Inter-Profile Similarity (IPS): A Method for Semantic Analysis of Online Social Networks

Matt Spear, Xiaoming Lu, Norman S. Matloff, and S. Felix Wu

University of California, Davis
batman900@gmail.com, lu@ucdavis.edu, matloff@cs.ucdavis.edu,
wu@cs.ucdavis.edu

Abstract. Online Social Networks (OSN) are experiencing an explosive growth rate and are becoming an increasingly important part of people's lives. There is an increasing desire to aid online users in identifying potential friends, interesting groups, and compelling products to users. These networks have offered researchers almost total access to large corpora of data. An interesting goal in utilizing this data is to analyze user profiles and identify how similar subsets of users are. The current techniques for comparing users are limited as they require common terms to be shared by users. We present a simple and novel extension to a word-comparison algorithm [6], entitled Inter-Profile Similarity (IPS), which allows comparison of short text phrases *even if they share no common terms.* The output of IPS is simply a scalar value in $[0, 1]$, with 1 denoting complete similarity and 0 the opposite. Therefore it is easy to understand and can provide a total ordering of users. We, first, evaluated the effectiveness of IPS with a user-study, and then applied it to datasets from Facebook and Orkut verifying and extending earlier results. We show that IPS yields both a larger range for the similarity value and obtains a higher value than intersection-based mechanisms. Both IPS and the output from the analysis of the two OSN should help to predict and classify social links, make recommendations, and annotate friends relations for social network analysis.

Keywords: Online Social Network, Semantic Analysis, Profile Similarity, Natural Language Processing.

1 Introduction

Online Social Networks (OSN) are experiencing an explosive growth rate and are becoming an increasingly important part of people's lives. There is an increasing desire to aid users in identifying potential friends, interesting groups, and compelling products. Furthermore, these networks have offered researchers almost total access to large corpora of data. An interesting goal in utilizing this data is to analyze user profiles and identify how similar subsets of users are. The current techniques for comparing users are limited as they require common terms to be shared.

J. Zhou (Ed.): Complex 2009, Part I, LNICST 4, pp. 320–333, 2009.

In this paper, we devise a simple, novel method which extends [6] to compare short-text snippets using Natural Language Processing (NLP). Inter-Profile Similarity (IPS) provides an application-independent mechanism to give a total ordering according to the similarity value. This algorithm requires no extra work from the user and provides a comparison of profiles[1], which OSN already have user's input. To our knowledge, this paper is the first to provide an NLP style similarity analysis on social network graph.

The benefits of using NLP to compare the similarity of users are twofold: (1) different words that possess the same meaning will be correctly identified, and (2) the number of terms in common decreases as the size of the vocabulary increases. As an example in [8] the authors show a difference between immediate friends and a random non-friend, but looking at the mean similarity, there is not a large difference between the two groups. Using an algorithm which utilizes NLP, e.g. IPS, should aid researchers in quickly evaluating the similarity of users despite there existing a large vocabulary set. We show that IPS yields both a larger range for the similarity values and obtains higher values than the intersection-based approaches.

IPS lends itself to many practical applications in OSN e.g.: (1) product recommendations to users based on their profiles, (2) personalizing the ordering of search results based on their profiles, and (3) building communities by recommending groups to users who based on their profiles. Sites such as Netflix and Amazon recommend items that they believe the user would enjoy given their prior history, but these algorithms are done in a closed fashion and are not directly compatible. Secondly, when executing searches for potential friends one is generally interested more in people similar to them. Therefore, it is preferable to provide a total ordering utilizing the user's similarity to the people in the result-set. Some current OSN already do this, but in a rather simplistic manner, such as utilizing the group and affiliation information of the users. Finally, being able to quickly find communities with people whom share interests with the user is an impoerant part of OSN.

The core of all of above problems is identifying "similar" users, whether it be due to their history of movie ratings, history of items bought, or activities/interests in common. IPS provides a common front-end to identify similar users utilizing their profiles which already exist in OSN. Once this is accomplished, all of the above problems can be solved.

We evaluated the effectiveness of IPS with a user-study and found that IPS's ordering is closely aligned to what users expect. We also applied IPS to Facebook and Orkut and were able to verify and extend earlier results. We showed, in agreement with [9], that there is a trend of decreasing similarity with increasing distance, and, further, showed that there was no significant between same gender and opposite gender neighbor similarities. Both IPS and the analysis of the two OSN should help to predict and classify social links, make recommendations, annotate friends relations for social network analysis.

[1] Where a profile is considered to be a set of short phrases, e.g. "play basketball", "read book".

