Performance of data forwarding in opportunistic social networks benefits considerably if one can make use of human mobility in terms of social contexts. However, it is difficult and time-consuming to calculate the centrality and similarity of nodes by using solutions of traditional social networks analysis, this is mainly because of the transient node contact and the intermittently connected link. In this paper, we are interested in the following question: Can we exploit some other stable social attributes to quantify the centrality and similarity of nodes? Aggregating GPS traces of human walks from the real world, we find that there exist two types of phenomena. One is public hotspot, the other is personal hotspot. Motivated by this observation, we propose Hotent (HOTspot-ENTropy), a novel data forwarding metric to improve the performance of opportunistic routing. First, we use the relative entropy between the public hotspots and the personal hotspots to compute node centrality. Second, we utilize the inverse symmetrized entropy of the personal hotspots between two nodes to evaluate their similarity. Third, we integrate the two social metrics by using the law of universal gravitation. Besides, we use the entropy of personal hotspots of a node to characterize its personality. Finally, we compare our routing strategy with the state-of-the-art works through extensive trace-driven simulations, the results show that Hotent largely outperforms other solutions, especially in terms of packet delivery ratio and the average number of hops per message.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

With the pervasive of hand-held mobile devices, such as smart phones, there arises the requirement to share content (e.g., news, photo, music, video clips, etc.) among those devices [1–4]. Such contents can be the offloaded data from virtual space, or the sensed information from physical world as shown in Fig. 1. People download and replicate content when they enter the communication range of access point (AP). When they are out of the AP’s coverage, they can request their interested contents from other peers. Thus, mobile devices form an opportunistic social network (OSN) [5–7], for example via WiFi, to exchange/share content by intermittent contacts.

In the above scenario, routing is the base of content sharing. It is also one of the most challenging problems, due to the lack of an end-to-end path between source and destination. This new feature leads to a considerable performance degradation for conventional wireless routing protocols such as AODV or DSR, since they are originally designed for stable scenarios. Hence, new data forwarding algorithms are desired for opportunistic scenarios.
Several routing schemes (e.g., epidemic [8] and data MULEs [9]) have been proposed to deal with this problem in the past few years. Among them, the epidemic scheme seems to be a feasible solution to forward content from a sender to a potential receiver when nothing is known about the mobility of nodes, this is mainly because it tries to send content over all possible paths in the network. This scheme therefore achieves the optimal performance in terms of mean delivery delay (MDD) and packet delivery ratio (PDR), simultaneously, it requires more buffer space than other schemes, so as to store the large amount of redundant copies.

This deficiency has motivated researchers to develop other data forwarding algorithms, including PROPHET [10], CAR [11], MobySpace [12], Spray-and-Wait [13] and delegation forwarding [14], etc. For these algorithms, the contexts they used to estimate the relay nodes play a big role in routing performance. We notice that most existing schemes only take the physical contexts (e.g., contact number and duration) into account and neglect the impact of social contexts on network performance. In fact, the network performance depends heavily on human walks [15], since devices may lose connection when people move around. Hence, the social contexts acquired by mobility characterization techniques are of great importance on designing data forwarding metrics.

There exist a few works that explicitly consider some social contexts in opportunistic routing, for example SimBet [16] and Bubble [17]. The two schemes computed node’s centrality/similarity by using traditional social network analysis technology [18], which is time consuming or even impractical in OSN [19]. For instance, SimBet has a high time complexity since it needs to calculate the cube of adjacency matrix [20]. Moreover, as we have shown later, SimBet exploits the growing time window to aggregate node contacts, resulting in a homogeneous issue. On the other hand, considering the fact that the shortest path for each source–destination pair may not exit or vary from time to time in OSN, betweenness centrality adopted in Bubble has to collect the shortest path in an off-line way, which makes it impractical in distributed scenarios.

Considering these facts, we are interested in the following question: Can we explore some other stable social attributes to quantify the centrality and similarity of nodes? By analyzing GPS traces of human walks from the real world, we confirm that there also exist two known phenomena as the indications in [21]. One is that people always move around a set of well popular locations which are called public hotspots, instead of purely random motions. The other is that each person shows preference for some particular locations which are called personal hotspots in this paper. We believe that both kinds of hotspots are more stable than underlying social contexts mentioned above, as public hotspots are formed by superimposing personal hotspots together and personal hotspots/habits are stable over time and across situations [22].

