Consiglio Nazionale delle Ricerche

Tag-based Recommender System for Context-Aware Content Dissemination in Opportunistic Networks

V.Arnaboldi, M.Campana, F. Delmastro, E.Pagani

IIT TR-07/2015

Technical Report

giugno 2015

Istituto di Informatica e Telematica

Tag-based Recommender System for Context-Aware Content Dissemination in Opportunistic Networks

Valerio Arnaboldi[*1], Mattia Campana[†2,1], Franca Delmastro[‡1] and Elena Pagani[§2]

[1]IIT-CNR, Via G. Moruzzi 1, 56124, Pisa, Italy
[2]Department of Computer Science and Communication, University of Milan, Via Comelico, 39/41, 20135 Milan, Italy

Abstract

Content dissemination in opportunistic networks is a hot research topic that attracted a lot of interest in the last few years. The key idea is to optimise the diffusion of content among nodes in opportunistic networks to ensure that users are always able to obtain the most relevant items according to their interests. The classical approach is to statically define a set of interests for each user, and make sure that they receive items matching those interests. In this paper, we propose a novel approach, based on the dynamic and automatic identification of interests. To do so, we exploit the tags that users assign to the items they create, and the tags of the items that they download. We model these actions through a folksonomy and the related tripartite graph, with different nodes for users, items, and tags. We use this graph as the basis for identifying the relevance of the items. Specifically, we use a tag-based recommender system on the graph, called PLIERS, that is able to calculate the relevance of an item for a certain user, with respect to the items that are already linked to this user.

We validate our approach through a series of simulations. We emulate the presence of a variable number of agents which randomly move, create and tag items, and possibly encounter other agents. Each agent maintains a tripartite graph locally, representing its actions, and it integrates this graph with information received from other encountered nodes. The agents use PLIERS on their local graph to assess the relevance of the items they find, and they decide whether these items are relevant for them

---

[*]v.arnaboldi@iit.cnr.it

[†]mattia.campana@studenti.unimi.it

[‡]f.delmastro@iit.cnr.it

[§]pagani@di.unimi.it

or not. We evaluate the accuracy of the results by comparing the recommendations on the local graphs with the relevance of the items (calculated through PLIERS) on a global graph obtained by merging together all the local graphs of the nodes. This graph represents the complete knowledge of all actions in the network and it allows us to obtain the best possible recommendations for a target user, that could be obtained if all the nodes had the full knowledge of the actions of other nodes. The results indicate that the recommendations on the local graph are accurate and that the local knowledge of nodes reaches the global knowledge in the network through a sufficiently high number of contacts.

1 Introduction

In an opportunistic environment, devices encounter and exploit all the communication opportunities offered by the available wireless technologies to communicate and exchange data, also under intermittent connectivity conditions. In this scenario, user mobility represents mainly an additional communication opportunity that the system can exploit in order to transfer data towards unreachable users and devices (i.e., data dissemination). To this aim, it is essential for the system to dynamically discover the context characterising the local user and her device, in terms of user's interests in content and resources available on the network, user's social behavior, and the surrounding environment, in order to appropriately define efficient data dissemination algorithms.

As shown in [5] and similar works, the effectiveness of opportunistic routing and forwarding algorithms greatly increases if they take into account context information as defined above, and similarly for content dissemination. However, one of the main limitations of actual context-aware systems, in opportunistic networking scenarios in particular, is the lack of mechanisms for the *automatic* and *dynamic* definition and classification of context data. Most of the existing solutions manually define a static "knowledge base", that represents the corpus of context entities of the system and their relations, the basis on which context-reasoning is performed. Actually, in a highly dynamic environment like opportunistic networks, in which it is not possibile to have a complete knowledge of the network and of the available content/resources, services and applications must be able to adapt in order to provide appropriate feedback to the users. Content dissemination represents a big example in this direction: if the system is able to detect changes in the context and in the users interests, they can be used to automatically discover new interesting content in the network and/or to limit the transmission of undesired or useless content between peers, and thus avoiding to waste resources.

In this paper, we propose a novel solution for content dissemination in opportunistic networks in which the context is represented by the information that characterises the users' interests and the available content. The context is not known a priori and it dynamically changes over time, depending on the single user's action. This solution is based on the definition of *folksonomies* (i.e., a user-defined taxonomy in order to allow users to freely assign tags to

resources/content[12]), on the use of a recommender system for reasoning upon these information representing the context of a single user and of the network. This solution is inspired by the way content and users are categorised and associated in Online Social Networks (hereinafter OSNs). The advent of OSNs fostered the users generation of annotations on web content. Thanks to OSNs, users can easily create, share, and access multimedia content. Users interact with a content by leaving annotations on it. For example, they can post comments or reviews concerning products, places, people, or other. Perhaps more importantly, users can actively contribute in the categorisation of content by adding tags (i.e., short text labels that assign the content to user-defined categories). These tags are useful both to identify the content type (e.g. "mov", "jpg", "text", etc...) and to give a semantic meaning to the data (e.g. "cat", "politics", "football", etc...). This metadata can be automatically generated by the applications or manually created by the users.

In contrast with the traditional taxonomies and ontologies used to model a complex knowledge, a folksonomy does not require any explicit relationship between the terms that constitute it (e.g., generalization, specialization and correlation). Therefore, starting from a folksonomy characterising users' interests and generated content, we want to identify in an opportunistic network the potential new interests of the local user related to content generated by other encountered users. To this aim, we define an optimised *tag-based recommender system* for opportunistic network, based on the success of these systems on the Web [17] (especially for commercial applications), but able to overtake their limitations in a highly dynamic environment. This system is called PLIERS and it has been defined in [11] after a deep analysis of tag-based recommender systems already proposed in literature, which have been also used to validate PLIERS performances in a Web-based scenario.

