Diverging towards the common good

Heterogeneous Self-Organisation in Decentralised Recommenders

Anne-Marie Kermarrec
INRIA Rennes Bretagne Atlantique, Rennes, France
anne-marie.kermarrec@inria.fr

François Taïani
Lancaster University, Lancaster LA1 4YW, UK
f.taiani@lancaster.ac.uk

Abstract

Decentralised social networks promise to deliver highly personalised, privacy-preserving, scalable and robust implementations of key social network features, such as search, query extensions, and recommendations. Such systems go beyond traditional online social networks by leveraging implicit social ties to implement personalised services. Yet, current decentralised social systems usually treat all users uniformly, when different sub-communities of users might in fact work best with different mechanisms. In this paper, we look at the specific case of decentralised social networks seeking to cluster users exhibiting similar behaviours to provide decentralised recommendations. These decentralised recommendation systems typically rely on a single metric applied uniformly to all users to extract similarities, while it seems natural that there is no such one-size-fits-all approach. More specifically we show in this paper, using a real Twitter trace, that (i) individual users can benefit from a personalised strategy in the context of decentralised recommendation systems, and that (ii) overall system performance is improved when the system accounts for the varying needs of its users *i.e.* when each user is allowed to diverge and use its optimal strategy.

Categories and Subject Descriptors C.2.4 [*Distributed Systems*]: Distributed applications; D.4.7 [*Organization and Design*]: Distributed systems

General Terms Algorithms, Design, Experimentation

Keywords Social Networks, Gossip protocols, Decentralisation, Recommendation

1. Introduction

The past decade has witnessed a dramatic shift in the way the Web is used. While social networks have exploded, many other forms of social networking have also emerged, *e.g.* user-generated content services (Flickr, Youtube, Delicious), micro-blogging platforms (Twitter, Tumblr), and geo-located services (foursquare, Gowalla—now part of Facebook). This new context, where a huge amount of information is dynamically produced, shared, and searched globally, represents a fantastic potential to leverage insights about each and every user and use it to provide better, more personalised, services. Although traditional social networks have been extremely successful, they today only encompass a tiny share of the data associated with each user, and are therefore limited in their ability to provide highly personalised services. Decentralised alternatives that are able to fully harvest user data where it is produced—at the user—are therefore appealing over more traditional social networks to provide a better personalised online experience [1, 6]. In addition, decentralisation holds the promise of providing users with more control on their privacy, and of delivering more scalable architectures, a key issue as illustrated by Twitter's not-so-infrequent outages.

To implement these decentralized (implicit) social networks where each user typically gets connected to users resembling her, gossip protocols appear as an appealing solution, as they intrinsically tend to be highly resilient, efficient, and scalable. Social applications based on gossip have however been limited so far to relatively homogeneous systems: They typically rely on one similarity metric [8] to self-organise large numbers of distributed users in implicit communities, and offer powerful means to search, mine, and serve personalised data in a distributed manner. Figuring out the right similarity metric that best fits the needs of a large collection of users is however challenging. As in many other areas in computer science, there is no such a thing as a one-size-fits-all approach.

In this paper, we argue that, as social systems grow, and encompass an increasing amount of diverse information, homogeneous approaches will reach their limitations. Our intuition is that in very large social networks, different sub-communities will require different mechanisms to best benefit from social network features (e.g. to spread news notifications, or to recommend new contacts). Using diverging mechanisms in a decentralised system is however fraught with challenges, as different mechanisms might become incompatible, or counter productive, and might decrease the overall's system cohesion and efficiency.

In the following we explore this tension between local needs (which may diverge) and overall cohesion in the case of gossip-based self-organizing social networks. We look at the particular case of content recommendation, a key feature of most social networks, and investigate how different similarity metrics might best work for different subsets of users. We do so in the context of a Twitter trace. We then evaluate the performance of a heterogeneous self-organising social network, in which each user runs its optimal similarity, and show that divergence is a viable way forward to improve the quality of recommendation in decentralised social networks, improving recall from 78.0% to 85.2% in our experiment without compromising convergence. These results argue for *personalising the personalisation process*, and we believe open several promising avenues of research yet to be explored.

