# Multiperspective Graph-Theoretic Similarity Measure

Dung D. Le
Singapore Management University
Singapore
ddle.2015@phdis.smu.edu.sg

Hady W. Lauw
Singapore Management University
Singapore
hadywlauw@smu.edu.sg

## ABSTRACT

Determining the similarity between two objects is pertinent to many applications. When the basis for similarity is a set of object-to-object relationships, it is natural to rely on graph-theoretic measures. One seminal technique for measuring the structural-context similarity between a pair of graph vertices is SimRank, whose underlying intuition is that two objects are similar if they are connected by similar objects. However, by design, SimRank as well as its variants capture only a single view or perspective of similarity. Meanwhile, in many real-world scenarios, there emerge multiple perspectives of similarity, i.e., two objects may be similar from one perspective, but dissimilar from another. For instance, human subjects may generate varied, yet valid, clusterings of objects. In this work, we propose a graph-theoretic similarity measure that is natively multiperspective. In our approach, the observed object-to-object relationships due to various perspectives are integrated into a unified graph-based representation, stylised as a hypergraph to retain the distinct perspectives. We then introduce a novel model for learning and reflecting diverse similarity perceptions given the hypergraph, yielding the similarity score between any pair of objects from any perspective. In addition to proposing an algorithm for computing the similarity scores, we also provide theoretical guarantees on the convergence of the algorithm. Experiments on public datasets show that the proposed model deals better with multiperspectivity than the baselines.

**ACM Reference Format:**
Dung D. Le and Hady W. Lauw. 2018. Multiperspective Graph-Theoretic Similarity Measure. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3269206.3271758

## 1 INTRODUCTION

Determining whether two objects are similar is a fundamental task in many real-world applications. When browsing a product online, a customer may be presented other similar products to consider. On Pinterest, a visual discovery platform, as users pin images onto boards, they may wish to discover other related images. Search engines, such as Google, support *similar* button under a result link, which would return similar pages to the link. Similarity is also an elemental component of data-driven tasks such as clustering [9], entity resolution [1], retrieval [16], recommendation [2], etc.

There are various ways to measure similarity. Some, such as cosine similarity, are based on content or features, e.g., whether two documents contain the same words, or two products have the same attributes. Others, such as KL-divergence, are based on probability distributions. Yet other measures may be domain-specific, such as sequence alignment [5]. These diverse types of similarity are orthogonal, reflecting various aspects. They are not so much alternatives as complements, and indeed they have been used in conjunction in some applications such as entity resolution.

**Problem.** In this work, we focus on the notion of *graph-theoretic similarity*, based on object-to-object "relationships" (the specific definition of which may be domain-dependent). For instance, a Web page may link to another; two images may belong to the same Pinterest's board. In each case, object-to-object relationships become the basis for inferring the similarity between any two objects of interest (pages, images). Naturally, such notion of relationship-based similarity lends itself well to a graph-based formulation, with vertices for objects, and edges for relationships between objects.

SimRank [11] lays a foundation for graph-based similarity measurement, premised on the intuition that the similarity between a pair of objects is dependent on the similarity of other object pairs. We consider two objects $(i, j)$ *similar*, if the two objects are respectively related to other objects $k$ (related to $i$) and $l$ (related to $j$) that are themselves *similar*. Under this definition, two Web pages are similar if they respectively link to two other pages that are similar. Two images on Pinterest are similar if they respectively belong to the same boards as two other images that are themselves similar. Two users are similar if they respectively adopt similar products.

However, SimRank is a *uniperspective* measure. It assumes only one perception of similarity. In some scenarios, there are actually multiple perspectives of similarity. What may be similar according to one perspective may be different according to another. This may arise due to different facets of relationships, e.g., two products may be "related" in different ways: browsed together, purchased together, same manufacturer, etc. This may also arise due to different agents that express the relationships, e.g., someone may group tourist attractions based on activities (strolling, amusement park), while another based on artistic value (architecture, museums) or neighborhoods [22]. A uniperspective approach (e.g., SimRank) is not designed for capturing "different strokes for different folks".

How then do we cope with the presence of multiple perspectives? There are a couple of *naive* approaches. One is to ignore the multiplicity, creating a uniperspective measure by merging the disparate relationships into a single graph and applying the SimRank on this one graph. Another is to isolate each perspective, creating multiperspective measures by maintaining a distinct graph for each perspective and applying SimRank on each graph separately.

*CIKM '18, October 22–26, 2018, Torino, Italy*

ACM ISBN 978-1-4503-6014-2/18/10...$15.00
https://doi.org/10.1145/3269206.3271758

The former may underfit, due to a lack of capacity to model idiosyncratic nuances of similarity. The latter may overfit, due to the sparsity of relationships within each perspective and the potential to capture incidental relationship instances that may not generalize.

**Proposed Approach.** Therefore, we propose a natively *multi-perspective* approach to measuring graph-theoretic similarity. As input, we are given not one graph, but multiple graphs corresponding to multiple perspectives, with each graph reflecting relationships among objects from a specific perspective. As output, we seek to measure the similarity between a pair of objects according to a particular perspective. The key intuition underlying this formulation is to model not only the perspective-specific *inter-object* similarity between any pair of objects, but also the *inter-perspective* similarity between any two perspectives. The latter allows the former to be learned simultaneously across all perspectives, rendering an advantage in sharing information across similar perspectives, which helps to address the sparsity of relationship instances.

**Contributions.** In this work, we make several contributions. *First*, to our best knowledge, we are the first to define the problem of multiperspective graph-theoretic similarity. *Second*, we propose a multiperspective formulation capable of expressing both intra-perspective similarity between two objects and inter-perspective similarity between two perspectives (Section 3). *Third*, we describe a straightforward solution that works as a pipeline whereby we first learn inter-perspective similarities followed by perspective-specific inter-object similarities (Section 4). *Fourth*, we further propose a computationally more efficient joint solution where both are learned simultaneously (Section 5). For each solution, we describe its learning algorithm and provide arguments for the existence of its solution. *Fifth*, in Section 6, we conduct experiments on public datasets to validate the effectiveness of the multiperspective graph-theoretic approach, showing its outperformance when compared to uniperspective graph-based baselines, and multiperspective non-graph baseline. *Finally*, Section 7 analyzes the storage and computational complexities of the models, and describes a clustering-based heuristic to approximate the multiperspective similarities, offering a trade-off between speed and accuracy.

## 2 RELATED WORK

Similarity measurement is a broad topic. Since our key thrust is incorporating multiperspectivity into graph-theoretic similarity measure, in the following we relate our work to the closest branches in the literature, namely other methods for graph-theoretic similarity as well as other notions of multiperspective similarity.

**Graph-Theoretic Similarity.** Most of the previous works in graph-based similarity are based on SimRank [11]. We first briefly review SimRank. Given a graph $G(V, E)$, SimRank measures the similarity between two vertices based on the graph structure. Each vertex represents one object. Formally, the SimRank similarity score $S(a, b)$ between two vertices $a, b$ is defined as follows:

$$S(a,b) = \begin{cases} \frac{C}{|N(a)||N(b)|} \sum_{i=1}^{|N(a)|} \sum_{j=1}^{|N(b)|} S(N_i(a), N_j(b)), & \text{if } a \neq b, \\ 1, & \text{if } a = b \end{cases} \tag{1}$$

in which $C$ is the damping factor between 0 and 1; $N(a)$ and $N(b)$ comprise the neighbors of $a$ and $b$ respectively. In other words, the SimRank score between $a, b$ is defined in terms of the SimRank scores of their neighbors. The base case is the similarity between a vertex and itself, which is always 1. If a vertex $a$ has no neighbor, then we have $S(a, b) = 0$ for any vertex $b \neq a$.

