($f^{LP}_{Out}$) and incoming degree ($f^{LP}_{In}$): We consider outgoing edges from $v_i$ and incoming edges to $v_j$. Formally we use $f^{LP}_{In}(v_j)$ to denote the number of incoming edges to $v_j$ and use $f^{LP}_{Out}(v_i)$ to denote the number of outgoing edges from $v_i$.

Relation similarity ($f^{RM}_{sim}$): Suppose we predict whether the relation ($v_i$, $v_j$) and the relation ($v'_i$, $v'_j$) match with each other. We consider the similarity between ($v_i$, $v_j$) and ($v'_i$, $v'_j$) in their labels. We use Jaccard to measure their similarity.

### 4.2 Inter-feature extraction

Besides the intra-features, we also extract the inter-features to disseminate information among variables of different tasks. In this paper, we take into account the inter-features between two tasks of ER, LP and RM.

(1) Inter-feature for ER+LP

As shown in Fig. 3(a), suppose there is an anchor link between $v_i$ and $v_j$, and there is a link between $v_j$ and $v_k$ in $G_2$. Our task is not only to predict the link between $v_i$ and $v_m$ in $G_1$, but also to identify whether there is an anchor link between $v_m$ and $v_k$. First, let's consider the effect of ER on LP. Suppose via ER, the link between $v_m$ and $v_k$ is identified as an anchor link (i.e., $x^{ER}_{mk}=1$). Then this will facilitate the existence of the link between $v_i$ and $v_m$ (i.e., $x^{LP}_{im}=1$). Similarly, as for the effect of LP on ER, if the value of $x^{LP}_{im}$ is 1, there will be a higher probability that the value of $x^{ER}_{mk}$ is 1 as well.

We use $f^{ER+LP}$ to denote the inter-feature between ER and LP (Eq. (2)). If it conforms to the above rule, it will be rewarded with a value of 1. If it violates the rule, it will be punished with a value of −1. Otherwise, it will be 0.

$$f^{ER+LP}(x^{ER}_{ij}, x^{ER}_{mk}, x^{LP}_{jk}, x^{LP}_{im}) =$$

$$\begin{cases} 1, & \text{if } (x^{ER}_{ij}, x^{ER}_{mk}, x^{LP}_{jk}, x^{LP}_{im}) = (1,1,1,1), \\ -1, & \text{if } (x^{ER}_{ij}, x^{ER}_{mk}, x^{LP}_{jk}, x^{LP}_{im}) = (1,0,1,1) \text{ or } (1,1,1,0), \\ 0, & \text{Otherwise.} \end{cases} \quad (2)$$

(2) Inter-feature for ER+RM

As shown in Fig. 3(b), suppose there is a relation ($v_i$, $v_m$) in $G_1$, and there is a relation ($v_j$, $v_k$) in $G_2$. Also there is an anchor link between $v_i$ and $v_j$. Our task is not only to identify whether there is an anchor link between $v_m$ and $v_k$, but also to match two relations (i.e., ($v_i$, $v_m$) and ($v_j$, $v_k$)). For the effect of ER on RM, if via the task of ER, $v_m$ and $v_k$ are identified as the same entity (i.e., $x^{ER}_{mk}=1$), obviously this will promote the matching degree between ($v_i$, $v_m$) and ($v_j$, $v_k$) (i.e., $x^{RM}_{(im,jk)}=1$), and vice versa.

We use $f^{ER+RM}$ to denote the inter-feature between ER and RM (Eq. (3)). Similarly, setting the value of $f^{ER+RM}$ to 1 indicates reward, while −1 indicates punishment.

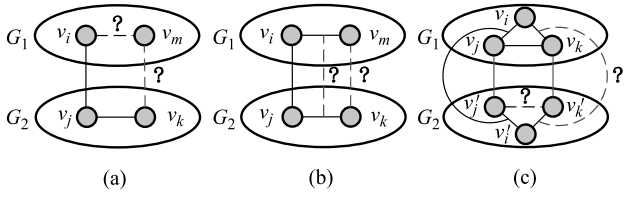


**Fig. 3** Examples of inter-features. (a) ER+LP; (b) ER+RM; (c) LP+RM

$$f^{ER+RM}(x^{ER}_{ij}, x^{ER}_{mk}, x^{RM}_{(im,jk)}) =$$

$$\begin{cases} 1, & \text{if } (x^{ER}_{ij}, x^{ER}_{mk}, x^{RM}_{(im,jk)}) = (1,1,1), \\ -1, & \text{if } (x^{ER}_{ij}, x^{ER}_{mk}, x^{RM}_{(im,jk)}) = (1,0,1) \text{ or } (1,1,0), \\ 0, & \text{Otherwise.} \end{cases} \quad (3)$$

(3) Inter-feature for LP+RM

As shown in Fig. 3(c), suppose there are two relations ($v_i$, $v_j$) and ($v_i$, $v_k$) in $G_1$, also there are two relations ($v'_i$, $v'_j$) and ($v'_i$, $v'_k$) in $G_2$. Assume that ($v_i$, $v_j$) and ($v'_i$, $v'_j$) match. Our task includes matching ($v_i$, $v_k$) and ($v'_i$, $v'_k$), and predicting the link between $v'_j$ and $v'_k$. For the effect of LP on RM, if via LP, the value of $x^{LP}_{j'k'}$ is 1, there will be a higher probability that ($v_i$, $v_k$) and ($v'_i$, $v'_k$) can match with each other (i.e., $x^{RM}_{(ik,i'k')}=1$), and vice versa.