The rest of the paper is organised as follow, we first provide some background on gossip-based implicit social networks in Section 2, we propose a heterogeneity-aware approach in Section 3.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SNS'12 April 10, 2012, Bern, Switzerland
Copyright © 2012 ACM ACM 978-1-4503-1164-9/12/04. . . $10.00

Figure 1. Some of the tweets received by Taiani

Experimental results are provided in Section 4, before we conclude with a discussion on open problems in Section 5.

2. Decentralised (implicit) Social Networks

To illustrate our approach, we focus in the rest of the paper on micro-blogging recommendations, a particular case of content recommendation in social networking applications. Before delving into the detail of our approach in Section 3, we introduce here the fundamental concepts underlying micro-blogging, social recommendations, and gossip-based decentralised social networks. We highlight in particular the crucial role of similarity in such decentralised solutions, and the challenges associated with choosing the right similarity metric.

2.1 Micro-blogging

A micro-blogging service (e.g. Twitter[1]) is a particular instance of social application that allows users to publish a stream of short online messages (known as *tweets* in Twitter). Other users can subscribe to a user's messages, in which case they become a *follower* of this user. The messages a user receives through these subscriptions, along with the ones she produces constitute her *home timeline* and are displayed on her twitter homepage (Figure 1).

Originally launched in 2006, Twitter is one of the leading micro-blogging services today, with 100 millions users announced in 2011[2]. Twitter offers a number of functionalities on top of its micro-blogging platform, including forwarding (re-tweeting), tweet responses, trending (following popular keywords in real-time), search (including geo-located searches), and recommendations, a fundamental service of most on-line social networks.

Twitter recommendations suggest accounts a user might be interested to follow based on a number of heuristics. More generally, advanced techniques exist to compute recommendations centrally, for instance based on link prediction and collaborative filtering [5, 7]. These methods typically assume global knowledge of all users' profiles, and might involve complex global parameter optimisation (such as minimising a matrix product over the system's entire dataset). They are as such difficult to implement efficiently in a distributed setting.

2.2 Decentralised recommendations

An alternative to centralised content recommendation consists in exploiting the ability of large-scale gossip-based systems to organise a large set of users (called peers) in local communities of similar

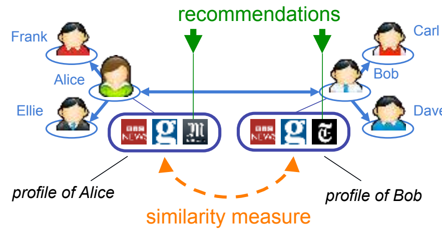

Figure 2. A typical gossip-based social network

users [1, 3, 8]. This type of decentralised content recommendation uses a peer-to-peer architecture in which each peer (a smart phone, a tablet, a laptop) is associated with a user (Fig. 2). (In the rest of the paper we assume a one-to-one mapping between a peer and a user for the sake of simplicity.) Peers can connect to each other using point-to-point networking, but they only have a partial and very limited view of the rest of the system: Typically a small-size neighbourhood of other peers. Each peer is characterized by a profile, for instance a list of friends, or a list of items tagged in a collaborative tagging service (delicious), or the set of the twitter feeds the user is subscribed to (twitter). Peers use this data to organise themselves in a distributed overlay network so that similar peers end up connected together. E.g. in Fig. 2, Alice and Bob have similar subscription profiles (they both follow the BBC and the Guardian on Twitter), and have therefore been selected to be neighbours of each other (using a mechanism we detail just below).

The resulting overlay can then be exploited to propose a range of personalised services such as search, query expansions and recommendations. The intuition behind this is that if Alice and Bob have subscribed to similar twitter feeds, then Alice's profile can be leveraged to provide a better service to Bob. Here for instance, Alice might know of twitter feeds of interest to Bob (here that of the newspaper *Le Monde*), and reciprocally (the New York Times).