In order to validate the efficacy of the proposed solution, in this paper we describe the state of the art on context-aware content dissemination for opportunistic networks and how PLIERS can be used in this environment. Then, we provide a detailed performance evaluation in a simulation scenario.

2 Related Work

Content dissemination systems have been proposed both for the legacy Internet networks [3], and for conventional MANETs [14]. In general, these systems assume that network paths are rather stable, and in some cases generate a significant amount of traffic to maintain knowledge of the other nodes and caches. Therefore, they are not suitable to opportunistic networks. In the last few years, researchers proposed some solutions also for opportunistic networks, where there is no stable path, and forwarding algorithms are mainly based on the *store-carry-forward* paradigm [6]. Specific attention has been paid to context-aware forwarding and content dissemination but mainly starting from a predefined knowledge of users' interests and their associations with generated content. In these cases the main idea is to use context information to characterise the proba-

bility of users' encounters and the possibility to move messages closer and closer to their destinations following a path based on the social interactions between nodes (as in the famous *six degrees of separation* experiment [13]). Although this approach is feasible for forwarding, since messages have a specific destination node, it is not for content dissemination solutions, which mainly follow the User Generated Content approach. In this case content generators might be unaware of the nodes interested in their data, and so might be the content consumers about the nodes that generate the content they are interested in. For this reason some solutions exploit the approach of Pub/Sub systems: users explicitly declare their interests in a set of content (i.e., channel subscription) and each content is registered as belonging to a specific set. Then, additional context information is used to optimise the dissemination among the network nodes. PodNet project [9] is one of these solutions in which users subscribe to channels they are interested in (i.e., content category) and, upon pair-wise contacts, they exchange their interests and select which data objects to exchange. Content-Place [4] further extends this approach by exploiting social information related to users' encounters and sharing interests as context information to enable the communication and to disseminate potentially interesting content. Specifically, it exploits a community detection algorithm defined in [15] to classify users' relationships and define social communities of users. In addition, it collects information about the available content and related channels during pair-wise contacts and, based on this information, it defines an utility function of each data object for all the social communities any given user is in touch with. In this way, it is able to drive the content dissemination through the network following various socially-inspired policies mainly based on the social profile of the local user and her social communities.

In the solution we propose, we approach the content dissemination problem from a completely different perspective: we assume that there is no a-priori knowledge of the user's interests but it is dynamically created by using a folksonomy defined by the network users while generating and sharing their own content. The folksonomy represents the set of information characterising a user, her generated content, and the associated tags. Then, through opportunistic communications, users exchange their local information, defining thus on each node a "knowledge base" that can be naturally represented through graphs. Specifically, users, items and tags can be seen as nodes of the graph and their relations (i.e, user-item, user-tag links) represent the interests of each user in specific items and tags. This is due to the fact that the user generated the content and/or she downloaded it from another user. Moreover, links between items and tags indicate that a certain item has been tagged with one or more tags, thus classifying it into user-defined categories. Since users, items and tags belong to different categories, tripartite graphs are typically used, assuming that the links between a user and a set of item reflect the user's interests in the tags associated to the same items. In this representation, nodes of different types can be linked to each other, but no links can exist between nodes of the same type. Separate bipartite graphs could also be used to express relations between two out of the three types of nodes (e.g., user-item and user-tags relations).

4

Once defined the use of a folksonomy and its representation through bipartite and tripartite graphs, we decided to use a recommender system able to reason upon this type of knowledge in order to identify relevant content for a target user within the context of an opportunistic network. This type of recommender systems are generally defined as *tag-based recommender system*. In the next section we briefly explain the basic principles of those systems and our optimised solution called PLIERS. Then, we detail the use of PLIERS in an opportunistic environment.

3   PLIERS: PopuLarity-based ItEm Recommender System

In the literature, several approaches have been proposed for Tag-based Recommender Systems [17]. The most promising solutions are based on the definition of a score value for each item on the network calculated through the diffusion of fictitious resources generally on a bipartite graph, from a node representing a target user (i.e., the target of the recommendation) and the items/tags on the graph. These solutions cab be categorised as *network- and diffusion-based* algorithms. This allows the recommender system to identify relevant items/tags in the network that are indirectly connected to the target user via other users sharing with it one or more connections. The higher the number of links connecting an item to the items of the target user, the higher the score that this item will receive for the recommendation. In this way, the recommender system exploits the structure of the network to identify the most relevant content for the user.

ProbS [18] and *HeatS* [16] represent two of the most used tag-based recommender systems. Both these algorithms have strong limitations that are mainly related to the fact that they base their scores on the general popularity of a content on the network, independently of the impact of the target user interests on the network structure. As a result of their algorithms, *ProbS* tends to recommend the most popular content among those available on the network and *HeatS*, on the contrary, tends to highlight those with minimal popularity (i.e., with the smallest possible number of users connected to them). To overcome these limitations, a hybrid *ProbS + HeatS* solution has been recently proposed [10]. It calculates a linear combination of ProbS and HeatS, inheriting thus both the positive and the negative effects of them. In addition, it requires the definition and the dynamic tuning of a parameter representing the weight of each algorithm in the linear combination.