We use $f^{LP+RM}$ to denote the inter-feature between LP and RM (Eq. 4). If they promote each other, the value of $f^{LP+RM}$ is set to 1. Otherwise, the value of $f^{LP+RM}$ is set to −1 or 0.

$$f^{LP+RM}(x^{LP}_{jk}, x^{RM}_{(ij,i'j')}, x^{LP}_{j'k'}, x^{RM}_{(ik,i'k')}) =$$

$$\begin{cases} 1, & \text{if } (x^{LP}_{jk}, x^{RM}_{(ij,i'j')}, x^{LP}_{j'k'}, x^{RM}_{(ik,i'k')}) = (1,1,1,1), \\ -1, & \text{if } (x^{LP}_{jk}, x^{RM}_{(ij,i'j')}, x^{LP}_{j'k'}, x^{RM}_{(ik,i'k')}) = (1,1,0,1) \text{ or } (1,1,1,0), \\ 0, & \text{Otherwise.} \end{cases} \quad (4)$$

## 5 Weight learning

In this section, we use CRF model to learn the weight of each feature. CRF is a graphical model encoding the conditional probability of a set of output variables $X$ given a set of evidence variables $Y$ [19]. In this paper, $X$ is the union of the results of three tasks, i.e., $X^{ER} \cup X^{LP} \cup X^{RM}$, $Y$ is a collection of observed entities and relationships among them in the information networks ($G_1, \ldots, G_n$). We use $P(X^{ER}, X^{LP}, X^{RM}|Y)$ to represent the joint probability distribution over all the random variables in $X^{ER} \cup X^{LP} \cup X^{RM}$ (Eq. (5)). Here $Z_Y$ is a normalizer. Each potential function $\phi$ is represented more compactly as a log-linear combination over a set of features extracted in Section 4 (Eq. (6)). Here $f_l$ means the value of the $l$-th feature for $\phi_Q$ with the weight $w_l$. These potential functions together determine a joint probability distribution over all the random variables in the potential functions (i.e., $X^{ER} \cup X^{LP} \cup X^{RM}$).

$$P(X^{ER}, X^{LP}, X^{RM}|Y) = \frac{1}{Z_Y} \prod_{Q \in N} \phi_Q(X^{ER}_Q, X^{LP}_Q, X^{RM}_Q, Y_Q), \quad (5)$$

$$\phi_Q(X^{ER}_Q, X^{LP}_Q, X^{RM}_Q, Y_Q) = \exp(\sum_l w_l f_l(x^{ER}_Q, x^{LP}_Q, x^{RM}_Q, Y_Q)). \quad (6)$$

### 5.1 Factor graph construction

A CRF can be viewed as a template for constructing factor graphs. As defined in [31], a factor graph is a set of factors $\Phi = \{\phi_1, \ldots, \phi_N\}$, where each factor $\phi_i$ is a potential function defined in Eq. (6) to indicate the causal relationships among the random variables in $X^{ER} \cup X^{LP} \cup X^{RM}$. For each factor, a factor graph is constructed, in which a factor is represented as a square with its variables represented by its neighboring circles.

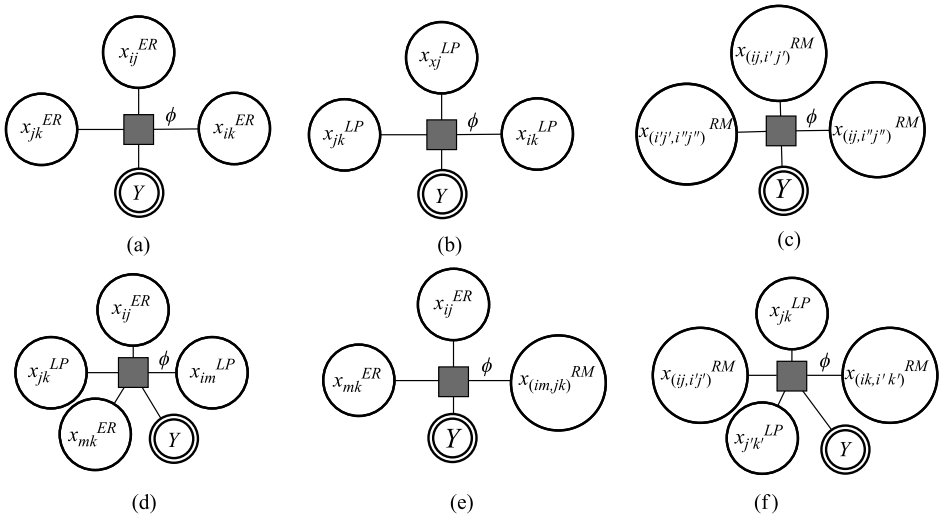


**Fig. 4** Factor graph construction

Based on the extracted features in Section 4, we take into account nine types of factor graphs, each of which corresponds to one kind of factor. For each task, based on its intra-features (except for $f_{T-sim}^{ER}$, $f_{T-sim}^{LP}$, and $f_{T-sim}^{RM}$), we can construct a factor graph respectively, which includes only one factor, one random variable ($x^{ER}$, $x^{LP}$, or $x^{RM}$) and the evidence variables $Y$. These are simple cases because there is no joint probability distribution. Due to space, we only focus on introducing the factor graphs for the other six cases (Fig. 4). For presentation simplicity, we uniformly use $Y$ (denoted as a double circle in factor graphs) to represent the observed evidence variables for each potential function (In fact, they may be different.).

(1) Factor graph based on $f_{T-sim}^{ER}$

As shown in Fig. 4(a), based on the feature $f_{T-sim}^{ER}$, we can construct a factor graph including one factor (i.e., $\phi(x_{ij}^{ER}, x_{jk}^{ER}, x_{ik}^{ER}, Y)$), three random variables (i.e., $x_{ij}^{ER}$, $x_{jk}^{ER}$, and $x_{ik}^{ER}$) and the evidence variables $Y$. We use the factor graph to represent the associated potential, i.e., the joint probability distribution over $x_{ij}^{ER}$, $x_{jk}^{ER}$, and $x_{ik}^{ER}$.

(2) Factor graph based on $f_{T-sim}^{LP}$

Similar to the factor graph based on $f_{T-sim}^{ER}$, the factor graph based on $f_{T-sim}^{LP}$ includes one factor (i.e., $\phi(x_{ij}^{LP}, x_{jk}^{LP}, x_{ik}^{LP}, Y)$), three random variables (i.e., $x_{ij}^{LP}$, $x_{jk}^{LP}$ and $x_{ik}^{LP}$) and the evidence variables $Y$ (shown in Fig. 4(b)). We use it to represent the associated potential, i.e., the joint probability distribution over $x_{ij}^{LP}$, $x_{jk}^{LP}$, and $x_{ik}^{LP}$.