2.3 Constructing the overlay

To construct a similarity overlay, a gossip-based social network uses a two-layer architecture (Fig. 3) [1, 3, 8]. Both layers maintain an overlay, in which peers have a fixed list of neighbours. The bottom (random) layer implements a Random Peers Sampling (RPS) protocol, ensuring the connectivity of the whole network through a random graph-like topology. This is achieved by having peers exchange and shuffle their neighbours list in *periodic gossip rounds* to maximise the randomness of the RPS graph over time [4]. The upper (clustering) layer clusters peers according to their similarity.

To achieve this clustering, the clustering layer implements a local greedy optimisation procedure that leverages neighbours returned both by the RPS layer, and by the clustering views [3, 8]. A peer (say Alice in Fig. 3) will periodically update its list of similar neighbours with new neighbours found to be more 'similar' to her in the RPS layer. This guarantees convergence under stable conditions, but can be particularly slow in large systems. This mechanism is therefore complemented by a swap procedure in the

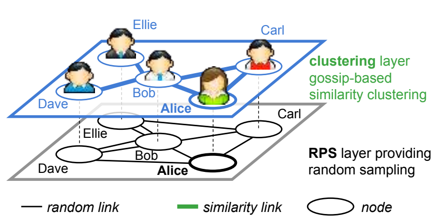

Figure 3. Gossip-based distributed clustering

[1] https://https://twitter.com/

[2] http://blog.twitter.com/2011/09/one-hundred-million-voices.html

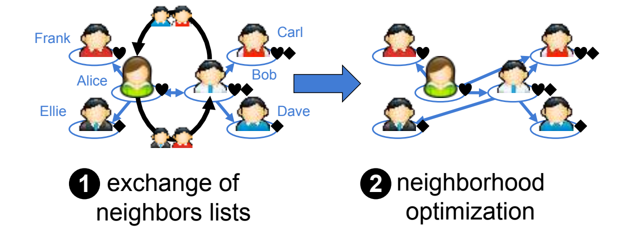

Figure 4. Peer-to-peer clustering mechanism

clustering layer (Fig. 4), whereby two neighbouring peers (here Alice and Bob) exchange their neighbours lists (Step 1), and seek to construct a better neighbourhood based on the other peer's information (Step 2).

In Fig. 4 for instance, Alice has a thing for hearts, and prefers like-minded people. Bob, on the other hand, has a minor interest in hearts, but is much more interested in diamonds. When Alice sends to Bob her current list of neighbours, and Bob sends Alice his (Step 1), both discover new potential neighbours closer to their interests. Alice thus drops Ellie for Carl, and Bob drops Alice for Ellie.

The whole process relies on one *homogeneous* similarity metric (Jaccard or cosine similarity or even a simple overlap) which is used by each peer to compute its similarity with other peers across the whole system. The clustering protocol converges fairly rapidly precisely because all peers rely on this very same metric. This leverages the fact that if Alice is similar to Bob, Bob's neighbours in the clustering topology are likely to be good candidates for Alice too. Yet, the larger a system grows, the less likely one single similarity metric will be able to match every and each user's requirements.

3. Investigating Divergence: Approach

In order to confirm our intuition that a single global metric is inadequate to organise an entire decentralised system, we investigate on a real workload the behaviour of several similarity metrics. We analyse in particular whether any of these metrics can be considered optimal. In this section, we first present our dataset, the various similarity metrics we consider and describe the experiments we conducted to verify our intuition.

3.1 Dataset

For this exploratory study, we use a Twitter dataset containing 1000 users, which is representative of a small internal social network, for instance in a Small or Medium Enterprise or a university. The dataset was crawled between May and October 2011, mainly from the San Francisco bay area using Twitter's geolocated search, and information from Foursquare, a geolocated social network. We recorded for each user u_i the set of Tweeter subscriptions $subs_i^{total}$ to which u_i is subscribed (her *subscription set*).

Our original dataset contains 392,526 subscriptions to 250,286 unique feeds. The 15 most popular feeds are shown on Table 1. The distribution of feed occurrences among users is shown on Figure 5, and clearly shows a power-law behaviour. The vast majority of feeds (81.5%, or 203,957 out of 250,286) are only subscribed by one user each, with only a minority (0.3%, or 921) shared across at least 20 users.

The distribution of the number of subscriptions per user in our original dataset is shown on Figure 6 (solid black line). Not surprisingly, this shows a wide spread of user behaviours, with 90% of the users having between 28 and 1337 subscriptions, and the median falling at 198 subscriptions per user.