In [11] we propose a new tag-based recommender system called *PLIERS: PopuLarity-based ItEm Recommender System* that solves the dilemma between the choice of popular or unpopular items in a more natural way with respect to the hybrid ProbS+HeatS solution, and without requiring any parameters to tune. It guarantees that the popularity of the recommended items is always compatible and comparable with the popularity of items belonging the target user. This

introduces an additional evaluation of the item relevance for the target user by assuming that she will be interested (with a high probability) in items that have similar popularity in the network to those generated by the user herself. By following the user behavior on OSN we can assume that a user interested in a very popular item or tag (i.e., connected to many users), can semantically relate to a more "generic" topic, compared to a less popular item that, instead, is intended to describe a more "specific" topic. For example, any content related to the football club Millwall can be tagged with both tags *"Millwall"* and *"Football"* but the opposite is not always true: all content concerning football will not always be tagged with *"Millwall"*. According to this assumption, we can therefore say that the tag *"Football"* refers to a more generic topic than that referred by the tag *"Millwall"*. Users interested in the Millwall football club, but not connected to items tagged with *"Football"*, are clearly not interested in all the items tagged with the latter tag, as these could contain information of other football clubs. Recommending items with popularity compatible to that of the items of the users permits to overcome this problem, and leads to better recommendations. Simulation results on real and heterogeneous datasets in [11] show that PLIERS outperforms the other solutions proposed in literature by providing more accurate recommendations to the target users.

4   Using PLIERS to Improve Content Dissemination in Opportunistic Environments

In the reference scenario of content dissemination in opportunistic networks, a user generally performs a restricted set of actions: she may move in the environment, encounter other users, generate and share items, download items from other encountered users through device-to-device communications (as an explicit user action or as an automatic system operation), tag items, both generated by herself or by others. We assume that tags identify user interests in specific topics and content. Thus, when a user generates a new item and she tags it, we assume that she is actually interested in the other content available in the network with the same tags. Moreover, when a user downloads an item from another node, she can be also interested in all the other items in the network with the same tags. This means that the interests of the user may be automatically and dynamically discovered by looking at the tags of the items she generated or downloaded, and they can change over time. This is also a consequence of the use of a recommender system, allowing the user to become aware of the availability of content that are not strictly related to her tags. The ensemble of the actions users can perform in the opportunistic environment can define the context data that are relevant for the purpose of content dissemination and it can be fully described by a folksonomy modelled through bipartite and tripartite graphs. Clearly, these graphs are not static, and they evolve over time as nodes perform their actions. Therefore, to maintain an updated version of the knowledge graph of a target user in terms of own-generated content and those

downloaded from others, each node must be aware of the actions performed by the other nodes of the network.

To this aim we implement a context-aware content dissemination protocol that exploits the recommending features of PLIERS based on the dynamic generation of a local knowledge of the users' interests and the available content. Specifically, the protocol is implemented on each node as a software agent that allows nodes to exchange and integrate their local graph with those of the nodes that they meet through the following actions:

a. **Context-information exchange**: Each node sends its local graph to the other node.

b. **Local knowledge update**: Each node merges its local graph with the information received from the other node.

c. **Execution of PLIERS**: Each node runs PLIERS to evaluate whether the new items carried by the other node may be of interest for it.

d. **Download**: Each agent asks to download relevant items (if any), according to the scores obtained through the use of PLIERS.

Figure 1 depicts the operations performed by the content dissemination protocol running on two nodes when they come across each other.
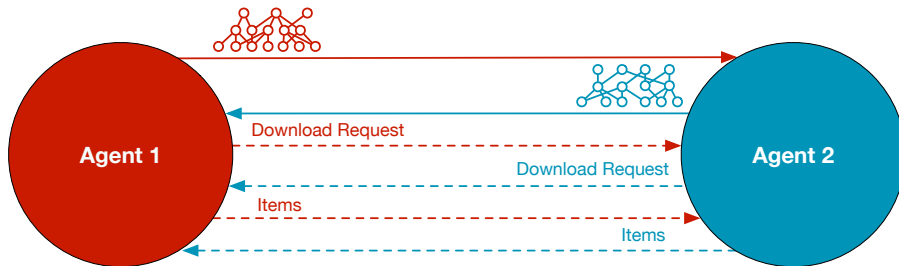

Figure 1: Sequence of actions during an opportunistic contact between a pair of nodes.

Therefore, each agent is in charge of maintaining and updating the knowledge graph of the local user every time she encounters a new user on the network and to evaluate the recommending score of each available content through the execution of PLIERS. Naturally, each agent can have only a limited view of the actions occurring in the whole network that is bounded to the set of nodes that each user directly encounters over time. For this reason, the graph representing user-item-tags relations that each node carries (called *local knowledge graph*) only partially reflects the whole knowledge of actions in the network (i.e., the *global knowledge graph*). In addition, we assume that each agent is responsible for the creation of links in the local graphs between the local user and the generated and/or downloaded items and between those items and related tags.

By using PLIERS the content dissemination algorithm is thus able to dynamically model the user's interests based on her generated content (if any) and on the download of additional content available on the network and suggested by the system itself. In this way, the user's interests can change over time together with the availability and popularity of the content, and the dissemination on the network dynamically adapts to the user requirements. In this case the folksonomy represents the main context information used to optimise the content dissemination but, at the same time, it can be further extended through additional context information, like social information and resource availability, as presented in previous works on context-aware content dissemination. In addition, the methodology defined by PLIERS can also be adopted to reason upon other types of context information that can be modelled through a folksonomy or simply through a bipartite or tripartite graph.

However, the use of PLIERS in the opportunistic environment and especially on mobile devices introduces a set of issues mainly related to: (i) the accuracy of PLIERS recommendations in a highly variable environment, and (ii) the management of the local knowledge graph. In the following sections we present specific features of PLIERS that we introduce to optimise its behavior in the opportunistic environment, while for details on the definition of PLIERS algorithm and its performances in a Web-based scenario please refer to [11].