Taking all above issues into account, we design data forwarding metric by exploiting hotspot distribution of nodes. In specific, we investigate the following two kinds of hotspots. (i) The public hotspots: this implies that there exists a bigger chance to meet the destination in these landmarks than other places. In other words, a node frequently wandering among these hotspots can reach more other nodes in the network, which is consistent with the nature of node centrality. Hence, we have to address how to identify these nodes with a higher centrality than others. (ii) The personal hotspots: this implies that if we can deliver content to one of the top $k$ popular personal hotspots of the destination, the content will be quickly received by the destination. As such, we have to answer the problem of how to estimate the similarity between a potential relay and the destination. Besides, since each person has his/her unique mobile profile which we call node personality, we still need to incorporate this factor into the data forwarding process.

In this paper, we develop a novel data forwarding metric, called Hotent (HOTspot ENTropy), to address these challenges. We first use the relative entropy [23] between public hotspots and personal hotspots to evaluate centrality of nodes. Then we utilize the inverse symmetrized entropy [24] of personal hotspots of two nodes to characterize their similarity. Third, we use the law of universal gravitation to integrate the two social metrics. Furthermore, different from the related works, we integrate a new factor, personality, into the Hotent metric and exploit the entropy of personal hotspots to estimate node personality. We also propose a method to ascertain the optimized size of hotspot. Our main contributions can be summarized as follows:

- We observe that the hotspots are bursty and stability. The bursty feature implies that we can decrease the number of hotspots required to exchange, and the stability feature means we can reduce the update frequency of hotspots. Both of them make Hotent lightweight.
- We employ the information entropy theory to compute node’s centrality, similarity and personality. Rather than exchanging neighbor’s adjacency matrix [16] or counting the number of the shortest delay paths [17], we use hotspot entropy to quantify the centrality and similarity of nodes, which guarantees Hotent with a low time complexity.
• We take personality of nodes into account, which makes Hotent predict more accurate than the existing method since each person has his/her own personal habit.
• We exploit the values of Hurst parameter to explore the optimized size of hotspots and reduce the influence of the number of hotspots on the bursty dispersion of traces.
• We conduct extensive experiments to compare Hotent and several state-of-the-art works through two data-sets, experimental results show that Hotent largely outperforms other solutions, especially in terms of packet delivery ratio and the average number of hops per message. For example, it achieves up to a 20% improvement in packet delivery ratio over SimBet and 35% over PeopleRank [25], and has a reduction of the number of hops by up to 5 and 0.67 factors compared to SimBet and PeopleRank respectively under the same conditions.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 first overviews our work and then presents the procedure for identifying hotspots. Section 4 describes our approaches to evaluate centrality, similarity, personality and Hotent metrics. In Section 5, we make a performance evaluation. Finally, we conclude our paper in Section 6.

2. Related work

The related works include the following two aspects: content sharing and data forwarding. We introduce them in detail.

2.1. Content sharing in OSN

With the recent population of hand-held mobile devices, sharing content among those devices becomes important. L. McNamara et al. [1] proposed a user-centric media sharing system. They first selected the best content source by inferring colocation information about fellow commuters from the historical data stored in RFID cards, and then forwarded the content via a hop-by-hop manner. The authors of [2] proposed CoCam, a framework for smart phones that enables uncoordinated real-time multimedia (image and video) sharing between different users. To improve the location recognition accuracy, H. Cheng et al. [3] used multiple smart phones to sense the same object from different positions, and then combined the correlated sensed data for location inference. E. Koukoumidis et al. [4] exploited windshield-mounted phones to opportunistically sense current traffic signals with their cameras, collaboratively communicate and predicate traffic signal schedule patterns. Drivers can then adjust speed so as to avoid a sudden stop.

2.2. Data forwarding in OSN

Routing is the base of content sharing. Several routing schemes have been proposed in the past few years. On the basis of contexts they used, we classify them into the following two categories: (i) data forwarding without social contexts, (ii) data forwarding with social contexts.

2.2.1. Data forwarding without social contexts

**Periodic information based forwarding:** Several schemes utilize the periodic information to forward content in opportunistic social networks. S. Merugu et al. [26] assumed that the global knowledge of node mobility can be predicted over a finite or indefinite time scale, due to the periodicity of node movement [27]. They delivered contents over a space–time routing table with knowledge of when the relay would be encountered. Likewise, S. Jain et al. [28] took a modified Dijkstra algorithm to compute the shortest path between source–destination pair, and then designed routing table based on intermediate nodes along those paths. They presented several schemes and evaluated their performance based on different knowledge oracles acquired from the network. On exploiting past traces of buses to predict their future behavior, the authors of [29] presented MaxProp, which achieves better performance than protocols that depend on proactive knowledge. Besides, the authors of [30] proposed a source routing for opportunistic social networks, they took the expected minimum delay (EMD) as a forwarding metric and applied the Markov decision process to derive the EMD of content at particular moments. Recently, U.G. Acer et al. [31] proposed a simple stochastic model for bus arrivals in quasi-deterministic mobility scenarios. Under this model, they designed an optimal single-copy routing algorithm to maximize the delivery probability.