Feed	User Count
BarackObama	331
ConanOBrien	300
TechCrunch	257
mashable	248
shitmydadsays	225
cnnbrk	223
foursquare	203
twitter	198
google	194
nytimes	181
BillGates	178
StephenAtHome	167
kevinrose	161
jack	158
ladygaga	158

Table 1. The 15 most popular feeds in our dataset

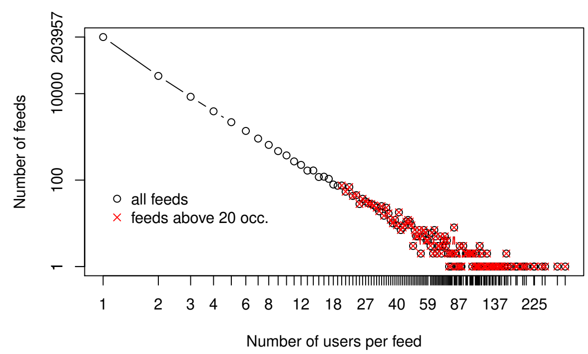

Figure 5. Distribution of feed popularity among users (log-log scale).

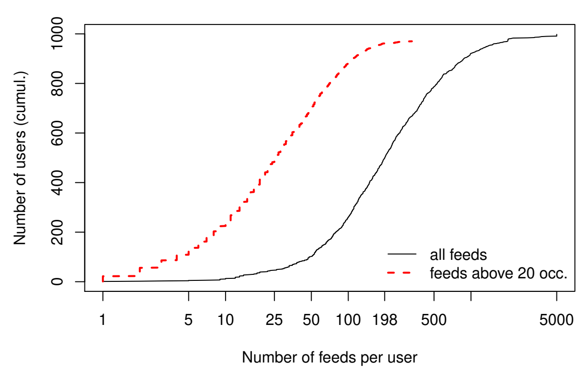

Figure 6. Distribution of the number of subscriptions per users (log scale on the x axis)

The intuition behind the recommendation mechanism presented in Section 2 assumes some minimum level of redundancy in the dataset: subscriptions that are only subscribed to by one user are unlikely to be of interest to many other users in the system. We therefore filter our original dataset and only keep those subscriptions that are shared between at least 20 users (red crosses on Figure 5), and only retain users with at least one subscription left after this filtering operation. This results in a final dataset of 970 users, having 38,789 subscriptions to 921 unique feeds. This reduces in particular the number of subscriptions per user to a median of 25, with 90% of users having between 2 and 129 subscriptions. The top 15 most popular feeds in this filtered dataset remains however those of Table 1, with the same user counts.

To test the quality of recommendation returned by each of our experiment, we use a simple cross-validation technique. We randomly split each user's u_i subscription set subs_i^{total} into two parts: a training set subs_i (80%), and a hidden set hidden_i (20%). The training set of subscriptions is used to run the experiments and organise peers in similarity neighbourhoods, while the remaining 20% are used to measure the quality of the returned predictions. In order to assess the impact of a given similarity metric, we first cluster users with that similarity metric using each user's training set (i.e. subs_i, 80% of the original trace profile). For each user, we then compute the recommendations for that user, which is the union of the subscriptions present in the profile of this user's neighbours in the resulting social network. In the rest of the paper we use recall[3] as our principal quality metric, defined as the proportion of correct recommendations (i.e. recommendations found in the hidden set hidden_i) among all recommendations. In particular we compute in each round of our experiment the average recall for all users:

$$\underset{u_i \in Users}{average}\left(\frac{|\mathsf{rec}_i \cap \mathsf{hidden}_i|}{|\mathsf{rec}_i|}\right)$$

3.2 Similarity metrics

To test our hypothesis, we consider four different similarity metrics, listed in Table 2. overlap and jaccard are two traditional metrics often used in information retrieval systems. overlap favours users who have a large number of subscriptions in common. overlap tends however to prioritise power-users, who are subscribed to many feeds, but might not represent the interests of other potential users. jaccard mitigates this effect by normalising overlap by the size of the union of the two sets of profiles.