The remainder of the paper is organized as follows: in Section 2 we describe the related work, in Section 3 we formalize the limitations of using the current approaches and introduce the IPS algorithm, then in Section 4 we provide the details and results of a user-study of IPS, next in Section 5 we apply IPS to analyze datasets from Facebook and Orkut, finally in Section 6 we note future work and present our conclusions.

2 Related Work

OSN have grown very rapidly and have become a popular mechanism for discovering (and rediscovering) friends and relationships [12]. For example, in its January 2004 debut, Orkut had over 50,000 communities, but by May, 2005 had over 1,500,000. Many OSNs now have members numbering in the tens-of-millions, and users are increasingly using them as a recommendation and community building application. Efficient and well-ordered search results are essential to the discovery of new friends in OSN.

There are a number of existing algorithms that provide the similarity of profiles (see [11] for a good survey), but there is no existing approach that utilizes NLP. Current similarity measure used in profile comparisons, documents clustering and search result presentations tend to be based on word intersection [2,3,4]. In [8], the similarity amongst neighboring nodes were compared to values of a node picked at a random in the network. It is interesting to note that the maximum intersection of any two users was below 0.16, which should intuitively imply that users are not related although they might be more related when comparing the semantic meaning of their image tags. User profiles tend to be short snippets and word-intersection algorithms might not achieve satisfactory results due to lack of common words [5]. The approach in [5] treats each snippet as queries, obtains documents related their snippets, and then compare these returned documents using a word/phrase intersection similarity measure such as cosine [3].

IPS extends [6], which provides a word-comparison algorithm using context-vectors in WordNet [10]. WordNet is a lexical dictionary that also has a hierarchy amongst the words. Words which have different meanings (or parts of speech) are designated by *senses*. The IPS similarity measure uses NLP, which allow a more flexible measure of similarity.

3 IPS Algorithm

In this section, first we analytically show why the use of a simpler algorithm which utilizes intersection will not provide good results. Then we present the IPS algorithm and describe its complexity, benefits and limitations.

3.1 Why Not Intersection?

The most obvious problem with intersection is that it requires both users to choose the same keyword. To help illustrate the issue, we will walk through a simple example to show the expected number of items in the intersection of two equally sized sets. We assume the keywords are chosen from a universe U of cardinality N, and are numbered $\{1, \ldots, N\}$ in order of decreasing popularity. We assume that the keyword popularity follows a Zipf distribution[2], i.e. the probability of drawing the k^{th} most popular item is $\Pr(k) \overset{\text{def}}{=} Ck^{-1}$, where C is a normalizing constant. We also assume keywords are chosen independently; this is not generally the case, e.g. if "JPOP" is drawn it is more likely "Ai Otsuka" will be drawn even if it is not a popular keyword overall, but this assumption makes the analysis much easier. Furthermore, we assume that the user profiles are all the same size[3].

We now consider the expected number of items in common amongst two sets of items picked from $\mathbf{U}$. Our derivation uses Wallenius' Multivariate Noncentral Hypergeometric Distribution [1], as each keyword has different weight and keywords from $\mathbf{A}$ and $\mathbf{B}$ are picked without replacement from $\mathbf{U}$:

$$\begin{aligned} \mathrm{E}[|\mathbf{A} \cap \mathbf{B}|] &= \sum_{k=1}^{N} 1 \cdot \Pr(k \in A \wedge k \in B) = \sum_{k=1}^{N} [\Pr(k \in A)]^2 \\ &= \sum_{k=1}^{N} \underbrace{\left[\sum_{\mathbf{x} \in \chi} \underbrace{\int_0^1 \prod_{j=1}^{N} \left(1 - t^{w_j/D}\right)^{x_j} dt}_{\Pr(k \in A|\mathbf{x}),\ \text{from WMNHD}} \right]}_{\Pr(k \in A)}^2 \end{aligned} \qquad (1)$$

where χ is the set of vectors of length N such that $x_k = 1$ and $\sum_j x_j = |A|$, w_j is the weight of the j^{th} item in $\mathbf{U}$ and $D = \mathbf{w} \cdot (1 - \mathbf{x}^i)$. Note that for any fixed $|A|$[4]:

$$\lim_{N \to \infty} \mathrm{E}[|\mathbf{A} \cap \mathbf{B}|] = 0$$

In Figure 1, we simulated $\mathrm{E}[|\mathbf{A} \cap \mathbf{B}|]$ where we drew $|A|$ items from the N (number of keys in the universe) possible for various $|A|$ and $|N|$. The simulation shows that the intersection of $\mathrm{E}[|\mathbf{A} \cap \mathbf{B}|]$ is very low with increasing N and decreasing $|A|$.