**Opportunistic information based forwarding:** The deficiency of epidemic scheme has motivated researchers to develop opportunity based data forwarding algorithms (e.g., [10–14]). Most of them make a better tradeoff between packet delivery ratio and the consumption of system resources by taking advantage of different contexts. For these schemes, the routing performance depends heavily on the contexts they used to select the better relay nodes to the destination. For instance, A. Lindgren et al. [10] presented PROPHET, a probabilistic routing protocol for opportunistic scenarios. They exploited the past histories of encounters to predict their future contact probability. Similar to [10], CAR (context aware routing) was proposed in [11], which collected the context information such as the changing rate of node neighbors, and then exploited Kalman filters to estimate the delivery probability. In addition, J. Leguay et al. presented MobySpace [12], a high-dimensional Euclidean space constructed by the past motion patterns of nodes. M. Xiao et al. [32] proposed a time-sensitive utility function for OSNs, where the benefit attached in each content decays over time.
2.2.2. Data forwarding with social contexts

Most aforementioned schemes do not take the social structures into account. However, with the recent popularization of personal hand-held mobile devices, human walks gradually play a critical role in the network performance, since devices may fail to connect with each other when people move around. A few works attempt to discover the underlying stable network structure in real traces by using social networks analysis technology. For example, SimBet [16] exploited centrality and social similarity of ego networks to differentiate nodes. Content will be forwarded to nodes with relatively big SimBet values to increase the probability of finding better relays to the final destination. The authors of [33] investigated the impact of community-structure environment on opportunistic routing performance. They concluded that the controlled replication-based routing schemes worked well at large entropy environments (i.e., random or chaotic mobility scenarios), and link-state protocols only at small entropy environments. P. Hui et al. proposed Bubble [17], which combined node betweenness centrality and community structure to make forwarding decisions. They assumed that each node had a global rank across the whole system and a local rank within its local community. When a content is out of the community of the destination, it is forwarded to the node with a high global rank, when it enters into the range of the destination community, the content is delivered to the node with a high local rank in that community. PeopleRank [25] assigned higher weights to nodes if one or more of their neighbors play big roles in the network, which is inspired by the PageRank idea. Recently, the authors of [7,34] employed the static nodes placed in the system “hot region” to relay contents. If the content entered into the “hot region”, the static node sprayed one replica of the content to any other node it encountered, otherwise, the content was sprayed in a binary way [14].

Different from the aforementioned works, we exploit the information entropy theory to evaluate node centrality and similarity. Furthermore, we take node personality into account [35].

3. Identifying and analyzing hotspot

We introduce the data-sets and give an overview of Hotent in Section 3.1. We give a detailed presentation about the hotspots division and weight computation in Section 3.2. In Section 3.3, we discuss the bursty dispersion and stability of hotspots.

3.1. Data-sets and architecture of Hotent

We use the following two real data-sets gathered by [21] over almost two years (from 2006-08-26 to 2008-04-18), referred to as KAIST and NCSU. The characteristics of these data-sets such as intra/inter-contact distribution have been explored in several studies (e.g., [21,36]) and applied into different scenarios (e.g., message deletion mechanism in [37] and the localization of mobile networks in [38]). Table 1 summarizes their main features.

Fig. 2 presents the architecture of Hotent. It consists of three main phases: (1) identifying hotspots, (2) analyzing hotspot and (3) computing Hotent. We will detail each phase in the following three sections.

3.2. Hotspots division and weight computation

In this section, we first clarify some terms such as GPS log, GPS trajectory, and stay point, and then present our solutions to hotspots division and weight computation.
Table 1
Statistics of collected real traces from the two sites.

<table>
<thead>
<tr>
<th>Site (volunteers)</th>
<th># of trajectory</th>
<th># of stay points (min/avg/max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KAIST (34)</td>
<td>92</td>
<td>511/1589/2800</td>
</tr>
<tr>
<td>NCSU (20)</td>
<td>35</td>
<td>206/1179/2604</td>
</tr>
</tbody>
</table>

GPS log and GPS trajectory: The data collected by the GPS devices carried by participants are form of GPS log, which is a sequence of three-tuples (Timestamp, X-coordinate, Y-coordinate). As depicted in Fig. 3, on a two dimensional plane, we can connect these three-tuples into a GPS trajectory according to their time sequences.