Users with very small profiles (one or two subscriptions for instance) might however fare badly both with overlap and jaccard. We before consider two further metrics: big takes the idea of overlap to its extreme, and assumes that power users (users with the largest number of subscriptions) might be a good source of recommendations. $\mathsf{over\_big}(u_i, u_j)$ is a variation of overlap and big that just adds the two metrics together.

It is important to note this set of selected metrics is somewhat arbitrary: There might be other choices, possibly more efficient, that could be considered. Rather than a detailed study of each metric, our goal here is to illustrate the potential for multiple heterogeneous metrics in the same self-organising network, and to demonstrate the viability of a gossip-based clustering system based on this principle.

$$\begin{aligned}
\mathsf{overlap}(u_i, u_j) &= |\mathsf{subs}_i \cap \mathsf{subs}_j| \\
\mathsf{big}(u_i, u_j) &= |\mathsf{subs}_j| \\
\mathsf{over_big}(u_i, u_j) &= \mathsf{overlap}(u_i, u_j) + \mathsf{big}(u_i, u_j) \\
\mathsf{jaccard}(u_i, u_j) &= \frac{\mathsf{overlap}(u_i, u_j)}{|\mathsf{subs}_i \cup \mathsf{subs}_j|}
\end{aligned}$$

Table 2. Similarities used in this study

3.3 Experiments

We perform two batches of experiments to explore the potential of heterogeneous similarities in self-organising gossip-based networks. In a first batch, we simulate a decentralised gossip-based recommendation system (Section 2) for each of the similarities of

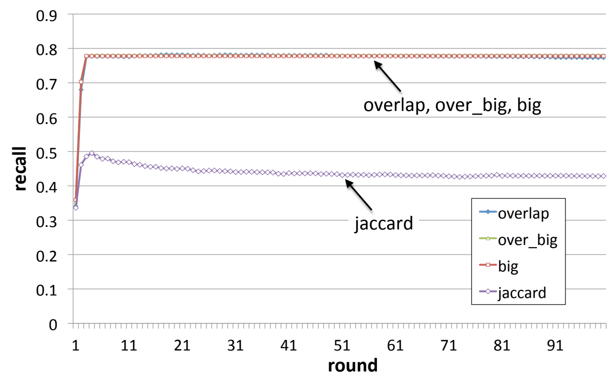

Figure 7. Convergence of recommendation recall for the similarities of Table 2

Table 2. We record for each peer and each similarity the recall obtained after 100 gossip rounds. We use a view size of 5 peers in the clustering layer, and 10 in the RPS layer. To accelerate our simulations, we also contact all of a peer's neighbours in each round to provide candidates for a new neighbourhood. We use this first set of experiments to compare similarities with each other, both globally (in terms of average recall), and at the level of each individual peer. The intuition here is that even if the system fares globally better with similarity f than with similarity g, some peers might in fact be better off (in terms of recall) with g than with f, demonstrating the existence of *heterogeneous needs* within the network.

In a second batch of experiments, we consider what would happen in a system if each peer uses its *optimal* similarity, rather than a uniformly imposed one. In effect this approach leads to using different similarities in the same self-organising system, with the actual similarity used for each peer pre-determined by our earlier experiment.

To assess the potential benefits of such a *heterogeneous similarity*, we first compute the 'ideal' recall to be obtained if each peer were able to reach its maximum recall for its optimal similarity. One key question is however whether such a heterogeneous similarity can work in practice. This is because self-organising gossip-based networks expect all peers to use the same metric. To assess the feasibility of a heterogeneous metric we therefore simulate a decentralised gossip based recommendation system with this heterogeneous similarity, and show that it converges close to the "ideal" recall (a result we have termed *converging heterogeneity*).

4. Experimental Results

4.1 In need of heterogeneity

The convergence behaviour of the four base metrics of Table 2 is shown on Figure 7. overlap, big and over_big deliver very similar performance, at around 77% of recall, while jaccard does markedly less well, at 43%. Beyond the actual measures (which are highly dependent of the dataset), and the specifics of each metric (overlap and big are rather similar in that power-users will tend to show high overlaps), this goes to show that it be can difficult to predict beforehand which similarity metrics might work best in a particular context.