3.2 IPS

The entirety of the IPS algorithm is shown in Algorithm (1). The main component of IPS is the ProfileSimilarity on line 1. This function takes two profiles A and B (each consisting of a set of phrases) and outputs a similarity value between 0 and 1. The phrases consist of a small set of words and, in general,

[2] This assumption is backed up by many studies on Peer-to-Peer (P2P) and our own studies into Facebook and Orkut.

[3] It is trivial to extend the probabilities to cover the case where they are not.

[4] This approaches 0 at a rate of $\log(N)$, so one could *in theory* multiply by $\log(N)$ to achieve a stable value, although calculating N would be extremely cumbersome.

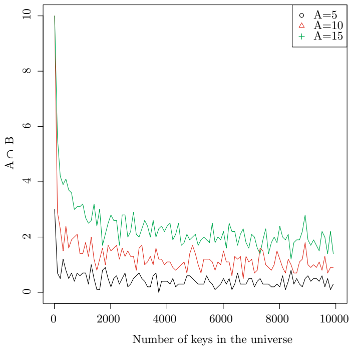


Fig. 1. Simulated E[$|\mathbf{A} \cap \mathbf{B}|$] for various $|A|$ and varying N

Algorithm 1. IPS

```
 1: Function ProfileSimilarity($\mathbf{A}$, $\mathbf{B}$)
 2:   $(s, c) \leftarrow (0, 0)$
 3:   ForEach $a \in \mathbf{A}$ Do
 4:     $(t, b) \leftarrow \max_{b \in \mathbf{B}} PhraseSimilarity(a, b)$ ; $(s, c) \leftarrow (s + t, c + 1)$; $\mathbf{B} \leftarrow \mathbf{B} - b$
 5:     Break If $|\mathbf{B}| = 0$
 6:   EndFor
 7:   Return $\frac{s}{c}$
 8: End
 9: Function PhraseSimilarity($\mathbf{A}, \mathbf{B}$)
10:   /* $a, b$ from line 4 were split on spaces */
11:   If $|A| = 1$ AND $|B| = 1$ Then
12:     Return SIMILARITY($A_0, B_0$)
13:   EndIf
14:   $(\mathbf{adj}_A, \mathbf{noun}_A, \mathbf{verb}_A) \leftarrow WSD(\mathbf{A})$ ; $(\mathbf{adj}_B, \mathbf{noun}_B, \mathbf{verb}_B) \leftarrow WSD(\mathbf{B})$; $\mathbf{S} \leftarrow \emptyset$
15:   ForEach $T \in \{adj, noun, verb\}$ Do
16:     $(s, c) \leftarrow (0, 0)$
17:     ForEach $a \in \mathbf{T}_A$ Do
18:       $(t, b) \leftarrow \max_{b \in \mathbf{T}_B} SIMILARITY(a, b)$; $(s, c) \leftarrow (s+t, c+1)$; $\mathbf{T}_B \leftarrow \mathbf{T}_B - b$
19:       Break If $|\mathbf{T}_B| = 0$
20:     EndFor
21:     $\mathbf{S}_T \leftarrow \frac{s}{c}$
22:   EndFor
23:   $(\mathbf{A}, \mathbf{B}) \leftarrow \bigcup_{T \in \{adj, noun, verb\}} (\mathbf{T}_A, \mathbf{T}_B)$; $(s, c) \leftarrow (0, 0)$
24:   ForEach $a \in \mathbf{A}$ Do
25:     $(t, b) \leftarrow \max_{b \in \mathbf{B}} SIMILARITY(a, b)$; $(s, c) \leftarrow (s + t, c + 1)$; $\mathbf{B} \leftarrow \mathbf{B} - b$
26:     Break If $|\mathbf{B}| = 0$
27:   EndFor
28:   $\mathbf{S}_{lo} \leftarrow \frac{s}{c}$
29:   Return weighted_avg($\mathbf{S}$)
30: End
```

are not full sentences; instead they are generally short-text snippets describing some activity or interest. In simplest terms, ProfileSimilarity repeatedly finds the maximally similar items from A and B and removes them until either A

or B is empty; then returns the average similarity value. This function utilizes PhraseSimilarity to compare phrases.

In PhraseSimilarity (line 9), the short-text snippets A and B need to be compared. This requires two functions from [6]: (1) WSD, and (2) SIMILARITY. Word Sense Disambiguation (WSD) is a common problem encountered in NLP—it consists of taking a sentence and identifying the sense, i.e. the part of speech (e.g. noun, adjective, verb) and the meaning, of each word. The second operation SIMILARITY provides the similarity of two words given their senses. We do not go into the details of the algorithm and refer the reader to [6] for a detailed explanation.