4.1   Suggestion buffer

It is worth noting that tag-based recommender systems generally operate on a complete, global knowledge graph derived from centralised datasets. This is not possibile in an opportunistic environment where the number of relevant and available items for the local user varies at each encounter, and the average value of their relevance with respect to the local user interest (i.e., PLIERS scores) could greatly vary over time depending on several factors: node's mobility, encountered nodes, items owned by the local node, and the items available on the other nodes. To ensure that users always obtain the most interesting items for them out of the set of items available on the network, we introduce the concept of *Suggestion Buffer* in PLIERS, used to calculate the average score of the items available for download.

Every time the content dissemination agent finds a new available item (during an opportunistic encounter), it calculates its PLIERS score and it adds this scores to the buffer together with the item identification number and the timestamp of its encounter. If the item is already present in the buffer, the agent updates the score and the timestamp. In this way, the agent will request for download only items with PLIERS score higher than the average score of the last n encountered items. This value is called *current_avg* and it is calculated as follows:

$$current\_avg = (1 - \beta) \cdot old\_avg + \beta \cdot new\_avg$$

where *old_avg* refers to the value of *current_avg* calculated at the previous step, *new_avg* is the average value of the items currently present in the buffer and

β controls the update rate for the new value with respect to the value at the previous step.

The value of *current_avg* is used to evaluate the relevance for the local user of the items which will be encountered in the future. In this way, the system is able not only to recommend the most relevant items for the local user at a certain time, but also to dynamically adapt the recommendations according to the availability of interesting content in the network.

The first time old_avg is equal to current_avg. After this first calculation, when new items are found, the least recently seen items are removed from the buffer to make room for the new ones, and the new value of current_avg is calculated at each insertion or update. When the buffer is still not completely full, the agent may decide to either download all the encountered items or none of them. For the sake of simplicity, in our simulations the agent does not download any item until the buffer is full.

Figure 2 depicts an example of how the buffer and the calculation of current_avg work. At t_0, the buffer is empty. At t_2, the node has encountered other nodes, and it has seen two items ($I1$ and $I2$), and calculated their PLIERS scores (0.02 and 0.03). At this step, the buffer is still not completely full, and no calculations are performed on current_avg. The agent does not download the items, but it only stores their PLIERS scores (s in the figure) and the timestamp related to their calculations (ts in the figure). After adding other items to the buffer, when the buffer is full (t13 in the figure), the new value of current_avg is calculated, and the items whose score is higher than this value are identified as relevant.
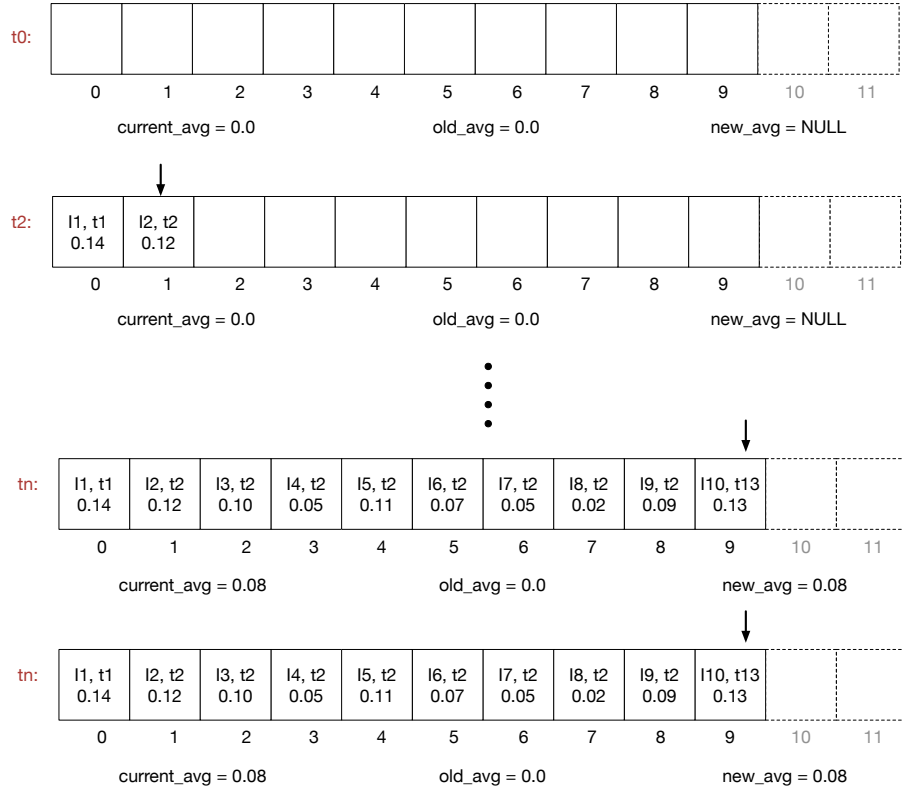
Figure 2: Suggestion buffer.

4.2   Local graph management

As previously described, when nodes come across each other, they exchange and merge their local knowledge graph. These graphs could continuously grow over time, and this could saturate the nodes' resources. To prevent this, we propose two possible solutions to maintain a limited size of the graph.