Figure 7 is based on average recall values, and therefore only provides an aggregate measure of success, which does not necessarily hold for individual peers. This is made apparent by the two charts in Figures 8 and 9. Figure 8 shows a scatter-plot matrix that compares the recalls obtained by each similarity for individual peers. For instance, the plot with the arrow (first line, second column) compares the recalls obtained by peers when using overlap

[3] We ignore in particular precision in first approximation.

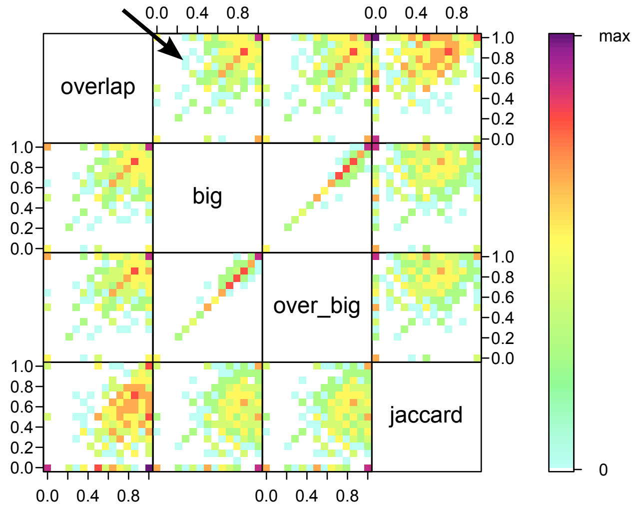

Figure 8. Recall results correlation between each pair of similarities. Each graph is a scatter plot of individual users presented as a density map. Each user is plotted according to the recall values obtained with the two similarities being compared (e.g. with overlap (y-axis) and big (x-axis) for the first graph of the first line).

(on its y-axis) and big (on its x-axis). Because individual peers tend to overlap in these plots, the diagram uses density plots rather than traditional scatter plots: each coloured square represents a number of peers found in this area of the plot, with the colour scale on the right following a logarithmic progression. For instance, the point under the arrow represents a peer that achieved a recall of 0.75 with overlap and 0.25 with big.

The main message from this graph is that no single similarity metric succeeds in being optimal for all peers. Interestingly even jaccard happens to beat each of the three stronger similarities for at least a few peers. For instance, coming back to the plot with the arrow: any peer above the diagonal obtains a better recall with overlap than with big, and oppositely for peers under the diagonal. To be globally optimal, a similarity would need to have all peers above the diagonal in all the scatter-plots on its line.

Each similarity thus partitions the set of peers into two non-empty subsets: a set for which this similarity yields the highest recorded recall, and a set for which this similarity does not deliver the highest recall. To explore this partitioning in more detail, we record for each peer (i.e. each user), the similarity (or similarities) for which this peer obtains its highest recall. In this case we call these similarities *optimal* for this user. As a short-cut we will also say that the neighbourhood constructed with these similarities is *optimal*[4]. In case of ties, we recorded all optimal similarities. The results are shown on Figure 9. Although overlap, big, and over_big manage to yield optimal recalls for a majority of peers (between 66.4% for big and 69.6% for overlap), no similarity achieves to be optimal for all peers. Furthermore even jaccard succeeds in being optimal for 15.2% of peers.

This phenomenon is looked at in more detail in Table 3 (ignoring the last row for the moment). The first column of Table 3 repeats the information of Figure 9, and shows the proportion of peers for which a similarity is optimal (e.g. 69.5% for overlap). The second columns shows the average recall obtained when only considering those peers for which the similarity is optimal. Conversely, the third columns shows the same average recall, but computed over the remaining "sub-optimal" peers, and the final column shows the overall average recall obtained over all peers. All similarities do substantially better on their "optimal" peers (second column) than on their suboptimal ones (third), and the difference is quite important, ranging from 20.97% for overlap to 35.55% for jaccard.

[4] *Optimal* is here to be taken with a grain of salt. It just means that the corresponding neighbourhood yields the highest recall among the neighbourhoods constructed with the other similarities. This neighbourhood might not, however, be optimal among all possible neighbourhoods.