(3) Factor graph based on $f_{T-sim}^{RM}$

The factor graph based on $f_{T-sim}^{RM}$ includes one factor (i.e., $\phi(x_{(ij,i'j')}^{RM}, x_{(i'j',i''j'')}^{RM}, x_{(ij,i''j'')}^{RM}, Y)$), three random variables (i.e., $x_{(ij,i'j')}^{RM}$, $x_{(i'j',i''j'')}^{RM}$, and $x_{(ij,i''j'')}^{RM}$) and the evidence variables $Y$ (shown in Fig. 4(c)). We use the value of the factor to measure the joint probability distribution over $x_{(ij,i'j')}^{RM}$, $x_{(i'j',i''j'')}^{RM}$, and $x_{(ij,i''j'')}^{RM}$.

(4) Factor graph based on $f^{ER+LP}$

The above factor graphs are based on the intra-features within one task, which are used to represent the interaction among the random variables within the same task. Besides them, we can construct some factor graphs to represent the interaction among the random variables across different tasks. As shown in Fig. 4(d), the factor graph based on $f^{ER+LP}$ is used to represent the interaction between ER and LP, which includes one factor $\phi(x_{ij}^{ER}, x_{mk}^{ER}, x_{jk}^{LP}, x_{im}^{LP}, Y)$, four random variables ($x_{ij}^{ER}$, $x_{mk}^{ER}$, $x_{jk}^{LP}$, and $x_{im}^{LP}$) and the evidence variables $Y$. The factor is used to measure the joint probability distribution over $x_{ij}^{ER}$, $x_{mk}^{ER}$, $x_{jk}^{LP}$, and $x_{im}^{LP}$.

(5) Factor graph based on $f^{ER+RM}$

As shown in Fig. 4(e), the factor graph based on $f^{ER+RM}$ is used to represent the interaction between ER and RM. There are one factor $\phi(x_{ij}^{ER}, x_{mk}^{ER}, x_{(im,jk)}^{RM}, Y)$, three random variables ($x_{ij}^{ER}$, $x_{mk}^{ER}$, and $x_{(im,jk)}^{RM}$) and the evidence variables $Y$. Similarly, the factor is used to measure the joint probability distribution over $x_{ij}^{ER}$, $x_{mk}^{ER}$, and $x_{(im,jk)}^{RM}$.

(6) Factor graph based on $f^{LP+RM}$

Based on the inter-feature $f^{LP+RM}$, we can construct the factor graph to represent the interaction between LP and RM. As shown in Fig. 4(f), there are one factor $\phi(x_{jk}^{LP}, x_{(ij,i'j')}^{RM}, x_{j'k'}^{LP}, x_{(ik,i'k')}^{RM}, Y)$, four random variables ($x_{jk}^{LP}$, $x_{(ij,i'j')}^{RM}$, $x_{j'k'}^{LP}$, and $x_{(ik,i'k')}^{RM}$) and the evidence variables $Y$. It is used to measure the joint probability distribution over $x_{jk}^{LP}$, $x_{(ij,i'j')}^{RM}$, $x_{j'k'}^{LP}$, and $x_{(ik,i'k')}^{RM}$.

In addition, by combining the above factor graphs, we can also construct more complex factor graphs. Due to space, here we will not list them all.

### 5.2 CRF-based weight learning algorithm

Based on the factor graphs constructed above, a conditional probability distribution over these random variables can be calculated. However, it is a hard problem to maximize $P$ in Eq. (5). And it requires time that is exponential in $|X^{ER}| \times |X^{LP}| \times |X^{RM}|$. So we adopt an approximation method proposed in [2] to estimate the joint probability distribution (Eq. (7)). Here $X \backslash x^{ER}$ means the remainder random variables after we get rid of $x^{ER}$ from $X$. So do $X \backslash x^{LP}$ and $X \backslash x^{RM}$. Note we only sum over the possible values of ($x^{ER}$, $x^{LP}$, $x^{RM}$). Hence evaluating the normalization constants of all terms only requires time that is linear

in $|X^{ER}| \times |X^{LP}| \times |X^{RM}|$.

$$P^*(X^{ER}, X^{LP}, X^{RM}|Y) = \left(\prod_{x^{ER} \in X^{ER}} P(x^{ER}|X \backslash x^{ER}, Y)\right) \times \left(\prod_{x^{LP} \in X^{LP}} P(x^{LP}|X \backslash x^{LP}, Y)\right) \times \left(\prod_{x^{RM} \in X^{RM}} P(x^{RM}|X \backslash x^{RM}, Y)\right). \quad (7)$$

For each term in Eq. (7), the weights for features can be learned by maximizing the marginal (Eq. (8)), where $X_P$ and $X_N$ are the positive instance set and the negative instance set respectively, i.e., the correct prediction and the incorrect prediction respectively. The weights should maximize the ratio between the conditional probability of the correct prediction and the conditional probability of the incorrect prediction. It is equivalent to maximize the margins of log-linear combination over a set of features.

$$\begin{aligned} W &= argmax_W \left(\prod_{x \in X_P, x' \in X_N} P(x|X \backslash x, Y)/P(x'|X \backslash x', Y)\right) \\ &= argmax_W \Bigg(\sum_{l, Q, x_Q \in X_P, x'_Q \in X_N} w_l(f_l(x_Q, X \backslash x_Q, Y_Q) - \\ &\quad f_l(x'_Q, X \backslash x'_Q, Y_Q))\Bigg). \end{aligned} \quad (8)$$

We propose a CRF-based weight learning algorithm to learn the weights of features, which is shown in Algorithm 1. Given a set of intra-features $f_{intra}$, a set of inter-features $f_{inter}$, a positive instance set $X_P$, a negative instance set $X_N$, a set of predicted variables $X_R$ and a set of evidence variables $Y$, the goal is to return a set of weights $W$ for both $f_{intra}$ and $f_{inter}$.

**Step 1** Learn the initial weights for the intra-features. In this step, we only assign the weights for the intra-features because the ground truth values are usually seldom available resulting in being impossible to calculate the weights for the inter-features.