SimRank has been extended in diverse directions, of which we cite a few here, as a complete enumeration would not have been feasible. [13] proposed non-iterative computation for dynamically changing graphs. [8] parallelized the similarity computation using GPUs. [14] optimized the computation when the target was computing the similarity of a single pair of objects. [12] sought to speed up the computation for extremely large graphs. In the context of translation lexicons, [4] presented a modification of SimRank to measure similarity across two graphs (one object in each graph); this is distinct from the notion of multiperspective as there is only one perspective. We are not aware of any SimRank extension incorporating multiperspective similarity as we are proposing.

Besides SimRank, there are other notions of graph-based similarity. Most are based on random walk variants [19] (e.g., Personalized PageRank [10, 17] or hubs and authorities [7]). [18] was concerned with metapaths in heterogeneous information networks.

**Multiperspective Similarity.** Outside of graph-theoretic methods, there exists other methods that could be interpreted as learning multiperspective similarities. The closest is [22], which uses matrix factorization to learn personalized clustering of objects; each person could be seen as a perspective and two objects are similar if they belong to the same cluster. However, instead of graph theory, it is framed in terms of similarity learning [3], where the objective is to derive a function mapping features to similarity labels. In the absence of features, it turns into learning latent representations from similar labels. In Section 6, we compare to the latter.

The notion of crowdsourced clustering [20, 21] deals with the problem of learning clustering from similarity labels generated by Turkers. There, the focus is not so much to reflect a variety of perspectives as to arrive at the common consensus.

## 3 OVERVIEW

In this section, we provide an overview of the problem formulation and solutions. After formally introducing our notations and defining the data representation in terms of our hypergraph formulation, we outline the solution framework for deriving the perspective-specific inter-object similarities and inter-perspective similarities.

### 3.1 Problem Formulation

Let $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$ be the universal set of objects for which we seek to infer similarities. Suppose that we are interested in modeling $m$ different perspectives $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$ over the similarities of objects in $\mathcal{O}$. For each perspective $p \in \mathcal{P}$, we are given a graph $G_p(\mathcal{O}, E_p)$, where $E_p \subseteq \mathcal{O} \times \mathcal{O}$ comprises edges between pairs of objects that $p$ considers related. The collection of such graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_m\}$ make up the input to the problem.

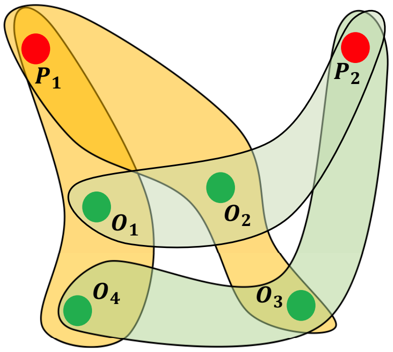

**Figure 1: Illustration of the Hypergraph Representation**

Because the respective $G_p$'s are defined over the same set of vertices $\mathcal{O}$, we seek a unified representation that allows the integration of the $m$ separate graphs. There are several equivalent representations for what is essentially the same data. One is a multi-labeled graph, with perspectives serving as edge labels. Another is a bipartite graph, with perspectives as one type of vertices, and object pairs as the other type. Since we are as concerned with the inter-perspective similarity as we are with the inter-object similarity, for most of the subsequent discussions, we resort to a representation where perspectives and objects are both vertices. A natural candidate for such a representation is a 3-uniform hypergraph, whereby each edge relates exactly three vertices: one *perspective* vertex and two *object* vertices considered similar by the former.

From the input $\mathcal{G}$, we construct a 3-uniform hypergraph $\mathcal{H} = (\mathcal{X}, \mathcal{E})$ consisting of a set of vertices $\mathcal{X} = \mathcal{P} \cup \mathcal{O}$ and a set of hyperedges $\mathcal{E} = \{(p_k, o_i, o_j) : 1 \leq k \leq m; 1 \leq i, j \leq n\}$, in which $(p_k, o_i, o_j) \in \mathcal{E}$ means that $o_i$ and $o_j$ are related according to perspective $p_k$, i.e., $(o_i, o_j) \in E_{p_k}$ in $G_{p_k}$. Figure 1 illustrates an example hypergraph with two perspectives $\mathcal{P} = \{p_1, p_2\}$ (red) and four objects $\mathcal{O} = \{o_1, o_2, o_3, o_4\}$ (green). Hyperedges $(p_1, o_1, o_4)$ and $(p_1, o_2, o_3)$ indicate that according to perspective $p_1$, object $o_1$ is related to object $o_4$, while object $o_2$ is related to object $o_3$. In contrast, according to $p_2$, $o_1$ is related to $o_2$, and $o_3$ is related to $o_4$.

Given a multiperspective hypergraph $\mathcal{H}(\mathcal{X}, \mathcal{E})$, the similarity score of two objects $o_i, o_j \in \mathcal{O}$ according to perspective $p \in \mathcal{P}$ is denoted as $S_p(o_i, o_j)$, whose value is bounded by $[0, 1]$. A special case is $S_p(o_i, o_j) = 1$ when $i = j$. We are now ready to state the problem formally as follows.

**Problem 1 (Multiperspective Similarity).** *Given a multiperspective hypergraph $\mathcal{H}$, determine the similarity score $S_p(o_i, o_j)$ for each perspective $p \in \mathcal{P}$ and pair of objects $o_i \neq o_j \in \mathcal{O}$.*

### 3.2 Framework for Multiperspective Solutions

A naive solution to multiperspectivity (i.e., Problem 1) is to run SimRank (Equation 1) on each perspective's component graph $G_p$ separately. We refer to this solution as *Disjoint-SimRank*. While this produces perspective-specific inter-object similarities, the main issue is that there may not be sufficient information within each $G_p$ to learn the similarities among objects effectively. If every perspective is distinct and unique, then perhaps we could do no better than this. However, realistically, the various perspectives may share some degree of agreement in how they perceive the similarities among objects. If so, then there would be an opportunity to let a perspective collaborate with other similar perspectives, filling the gaps in each other's knowledge of object similarities.

**Table 1: List of Notations**

| Symbols | Description |
|---|---|
| $\mathcal{P}$ | $\{p_1, p_2, \ldots, p_m\}$-perspective vertices |
| $\mathcal{O}$ | $\{o_1, o_2, \ldots, o_m\}$-object vertices |
| $\mathcal{X}$ | set of all vertices, i.e., $\mathcal{P} \cup \mathcal{O}$ |
| $\mathcal{E}$ | set of all hyperedges, i.e.,<br>$\{(p, o_i, o_j) : o_i$ and $o_j$ are related, according to $p\}$ |
| $m$ | number of perspectives |
| $n$ | number of objects |
| $N_p(o_i)$ | $\{o_j \in \mathcal{O} \mid (p, o_i, o_j) \in \mathcal{E}\}$ |
| $S_p(o_i, o_j)$ | similarity score between vertices $o_i$ and $o_k$, according to perspective $p$ |
| $\mathcal{S}_p$ | $[S_p(o_i, o_j)]_{n \times n}$: perspective-specific inter-object similarity score matrix |
| $W_p$ | Column-normalized matrix (size $n \times n$) of the adjacency matrix with respect to perspective $p \in \mathcal{P}$ |
| $\text{sim}(p, p')$ | similarity between two perspective vertices $p, p'$ |
| $C$ | damping factor |
| $\text{I}_n$ | identity matrix with size $n$ |

Therefore, for a truly multiperspective solution, we advocate enabling information sharing across perspectives, to a degree correlated with the similarity among the corresponding perspectives. Let's denote $\text{sim}(p, p') \in [0, 1]$ to be the similarity between two perspectives $p, p' \in \mathcal{P}$. How these values may be derived for $p, p' \in \mathcal{P}$ will be discussed shortly.