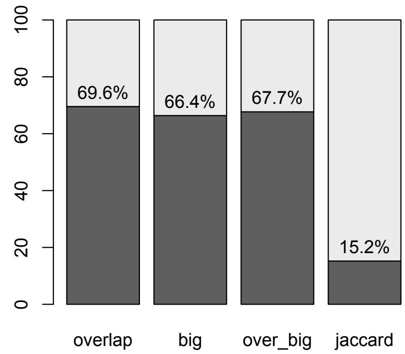

Figure 9. Percentage of peers for which a similarity is optimal. Several similarities might be in a tie for a peer, hence the total above 100%.

Both this table and Figure 9 substantiate our intuition that in reasonably complex decentralised social networks a single metric is unlikely to work optimally for all users. The natural follow-up question is thus whether this observation can be used to construct a better system, that tries to cater for each user's specifics needs. This is the question we turn to in the next section.

4.2 Heterogeneous convergence

Our proposal is to consider a self-organising overlay in which each peer is free to use whatever metrics happens to work best for it. To explore the potential benefits of such an approach, we assume that each peer knows beforehand which optimal similarity (from those of Table 2) to use, using the first best similarity in case of ties. We return to how this might be obtained in practice in Section 5. The resulting similarity, which we term heterog, is *composite*, being made of the four original metrics we considered.

We first consider the theoretical recall that might be obtained with heterog if each peer were able to reconstruct the same optimal neighbourhood as in the previous experiment. In such an overlay, each peer would obtain the same best recall as the one we recorded in Section 4.1, yielding an overall theoretical recall of 86.23%, shown on the last line of Table 3 (heterog_ideal). By construction, this ideal version of heterog is optimal for all peers, and harnesses the best performance of all metrics of Table 2.

	optimal for	*recall (%, 100 rounds)*		
	(% peers)	**optimal**	**suboptimal**	**overall**
overlap	69.55	85.08	59.57	77.32
big	66.37	84.82	63.85	77.77
over_big	67.69	84.86	63.62	77.99
jaccard	15.22	73.01	37.46	42.87
heterog_ideal	100.00	N/A	N/A	86.23

Table 3. Proportion of peers for which a similarity is optimal (1^{st} column), average recall obtained by each similarity for the peers for which they are optimal (2^{nd} column), resp. sub-optimal (3^{rd} column), and overall average recall (final column)

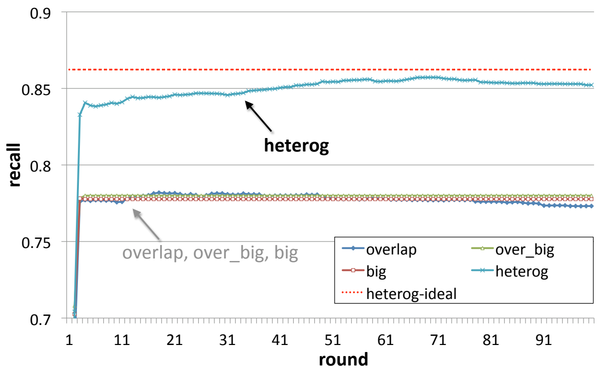

Figure 10. Convergence of the composite similarity heterog compared to the base similarities.

In practice there is however no guarantee that heterog will yield such a high recall. This is because of at least two reasons. First, as mentioned earlier, self-organising gossip-based networks expect all peers to use the same metric, and the use of a heterogeneous metric that returns values that are not necessarily comparable might disrupt the convergence process. Second, because of the cross-validation we use, a peer (say Alice) might obtain the same high similarity with several other peers (let's call them Bob and Dave). From the point of view of heterog, Bob and Dave are indistinguishable, and once the system has stabilised, might keep entering and leaving Alice's neighbourhood. If however Bob and Dave have different hidden sets, they will contribute differently to Alice's recall, preventing heterog to reach the optimal configuration represented by heterog_ideal.

To investigate these potential limitations, we simulate heterog under the same conditions as the other metrics. The resulting convergence curve is shown in Figure 10. The final recall values (in round 100) along with the proportion of peers for which heterog yields an optimal neighbourhood are summarised in Table 4.