**Step 2** Based on the learned weights for the intra-features, the values of random variables in $X_R$ are assigned by maximizing the joint probability (Eq. (9)). Here we use maximum a posteriori (MAP) [19] to estimate the values of the random variables (see Section 6). For the fourth equation in Eq. (9), the first term can be solved based on maximum likelihood estimation [32], and the second term represents the prior probability of $x$ which can be estimated according to the distribution of the observed data. The variables $x$ to be learned should be a vector made up of binary integers, but MAP algorithm cannot handle the integer constraints on the variables. Therefore, here we relax the variables $x$ as real values to denote the probabilities. We consider these estimated values as the ground truth values too, so that we can acquire more ground truth values.

$$\begin{aligned} \widehat{x}_{MAP} &= argmax_x P^*(x|X \backslash x, Y) \\ &= argmax_x \log(P^*(x|X \backslash x, Y)) \\ &= argmax_x \log(P^*(X \backslash x, Y|x) P^*(x)) \\ &= argmax_x \log(P^*(X \backslash x, Y|x)) + \log(P^*(x)). \end{aligned} \quad (9)$$

**Step 3** Learn the weights for both the intra-features and the inter-features. Due to more ground truth values, we can acquire more evidence to learn the weights for features. On the one hand, the initial weights for the intra-features are refined. On the other hand, the weights for the inter-features can be learned with the aid of abundant evidence.

## 6 Iterative inference

In this section, we propose an iterative inference algorithm to infer the final results of three tasks (i.e., ER, LP, and RM) simultaneously by performing MAP inference in a CRF. We use MAP inference to estimate the values of the random variables in $X$ (i.e., $X = X^{ER} \cup X^{LP} \cup X^{RM}$).

The pseudocode is shown in Algorithm 2, denoted iterative inference. Given a set of intra-features $f_{intra}$ with their weights $W_{intra}$, a set of inter-features $f_{inter}$ with their weights $W_{inter}$, a set of target variables $X$, a set of evidence variables $Y$, the maximum number of iterations *maxIter* and the converging threshold $\varepsilon$, the goal is to return the values of the random variables in $X$.

**Step 1** Infer the values of the random variables in $X$ based on the intra-features $f_{intra}$ and their weights$W_{intra}$. In this step, we only use $f_{intra}$ and the evidence $Y$ to infer the values of the variables in $X^{ER}$, $X^{LP}$, and $X^{RM}$.

**Step 2** Iteratively infer the values of the random variables in $X$ based on all the features including both $f_{intra}$ and $f_{inter}$. In this step, we use not only $Y$ but also the inferred variables in Step 1 as the evidence. For each iteration, we consider the inferred variables in the previous iteration as new evidence which will be applied to the current iteration. As the iterative inference progresses, the values of the variables in $X^{ER}$, $X^{LP}$, and $X^{RM}$ tend to converge. When the variables values converge (i.e., $\text{diff}(X, X') \leqslant \varepsilon$) or when a user-specified maximum number of iterations is reached (i.e., $i \geqslant maxIter$), the algorithm terminates.

The basic idea of iterative inference is shown in Fig. 5. We perform ER, LP, and RM at the same time. Each iteration uses

**Algorithm 1** CRF-based weight learning

Input: $f_{intra}$, $f_{inter}$, $X_P$, $X_N$, $X_R$ and $Y$
Output: $W$
1 $W_{intra} \leftarrow$ maxMargin($f_{intra}$, $X_P$, $X_N$, $X_R$, $Y$);
2 for each $x$ in $X_R$ do
3 $x \leftarrow$MAP($x$, $X_R$, $Y$, $f_{intra}$, $W_{intra}$);
4 $W \leftarrow$ maxMargin($f_{intra} \cup f_{inter}$, $X_P$, $X_N$, $X_R$, $Y \cup X_R$);
5 return $W$;

**Algorithm 2** Iterative inference

Input: $f_{intra}$, $f_{inter}$, $W_{intra}$, $W_{inter}$, $X$, $Y$, *maxIter* and $\varepsilon$
Output: $X$
1 for each $x$ in $X$ do
2 $x \leftarrow$MAP($x$, $X$, $Y$, $f_{intra}$, $W_{intra}$);
3 for ($i$=0; $i <$ *maxIter* && diff($X$,$X'$) $> \varepsilon$; $i$++)
4 $X' \leftarrow X$;
5 for each $x$ in $X$ do
6 $x \leftarrow$ MAP($x$, $X$, $Y \cup X'$, $f_{intra} \cup f_{inter}$, $W_{intra} \cup W_{inter}$);
7 return $X$;

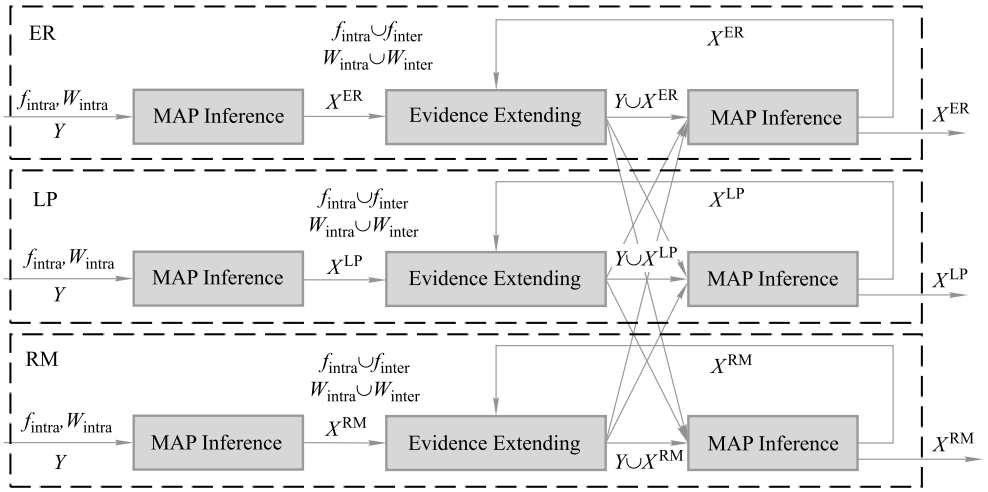


**Fig. 5** The basic idea of iterative inference

the result of its previous iteration as evidence and makes MAP inference to generate the new result which will be applied to the next iteration. Via disseminating information iteratively among different tasks, the predicted result is refined gradually.