Now that the two external functions have been described we detail the evaluation of the phrase-similarity. To begin with, each phrase is broken into its sets for each of its parts of speech (line 14). Then, for each part of speech, the maximally similar words from A and B are found and removed until there are no more words in the part of speech. Once all parts of speech have been compared any remaining words are compared without regard to the part of speech; this is done for two reasons: (1) WSD is not a perfect algorithm and will sometimes incorrectly identify a word, and (2) some phrases do not share any parts of speech (e.g. “eat” and “JPOP”), but it is still desirable to know their similarity. Once all the parts of speech and leftover words have been compared a weighted average is returned wherein more weight is placed on nouns than verbs.

3.3 Benefits and Limitation

In this section we discuss some of the benefits and current limitations of the IPS system.

The benefits are:

- **Identify similar concepts despite being expressed with different words.** Previous intersection-based methods do not scale due to the enormity of most languages. Employing IPS overcomes this fundamental limitation.
- **Provides a total ordering over any set of users with regard to a querier.** This is useful in many applications from data mining to recommendation systems, and IPS provides a simple common interface for doing so.
- **Handles phrases of varying length by ignoring words that do not match.** Phrases of differing lengths are fairly common, and IPS handles them by comparing the most similar words first and ignore the rest. For example: let A = “play basketball” and B = “basketball”, then IPS will report these users with a high similarity value.

The limitations are:

- **Ignores negation.** Phrases or sentences with negated weight such as “do not like basketball” are ignored to avoid confusion to algorithm computation.

- **The left-over words for phrases of varying length may be important.** To handle sentences of varying lengths IPS simply ignores the leftover words, this may not be the most appropriate course of action.

All of the limitations are currently future work and should be addressed in later iterations of IPS.

4 IPS User Study

To evaluate the effectiveness of IPS, we conducted a user-study[5] While IPS outputs a number between 0 and 1, it is very difficult to evaluate the "correctness" of this number, e.g. it is hard for a user to state that "basketball" and "volleyball" are 0.75 similar. To reduce the user's burden, we asked the users to order a set of phrases, and then compared this ordering with IPS' ordering.

4.1 User-Study Description

The user-study consisted of three sections: (1) word similarity, (2) phrase similarity, and (3) profile similarity. Each section presented the user with a base word, phrase, or profile and then three other words, phrases, or profiles to order with respect to the base word, phrase, or profile. The user-study was implemented as an online interface using JavaScript to allow the users to visually order the words, phrases, or profiles.

The words, phrases, or profiles are in bold and the remaining words, phrases, and profiles are ordered according to IPS' value (shown in parenthesis). In the actual user study, these values were hidden from the user and were presented in a random order.

An example of the word similarity questions asked in the questionnaire is:

basketball: (1) volleyball (0.99), (2) soccer (0.61), (3) California (0.35)
The number in parenthesis is determined by IPS and not presented to the user.

An example of the phrase similarity questions asked is:
play soccer: (1) play football (0.82), (2) watch football (0.52), (3) water garden (0.27).

An example of the profile similarity questions asked is:
educated partner, read book, has patience, stable job, good salary, play sports, enjoy travel, enjoy movie: (1) educated and well read, stable job and finance, nice salary, play sports,watch movies, has patience (0.66) (2) read book, watch movies, go to church, good job, hang out with friends, good dresser (0.58); (3) party and drink, listen to music, dining at fancy restaurant, shopping online, go to concert (0.37);

4.2 Analysis

We had 30 users complete the survey. Overall, the mode of the user responses always matched IPS' ordering. Also, 98% of the users were within one transposition from IPS' ordering, and 79% of the user's orderings agreed with IPS.

[5] Available at:
`http://wwwcsif.cs.ucdavis.edu/%7Elu/similarity/examples/test.html`

To evaluate the each individual questions, we numbered the possible responses according to a gray-code wherein IPS' ordering was in the center. A gray-code has the property that neighboring codes contain only a single transposition, which allows us to consider how close the response was to IPS. For example, if IPS stated the order was *bac* then we would use the ordering $\{cab, cba, bca, \mathbf{bac}, abc, acb\}$ so IPS' output appears at position 4 and each item has only a single transposition between each of its neighbors.

We now present the confidence intervals for each question in each section using a z value chosen for 0.01 confidence, and provide some statistical observations.

For the word similarity, 68% of the user's orderings completely agreed with IPS' ordering, and 97% of the users made at most one transposition from IPS' ordering. The aggregate of word similarity section yielded $\mu_{\text{word}} \in [3.98, 4.28]$, where the correcting ordering was numbered 4.

For the phrase similarity 87% of the user's orderings completely agreed with IPS' ordering, and 99% of the users made at most one transposition from IPS' ordering. The aggregate of word similarity section yielded $\mu_{\text{phrase}} \in [3.92, 4.06]$, where the correcting ordering was numbered 4.