Stay point: A stay point $P$ denotes a physical location where a participant stays more than a threshold. There are two categories of stay points. The first one denotes that a participant remains stationary for a while exceeding the threshold, and the second one denotes that a person wanders around within a certain small spatial region for a time period (in [21], the default values of the threshold and the radius of small region are 30 s and 5 m, respectively). The authors of [39] proposed an algorithm for stay point detection.

Hotspot division: Hotspot division is a well-studied topic in data mining regime [6,40,41]. Given a series of stay points and the deployed region, identifying hotspots can be regarded as a general case of object clustering. That is, if we view each hotspot as a community and each stay point as a node, we can apply Newman’s weighted network analysis [41] for our work. Whereas, it is not practical to take this method due to (1) it is designed for stable topology and (2) its high time and space complexity and the large amount of stay points we analyzed (the number of stay points we employed is close to ten thousands). Considering these facts, we exploit the Hurst parameter of the trajectories to explore this question. The Hurst parameter is used as a measure of bursty dispersion of time series [21]. The trajectories are said to be bursty dispersion (and therefore, self-similarity) if the Hurst value is greater than or equal to 0.5.

To compute the Hurst value, we first divide the deployed region into lots of small grids and each of them has an area of $d$ by $d$, we then compute weights of grids according to Eq. (1) (see the section “weight computation”). After that, we measure the aggregated variances in these weights and use polynomial fit (polyfit) to estimate the absolute slope $\psi$. The value of the Hurst parameter is thus $1 - \phi/2$.

The maximum Hurst value: Note that the choice of the value of $d$ has a number of implications. Making it too big can merge some original grids into a over-sized hotspot, while making it too small can split an original hotspot into some ordinary grids. That is, it directly impacts the weight of each grid, and indirectly influences the Hurst value. Since our goal is to reveal the bursty dispersion of human mobility, which is measured by the Hurst parameter. The bigger the Hurst value is, the more the bursty dispersion would be. That is, we should find the maximum Hurst value so as to show the burstiness degree of the traces. Considering the relationship between the Hurst value and the value of grid size $d$, it naturally exploits the maximum Hurst parameter to ascertain the “exact” value of $d$, where the “exact” value of $d$ denotes that we can get the maximum Hurst parameter if we set this value. Mathematically, let $D$ denote the set of $d$, let $H$ denote the Hurst parameter of trajectories and function $f$ denote the mapping from $D$ to $H$, we have $f : D \to H$, and there exists $d_{\text{exact}} \in D$, such that $h_{\text{max}} = \max(f)$, where $h_{\text{max}} \in H$. We present the above procedure in Algorithm 1.

We increase the grid size to observe its influence on Hurst parameter. Fig. 4 shows the results under the two scenarios. Although there is no significant linear relationship (e.g., monotonically increasing/decreasing) between the length of $d$ and the Hurst parameter, it clearly shows that the choice of $d$ has a big effect on the values of Hurst parameter. For example, at KAIST (Fig. 4(a)), the smallest Hurst value is near to 0.45, while the biggest one is 0.7, i.e., it may impair the bursty dispersion in these trajectories as the Hurst value may be smaller than 0.5 if we randomly choose a value of $d$.

Weight computation: As stated above, we divide the two scenarios by non-overlapping $d$ by $d$ grids, each grid indicates one hotspot. We use the weight of each hotspot to denote its popularity. The larger the weight value is, the more popular the hotspot is. There are several methods to estimate the weight of hotspots [42], we here take a simple but efficient solution, called count process. We count the number of stay points within each hotspot and then compute the weight of each hotspot by normalizing the sampled count.

Let $K$ denote the total number of hotspots in the network, let $n_i$ denote the number of stay points in hotspot $i$ and $w_i$ denote the weight of that hotspot, we have:

$$w_i = \frac{n_i}{\sum_{i=1}^{K} n_i}.$$  \hspace{1cm} (1)
Algorithm 1 Computing the maximum Hurst value

1: Input: a series of stay points and the deployed region with an area of $L \times L$
2: Output: the biggest Hurst value $h_{\text{max}}$
3: $D \leftarrow \emptyset$, $H \leftarrow \emptyset$
4: for $d = \text{initial value}; d \leq L/2; d = d + \text{step}$ do
5:   $W \leftarrow \emptyset$ (the set of grid weights)
6:   dividingGrids($d$)
7:   for each grid $i$ do
8:     $w_i = \text{computingWeight}(i)$
9:     $W = W \cup \{w_i\}$
10:   end for
11:   aggregatedVariance$(W)$
12:   $|\varphi| \leftarrow \text{polyfit}(), h \leftarrow 1 - |\varphi|/2$
13:   $H = H \cup \{h\}, D = D \cup \{d\}$
14: end for
15: $h_{\text{max}} = \max(H)$