To infer the similarity $S_p(o_i, o_j)$ between two objects $o_i$ and $o_j$ according to $p$, we propose to expand the definition in Equation 1 to incorporate inter-perspective similarity $\text{sim}(p, p')$, in such a way that $S_p(o_i, o_j)$ is expressed in terms of the corresponding object similarities according to other perspectives $p'$ as well, as shown in Equation 2. Here, $N_p(o_i)$ comprises the neighbors of $o_i$ in $G_p$.

$$S_p(o_i, o_j) = \frac{C}{|\mathcal{P}|} \sum_{p' \in \mathcal{P}} \text{sim}(p, p') \sum_{o_k \in N_{p'}(o_i)} \sum_{o_l \in N_{p'}(o_j)} \frac{S_{p'}(o_k, o_l)}{|N_{p'}(o_i)||N_{p'}(o_j)|}, \tag{2}$$

Equation 2 captures a couple of fundamental principles. First, the similarity between two objects depends on the similarities between other objects related to those objects of interest. Second, distinctly in our formulation, the similarity between two objects of interest according to a specific perspective also depends on the similarities between related objects as seen by similar perspectives.

Let $\mathcal{S}_p = [S_p(o_i, o_j)]_{n \times n}$ be the matrix representation of the perspective-specific inter-object similarity scores, and $W_p$ be the column-normalized matrix of the adjacency matrix with respect to $p \in \mathcal{P}$. We can express Equation 2 in matrix form as follows:

$$\mathcal{S}_p = \frac{C}{|\mathcal{P}|} \sum_{p' \in \mathcal{P}} \text{sim}(p, p').{W_{p'}}^T \mathcal{S}_{p'} W_{p'} \tag{3}$$

In this multiperspective framework, one important component is the inter-perspective similarity $\text{sim}(p, p')$, determining the degree to which information is shared between one perspective and another. The straightforward solution is to treat it as a pipeline: first compute the similarity between perspectives, then solve Equation 2 to compute the perspective-specific inter-object similarities. We refer to this as Pipelined-SimRank (Section 4). In Section 5, we further propose a refined formulation MP-SimRank to compute the inter-perspective similarities and the perspective-specific inter-object similarities simultaneously. We expect that jointly learning both types of similarities would reinforce the performance of the framework at lower complexities than the former solution.

# 4 STRAIGHTFORWARD SOLUTION: PIPELINED-SIMRANK

In this section, we describe Pipelined-SimRank that still enables information sharing across perspectives through a pipelined solution. The key idea is to induce unidirectional dependency from the inter-perspective $\text{sim}(p, p')$ to the inter-object $S_p(o_i, o_j)$, but not the other way around. This directionality implies that $\text{sim}(p, p')$ has to be inferred from the multiperspective hypergraph $\mathcal{H}$ itself.

## 4.1 Inter-Perspective Similarity

As mentioned in Section 3, each perspective $p$ is associated with a graph of object-to-object relationships $G_p$. Intuitively, we consider two perspectives $p$ and $p'$ to be similar, if their corresponding graphs $G_p$ and $G_{p'}$ are similar, which implies that when $p$ considers two objects related, it is likely that $p'$ does as well. We express this intuition in graph-theoretic form as follows.

Let us transform the input hypergraph $\mathcal{H} = (\{\mathcal{P}, \mathcal{O}\}, \mathcal{E})$ into a bipartite graph $\mathcal{B}$ with two types of vertices, as illustrated in Figure 2 (unrelated to Figure 1). The first type are perspective vertices $\mathcal{P}$ (left). The second type are "object-pair" vertices $\mathcal{O} \times \mathcal{O}$, formed from all pairs of non-identical objects (right). An edge from a perspective $p$ to an object-pair vertex $o_{ij}$ exists in this bipartite graph $\mathcal{B}$ iff $(p, o_i, o_j) \in \mathcal{E}$ in the original hypergraph $\mathcal{H}$.

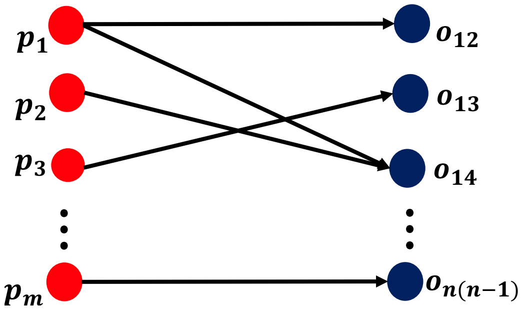

**Figure 2: Pipelined-SimRank: Bipartite graph for computing similarity between perspective nodes**

Once the bipartite graph $\mathcal{B}$ is in place, we can apply a graph-theoretic measure such as the bipartite variant of SimRank [11] to compute the inter-perspective similarity $\text{sim}(p, p')$, which will be used in the next phase for computing $S_p(o_i, o_j)$.

## 4.2 Learning Algorithm

Algorithm 1 encapsulates the pipelined solution Pipelined-SimRank, which involves two phases. In the first phase, we compute inter-perspective similarities $\text{sim}(p, p')$ for all $p, p' \in \mathcal{P}$ as described above. Thereafter, in the second phase, we use these inter-perspective similarities in Equation 2 to compute the inter-object similarities for each perspective. Note that $\text{sim}(p, p')$ is now fixed in the second phase. The initial values $S_p^{(0)}(*, *) \forall p \in \mathcal{P}$ at the start of the iterations (line 7) of Algorithm 1 are specified in Equation 4 below:

$$S_p^{(0)}(o_i, o_j) = 0 \text{ if } i \neq j \text{ and } 1 \text{ if } i = j \tag{4}$$

The solution to Equation 2 can be reached by iteration to a fixed-point (lines 8–9). Finally, the algorithm returns the converged inter-object similarities, as well as the inter-perspective similarities.

**Algorithm 1** Pipelined-SimRank

```
Require: Hypergraph $\mathcal{H}$ (defined as in Section 3)
1: /*— create bipartite graph from hypergraph — */
2: $\mathcal{B} \leftarrow \text{bipartiteTransform}(\mathcal{H})$
3: /*— compute the similarity between perspectives — */
4: $\{\text{sim}^{(*)}(p, p')\}_{\forall p, p' \in \mathcal{P}} \leftarrow \text{bipartiteSimRank}(\mathcal{B})$
5: Initialize $S_p^{(0)} \leftarrow \text{I}_n, \forall p \in \mathcal{P}$
6: while not converged do {
       $S_p^{(t+1)}(o_i, o_j) = \frac{C}{|\mathcal{P}|} \sum_{p' \in \mathcal{P}} \text{sim}^{(*)}(p, p') \times \sum_{o_k \in N_{p'}(o_i)} \sum_{o_l \in N_{p'}(o_j)} \frac{S_{p'}^{(t)}(o_k, o_l)}{|N_{p'}(o_i)||N_{p'}(o_j)|},$   (5)
       (for $1 \leq i \neq j \leq n$)
     and $S_p^{(t+1)}(o_i, o_i) = 1($ for $1 \leq i \leq n)$
7: Return $\{S_p^{converged}(o_i, o_j), \forall p \in \mathcal{P}, o_i, o_j \in \mathcal{O}\}$
8: and $\{\text{sim}^{(*)}(p, p'), \forall p, p' \in \mathcal{P}\}$.
```

## 4.3 Convergence Property

We now prove that Algorithm 1 will eventually converge, showing the existence of a simultaneous solution of Equation 2.

Lemma 4.1. *The sequence of perspective-specific similarity score produced by Algorithm 1 is non-decreasing and bounded by* $[0, 1]$, *i.e., for* $p \in \mathcal{P}, o_i, o_j \in \mathcal{O}, t \geq 0$.

$$1 \geq S_p^{(t+1)}(o_i, o_j) \geq S_p^{(t)}(o_i, o_j) \geq 0,$$

*Proof:* From the initialization step and update equations (5) (described in Algorithm 1), it is straightforward to see that:

$$S_p^{(1)}(o_i, o_j) \geq 0 = S_p^{(0)}(o_i, o_j), \forall p \in \mathcal{P}, o_i \neq o_j \in \mathcal{O}$$

$$\text{and } S_p^{(1)}(o_i, o_i) = 1 = S_p^{(0)}(o_i, o_i), \forall p \in \mathcal{P}, o_i \in \mathcal{O}.$$

That means Lemma 4.1 is true for $t = 0$. By induction, one can verify the statement in Lemma 4.1 still hold true for $\forall t \geq 1$.