Finally, for the profile similarity 74% of the user's orderings completely agreed with IPS' ordering, and 96% of the users made at most one transposition from IPS' ordering. The aggregate of word similarity section yielded $\mu_{\text{profile}} \in [3.91, 4.27]$, where the correcting ordering was numbered 4.

In all cases the confidence interval is much smaller than 1 (the minimum transposition distance). This in conjunction with the fact that 98% of the responses were within one transposition distance shows that IPS does a good job of providing orderings that would be consistent with what a user would expect.

5 Applying IPS in an OSN Study

We applied IPS to evaluate two popular social networks: Facebook and Orkut. We show how similarity correlated with topological distance with various subgrouping; part of this is validating the results from [9] using NLP instead of the intersection based approach they utilized and part is extending said work with flow inside affiliations and across genders. Our emphasis is to show that IPS provides more logical results for similarity values than prior work and to aid in developing a deeper insight into OSN growth. As such, we also show how IPS compares with the standard L1 intersection method.

5.1 Data Overview

The overall statistics of the two networks are shown in Table 1. We refer implicitly to this table throughout Section 5.1. Facebook allows users to describe themselves in a number of different categories; however, we concentrate only on the following categories: (1) activities, (2) interests, (3) gender, and (4) networks (affiliations). To help ensure clarity throughout the paper we will refer to the networks category as affiliations. Using the affiliations, users are able to restrict

Table 1. Overall Statistics

Attribute	Facebook	Orkut
Vertices	1265	15329
Edges	7827	61738
Average degree	12.4	8.1
Average number of keywords	5.17	29.5
Average number of words per keyword	5.2	1.99
Fraction of the network within 4 hops	74.4%	32.9%

the set of people that can view their profile, the default policy is that only those in the same affiliation or are immediate friends may view each others profiles. As a result crawling a large sample users in Facebook is considerably harder (hence the smaller graph size).

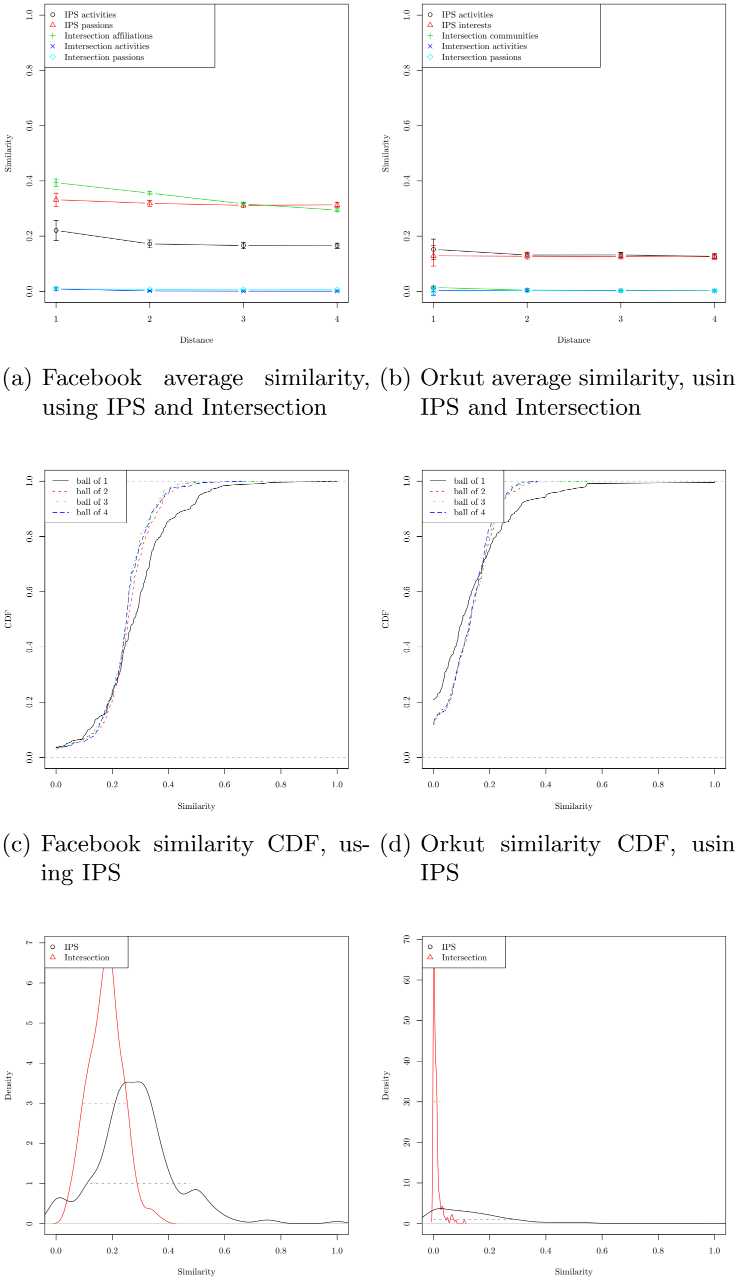
Similarly, in Orkut we concentrate on the following categories: (1) activities, (2) passions, (3) sex, and (4) communities. We chose to focus on these few categories as they generally use terms that exist in the dictionary (versus, e.g., Movies). Communities and affiliations were not compared using NLP, but instead used L1 intersection as the choices were from a fixed set with no semantic meaning to the keys. Furthermore, these categories allow cross-network comparison to ensure trends are consistent. For the NLP similarity comparison, we created rings for distances between 1 and 4, so we also present the fraction of the graphs that this covers.