Similarly, let $n_{ji}^p$ denote the number of the $i$th person’s stay points in $j$th hotspot and $w_{ji}^p$ denote the weight of that hotspot influenced only by the $i$th person, we have:

$$
w_{ji}^p = \frac{n_{ji}^p}{\sum_{j=1}^{K} n_{ji}^p}.
$$
(2)

An online approach to identify public hotspots: Note that it is not possible for a user to acquire a global knowledge (e.g., the public hotspots) in opportunistic scenarios. We here exploit the aggregated personal hotspots to identify the weight of public hotspots. That is, each node carries a hotspot matrix $M_{N \times K}$ with initial elements $m_{ij} = w_{ji}^p$ and 0 otherwise (where we take node $i$ as an example and $N$ is the number of nodes). When two nodes meet up, they first exchange the hotspot matrix $M$. After that, they estimate the weight $w_j$ of the $j$th public hotspot by summing the elements in $V_j$, where $V_j$ is the $j$th column of $M$. Finally, they normalize each $w_j$ and use them to compute the betweenness centrality (please refer to Section 4.4). We now analyze the time threshold $T$ used to fill the hotspot matrix $M_{N \times K}$. Take node $A$ as an example, we give the following upper bound on the time threshold.

**Theorem 1 (The Upper Bound on the Time Threshold $T$).** Let $\beta$ denote the average contact ratio between nodes. We have

$$
T \leq \log(N)/\beta.
$$
(3)

**Proof.** Suppose any two nodes exchange the matrix $M$ only once. The swap process therefore can be represented by a fully inverse binary tree as shown in Fig. 5, where $L$ denotes the depth of the binary tree.

Considering the fact that nodes in the same layer exchange the matrix in parallel, the collection duration depends on the number of swap between node $A$ and other nodes, which equals $L - 1 = \log(N)$ (recall that $N = 2^{L-1}$). Multiply the intratime between nodes $1/\beta$, the collection duration is $\log(N)/\beta$. Now we relax the assumption and permit any two nodes to
exchange the matrix multiple times as we discussed in the above section, the collection process should be further shortened. That is, \( \log(N)/\beta \) is the upper bound of \( T \).

Take the KAIST scenario as a sample, we compute the average collection duration of nodes. The values of \( N \) and \( \beta \) at KAIST are 90 and 0.0045, respectively, the upper bound on the average collection duration is therefore 1443 s. As we have shown later in simulation setup, we use the first \( 1/10 \) of traces (1500 s) as the learning phase. We think this duration is enough long to complete the matrix \( M \).

3.3. Bursty Dispersion and Stability of Hotspots

The phenomenon of bursty dispersion (i.e., self-similarity) of hotspot implies that people always tend to swarm near to a few very popular locations, which means we can characterize the individual trajectory with these particular landmarks. In addition, the size of control packets will be reduced considerably if nodes swap little hotspot information. The stability feature means we can reduce the update frequency of hotspot. Both of them make Hotent lightweight.

Bursty Dispersion of Hotspots: We first analyze the bursty dispersion of public hotspots. The bursty of public hotspots means that popular locations become more popular as individual bursty trajectories are superimposed together. Fig. 9 presents the distribution of public hotspots of the two scenarios, which has a clear bursty pattern and coincides with the theory of preferential attachment proposed in [43]. To further reveal this phenomenon, we statistically analyze the complementary cumulative distribution function (CCDF) and the aggregated weights of top \( k \) public hotspots as shown in Figs. 6 and 7. It is clear to see that few grids dominate the total system. For example, at KAIST, about \( 1/90 \) (top 4 hotspots out of 361) hotspots occupy 90% system weight and there exists a super “hotspot” whose weight is close to 40%.