Hence, each sequence $\{S_p^{(t)}(o_i, o_j)\}_{t \geq 0}$ is non-decreasing and bounded. By the Completeness Axiom of calculus, each sequence $\{S_p^{(t)}(o_i, o_j)\}_{t \geq 0}$ therefore converges to a limit $S_p(o_i, o_j) \in [0, 1]$. Moreover, $\{S_p(o_i, o_j)\}$ and $\{\text{sim}^{(*)}(p, p')\}$ are the solution for Eq. 2.

# 5 JOINT SOLUTION: MP-SIMRANK

We now describe our proposed *joint* solution MultiPerspective SimRank or MP-SimRank. The key idea is to induce bidirectional dependencies between the inter-perspective $\text{sim}(p, p')$ and the inter-object $S_p(o_i, o_j)$ similarities.

## 5.1 Inter-Perspective Similarity

The dependency from the inter-perspective $\text{sim}(p, p')$ to inter-object $\mathcal{S}_p$ is already encoded in Equation 2. To induce the dependency in the opposite direction, we need to define $\text{sim}(p, p')$ in terms of $\mathcal{S}_p$. While there could be many possible definitions, we propose the following definition in Equation 6, which, as we will show later would still preserve the convergence property.

$$\text{sim}(p, p') = 1 - \frac{\left\| \mathcal{S}_p - \mathcal{S}'_p \right\|_F}{n}, \tag{6}$$

The similarity between two perspectives $p, p'$ is inversely proportional to the Frobenius norm between $\mathcal{S}_p$ and $\mathcal{S}_{p'}$. If they are similar, i.e., $\frac{\left\| \mathcal{S}_p - \mathcal{S}'_p \right\|_F}{n}$ is close to 0 then $\text{sim}(p, p')$ is close to 1. Otherwise, if $\mathcal{S}_p$ and $\mathcal{S}_{p'}$ are extremely different, i.e., $\frac{\left\| \mathcal{S}_p - \mathcal{S}'_p \right\|_F}{n}$ is close to 1, then $\text{sim}(p, p')$ is close to 0.

## 5.2 Learning Algorithm

Algorithm 2 shows the joint-learning solution for Equation 2. We initialize the perspective-specific similarity score $S_p^{(0)}(*, *) \forall p \in \mathcal{P}$ as in Equation 4. For the similarity between perspectives, we initialize $\text{sim}^{(0)}(p, p') \forall p, p' \in \mathcal{P}$ as in Equation 8 below.

$$\text{sim}^{(0)}(p, p') = 0 \text{ if } p \neq p' \text{ and } 1 \text{ if } p = p' \tag{8}$$

**Algorithm 2** MP-SimRank

```
Require: Hypergraph $\mathcal{H}$ (defined as in Section 3)
1: Initialize $\mathcal{S}_p^{(0)} \leftarrow I_n, \forall p \in \mathcal{P}$
2: Initialize $\text{sim}^{(0)}(p, p') = 1$ if $p = p'$ and 0 if $p \neq p'$
3: while not converged do {
4:
       $$\mathcal{S}_p^{(t+1)}(o_i, o_j) = \frac{C}{|\mathcal{P}|} \sum_{p' \in \mathcal{P}} \text{sim}^{(t)}(p, p') \times \sum_{o_k \in N_{p'}(o_i)} \sum_{o_l \in N_{p'}(o_j)} \frac{\mathcal{S}_{p'}^{(t)}(o_k, o_l)}{|N_{p'}(o_i)||N_{p'}(o_j)|}, \tag{7}$$
       (for $1 \leq i \neq j \leq n$)
     and $S_p^{(t+1)}(o_i, o_i) = 1$ (for $1 \leq i \leq n$)
5:     $\text{sim}^{(t+1)}(p, p') = 1 - \frac{\left\| \mathcal{S}_p^{(t+1)} - \mathcal{S}_{p'}^{(t+1)} \right\|_F}{n}, \forall p, p' \in \mathcal{P}$
   }
6: Return $\{S_p^{converged}(o_i, o_j), \forall p \in \mathcal{P}, o_i, o_j \in \mathcal{O}\}$
7: and $\{\text{sim}^{converged}(p, p'), \forall p, p' \in \mathcal{P}\}$.
```

In contrast to the two-phase Algorithm 1, in this Algorithm 2 we iterate the computation of inter-object similarity in line 4 and that of inter-perspective similarity in line 5 until both converge.

## 5.3 Convergence Property

For MP-SimRank, we show that the computations for both types of similarities will converge to a fixed point.

Lemma 5.1. *The sequence of similarity between perspectives produced by Algorithm 2 is non-decreasing and bounded by* $[0, 1]$*, i.e., for* $t \geq 1$*,*

$$1 \geq \text{sim}^{(t+1)}(p, p') \geq \text{sim}^{(t)}(p, p') \geq 0, \forall p, p' \in \mathcal{P}. \tag{9}$$

*Proof:* Proving that, for $t \geq 0$:

$$\left\| \mathcal{S}_p^{(t+1)} - \mathcal{S}_{p'}^{(t+1)} \right\|_F \leq \left\| \mathcal{S}_p^{(t)} - \mathcal{S}_{p'}^{(t)} \right\|_F, \tag{10}$$

From Equation 3, $\forall p, p' \in \mathcal{P}$ we have:

$$\begin{aligned}
&\left\| \mathcal{S}_p^{(t+1)} - \mathcal{S}_{p'}^{(t+1)} \right\|_F \\
&= \left\| \frac{C}{|\mathcal{P}|} \sum_{p'' \in \mathcal{P}} \left( \text{sim}^{(t)}(p, p'') - \text{sim}^{(t)}(p', p'') \right) W_{p''}^T \cdot \mathcal{S}_{p''}^{(t)} \cdot W_{p''} \right\|_F \\
&\leq \frac{C}{|\mathcal{P}|} \sum_{p'' \in \mathcal{P}} \left| \text{sim}^{(t)}(p, p'') - \text{sim}^{(t)}(p', p'') \right| . \left\| W_{p''}^T \cdot \mathcal{S}_{p''}^{(t)} \cdot W_{p''} \right\|_F \\
&= \frac{C}{|\mathcal{P}|} \sum_{p'' \in \mathcal{P}} \left| \left\| \mathcal{S}_p^{(t)} - \mathcal{S}_{p''}^{(t)} \right\|_F - \left\| \mathcal{S}_{p'}^{(t)} - \mathcal{S}_{p''}^{(t)} \right\|_F \right| . \frac{\left\| W_{p''}^T \cdot \mathcal{S}_{p''}^{(t)} \cdot W_{p''} \right\|_F}{n} \\
&= \frac{C}{|\mathcal{P}|} \sum_{p'' \in \mathcal{P}} \left\| (\mathcal{S}_p^{(t)} - \mathcal{S}_{p''}^{(t)}) - (\mathcal{S}_{p'}^{(t)} - \mathcal{S}_{p''}^{(t)}) \right\|_F . \frac{\left\| W_{p''}^T \cdot \mathcal{S}_{p''}^{(t)} \cdot W_{p''} \right\|_F}{n} \\
&\leq \frac{C}{|\mathcal{P}|} \sum_{p'' \in \mathcal{P}} \left\| \mathcal{S}_p^{(t)} - \mathcal{S}_{p'}^{(t)} \right\|_F < \left\| \mathcal{S}_p^{(t)} - \mathcal{S}_{p'}^{(t)} \right\|_F \\
&\Rightarrow \text{sim}^{(t+1)}(p, p') \geq \text{sim}^{(t)}(p, p').
\end{aligned}$$

Since, $0 \leq \frac{\left\| \mathcal{S}_p^{(t)} - \mathcal{S}_{p'}^{(t)} \right\|_F}{n} \leq 1, \forall t \geq 1$ and $p, p' \in \mathcal{P}$, we also have $\text{sim}^{(t)}(p, p') \in [0, 1], \forall t \geq 0$ and $p, p' \in \mathcal{P}$. By the Completeness Axiom of calculus, $\text{sim}^{(t)}(p, p')$ converges to a limit $\text{sim}(p, p')$.