5.2 Semantic Analysis

We now turn to investigate the similarity of users using IPS. We are primarily interested in how similarity between pairs of users changes with increasing topological distance. To compute these similarity distributions for every node in the network, we constructed the ring of nodes at a distance 1, then those at a distance 2 but not at a distance of 1 and so on until distance 4. For each set of nodes in the ring we applied the similarity algorithm to get the pairwise similarity with varying distances.

Figures 2(a) and 2(b) shows the average profile similarity versus topological distance and also compares IPS with intersection. Both IPS and intersection show a trend of decreasing similarity with increasing distance, which agrees with [9]. IPS provides much higher similarity values than intersection. Note that the affiliations (communities) was compared using intersection as the values were chosen from a fixed vocabulary set with no semantic meaning in the identifiers.

Figures 2(c) and 2(d) shows the CDF ($\Pr(X \leq x)$) of IPS similarity distribution at different distances, ball of 1, 2, 3 and 4 (where ball of 1 means all the nodes 1-hop away, ball of 2 means all the nodes 2-hop away and not 1-hop away, and so on). Given a similarity value, say 0.4, the CDF of ball 2, 3, 4 are all closer to 1 than the CDF of ball 1 (for Facebook, it is around 0.83, for Orkut, it is around, 0.86. This shows neighboring nodes tend to have a higher similarity value than non-neighboring nodes.