We then analyze the bursty dispersion of personal hotspots. The bursty of personal hotspots implies that individual user spends his/her most time in some special locations consciously or unconsciously. On average, only about 1% and 1.8% hotspots are visited by each participant in the two scenarios, respectively (we here only consider the top \( k \) hotspots whose sum of weights is greater than or equal to 0.9 as shown in Fig. 8, which has at least 90% confidence guarantee, we will analyze the lost information encoded in the rest \( K-k \) hotspots in Section 4.4). Fig. 10 depicts the distribution of personal hotspots of NCSU, which also shows a bursty pattern as that of the public hotspots. Notice that there exist two phenomena in these
figures (Figs. 9 and 10). One is that different people may have different preferred locations, i.e., different personal habits (see Figs. 10(a), (b), the node (ID = 9) spends its most time in one location, while the other node (ID = 29) wanders among several locations), the second is that the bursty dispersion of personal hotspots is more obvious than that of the public hotspots. Both the two phenomena inspire us to estimate the centrality, similarity and personality of people.
Stability of hotspots: In order to verify the stability of hotspots, we set 19 observation time points evenly in the two datasets. At each observation point, we generated the top $k$ hotspots list and computed the ratio of the same top $k$ hotspots as

$$\frac{|H_i^j|}{|H_i|},$$

where $|H_i^j|$ denotes the number of hotspot that is the $j$th hotspot at two consecutive observation points, and $H_i$ denote the set of top $k$ hotspots at observation point $i$. We measured the ratio when $k$ equals 4 and 6, respectively. The average ratios of all personal hotspots are shown in Fig. 11(a). We can see that at most observation points, the average ratio keeps very high (greater than 80%). This result confirms that the top $k$ hotspots are relatively stable.

We further analyzed the weight change of hotspots. The weight change of a hotspot is measured as the absolute value of $w_{i+1}^j - w_i^j$, where $w_i^j$ denotes the normalized weight of $j$th hotspot at observation point $i$. Fig. 11(b) shows the average of the top two hotspots. We see that the weight change is less than 3%. This result verifies that weights of top $k$ hotspots are also very stable.

4. Implementing hotspots into Hotent

We present our solution in this section. In Section 4.1, we explore the centrality of a node. We analyze the similarity between nodes in Section 4.2. In Section 4.3, we present personality. We exploit hotspot entropy to design Hotent in Section 4.4. Finally, we have a theoretic analysis in Section 4.5.

4.1. Centrality

Node centrality reflects the relative importance of nodes in the network (i.e., how popular a person is within a social network). The more important the person is, the bigger the chance to meet other people is. Freeman [44,45] proposed three widely used methods to estimate node centrality, called degree, closeness and betweenness measures.
Degree centrality: Degree centrality is measured as the number of one-hop neighbors of a given node $i$, which reflects the direct relationship between the node $i$ and its neighbors. A node with higher degree centrality means it can directly contact with more other nodes. Degree centrality of node $i$ is counted as:

$$C_D^i = \sum_{j=1, j \neq i}^{N} p_{ij}$$

(4)

where $p_{ij} = 1$ if node $j$ is one of neighbors of node $i$, otherwise, $p_{ij} = 0$.

It is not easy to compute node degree in opportunistic social networks as the number of direct contacts that involve a node is varying from time to time. An optional method is that we can set a time window and count the number of neighbors of nodes within it. However, the issue of the optimal size of time window arises.

Closeness centrality: Closeness centrality shows the “closeness” of a node to all other reachable nodes. Freeman took the reciprocal of the average geodesic length $d(i, j)$ (i.e., the shortest path from node $i$ to all other reachable nodes) to measure it [45]. Closeness centrality of a node reflects the node’s freedom from the network, which is calculated as:

$$C_C^i = \frac{N - 1}{\sum_{j=1, j \neq i}^{N} d(i, j)}.$$ 

(5)

In opportunistic scenarios, it is hard to work out the geodesic length $d(i, j)$, due to the unguaranteed end-to-end path between node $i$ and node $j$.

Betweenness centrality: Betweenness centrality reflects the controlling capability of a node to other nodes, which measures the extent to which a node falls on the shortest path between two other nodes. The higher the betweenness centrality of a node is, the bigger the ability it has to facilitate communication to other nodes within the network is. Betweenness centrality of a node $i$ is computed as:

$$C_B^i = \sum_{j=1}^{N} \sum_{k=1}^{N} \frac{g_{jk}(i)}{g_{jk}}$$

(6)

where $g_{jk}$ is the total number of shortest path between node $j$ and node $k$, and $g_{jk}(i)$ is the number of paths including node $i$.