Limited update During the update of the local graph (i.e., during each opportunistic contact), the agent adds or updates only the new nodes related to items that are connected to users or tags that share at least one item with the local user. In fact, items not sharing any link to users or tags connected to at least one of the items of the local user are not considered by PLIERS for the recommendations (PLIERS score is equal to 0). In this way, the growth of the graph can be limited by not considering useless information for the actual recommendations. A downside of this approach is that the discarded information could be useful for the local user in the future, as some discarded nodes could become connected to the user once she changes her interests.
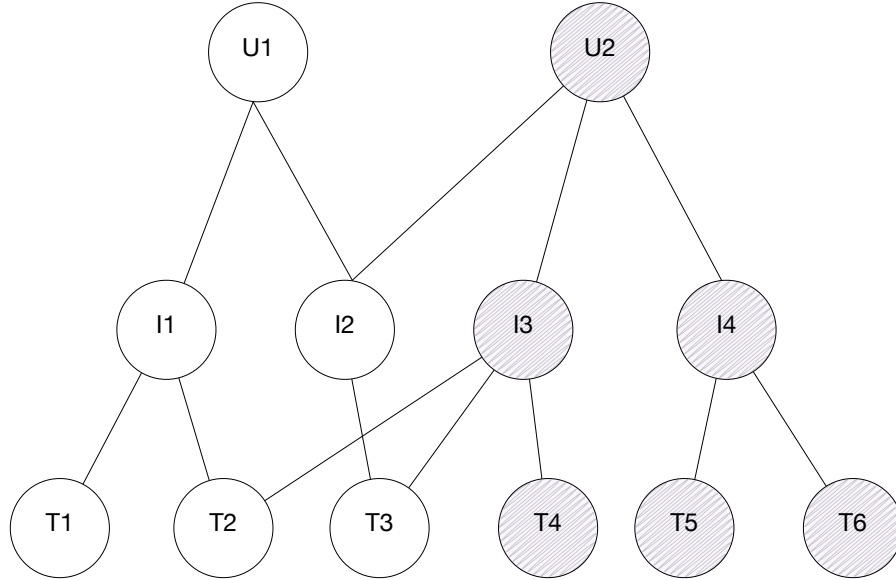
10

Figure 3: Pruning of the tripartite graph.

Pruning With this solution, each node maintains a maximum number of nodes and links in memory. To do so, the local graph is enriched with temporal information. Specifically, each link in the graph is associated with a timestamp. When a user generates an item, or the system downloads it from other nodes, the agent creates the related nodes in the graph if they do not exist yet, it adds a link in the local graph connecting the local user to the item, and it associates it with the current timestamp. When a node meets other nodes and it receives their local graphs, it checks, for each link user-item in the graph, if it is already present in its local graph. If not, it adds the link and, in case, the nodes at their endpoints. If it already has the link, it compares the timestamps and maintains the most recent one. When the maximum number of nodes and links is reached, the node checks whether the graph received from another node contains more recent information than the graph it owns and, if so, it deletes the links with the oldest timestamps to make room for the new links. This procedure implements, as defined in [8], the *Most Recent Contacts (MR)* policy. Alternatively, the *Most Frequent Contacts (MF)* policy could be implemented by maintaining the information related to the nodes most frequently seen.

For those pruning policies it is important that the local graph maintains a certain level of consistency. Specifically, for the proper execution of PLIERS, at the end of the pruning, no node can remain disconnected from the other components of the graph. Referring to the example in Figure 3, if U2 (User 2) is deleted by pruning the graph, the items linked to it and not connected to other users must be deleted as well, along with the tags that will eventually remain disconnected.

Table 1: Simulation parameters

N. of Agents	Size of the area	N. of contacts
50	500 m x 500 m	95,316
50	1,000 m x 1,000 m	24,296
50	2,500 m x 2,500 m	3,932
50	5,000 m x 5,000 m	1,008
100	1,000 m x 1,000 m	94,872
100	2,000 m x 2,000 m	23,808
100	5,000 m x 5,000 m	3,792
100	10,000 m x 10,000 m	948
200	2,000 m x 2,000 m	93,636
200	4,000 m x 4,000 m	23,696
200	10,000 m x 10,000 m	3,788
200	20,000 m x 20,000 m	916

5 Performance evaluation through opportunistic simulations

In order to validate the proposed solution and to test the accuracy of the recommendations of PLIERS, we performed a set of experiments in a simulated opportunistic environment. Specifically, we simulated the presence of 50, 100, and 200 agents moving in a limited area, and possibly encountering each other. The parameters of the performed simulations are reported in Table 1. Each agent emulates the actions of a node in our content dissemination protocol, and maintains its own local graph. We simulate the nodes mobility in the opportunistic network by assigning a random mobility trace to each agent, generated through a random walk model, with the parameters reported in Table 2. When agents come across each other (i.e., their distance is lesser than or equal to $30m$), they exchange and merge their local graphs, and they download relevant items from each other. We simulate the creation of content for each node by using real traces obtained from Twitter. Specifically, we used the same dataset used in [7, 1, 2]. We sample from the Twitter dataset a number of users equal to the agents in each simulation setting. In Table 3, we report the number of items (tweets) and tags (hashtags) extracted from the dataset for the different configurations. In Twitter, the creation of tweets, that we consider as items, is associated with a timestamp. We normalized the timestamps of the tweets in the dataset to fit the simulation time of $28,800s$. Then, we associated each agent of the simulator with a Twitter user. During the simulations, each agent creates items at the times related to the creation of the Tweets. When an agent creates an item, it updates its local graph, adding a link between the node representing itself and the created item, and between the item and the tags identified by the *hashtags* of the tweet. The simulations end when the agents generate all their items.

Table 2: Simulation parameters

Parameter	Value
Simulation time	28,800 s
Minimum node speed	1.0 m/s
Maximum node speed	1,86 m/s
Transmission range	30 m

We performed two types of simulations: (i) each agent updates the interests of the local user when it creates an item, by adding a link in its local graph between the user and the related item. (ii) Each agent updates the interests of the local user also when it downloads items from other nodes.