(a) Facebook average similarity, using IPS and Intersection

(b) Orkut average similarity, using IPS and Intersection

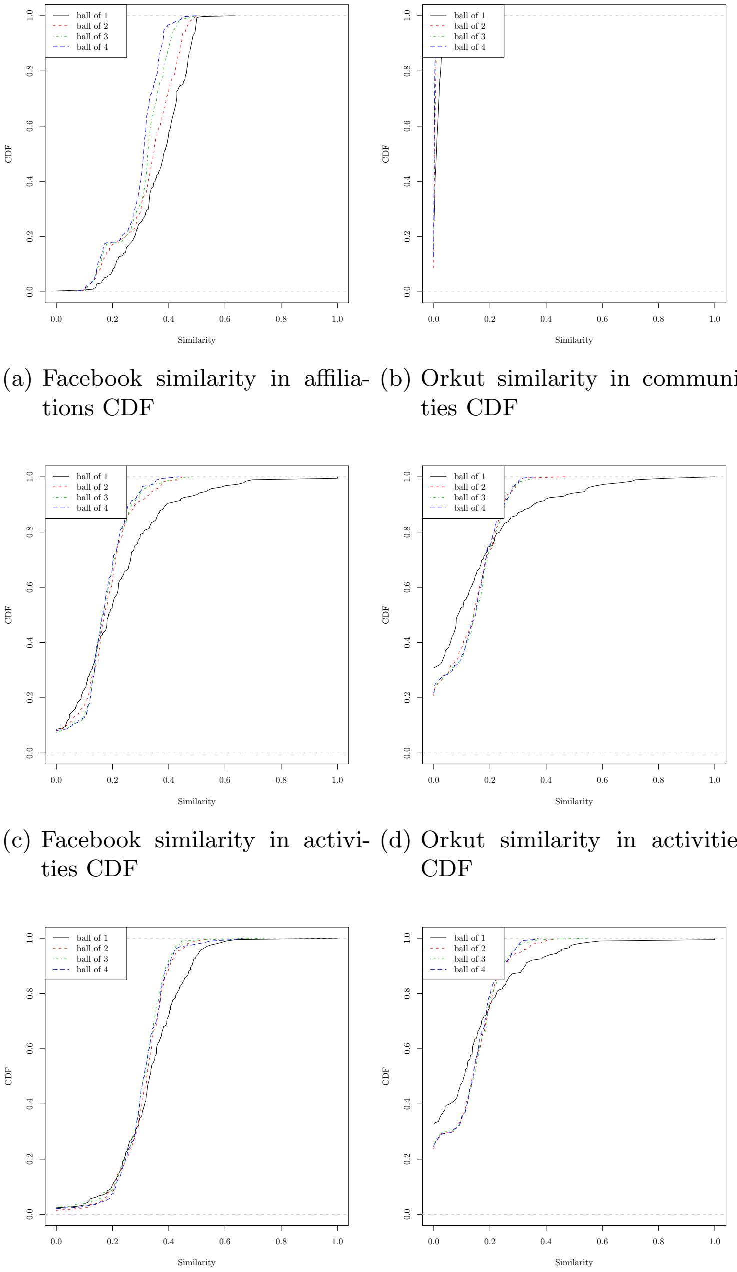
(c) Facebook similarity CDF, using IPS

(d) Orkut similarity CDF, using IPS

(e) Facebook similarity density, using IPS and Intersection

(f) Orkut similarity density, using IPS and Intersection

Fig. 2. Similarity comparisons over all profiles using Facebook and Orkut data



(a) Facebook similarity in affiliations CDF

(b) Orkut similarity in communities CDF

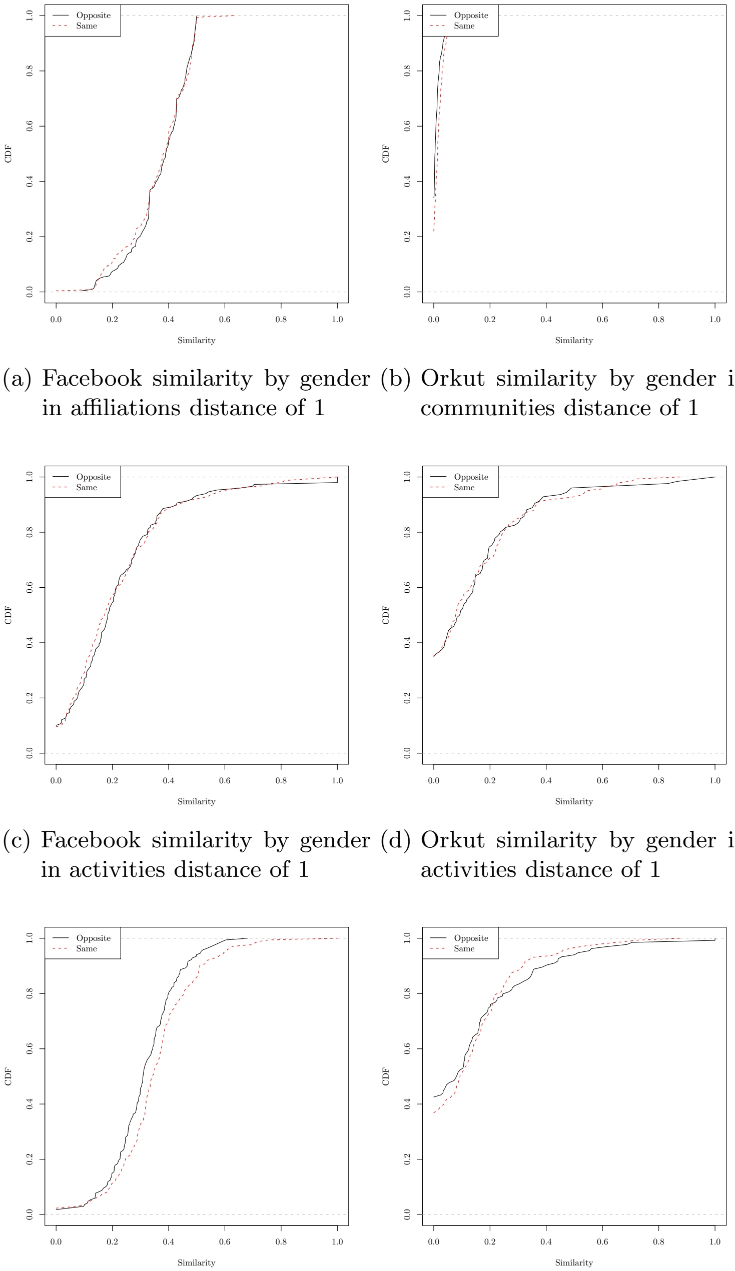
(c) Facebook similarity in activities CDF

(d) Orkut similarity in activities CDF

(e) Facebook similarity in interests CDF

(f) Orkut similarity in passions CDF

Fig. 3. IPS Similarity comparisons in affiliations, activities, and interests/passions using Facebook and Orkut data