Obviously, the betweenness centrality is difficult to be evaluated with the increasing number of nodes, due to the high time complexity. Besides, similar to that of closeness centrality, it is more difficult to work it out in opportunistic networks. For example, the authors of [16] used an adjacency matrix $A$ to represent node contacts, which has elements $A_{ij} = 1$ if there has been at least one contact between nodes $i$ and $j$ at any past time and $A_{ij} = 0$ otherwise. The betweenness centrality thus can be estimated as:

$$A^2[1 - A]_{i,j}.$$ 

(7)

Apparently, the matrices will get more and more identical with the contacts aggregation as shown in Fig. 12, which shows the evolution of adjacency matrix $A$ at different time slots (when two nodes meet up each other, they exchange the neighbor list to update the matrix). We can see that elements of $A$ tend to one as time elapses. As a consequence, heterogeneity of nodes cannot be well reflected, which in turn impairs the network performance (please refer to Section 5). On the other hand, if we use the sliding time window as the authors of [17] did, we have to ascertain the optimal size of time window, whereas, answering this question is non-trivial as well. We discuss how to exploit hotspot to solve this problem in Section 4.3.

4.2. Similarity

Similarity reflects the associations between nodes in the network. Sociologists have observed the phenomenon long before, which is called “clustering” in physics, that if two people have one or more common friends, they can also be friends with high probability.
The number of common neighbors between nodes has an important influence on the dissemination speed of messages. When the neighbors of nodes contact each other frequently, the message diffusion process can be expected to take faster than when the association between nodes is weaker. That is, nodes having a stronger association with a given node are good relay candidates for message diffusion to that node. The generalized method exploits some contexts to estimate the degree of association. For example, the authors of [46] took advantage of the mail list to match the relationship between people in real world. The authors of [47] identified the associations between bloggers by analyzing the linking objects existing in the large number of blogs.

However, it is difficult to count the number of common neighbors (or others such as the common mail list items [46] or common linking objects [47]), due to the same reasons discussed in Section 4.1.

4.3. Personality

The aforementioned sections show that different people have different mobile profiles, we here use the term “personality” to reflect the uniquely mobile characteristic (or behavior) of a given person. The famous psychologist G.W. Allport [22] defined the personality as “a general neuropsychic structure unique to the individual with the capacity to render many stimuli functionally equivalent and to instigate and guide consistent (equivalent) forms of adaptive and stylistic behavior”. He suggested that personality characteristics are relatively stable over time and are stable across situations.

The personality mainly includes tendentiousness, complexity, uniqueness, positiveness and stability etc. We believe that the personal hotspots at least reflect the tendentiousness, uniqueness and stability of personality as shown in Figs. 9–11, since public hotspots are superimposed by personal hotspots, and moreover, each person has his/her own personal habit and the personal habit is stable once it is formed. Hence, it is necessary and significative to exploit personal hotspots to make comparisons across people. We next discuss the impact of personality on content diffusion.

In the later section, we use information entropy to evaluate node personality (please refer to Eq. (12), Section 4.4). From the maximum entropy theory, the personality of a node will get the maximum value if the node visits each location with the same probability as shown in Fig. 13(a), where a node regularly roams among five locations and the entropy value is $2.3219 - \log(0.2) = 2.3219$. Fig. 13(b) shows another scenario, where a node always stays in one location (the $L_3$) and seldom visits other locations. The entropy value is $-\log(0.05)*0.2 - \log(0.8) = 1.1219$. Based on the personality of nodes, we classify nodes into two types: roaming and homing. We define a roaming node as a node that belongs to the top $x\%$ nodes with the highest personality in the whole network, and select the last $x\%$ as homing nodes. The roaming nodes intuitively play a big role in content dissemination since they spread the content from one location to another. We validate this intuition by excluding the two types of nodes from the content diffusion process, respectively, and analyze their influence on content dissemination speed. Fig. 14 shows the experimental results. It is clear to see that the roaming nodes have a positive influence on content diffusion, removing them leads almost to 33% increase in content diffusion delay at KAIST (when the top and last 10% nodes are removed, respectively). Even in the very sparse scenario NCSU, the content diffusion delay still increases 10% when we remove the roaming nodes. This phenomenon confirms the efficiency of node personality, i.e., nodes with high personality should be selected as relays. We next discuss how to incorporate this new metric into the Hotent metric.

4.4. Hotent

In this subsection, we use the information entropy theory to compute betweenness centrality, similarity and personality of nodes. More specifically, we utilize the relative entropy between the public hotspots and the personal hotspots to evaluate betweenness centrality of a node, we then exploit the inverse symmetrized entropy of personal hotspots between two nodes to compute their similarity, we finally use the entropy of personal hotspots of a node to estimate its personality.

Let random variable $X_i$ denote the distribution of personal hotspots of node $i$, let random variable $Y$ denote the distribution of public hotspots. We have $Y = w_1, w_2, \ldots, w_k$ and $X_i = w_{p_1}^i, w_{p_2}^i, \ldots, w_{p_k}^i$.