Figure 10 shows that under our experimental settings heterog rapidly converges to a recall value well above that of the other metrics (round 3, 83.2%). heterog then slowly grows to peak at 85.72% in round 70, before stabilising back to 85.22% in round 100. These recall values are within 1.01% of the "ideal" recall computed in Table 3 (86.23%, shown at the top-most dashed line on Figure 10), and show that heterog comes close to reconstructing optimal neighbourhoods for most peers. This is confirmed by Table 4, that shows that heterog yields optimal recalls for 90.47% of the peers, a proportion that is 20.92% above the closest of the original four similarities (overlap at 69.55%, see Table 3).

	optimal for	*recall (%, 100 rounds)*		
	(% peers)	**optimal**	**suboptimal**	**overall**
heterog	90.47	87.54	63.18	85.22

Table 4. Average recall obtained by the composite similarity 'heterog' on the peers for which it is optimal (2^{nd} column), resp. suboptimal (3^{rd} column), and overall average recall (final column)

5. Discussion and Conclusion

Our experiments show that the use of multiple co-existing similarities in self-organising gossip-based networks have the potential to provide a better service (in our case recommendation) in decentralised social applications.

Although promising, our experiment is limited in a number of ways, which each hint at possible avenues for future research. First, we have set beforehand which similarity each peer should use, based on a previous experiment. A natural question is therefore whether and how each peer's optimal similarity could be learnt *on the fly* at runtime, using available contextual information (for instance the user's current feedback on previously made recommendations).

Second, the user set we have used is quite small. Although it can be representative of an enterprise social network in an SME, our dataset is just not comparable to the data available to leading social applications (Facebook, Twitter, Foursquare). A key question is to understand how our approach would fare in these much larger networks. Intuitively, some of our metrics would probably no longer work: The 'big' metric for instance is likely to degrade as the number of users grows, along with that of niche content and of sub-communities.

Larger and richer datasets should however also provide more opportunities to explore a larger range of similarity metrics. In our example, user profiles are limited to sets of items, from which only a limited amount of features might be extracted (size, intersection, union). The profiles available to social networking applications are however much richer: they usually encompass different sets of data (friends, locations, content) which can be combined arbitrarily, and usually possess some additional semantics. For instance locations or content might be organised in categories, which can be used adapt the similarity of each user (for instance by adapting the weight of each category for each user) [2].

Acknowledgments

This work has been partially supported by the ERC Starting Grant GOSSPLE number 204742.

References

[1] M. Bertier, D. Frey, R. Guerraoui, A.-M. Kermarrec, and V. Leroy. The gossple anonymous social network. In *Proceedings of the ACM/IFIP/USENIX 11th International Conference on Middleware (Middleware'2010)*, pages 191–211, Bangalore, India, 2010.

[2] J. Carretero, F. Isaila, A.-M. Kermarrec, F. Taïani, and J. M. Tirado. Geology: Modular georecommendation in gossip-based social networks. In *IEEE 32^{nd} Int. Conf. on Dist. Computing Sys. (ICDCS 2012)*, Macau, China, June 2012.

[3] M. Jelasity and O. Babaoglu. T-man: Gossip-based overlay topology management. In *3rd Int. Workshop on Engineering Self-Organising App.*, 2005.

[4] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M. van Steen. Gossip-based peer sampling. *ACM ToCS*, 25, August 2007.

[5] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '08)*, pages 426–434, Las Vegas, Nevada, USA, 2008.

[6] J. M. Pujol, V. Erramilli, G. Siganos, X. Yang, N. Laoutaris, P. Chhabra, and P. Rodriguez. The little engine(s) that could: scaling online social networks. In *Proceedings of the ACM SIGCOMM 2010 conference (SIGCOMM 2010)*, pages 375–386, New Delhi, India.

[7] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proc. of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '11)*, pages 1046–1054, 2011.

[8] S. Voulgaris and M. v. Steen. Epidemic-style management of semantic overlays for content-based searching. In *Proceedings of the 11th international Euro-Par conference on Parallel Processing (Euro-Par'05)*, pages 1143–1152.