Based on the inferred values of the random variables in $X^{ER}$, $X^{LP}$, and $X^{RM}$, we can construct the hidden network $G_H$. First, based on the values of the random variables in $X^{ER}$, we can construct the node set $V_H$ of $G_H$ by merging the nodes in ($G_1$, …, $G_n$) referring to the same entity. Second, besides the original edges in ($G_1$, …, $G_n$), based on the values of the random variables in $X^{LP}$, we can add some new edges to $G_H$ which are predicted via the task of LP. Finally, based on the values of the random variables in$X^{RM}$, the equivalent relations are labeled as the same relation type.

## 7 Experimental evaluation

### 7.1 Dataset

We implement the experiments on a PC with Intel Core i7-2600 @ 3.40GHZ and 8GB main memory. We use two datasets, each of which contains two information networks with intersections. Our goal is to fuse them for each dataset.

For the first dataset, the information networks contained by it are constructed based on ReVerb [33] and Freebase. Twenty percent of triples in ReVerb have been mapped to Freebase relation triples. Therefore we can model ReVerb as a graph and divide it into two information networks (denoted as ReVerbI and ReVerbII respectively). The process of constructing the information networks is as follows. We choose 150 entities from FreeBase, each of which has at least two different representations in ReVerb. We put these different representations and their neighbor nodes within 3 hops into two information networks respectively. The links between these different representations constitute a set of anchor links. Finally we get two information networks which include 8500 nodes and 34000 nodes respectively. Not only the anchor links, but also the ground truth of relation matching between these two information networks can be known according to the mappings from ReVerb to Freebase. So the ground truth for ER and RM is available. We also randomly remove some edges from the information networks. The removed edges are considered as the links to be predicted and they constitute the ground truth for LP.

For the second dataset, we choose 2795289 nodes from YAGO [34] and 2365777 nodes from DBPEDIA [35] to construct an information network respectively. Both of them use WIKIPEDIA identifiers for their instances, so the ground truth for ER can be available. As for the ground truth for RM, it is generated by manually labeling. The generation of the ground truth for LP is the same as that for the first dataset.

### 7.2 Evaluation

We varied the percentage of evidence, that is, the percentage of observed anchor links for ER, observed matching relations for RM and observed links for LP (set as 20%, 40%, 60% and 80% respectively). The model is trained according to the observed part of the network and the remaining parts of the network is predicted. We use precision, recall and F1 performance to evaluate the quality of the results produced by the following different models. These models take into account different combination of features (shown in Table 2).

(1) Baseline1: It considers each task as an independent individual. For each task,, it is also independent for the predictions aiming at different node pairs, links or relations. For each task, we only use their intra-features except for the features representing similarity transferred.

**Table 2** Combination of features considered by each model

| Model | ER | LP | RM |
|---|---|---|---|
| Baseline1 | $f_{sim}^{ER}$, $f_{N-sim}^{ER}$ | $f_{sim}^{LP}$, $f_{CN}^{LP}$, $f_{Out}^{LP}$, $f_{In}^{LP}$ | $f_{sim}^{RM}$ |
| SingleTask | $f_{sim}^{ER}$, $f_{N-sim}^{ER}$, $f_{T-sim}^{ER}$ | $f_{sim}^{LP}$, $f_{CN}^{LP}$, $f_{Out}^{LP}$, $f_{In}^{LP}$, $f_{T-sim}^{LP}$ | $f_{sim}^{RM}$, $f_{T-sim}^{RM}$ |
| SequenceTasks | $f_{sim}^{ER}$, $f_{N-sim}^{ER}$, $f_{T-sim}^{ER}$, $f^{ER+LP}$, $f^{ER+RM}$ | $f_{sim}^{LP}$, $f_{CN}^{LP}$, $f_{Out}^{LP}$, $f_{In}^{LP}$, $f_{T-sim}^{LP}$, $f^{ER+LP}$, $f^{LP+RM}$ | $f_{sim}^{RM}$, $f_{T-sim}^{RM}$, $f^{ER+RM}$, $f^{LP+RM}$ |
| Baseline2 | $f_{sim}^{ER}$, $f_{N-sim}^{ER}$ | Meta-path based features | $f_{sim}^{RM}$ |
| MC-INFM | $f_{sim}^{ER}$, $f_{N-sim}^{ER}$, $f_{T-sim}^{ER}$, $f^{ER+LP}$, $f^{ER+RM}$ | $f_{sim}^{LP}$, $f_{CN}^{LP}$, $f_{Out}^{LP}$, $f_{In}^{LP}$, $f_{T-sim}^{LP}$, $f^{ER+LP}$, $f^{LP+RM}$ | $f_{sim}^{RM}$, $f_{T-sim}^{RM}$, $f^{ER+RM}$, $f^{LP+RM}$ |

(2) SingleTask: Within each task, SingleTask performs collective prediction. Therefore, based on Baseline1, SingleTask also uses $f^{ER}_{T-sim}$, $f^{LP}_{T-sim}$, and $f^{RM}_{T-sim}$ to transfer similarity from other node pairs, links or relations. That is, while processing a pair of nodes (or links, or relations), it also considers the influence from other node pairs (or links, or relations) on it. However, SingleTask still considers multiple tasks as separate ones.

(3) SequenceTasks: SequenceTasks considers both the intra-features and the inter-features. Also it considers the multiple tasks as related ones. But it performs tasks one at a time in a fixed order. In our experiment, we performs them in such an order: first ER, then LP, finally RM. The later tasks can use the predictions of earlier tasks. But the reverse is not true.

(4) Baseline2: Besides the above variants of our proposed model, we also compare our model against previous work on each task respectively. As for ER, we choose the method proposed in [7] as Baseline2, which formulates the inference problem for anchor links as a stable matching problem between the two sets of user accounts in two different networks. As for LP, Baseline2 represents the meta-path based multi-network link prediction method proposed in [17]. Here we randomly select three relations from relation set. For each relation, some inter-network meta paths and inter-network meta paths are constructed. As for RM, Baseline2 is based on the idea proposed in [27] which matches the relationships by clustering them.

(5) MC-INFM (our model): Both the intra-features and the inter-features are used in MC-INFM. Based on CRF model, the weight of each feature is learned and the results of these tasks are inferred simultaneously.