From Lemma 5.1 and by induction, we can prove that Lemma 4.1 still holds true for the perspective-specific sequences produced by Algorithm 2. That means $\{\mathcal{S}_p^{(t)}(o_i, o_j)\}_{t \geq 0}$ converges to a limit $\mathcal{S}_p(o_i, o_j) \in [0, 1]$ and $\{\text{sim}^{(t+1)}(p, p')\}_{t \geq 0}$ converges to a limit $\text{sim}(p, p')$. Moreover, $\mathcal{S}_p(o_i, o_j)$ and $\text{sim}(p, p')$ solve Equation 2.

# 6 EXPERIMENTS ON EFFECTIVENESS

Our experimental objectives are to study the comparative performance of the proposed graph-theoretic multiperspective approach against comparable baselines, and to investigate the role and effectiveness of inter-perspective similarities.

## 6.1 Experimental Settings

**Datasets.** For experiments, we seek publicly available datasets that could reflect the notion of multiperspectivity. We identify the following three datasets, whereby the first two model multiperspectivity due to different facets or attributes of objects, and the third models multiperspectivity due to different agents.

*Zoo*[1] contains 101 animals with 17 attributes (excluding name), e.g., *#legs*, *type* (mammals, birds, etc.). We treat attribute as perspective and animal as object, and model the varying similarity of animals according to attributes. We form a hyperedge $(p, o_i, o_j)$ if $o_i$ and $o_j$ have the same value for $p$. For example, one hyperedge is (*#legs*, *elephant*, *giraffe*), since elephant and giraffe have four legs.

*Congressional Voting Records (or HouseVote)*[2] contains 435 instances (congress members) and 16 attributes (votes). After excluding instances with missing values, we get a dataset with 232 instances. Considering each attribute as a perspective, we generate hypergraph in the same way as we do with *Zoo* dataset.

*Paris Attractions*[3] has 237 users organize 250 attractions in Paris into clusters. Each is a group of similar attractions from the perspective of a user. We induce hyperedges involving two attractions $i$ and $j$ that the user (perspective) puts into the same cluster.

The density ratio is measured by dividing the number of present hyperedges by the maximum number of hyperedges possible, i.e., $m * n^2$. Paris Attractions has the lowest density at 0.16%, as compared to 57.2% for Zoo and 52.3% for HouseVote.

**Task and Metrics.** We evaluate similarity methods as follows. In each dataset, a perspective is associated with a clustering of objects (based on attribute values or groupings). For each cluster, we sample 70% of objects for training, and keep the 30% hidden for testing. From the training set, we induce a hypergraph, and learn the similarity scores. At the prediction stage for each perspective, we measure the affinity between a hidden object and the clusters, and assigns the object to the highest-affinity cluster. Here, affinity is the average similarity (as measured by the comparative method) between the hidden object and the known objects in the cluster.

While presence of hyperedges indicate similarity, absences may not necessarily indicate dissimilarity (maybe missing values). Thus, we evaluate predictions via two recall-oriented metrics. We conduct stratified sampling to maintain the same ratio for each perspective and report the average results over ten train/test splits.

*Recall:* For a $p \in \mathcal{P}$, hiding an object from one of its clusters essentially creates hidden hyperedges in the test set involving the perspective, the hidden object, and other objects in the cluster. Correspondingly, at prediction stage, the assignment of a hidden object to the highest-affinity cluster "predicts" another set of hyperedges. Let $\mathcal{E}_p^{\text{hid}}$ denote the former, and $\mathcal{E}_p^{\text{pred}}$ the latter. *Recall* is the fraction of $\mathcal{E}_p^{\text{hid}}$ recovered by $\mathcal{E}_p^{\text{pred}}$, averaged across perspectives.

$$\text{Recall} = \frac{1}{m} \sum_{p \in \mathcal{P}} \frac{|\mathcal{E}_p^{\text{pred}} \cap \mathcal{E}_p^{\text{hid}}|}{|\mathcal{E}_p^{\text{hid}}|} \tag{11}$$

*PRES:* As the recall measure above relies on discrete assignments, we use a second metric that relies on rankings. For a cluster, we rank the candidate objects based on the affinity scores. We then evaluate the rank positions of the ground-truth hidden objects using *PRES* (Patent Retrieval Evaluation Score) [15], which had been designed for recall-oriented retrieval tasks. Equation 12 shows the formula for a cluster of a given perspective, where $n$ is the number of ground-truth objects hidden from this cluster, $r_i$ is the rank order of each ground-truth object in the output, and $N_{max}$ is the total number of candidates. To report the overall result, we average it across the clusters of a perspective, and then across perspectives.

$$\text{PRES} = 1 - \frac{\frac{\sum r_i}{n} - \frac{n+1}{2}}{N_{max}} \tag{12}$$

**Methods.** We compare the two methods[4] described in this paper: Pipelined-SimRank and MP-SimRank to several baselines. Since our work is related to SimRank, and the key contribution is to incorporate native multiperspectivity, our main baselines are variants of SimRank. For all the graph-theoretic methods, including ours, the damping factor $C$ is set to 0.8, as recommended in [11].

The first two are *uniperspective* SimRank-based methods. *Merged-SimRank* is obtained by taking the union of graphs due to different perspectives, and applying SimRank on the merged graph. *Average-SimRank* is obtained by running SimRank on each perspective's graph independently, and then averaging the SimRank scores to be used as a common inter-object score. Comparing to these uniperspective variants allows us to see the effect of multiperspectivity.

*Disjoint-SimRank* recognizes multiperspectivity, but assumes they can be obtained separately. For each perspective, we create a single graph to represent its own similarity viewpoint. We then run classic SimRank on each graph independently. In this mode, each perspective can only learn from its own graph, without collaborating with others. Comparing to this variant allows us to see the effect of inter-perspective collaboration that underpins both Pipelined-SimRank and MP-SimRank.

The final method *Personalized Collaborative Clustering* or *PCC* [22] is not graph-theoretic per se. Given our focus, strictly speaking it is not a baseline. However, it is included for completeness because it supports some notion of multiperspectivity, but relies on matrix factorization. We tune the parameters of PCC (learning rate, dimension of latent space) for its best performance.

## 6.2 Comparison to Baselines

We now discuss the experimental results, focusing on the similarity values among objects. Figure 3 shows the *Recall* of all comparative methods across the three datasets.

*Disjoint-SimRank* is consistently the weakest. Its *Recall* for *Zoo*, *HouseVote*, and *Paris Attractions* are 24.65%, 25.14%, and 0.65%. We attribute this to the lack of information within each perspective, since each only runs SimRank on its own graph.

*Merged-SimRank* achieves slightly better *Recall* than *Disjoint-SimRank* on three datasets: 26.0% for *Zoo*, 41.7% for *HouseVote*, and 3.4% for *Paris Attractions*. By pooling together all the perspectives, it learns the consensus view. *Average-SimRank* interestingly achieves the best *Recall* values among all the baselines: 50.7% for *Zoo*, 55.3% for *HouseVote*, and 3.7% for *Paris Attractions*. Perhaps it captures

[1] https://archive.ics.uci.edu/ml/datasets/Zoo

[2] https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records

[3] http://projects.yisongyue.com/collab_cluster/

[4] Implementation codes are available at https://github.com/PreferredAI/multiperspective_simrank