5.1 Simulation 1: Interests updated when items are created

In this simulation, the agents generate items and tags, updating their local graphs consequently to these operations. Moreover, they exchange and merge their local graphs (without limiting the updates) with other encountered agents, but they do not download any item from them. We use this particular setting to assess the extent to which each agent is able to approximate the global knowledge graph with the information of the local graphs received from the other encountered agents. Figure 4 shows the results in terms of average similarity (calculated with the Jaccard's coefficient) between the local graph of the agents and the global knowledge graph, as a function of the number of opportunistic contacts occurring in the simulations. It is worth noting that the agents are able, after a certain number of contacts, to have a complete view of the global knowledge. However, the higher the density of agents in the area of the simulation, the higher the number of contacts needed to reach the global knowledge.

Table 3: Samples used for the opportunistic simulations.

Agents	Items	Tags
50	3193	2059
100	4520	2059
200	13213	7561
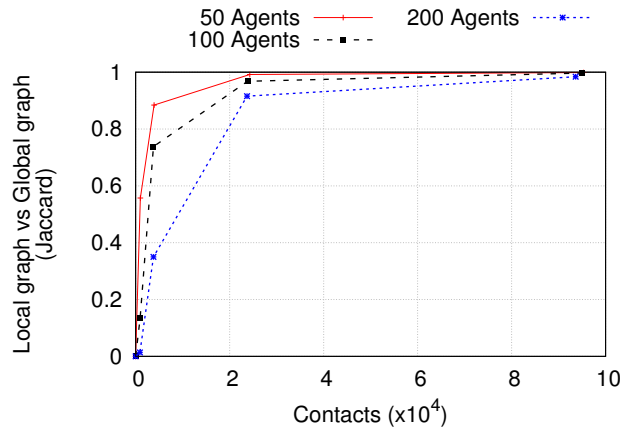
Figure 4: Similarity between local and global graphs

(a) $buffer\_size = 5$

(b) $buffer\_size = 10$

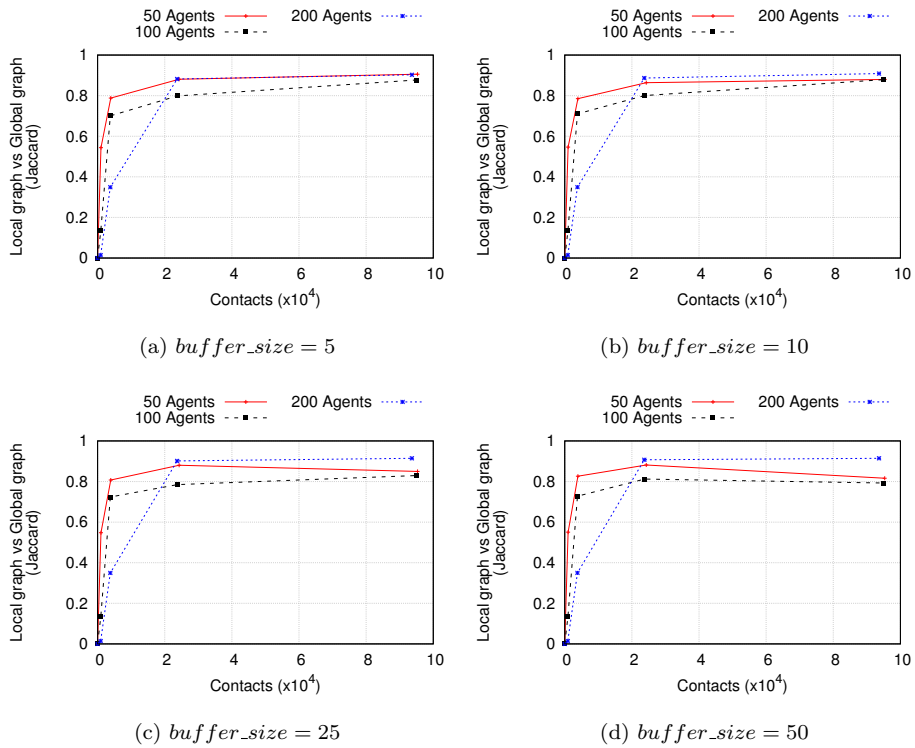(c) $buffer\_size = 25$

(d) $buffer\_size = 50$

Figure 5: Similarity between local and global graph varying the number of contacts and the suggestion buffer size. In blue it shows the trend for 50 agents, in black that for 100 and in orange for 200.
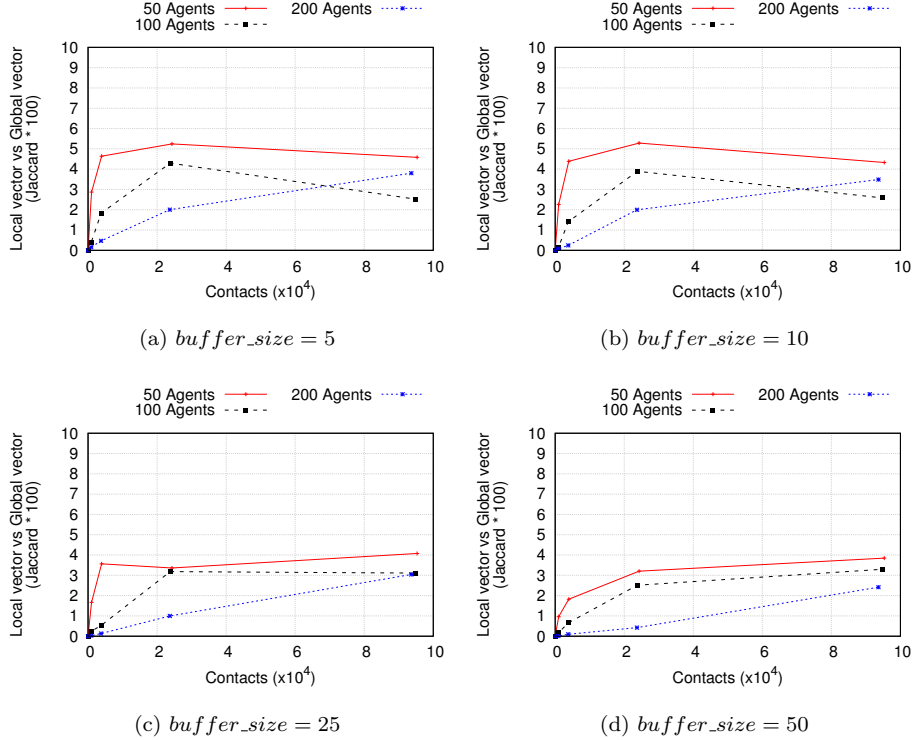
Figure 6: Mean percentage of items in the global resource vector that have been downloaded by agents at the end of the simulations, varying the number of contacts and suggestion buffer size. In blue it shows the trend for 50 agents, in black that for 100 and in orange for 200.