The precision, recall and F1 performance of different models are illustrated in Fig. 6, Fig. 7 and Fig. 8, respectively. The variation trend of experimental result on two datasets is similar. Baseline1 just makes a simple similarity comparison. SingleTask improves upon Baseline1 by taking into account the effects within one task, i.e., performing collective prediction. But it considers the multiple tasks as separate ones, resulting in ignoring the mutual promotion between tasks. SequenceTasks improves upon SingleTask by taking into account both the effects within one task and the effects among tasks. However, it does not allow earlier tasks to use the result of later ones, resulting in lower performance. Baseline2 improves some of the variants of our proposed model, but it greatly depends on the setting of meta-paths or the number of clusters. Also it does not take into account either the effects within one task or the effects between multiple tasks. MC-INFM further improves upon SequenceTasks by adopting our CRF-based weight learning algorithm and iterative inference algorithm. It also makes full use of the mutual promotion between tasks, leading to the best performance.

As for ER, because it is the first task performed in SequenceTasks which is not affected by the tasks performed later, SequenceTasks gets the same result as SingleTask. As for LP and RM on the second dataset, when the percentage of evidence is lower, the performance of SingleTask, SequenceTasks and MC-INFM is lower than those of Baseline1 and Baseline2. That is because the lack of evidence at this time makes the interaction between tasks (or the collective prediction within one task) may have little effects (even side effects) on the predicted result. But with the increase of the percentage of evidence, the performance increases gradually.

## 8 Conclusion

In this paper, we present a novel information networks fusion model based on multi-task coordination, which casts the fusion problem as a probabilistic inference problem, and collectively processes multiple tasks (including entity resolution, link

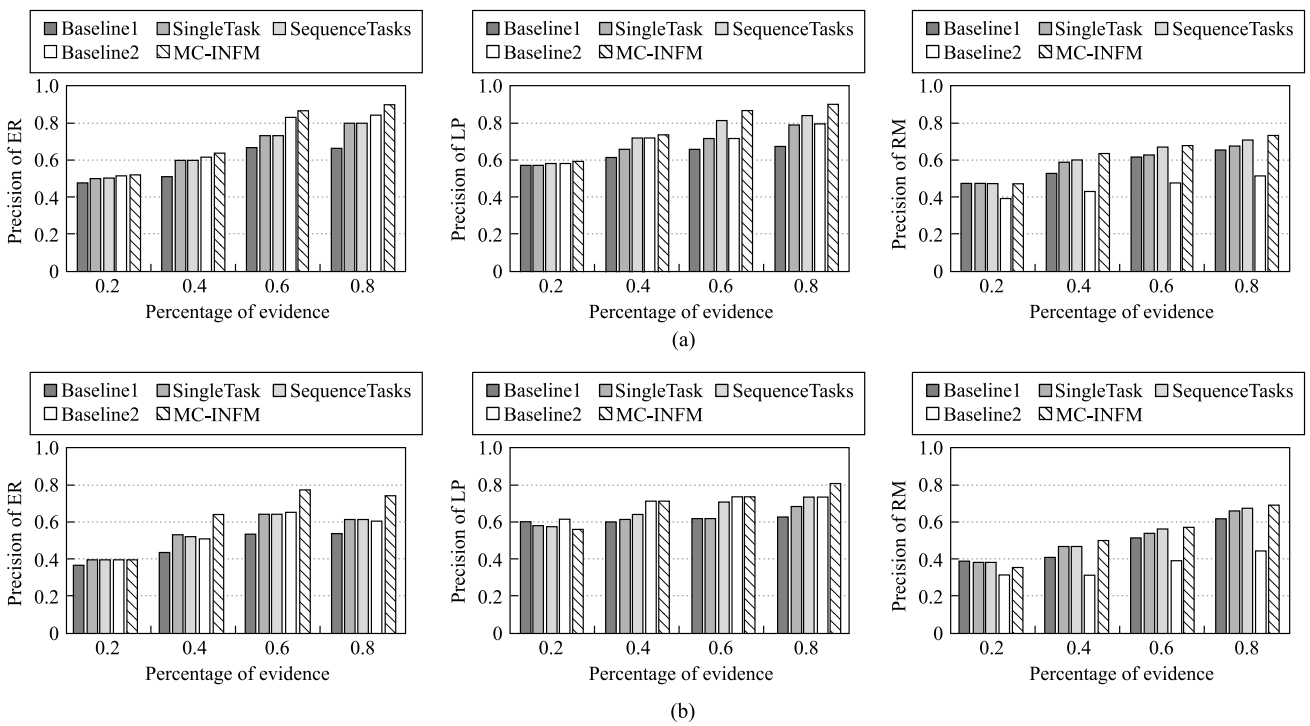


**Fig. 6** Comparison of different models on precision. (a) Fusion of ReVerbI and ReVerbII; (b) fusion of YAGO and DBPEDIA