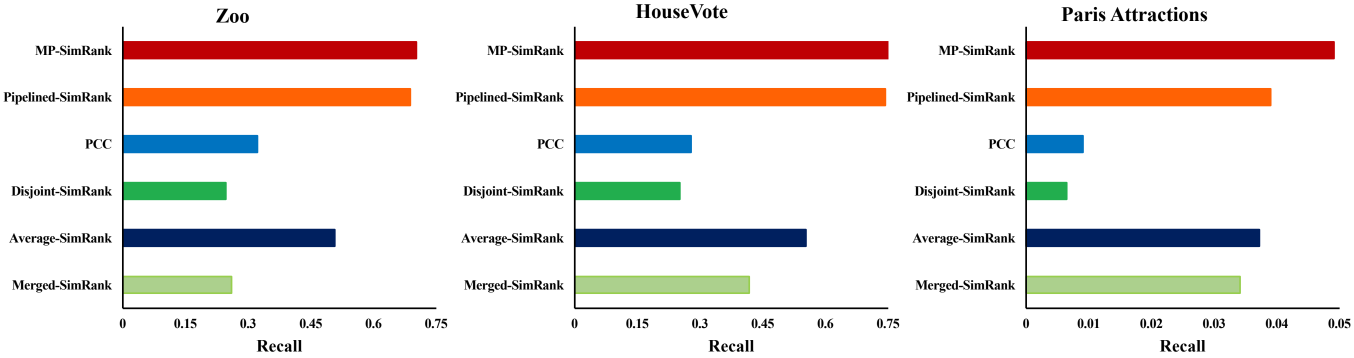

**Figure 3: Recall values of all models**

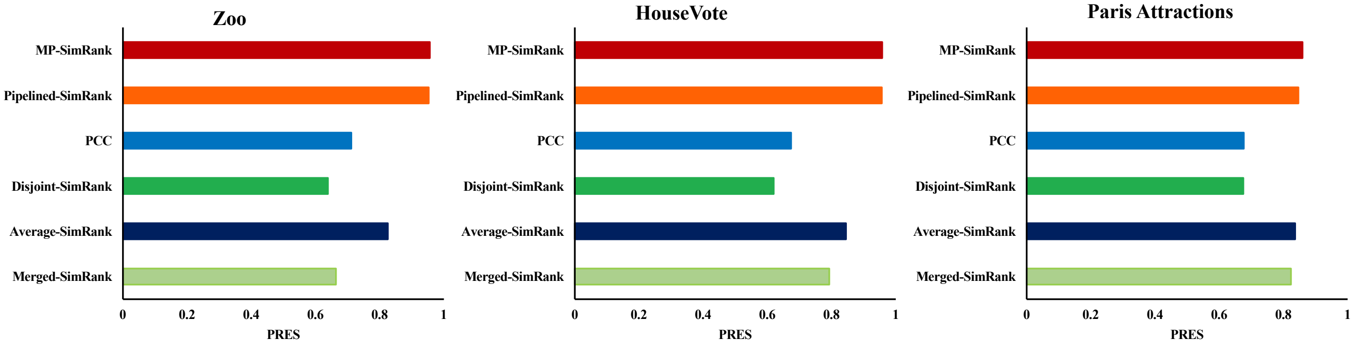

**Figure 4: PRES values of all models**

the dominant perspective, as this model would give higher similarity score for those pairs of objects that have been clustered as similar more frequently in the data than the score for other pairs.

The natively multiperspective models perform better. Pipelined-SimRank achieves *Recall* of 68.8% for *Zoo*, 74.3% for *HouseVote*, and 3.9% for *Paris Attractions*. MP-SimRank is even better, its *Recall* for the three datasets are 70.2%, 75.0%, and 4.9% respectively. This supports our intuition that by modeling multiple perspectives, we can capture nuances specific to some perspectives, and yet still allow collaboration among similar perspectives.

The results for non-graph theoretic *PCC* are middling and mixed, 32.2% for *Zoo*, 27.8% for *HouseVote*, and 0.9% for *Paris Attractions*. It is still better than *Disjoint-SimRank*, ostensibly due to the sharing across perspectives. However, it is not always better than *Merged-SimRank* and is generally worse than *Average-SimRank*.

Figure 4 presents the *PRES* values. The trends are consistent with the *Recall* values for all datasets, in terms of the relative performance of the comparative methods. Compared to *Recall*, the *PRES* values tend to be higher. Especially, the *PRES* of the two multiperspective models are close to 1. This indicates that while recalling all ground-truth objects may be challenging, those that we do recall tend to be ranked almost at the top of the candidates.

Comparing the three datasets in this paper, *Paris Attractions* is the most sparse, which explains why the *Recall* and *PRES* values of all models for this dataset are relatively lower than that of the two other datasets.

## 6.3 Inter-Perspective Similarities

As a byproduct of determining the similarities among objects, multiperspective models also produce the similarities among perspectives. We are interested in investigating the inter-perspective similarity $\mathrm{sim}(p, p')$, $\forall p, p' \in \mathcal{P}$ of the two models. Intuitively, for effective information sharing, two "similar" perspectives $p, p'$ should have higher $\mathrm{sim}(p, p')$ value than two "dissimilar" perspectives.

To attempt to understand how meaningful the $\mathrm{sim}(p, p')$ values are, we turn to the concept of Normalized Mutual Information or NMI [6]. In particular, for *Zoo* and *HouseVote*, each perspective corresponds to an attribute, whose values effectively define a clustering of objects. Supposing we see the full dataset, we can quantity how similar two attributes are, using NMI on the two clusterings over the same objects. For each $p \in \mathcal{P}$, we measure the Pearson correlation of its NMI scores and its inter-perspective similarities with other perspectives in $\mathcal{P}$. We do not include *Paris Attractions*, since each user only clusters a different subset of objects.

Table 2 and Table 3 show the Pearson correlation values of each perspective for Pipelined-SimRank and MP-SimRank respectively. Both achieve high correlation values between the NMI scores and the inter-perspective similarities for each perspective. That means both are able to reflect very well the underlying similarity between two perspectives. The joint learning MP-SimRank seems to better learn the similarity between two perspectives than the Pipelined-SimRank. This explains the improvement in the performance of MP-SimRank upon Pipelined-SimRank in the earlier experiment.

**Table 2: Correlation between NMI scores and inter-perspective similarities for *Zoo* (17 perspectives)**

| Perspective | MP-SimRank | Pipelined-SimRank |
|---|---|---|
| $p_1$ | 0.9531 | 0.6914 |
| $p_2$ | 0.8163 | 0.7702 |
| $p_3$ | 0.9576 | 0.6430 |
| $p_3$ | 0.9451 | 0.6051 |
| $p_5$ | 0.8954 | 0.8514 |
| $p_6$ | 0.9669 | 0.8670 |
| $p_7$ | 0.9952 | 0.9891 |
| $p_8$ | 0.9191 | 0.7410 |
| $p_9$ | 0.7920 | 0.7612 |
| $p_{10}$ | 0.8434 | 0.8208 |
| $p_{11}$ | 0.8574 | 0.9832 |
| $p_{12}$ | 0.8445 | 0.8391 |
| $p_{13}$ | 0.9049 | 0.7840 |
| $p_{14}$ | 0.9229 | 0.8490 |
| $p_{15}$ | 0.9139 | 0.9932 |
| $p_{16}$ | 0.9846 | 0.9400 |
| $p_{17}$ | 0.8492 | 0.6174 |

## 6.4 Illustrative Case Study

To gain an intuition of how multiperspectivity plays a part in the similarity measurement, here we include a small case study. For this example, we use the *Paris Attractions* dataset to showcase the role of multiperspective similarity measure.

Table 4 shows the clustering data of four users in the dataset, each represented by user id. Each cluster is separated from another by the symbol ||. Of particular interest to us are objects with id: 30, 50, 62, 76, and 88 (in bold). We can observe that users may cluster objects similarly or differently from one another. For example, both U53 and U86 place objects 30, 50, 62, and 88 in the same cluster. On the other hand, U94 places object 62 and 88 in the same cluster, but places object 50 in a different cluster. U168, however, places three objects: 30, 76, and 88 in three different clusters.

We apply the MP-SimRank on the full *Paris Attractions* dataset and investigate the inter-perspective similarities between the four mentioned users. In Figure 5, each big circle represents the clustering data of each user. Clusters are wrapped inside inner circles. The values on the dashed lines represent the Frobenius distance between perspectives (users). We observe that the distance between U53 and U86 is smaller than those between U53 and either U94, U86 or U94. This is expected since U53 and U86 have more similar perspectives. The inter-perspective distances reflect that U53 and U86 are more similar to U94 than to U168. This is reasonable, since U53, U86, and U94 place object 62 and object 88 in the same cluster.