5.2 Simulation 2: Interests updated also for downloads

In this type of simulation each agent acts as in the case of the previous simulation, but, in addition, it downloads from the agents it encounters the relevant items it finds, recommended by PLIERS. In this case, it updates its interests also by adding links between the node in its local graph representing itself and the downloaded items. For this kind of simulation, we assess, in addition to the similarity between the local graphs and the global graph, the goodness of the recommendation given by PLIERS comparing them with the recommendations which each item would receive if the agent had access to the global knowledge graph and it could use PLIERS on it. Figure 5, depicts the average similarity (the Jaccard's coefficient) between the local graphs of the agents and the global graph for the diffent number of agents in the simulations and for different buffer sizes. For relatively small buffer sizes (5 and 10), the agents reach the global knowledge, even if, compared to the previous simulations, the graphs receive a much higher number of updates, due to the changes associated to the down-

loads. On the other hand, with relatively large buffers (size 25 and 50), the local knowledge of the agents approaches the global knowledge, but, after a certain number of contacts, they start diverging from it. This can be explained by the fact that with larger buffers the system is more selective in the choice of relevant items, for the average score current_avg is calculated on a higher number of visited items and the probability to have an item with high PLIERS score in the buffer, that could significantly increase the value of the average, is higher. Figure 7 shows the degree to which the lists of relevant items recommended by PLIERS using the local graphs of the agents (local vectors) are similar to the lists of recommended items obtained from the global knowledge (global vectors). This similarity between the two lists is calculated through the Jaccard's index.

5.3 Adaptability of the suggestion buffer

We further analysed the performance of our solution by looking at the ability of the value of current_avg to vary over time in order to adapt to the availability of relevant items in the network. Figure 7 depicts the trends of the average value of current_avg during the simulations for three randomly selected agents and for different sizes of the buffer. It is worth noting that, for larger buffers, the value of curent_avg is more stable over time, while it varies with a higher frequency for small buffer sizes. Thus, the buffer smooths the adaptability of the value of current_avg and controls the sensitiveness of the system to changes in the items availability in the network.

6 Conclusions

In this work, we presented a novel protocol for context-aware content dissemination in opportunistic networks, based on the use of a new tag-based recommender system: PLIERS. The main innovation of our protocol is the ability to automatically and dynamically discover the users' interests in the network, by collecting and reasoning upon their context data. This is in contrast to existing solutions in the literature, where interests are generally assumed to be static and defined a priori by the users. We assume that users can generate items, download them from other encountered nodes, and tag them with user-defined labels. When a user creates or downloads an item with a set of tags, we consider that she could also be interested in all the other items in the network tagged by similar tags or items owned by similar users. Thus, we can represent the context relevant for content dissemination through a folksonomy that can be represented with a tripartite graph with three categories of nodes: users, items, and tags. Moreover, links between these nodes represent the actions concerning the generation or download of items in case of user-item links, and the presence of tags for items in case of item-tag links. On this tripartite graph, we apply a tag-based recommender system (PLIERS), suitable for identifying relevant content on these kinds of graph, whereby the similarity between items is calculated considering the overlap between the set of users and tags connected to

(a) current_avg for random agent 28

(b) current_avg for random agent 46

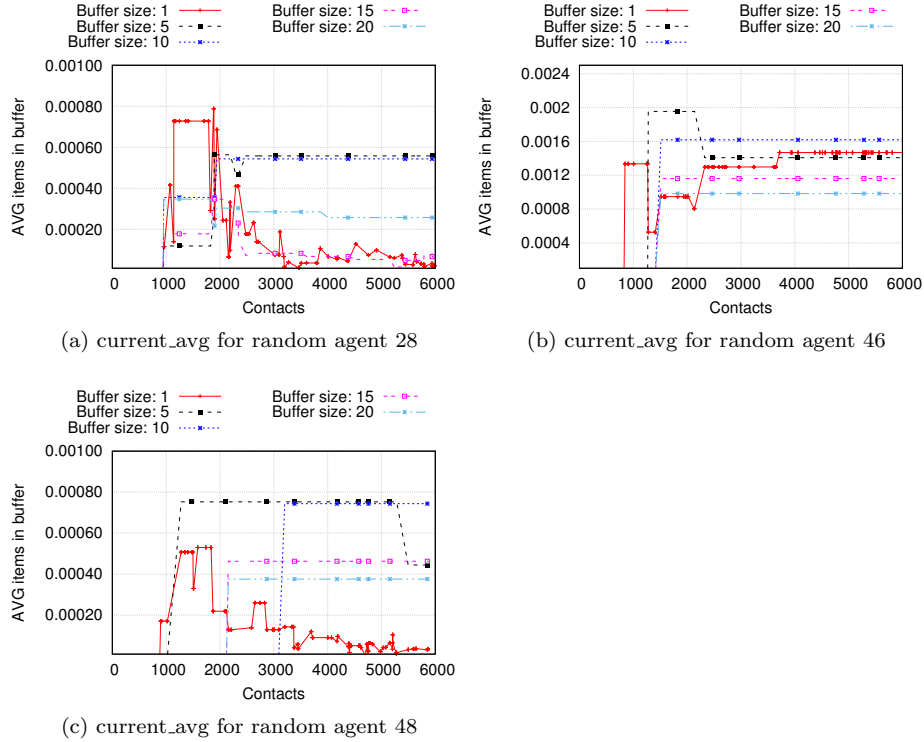(c) current_avg for random agent 48

Figure 7: Variation of current_avg over time in the simulations for three randomly selected agents.