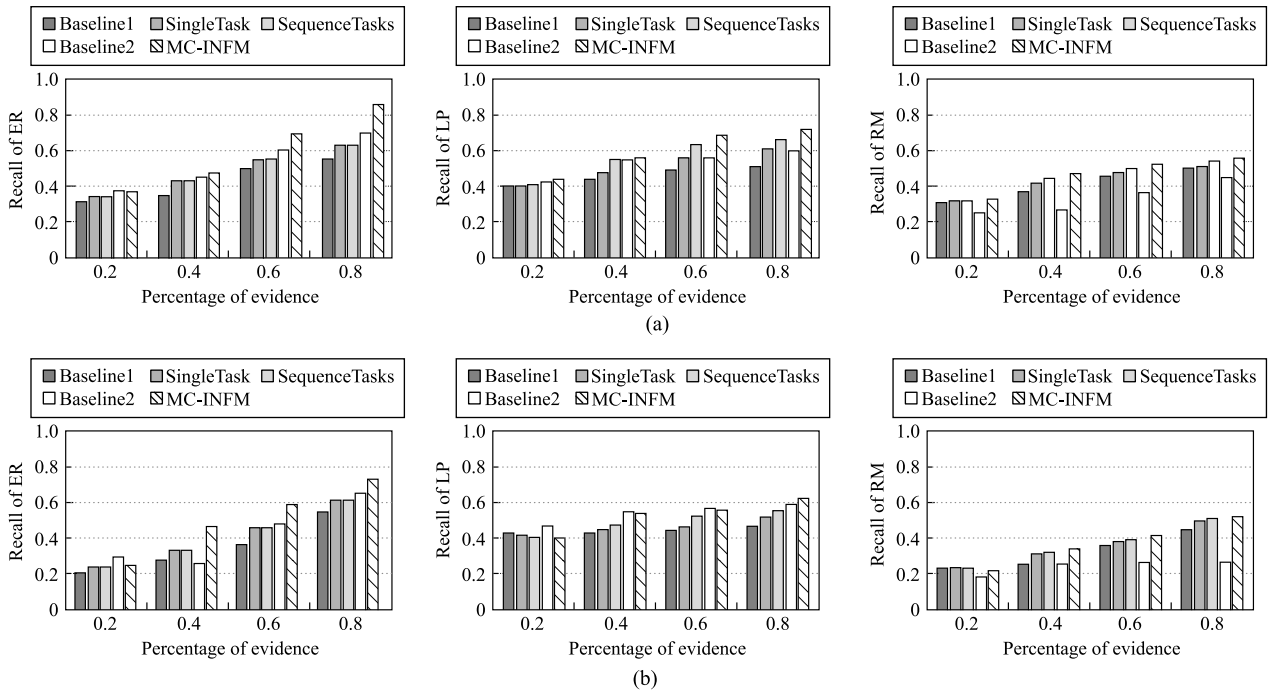


**Fig. 7** Comparison of different models on Recall. (a) Fusion of ReVerbI and ReVerbII; (b) fusion of YAGO and DBPEDIA

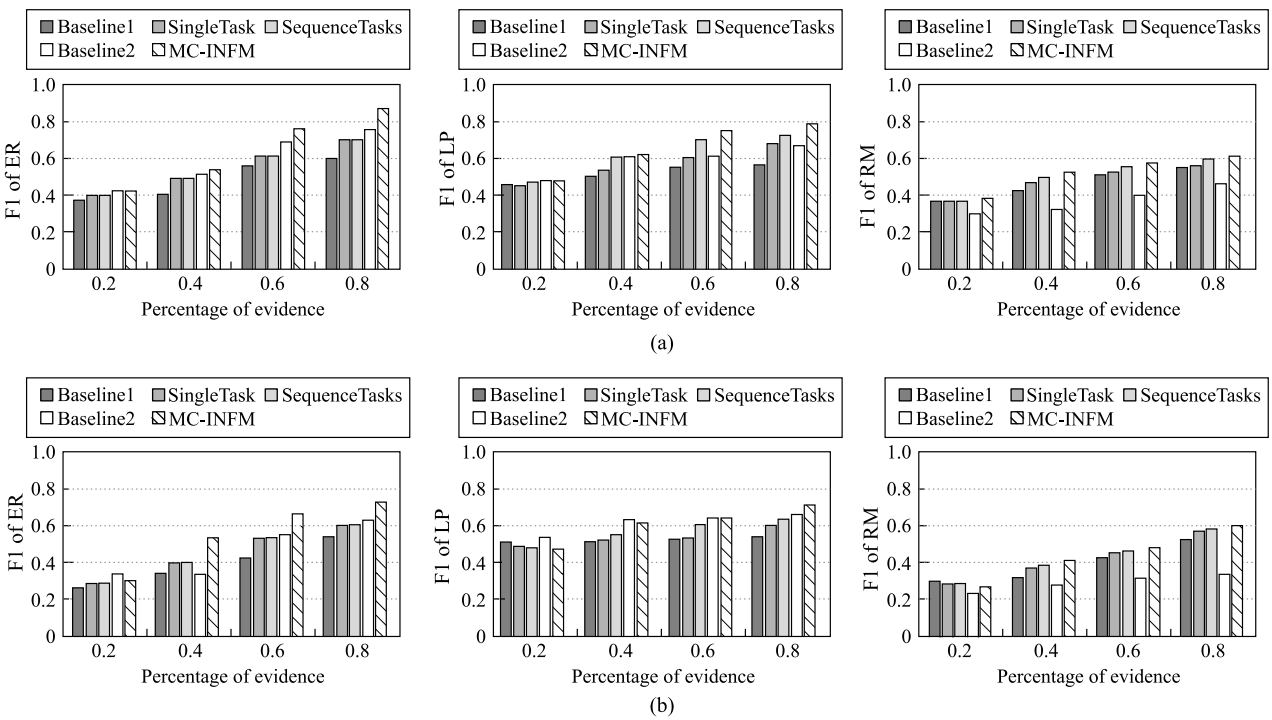


**Fig. 8** Comparison of different models on F1. (a) Fusion of ReVerbI and ReVerbII; (b) fusion of YAGO and DBPEDIA

prediction and relation matching) to infer the final result of fusion. First, we define the intra-features and the inter-features respectively and model them as factor graphs, which can provide abundant evidence to infer. Second, we propose a CRF-based weight learning algorithm to learn the weight of each feature. Then we propose an iterative inference algorithm to infer the results of these tasks simultaneously by performing the maximum probabilistic inference. Experiments demonstrate the effectiveness of our proposed model.

At present, our fusion model is based on the premise that the data from each information network is correct. In fact, there may be some wrong data in these information networks, which needs to be discarded during fusion. Therefore, in our future work, we will work on the fusion model on more complex information networks, especially on the networks including dirty data. In addition, now we focus on the coordination among the tasks of ER, LP, and RM. And we have not consider the problem of time consuming. Next we will further improve the

quality and efficiency of fusion by considering more tasks and performance optimization strategies.

**Acknowledgements** This work was supported by the National Key R&D Program of China (2018YFB1003404) and the National Natural Science Foundation of China (Grant Nos. 61672142, U1435216, 61602103).