## 7 DISCUSSION ON EFFICIENCY

In this section, we discuss the theoretical complexity and practical efficiency of the SimRank-based methods.

**Table 3: Correlation between NMI scores and inter-perspective similarities for *HouseVote* (16 perspectives)**

| Perspective | MP-SimRank | Pipelined-SimRank |
|---|---|---|
| $p_1$ | 0.9963 | 0.9763 |
| $p_2$ | 0.9999 | 0.9990 |
| $p_3$ | 0.9790 | 0.8295 |
| $p_4$ | 0.9764 | 0.7586 |
| $p_5$ | 0.9733 | 0.6552 |
| $p_6$ | 0.9789 | 0.8715 |
| $p_7$ | 0.9811 | 0.8732 |
| $p_8$ | 0.9749 | 0.7106 |
| $p_9$ | 0.9783 | 0.7335 |
| $p_{10}$ | 1.0000 | 0.9998 |
| $p_{11}$ | 0.9992 | 0.9975 |
| $p_{12}$ | 0.9763 | 0.7793 |
| $p_{13}$ | 0.9865 | 0.8866 |
| $p_{14}$ | 0.9641 | 0.8197 |
| $p_{15}$ | 0.9883 | 0.9335 |
| $p_{16}$ | 0.9713 | 0.9376 |

**Table 4: Cluster data of four users from *Paris Attractions***

| ID | Clustering Data |
|---|---|
| U53 | 14 21\|\|**30** 40 **50 62 76 88**\|\|17 156\|\|78 79 106 126 201 232 247 |
| U86 | 72 78 96 109 164 208\|\|2 **30 50 62 88** 178 224\|\|79 84 207 |
| U94 | 7 91 115 140 159 167 248\|\|34 49 **62** 73 79 **88** 142 151 238\|\|**50** 90 154 |
| U168 | 40 48 73 84 85 **88** 89 90 117 154 166 171\|\|45 51 61 **76** 111 116 126 133 146 200\|\|28 **30** 52 60 78 86 100 128 132 195\|\|21 |

**Table 5: Complexity analysis (per iteration) of all SimRank-based methods**

| **Methods** | **Storage** | **Time** |
|---|---|---|
| Merged-SimRank | $O\left(n^2\right)$ | $O\left(n^2 d_{\max}\right)$ |
| Average-SimRank | $O\left(n^2\right)$ | $O\left(mn^2 d_{\max}\right)$ |
| Disjoint-SimRank | $O\left(mn^2\right)$ | $O\left(mn^2 d_{\max}\right)$ |
| Pipelined-SimRank | $O\left(m^2 + n^4 + mn^2\right)$ | $O\left((m^2 + n^4 + mn^2) d_{\mathrm{bi}} + m^2 n^2 d_{\max}\right)$ |
| MP-SimRank | $O\left(m^2 + mn^2\right)$ | $O\left(m^2 n^2 d_{\max}\right)$ |

## 7.1 Complexity Analysis

First, we look into the theoretical storage and time complexities, which are summarized in Table 5. For the uniperspective *Merged-SimRank*, its complexities are the same as the original SimRank's, which is square to the number of object pairs, i.e., $n^2$. Suppose for a given perspective, $d_p$ is the average product of neighbor counts, i.e., $|N_p(o_i)|.|N_p(o_j)|$ across object pairs $o_i, o_j \in O$. Then $d_{\max}$ is the maximum such average among all perspectives $\forall p \in \mathcal{P}$.

For the methods that require computation for each perspective, *Average-SimRank* and *Disjoint-SimRank*, it is reasonable that the complexity will also scale with $m$ the number of perspectives. However, both of these act independently for each perspective.

For the natively multiperspective methods, there is a need to compute the inter-perspective similarities. For Pipelined-SimRank, this is done by inducing a bipartite graph of perspectives-by-object

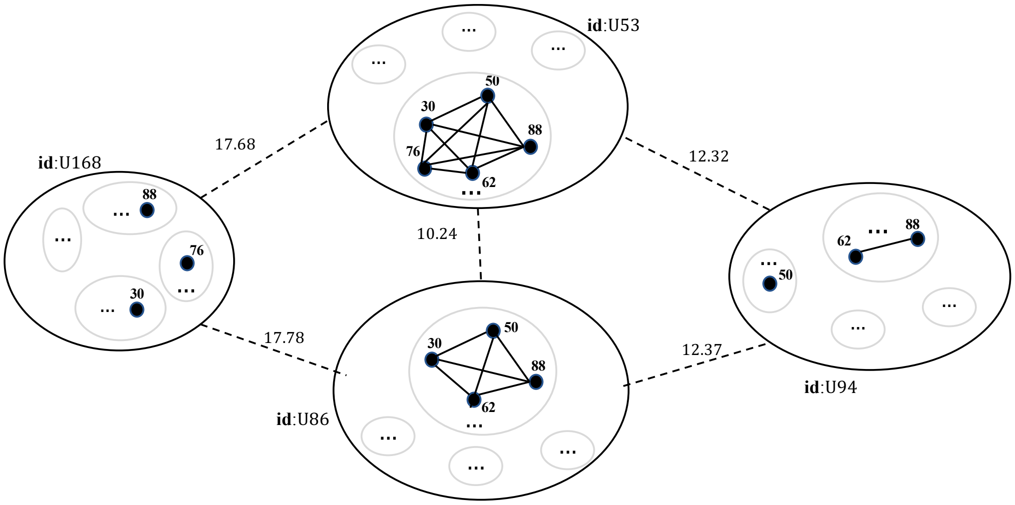

**Figure 5: Illustrative example of multiperspective similarity from *Paris Attractions* dataset.**

pairs. Therefore, in addition to the perspective-specific inter-object similarities ($mn^2$), we store and compute the inter-perspective similarities ($m^2$) and the similarity between any two object pairs ($n^4$). $d_{\text{bi}}$ is the average product of neighbor counts in the bipartite $\mathcal{B}$ (Section 4). In terms of time, we further need to consider the computation of perspective-specific inter-object similarities, iterated over all perspectives, i.e., $m^2n^2d_{max}$. This is computationally intensive, which motivates the development of MP-SimRank.

For MP-SimRank, the joint computation of both inter-perspective and inter-object similarities avoids the instantiation of the bipartite graph, dropping the $n^4$ term from the complexities. This dramatically improves the running time of MP-SimRank.

We are also interested in how many iterations are generally required for convergence in practice. The *convergence rate* of the algorithm is defined as follows:

$$D_t = \frac{1}{m} \sum_{p \in \mathcal{P}} \frac{\left\| \mathcal{S}_p^{(t+1)} - \mathcal{S}_p^{(t)} \right\|_F}{n},$$

as the algorithm converges, the value of $D_t$ should approach 0 as $t$ goes to infinity. Overall, both models converge after reasonably few iterations (less than 5 iterations for *Zoo* and *HouseVote*, and less than 8 iterations for *Paris Attractions*).

### 7.2 Heuristic for More Efficient MP-SimRank

Since our main focus is multiperspectivity, one possible avenue to further improve efficiency is to reduce the number of perspectives, by grouping similar perspectives into a cluster with one representative perspective. We test the feasibility of this concept here.

Algorithm 3 describes clusteredMP-SimRank that adopts the idea of clustering perspectives. We first run *Disjoint-SimRank* on each graph and produce $S_p^{\text{disjoint}}, \forall p \in \mathcal{P}$ (Step 1) with computational cost of $O(mn^2 d_{\max})$. Next, we compute the Frobenious distance between all perspectives, cluster them using the *k-medoids* algorithm ($k \leq m$ is given), and merge graphs of perspectives in the same cluster together (Step 2 and 3). A medoid here is defined as the perspective with the smallest average distance to all others in the same cluster. These two steps require a computational cost of $O(m^2 + km)$. We then run MP-SimRank on the new hypergraph $\mathcal{H}_c$, yielding the *cluster-specific inter-object* similarity $S_c$ (Step 4) with the cost of $O(k^2n^2d_{\max})$. Finally, perspectives of the same cluster share the same cluster-specific inter-object scores. The total computational cost is $O(mn^2d_{\max} + m^2 + km + k^2n^2d_{\max})$, less complex than the cost of MP-SimRank, i.e., $O\left(m^2n^2d_{\max}\right)$.