the considered items and the items of the user for which the recommendation is performed (i.e. the "target user"). PLIERS, compared to other existing tag-based recommender systems, is able to recommend items also minimising the difference in terms of popularity between the recommended items and the items of the target user.

The protocol is implemented as a software agent running on each user device and being able to maintain and update a graph representing the system/user actions *local graph*, augmented with the information derived from the exchange of the local graph of other users. The local graph represents the partial knowledge that the node has with respect to all the actions occurred in the network, represented by the global knowledge graph. In addition, every time a node encounters other nodes, it evaluates the relevance of the items that they are carrying calculating their PLIERS score on its local graph. In this way, it can autonomously discover interesting content in the network, without the need for the user to specify its interests in those content or in their tags.

We assessed the accuracy of the recommendations calculated on local graphs with respect to the recommendations that nodes would have obtained if they

had been able to access the global graph (i.e., the global knowledge of actions occurred in the network). Of course, the recommendations on local graphs are suboptimal to the recommendations on the global graph since, in opportunistic networks, each node can access a limited amount of items, depending on the number of other nodes it encounters, and the relevance of the items it sees could be different if it considered all the information available in the network.

We performed a series of experiments to prove the accuracy of our protocol, simulating the behavior of nodes in opportunistic networks. Specifically, in our simulations, we assume the presence of a variable number of users, moving within a limited area according to a random walk model. Moreover, each node generates items in accordance with the behavior of users in Twitter. We performed two types of simulations. In the first one, each node updates its interests only when it generates new items. In the second one each node updates its interests also when it downloads items from the network. Then, we assessed the degree to which the local graph of the nodes approximates the global graph of the network. This allowed us to discover that, after a certain number of contacts, all the nodes are able to have a complete view of the global knowledge for all the explored configurations. In addition, we compared the lists of recommended items for each node calculated on their local graphs with the list of recommendations that they would have received if they had been able to access the global knowledge. The results indicate that the accuracy of the recommendations increases with the number of contacts between nodes.

References

[1] Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Robin Dunbar. Dynamics of personal social relationships in online social networks: A study on twitter. In *Proceedings of the first ACM conference on Online social networks*, pages 15–26. ACM, 2013.

[2] Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Fabio Pezzoni. Ego networks in twitter: an experimental analysis. In *INFOCOM, 2013 Proceedings IEEE*, pages 3459–3464. IEEE, 2013.

[3] A. Balamash and M. Krunz. An overview of web caching replacement algorithms. *IEEE Comm. Surv. & Tut.*, 6(2):44–56, 2004.

[4] C. Boldrini, M. Conti, and A. Passarella. ContentPlace: social-aware data dissemination in opportunistic networks. In *ACM MSWiM*, 2008.

[5] Chiara Boldrini, Marco Conti, and Andrea Passarella. Exploiting users' social relations to forward data in opportunistic networks: The hibop solution. *Pervasive and Mobile Computing*, 4(5):633–657, 2008.

[6] Chiara Boldrini and Andrea Passarella. Data dissemination in opportunistic networks. *Mobile Ad Hoc Networking: Cutting Edge Directions,*, pages 453–490, 2013.

[7] RIM Dunbar, Valerio Arnaboldi, Marco Conti, and Andrea Passarella. The structure of online social networks mirrors those in the offline world. *Social Networks*, 43:39–47, 2015.

[8] Theus Hossmann, Thrasyvoulos Spyropoulos, and Franck Legendre. Know thy neighbor: Towards optimal mapping of contacts to social graphs for dtn routing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[9] Vincent Lenders, Gunnar Karlsson, and Martin May. Wireless Ad hoc Podcasting. In *IEEE SECON*, 2007.

[10] Chuang Liu and Wei-Xing Zhou. An improved heats+ probs hybrid recommendation algorithm based on heterogeneous initial resource configurations. *arXiv preprint arXiv:1005.3124*, 2010.

[11] M. Campana, V. Arnaboldi, F. Delmastro, E. Pagani. PLIERS: PopuLarity-based ItEm Recommender System. Technical Report TR-06-2015, IIT-CNR, 2015.

[12] Adam Mathes. Folksonomies-cooperative classification and communication through shared metadata, 2004.

[13] S. Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[14] Liangzhong Yin and Guohong Cao. Supporting cooperative caching in ad hoc networks. *IEEE Trans. on Mob. Comp.*, 5(1):77–89, 2006.

[15] Eiko Yoneki, Pan Hui, ShuYan Chan, and Jon Crowcroft. A socio-aware overlay for publish/subscribe communication in delay tolerant networks. In *ACM MSWiM*, 2007.

[16] Yi-Cheng Zhang, Marcel Blattner, and Yi-Kuo Yu. Heat conduction process on community networks as a recommendation model. *Physical review letters*, 99(15):154301, 2007.

[17] Zi-Ke Zhang, Tao Zhou, and Yi-Cheng Zhang. Tag-aware recommender systems: a state-of-the-art survey. *Journal of computer science and technology*, 26(5):767–777, 2011.

[18] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, 2010.