## References

1. Zhang J. Social network fusion and mining: a survey. 2018, arXiv preprint arXiv:1804.09874
2. Namata G, Kok S, Getoor L. Collective graph identification. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2011, 87–95
3. Lacoste-Julien S, Palla K, Davies A, Kasneci G, Graepel T. SIGMa: simple greedy matching for aligning large knowledge bases. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2012, 572–580
4. Suchanek F, Abiteboul S, Senellart P. PARIS: probabilistic alignment of relations, instances, and schema. Proceedings of the VLDB Endowment, 2011, 5(3): 157–168
5. Niu F, Re C, Doan A, Shavlik J. Tuffy: scaling up statistical inference in Markov logic networks using an RDBMS. Proceedings of the VLDB Endowment, 2011, 4(6): 373–384
6. Lao N, Mitchell T, Cohen W. Random walk inference and learning in a large scale knowledge base. In: Proceedings of Conference on Empirical Methods in Natural Language Processing. 2011, 27–31
7. Kong X, Zhang J, Yu P. Inferring anchor links across multiple heterogeneous social networks. In: Proceedings of ACM International Conference on Information and Knowledge Management. 2013, 179–188
8. Koutra D, Tong H, Lubensky D. Big-align: fast bipartite graph alignment. In: Proceedings of International Conference on Data Mining. 2013, 389–398
9. Zafarani R, Liu H. Connecting users across social media sites: a behavioral-modeling approach. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2013, 41–49
10. Zhang J, Shao W, Wang S, Kong X, Yu P. PNA: partial network alignment with generic stable matching. In: Proceedings of IEEE International Conference on Information Reuse and Integration. 2015, 166–173
11. Zhang J, Yu P. Integrated anchor and social link predictions across partially aligned social networks. In: Proceedings of International Joint Conference on Artificial Intelligence. 2015, 1620–1626
12. Zhang J, Yu P. Multiple anonymized social networks alignment. In: Proceedings of International Conference on Data Mining. 2015, 599–608
13. Zhang J, Yu P. PCT: partial co-alignment of social networks. In: Proceedings of International World Wide Web Conference. 2016, 749–759
14. Zhang J, Kong X, Yu P. Predicting social links for new users across aligned heterogeneous social networks. In: Proceedings of International Conference on Data Mining. 2013, 1289–1294
15. Zhang J, Kong X, Yu P. Transfer heterogeneous links across location-based social networks. In: Proceedings of ACM International Conference on Web Search and Data Mining. 2014, 303–312
16. Zhang J. Link prediction across heterogeneous social networks: a survey. Dissertation, University of Illinois at Chicago, US. 2014
17. Zhang J, Yu P, Zhou Z. Meta-path based multi-network collective link prediction. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2014, 1286–1295
18. Richardson M, Domingos P. Markov logic networks. Machine Learning, 2006, 62: 107–136
19. Lafferty J, McCallum A, Pereira F. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of International Conference on Machine Learning. 2001, 282–289
20. Zhou T, Lv L, Zhang Y. Predicting missing links via local information. The European Physical Journal B, 2009, 71(4): 623–630
21. Lv L, Zhou T. Link prediction in complex networks: a survey. Physica A: Statistical Mechanics and its Applications, 2011, 390: 1150–1170
22. Lee J Y, Tukhvatov R. Evaluations of similarity measures on VK for link prediction. Data Science and Engineering, 2018, 3(3): 277–289
23. Hasan M, Chaoji V, Salem S, Zaki M. Link prediction using supervised learning. In: Proceedings of SIAM International Conference on Data Mining. 2006
24. Aditya K, Menon A, Elkan C. Link prediction via matrix factorization. In: Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. 2011, 437–452
25. Dunlavy D, Kolda T, Acar E. Temporal link prediction using matrix and tensor factorizations. ACM Transactions on Knowledge Discovery from Data, 2011, 5(2): 10
26. Tang J, Gao H, Hu X, Liu H. Exploiting homophily effect for trust prediction. In: Proceedings of ACM International Conference on Web Search and Data Mining. 2013, 53–62
27. Yates A, Etzioni O. Unsupervised methods for determining object and relation synonyms on the web. Journal of Artificial Intelligence Research, 2009, 34(1): 255–296
28. Dong X, Srivastava D. Knowledge curation and knowledge fusion: challenges, models and applications. In: Proceedings of the ACM SIGMDD International Conference on Management of Data. 2015, 2063–2066
29. Galarraga L, Heitz G. Canonicalizing open knowledge bases. In: Proceedings of ACM International Conference on Information and Knowledge Management. 2014, 1679–1688
30. Cohen W, Ravikumar P, Fienberg S. A comparison of string distance metrics for name-matching tasks. In: Proceedings of International Joint Conference on Artificial Intelligence. 2003, 73–78
31. Chen Y, Wang D. Knowledge expansion over probabilistic knowledge bases. In: Proceedings of International Conference on Management of Data. 2014, 649–660
32. Rossi R J. Mathematical Statistics: an Introduction to Likelihood Based Inference. New York: John Wiley & Sons, 2018
33. Fader A, Soderland S, Etzioni O. Identifying relations for open information extraction. In: Proceedings of Conference on Empirical Methods in Natural Language Processing. 2011, 1535–1545
34. Suchanek F, Kasneci G, Weikum G. Yago: a core of semantic knowledge. In: Proceedings of International World Wide Web Conference. 2007, 697–706
35. Lehmann J, Isele R, Jakob M, Jentzsch A, Kontokostas D, et al. DBpedia — a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web, 2013, 6(2): 167–195

Dong LI is PhD candidate. He received his Master degree in Computer Technology from Northeastern University, China in 2008. His research interests include social networks analysis and data mining.

Derong Shen received her PhD degree in Computer Software and Theory from Northeastern University, China in 2004. Currently, she is a professor in the School of Computer Science & Engineering, Northeastern University, China. Her research interests include social networks analysis and data integration. She is a member of senior CCF, IEEE, and ACM.

Yue Kou received her PhD degree in Computer Software and Theory from Northeastern University, China in 2009. Currently, she is an associate professor in the School of Computer Science & Engineering, Northeastern University, China. Her research interests include social networks analysis and data mining. She is a member of CCF, IEEE, and ACM.

Tiezheng Nie received his PhD degree in Computer Software and Theory from Northeastern University, China in 2009. Currently, he is an associate professor in the School of Computer Science & Engineering, Northeastern University, China. His research interests include data integration and data mining. He is a member of CCF, IEEE, and ACM.