Figure 6 shows the performance of clusteredMP-SimRank and its running time as we vary the number of clusters $k$. The horizontal axis shows the required running time in second and the vertical axis shows the peformances in terms of *Recall* (in blue) and *PRES* (in red). We observe that by choosing a small number of clusters, we can improve significantly the speed of the learning. As $k$ increases, clusteredMP-SimRank approaches the performance of MP-SimRank (when $k = m$). With a reasonable choice of $k$, we can speed up the learning process while still obtaining acceptable level of performances from the learnt similarity scores.

## 8 CONCLUSION

In certain real world applications, there is a need for expressing diverse perspectives of similarity. We propose a multiperspective graph-based framework for learning similarity from data. The proposed framework relies on a unified hypergraph representation of object-to-object relationships. The key is to learn not only the similarity between two objects for each perspective, but also the similarities across perspectives so as to allow information sharing across perspectives. We present two models, Pipelined-SimRank and MP-SimRank, and provide their proof of convergence. Experiments on publicly available datasets show that multiperspective similarity models outperform baseline models that either ignores multiplicity of perspectives or treats each perspective separately. As future work, we will investigate strategies for improving the efficiency of the proposed framework, towards creating potential applications involving large-scale networks.

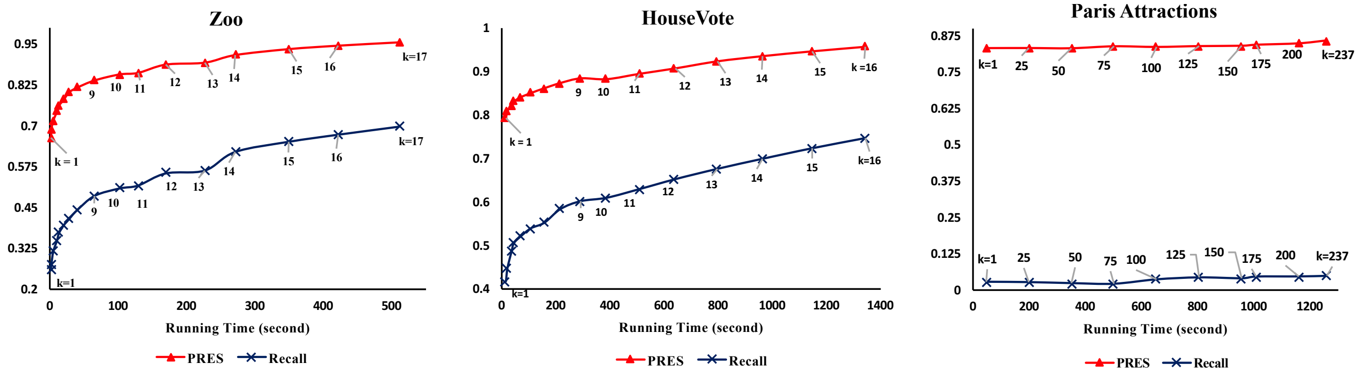

**Figure 6: PRES, Recall, and running time of CLUSTEREDMP-SIMRANK with different number of clusters $k$.**

**Algorithm 3** CLUSTEREDMP-SIMRANK

```
Require: Hypergraph $\mathcal{H}$ and number of clusters $k$
/*– Step 1: run disjoint-simrank on each perspective graph – */
$S_p^{\text{distjoint}} \leftarrow \text{Disjoint} - \text{SimRank}(G_p), \forall p \in \mathcal{P}.$

/*- Step 2: compute Frobenius distances between perspectives - */
$\mathcal{F} = [F(p,p')]_{p,p' \in \mathcal{P}}$, where
        $F(p,p') = \left\| S_p^{\text{distjoint}} - S_{p'}^{\text{distjoint}} \right\|_F$

/*– Step 3: cluster perspectives and merge graphs – */
$C \leftarrow \text{K} - \text{Medoids}(\mathcal{F}, k);\ \mathcal{H}_c \leftarrow \text{merge} - \text{graph}(\mathcal{H}, C)$

/*– Step 4: run MP-SIMRANK on the new hypergraph $\mathcal{H}_c$ – */
$\{S_c\}_{c \in C} \leftarrow$ MP-SIMRANK$(\mathcal{H}_c)$

/*– Step 5: assign each perspective the inter-object similarity –
– of the cluster it belongs to–*/
$S_p \leftarrow S_c, \forall p \in \mathcal{P}, c \in C$, and $p \in c$

Return $\{S_p, \forall p \in \mathcal{P}\}$
```

## ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its NRF Fellowship Programme (Award No. NRF-NRFF2016-07).

## REFERENCES

[1] Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007), 5.

[2] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction* 12, 4 (2002), 331–370.

[3] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. 2010. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research* 11, Mar (2010), 1109–1135.

[4] Beate Dorow, Florian Laws, Lukas Michelbacher, Christian Scheible, and Jason Utt. 2009. A graph-theoretic algorithm for automatic extension of translation lexicons. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*. Association for Computational Linguistics, 91–95.

[5] Robert C Edgar. 2004. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC bioinformatics* 5, 1 (2004), 113.

[6] Pablo A Estévez, Michel Tesmer, Claudio A Perez, and Jacek M Zurada. 2009. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks* 20, 2 (2009), 189–201.

[7] Floris Geerts, Heikki Mannila, and Evimaria Terzi. 2004. Relational link-based ranking. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 552–563.

[8] Guoming He, Haijun Feng, Cuiping Li, and Hong Chen. 2010. Parallel SimRank computation on large graphs with iterative aggregation. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 543–552.

[9] Anna Huang. 2008. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*. 49–56.

[10] Thad Hughes and Daniel Ramage. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.

[11] Glen Jeh and Jennifer Widom. 2002. SimRank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 538–543.

[12] Mitsuru Kusumoto, Takanori Maehara, and Ken-ichi Kawarabayashi. 2014. Scalable similarity search for SimRank. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 325–336.

[13] Cuiping Li, Jiawei Han, Guoming He, Xin Jin, Yizhou Sun, Yintao Yu, and Tianyi Wu. 2010. Fast computation of simrank for static and dynamic information networks. In *Proceedings of the 13th International Conference on Extending Database Technology*. ACM, 465–476.

[14] Pei Li, Hongyan Liu, Jeffrey Xu Yu, Jun He, and Xiaoyong Du. 2010. Fast single-pair simrank computation. In *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM, 571–582.

[15] Walid Magdy and Gareth JF Jones. 2010. PRES: a score metric for evaluating recall-oriented information retrieval applications. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 611–618.

[16] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. 2008. *Introduction to information retrieval*. Vol. 1. Cambridge university press Cambridge.

[17] Sascha Rothe and Hinrich Schütze. 2014. Cosimrank: A flexible & efficient graph-theoretic similarity measure. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1392–1402.

[18] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment* 4, 11 (2011), 992–1003.

[19] Hanghang Tong, Christos Faloutsos, and Jia Yu Pan. 2006. Fast random walk with restart and its applications. In *6th International Conference on Data Mining, ICDM 2006*.

[20] Ramya Korlakai Vinayak and Babak Hassibi. 2016. Crowdsourced clustering: Querying edges vs triangles. In *Advances in Neural Information Processing Systems*. 1316–1324.

[21] Jinfeng Yi, Rong Jin, Shaili Jain, Tianbao Yang, and Anil K Jain. 2012. Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In *Advances in neural information processing systems*. 1772–1780.

[22] Yisong Yue, Chong Wang, Khalid El-Arini, and Carlos Guestrin. 2014. Personalized collaborative clustering. In *WWW*. 75–84.