(a) Facebook similarity by gender in affiliations distance of 1

(b) Orkut similarity by gender in communities distance of 1

(c) Facebook similarity by gender in activities distance of 1

(d) Orkut similarity by gender in activities distance of 1

(e) Facebook similarity by gender in interests distance of 1

(f) Orkut similarity by gender in interests distance of 1

Fig. 4. Gender Similarity comparisons

Finally, Figures 2(e) and 2(f) shows the similarity density distribution using IPS and intersection for the ball of 1. The dashed lines indicate the range which encompasses 80% of the similarity values. Two observations can be made: (1) The intersection method produces lower similarity values (centered at 0.2) than IPS (centered at 0.3). (2) The Intersection method produces a small range of similarity values (80% of similarity values are between $[0.1, 0.3]$, where as IPS produces larger range of similarity values (80% of similarity values are between $[0.1, 0.45]$. This means that IPS gives better similarity measures as the similarity values are less clustered and more meaningful to be interpreted.

One issue with doing analysis on the data is that there is dependence between data points; as an example consider that A and B are very similar as are B and C then the similarity of A and C is likely to also be high—it is not independent of the first two measurements. The bootstrap is a useful tool for data analysis arising in nonstandard situations, such as the correlated-data setting we have here [13,14].

Next, in Figures 3(a)–3(f) we investigate how similarity changes with increasing distance per category using IPS. In all cases the difference of means shows a trend towards closer pairs being more similar.

Next, we investigated the difference in means between males and females at distance 1 using IPS. Figures 4(a)–4(f) show the CDF for each of the Facebook and Orkut categories. In almost all cases the difference of means is almost perfectly symmetric around zero indicating it is likely there is not any meaningful difference. Interests and communities were an exception showing a trend towards same-gender having a slightly higher mean. As there was no consistent trend across networks, it lends doubt that there is any meaningful difference between the similarities amongst genders.

Finally, we investigated if there was any correlation between *geographic* distance and similarity. Due to space limitations we do not display the graph nor the means, but we did not see any significant correlation between geographic distance and similarity.

6 Conclusions and Future Work

We have presented IPS, a simple and novel extension to WordNet [6] can be used to evaluate the similarity of words, phrases and profiles. The total ordering produced agrees strongly with what a user expects, as verified through our user-study. IPS can be utilized in many existing applications to provide a simple, application-independent ordering of users based solely on existing profile data. We, also, applied IPS to evaluate both Facebook and Orkut graphs, which were geographically diverse, and obtained many pieces of data.

IPS has a few shortcomings as described in Section 3.3. In the future more study into better mechanisms to address these and optimizing IPS would be important. With the power of IPS, we also want to investigate how network properties change over time and see if similarity helps to explains why OSN growth exceeds the prediction by preferential attachment [7].

References

1. Wallenius' noncentral hypergeometric distribution, http://en.wikipedia.org/wiki/Wallenius_noncentral_hypergeometric_distribution
2. Hammouda, K., Kamel, M.: Phrase-based Document Similarity Based on an Index Graph Model. In: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), p. 203. IEEE Computer Society, Washington (2002)
3. Investigating Measures for Pairwise Document Similarity, http://www.ncstrl.org:8900/ncstrl/servlet/search?formname=detail&id=oai
4. Zamir, O., Etzioni, O., Karp, R.: In: Kamel, M.K. (ed.) Knowledge Discovery and Data Mining, pp. 287–290 (1997)
5. Sahami, M., Heilman, T.: A web-based kernel function for measuring the similarity of short text snippets. In: Proceedings of the 15th international conference on World Wide Web, pp. 377–386. ACM, New York (2006)
6. Patwardhan, S., Pedersen, T.: Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In: Proceedings of the EACL 2006 workshop, pp. 1–8 (2006)
7. Mislove, A., Marcon, M., Gummadi, K., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pp. 29–42. ACM, New York (2007)
8. Marlow, C., Naaman, M., Boyd, D., Davis, M.: HT 2006, tagging paper, taxonomy, Flickr, academic article, to read. In: Proceedings of the 17th conference on Hypertext, pp. 31–40. ACM, New York (2006)
9. Information Flow in Social Groups, http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0305305
10. Fellbaum, C.: WordNet: An Electronic Lexical Database (Language, Speech, and Communication). The MIT Press, Cambridge (1998)
11. Spertus, E., Sahami, M., Buyukkokten, O.: Evaluating similarity measures: a large-scale study in the orkut social network. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 678–684. ACM, New York (2005)
12. Wen, Z., Tzerpos, V.: Evaluating Similarity Measures for Software Decompositions. In: Proceedings of the 20th IEEE International Conference on Software Maintenance, pp. 368–377. IEEE Computer Society, Washington (2004)
13. Bradley, E.: Better Bootstrap Confidence Intervals. Journal of the American Statistical Association 82, 171–185 (1987)
14. Tan, P., Steinback, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Reading (2005)