**Betweenness centrality computation:** Relative entropy (also called Kullback–Leibler divergence) can be used to differentiate the divergence between two random distributions. If the relative entropy value equals zero, we call that the two random variables have the same distribution. Therefore, if the distribution of $X_i$ is the same as that of $Y$, we call that node $i$ has the highest betweenness centrality, since it can reach more nodes in the network, i.e., this definition (the relative entropy) is

---

1 Fig. 13 shows the mobility profile of users, which includes two parts. One is the landmarks he/she visits, the other is the visiting frequencies for each landmark. For example, in Fig. 13(b), the user visits five landmarks with a cumulative frequency $0.05, 0.05, 0.8, 0.05$ and $0.05$, respectively.
consistent with the nature of traditional centrality evaluation methods. Let $C'_b$ denote the betweenness centrality of node $i$, we have

$$C'_b = \left( \sum_{j=1}^{k} w^j_{p_i} \log(w^j_{p_i} / w^j_{p_j}) \right)^{-1}. \quad (8)$$

Compared with Eqs. (5) and (6), it is clear to see that our solution is more concise and has a low time complexity $\Theta(k)$, which is only related to the number of top $k$ hotspots and independent of the number of nodes in the network. We evaluate the accuracy of these centrality metrics in Section 5.3.

**Similarity computation:** Due to the nature of the logarithmic function, the relative entropy does not keep symmetry, i.e., the relative entropy of $X_i$ over $X_j$ does not equal to that of $X_j$ over $X_i$. We hence use inverse symmetrized entropy to estimate the similarity $\text{Sim}(i,j)$ between node $i$ and node $j$, we have

$$\text{Sim}(i,j) = \left( \text{Sim}(i/j) + \text{Sim}(j/i) \right)^{-1} \quad (9)$$

where $\text{Sim}(i/j)$ is the relative entropy of node $i$ about node $j$ and $\text{Sim}(j/i)$ is that of node $j$ about node $i$, we have,

$$\text{Sim}(i/j) = \sum_{l=1}^{k} w^l_{p_i} \log(w^l_{p_i} / w^l_{p_j}) \quad (10)$$

$$\text{Sim}(j/i) = \sum_{l=1}^{k} w^l_{p_j} \log(w^l_{p_j} / w^l_{p_i}) \quad (11)$$

**Personality computation:** Let $\text{Per}_i$ denote the personality of node $i$, according to the definition of entropy, we have

$$\text{Per}_i = -\sum_{l=1}^{k} w^l_{p_i} \log(w^l_{p_i}). \quad (12)$$

To make the above equations hardness, let $0 \log 0 = 0$.

**Hotent metric:** The Hotent metric is calculated by integrating the above three components. Hence, the question of selecting the best relay becomes a multi-objective optimization problem. In Section 4.3, we conclude that the personality has a positive influence on content diffusion speed, which shows the necessity for taking node personality into account. We now analyze the necessity for integrating the centrality and similarity. Many social-based solutions utilize node centrality/similarity as the forwarding metric to choose desired relays (see Section 2.2.2). The intuition behind the former is that people have different social status in real world, a good forwarding metric needs to choose nodes with high centrality scores in the network, so as to increase the probability to encounter destination nodes. The intuition behind the latter is that people have different social relationships (e.g., stranger or friend [5]) in real life, there exist also different similarity scores in the underlying network as a reflection. A good forwarding metric is naturally to select nodes having high similarity scores with the destination to speed up the delivery process, instead of purely forwarding copies to each encountered node. Thus the question of how to incorporate them quickly arises, as both of them help to improve the content delivery performance. We here take PeopleRank [25] (based on centrality) and MobySpace [12] (based on similarity) as two samples to illustrate merits and drawbacks of the two social-based forwarding schemes.

Fig. 15 shows the mean delivery delay and normalized cost of PeopleRank and MobySpace under KIAST scenario. We observe two different but expected phenomena. One is that the centrality-based scheme effectively decreases the forwarding
cost while increasing the delivery delay when the message TTL is longer than 10,000 s (about 2/3 of the traces). This is mainly because node centrality follows a power-law distribution in the network, which means that the number of nodes with high centrality is very few. At the same time, it is possible that these nodes have not close relationship with destinations, thus, prolonging the forwarding process. The other is that the similarity-based scheme is not very sensitive to the message TTL in terms of delivery delay, while aggravating the forwarding cost. For example, the delivery delay of similarity-based scheme increases stably rather than sharply as shown in Fig. 15(a), however, the forwarding cost increases almost by 30%. The reason behind this is that there may exist many intermediate nodes having high-similarity scores with destinations, especially in the condition where the initial carrier has a very low similarity with the destination. As a short conclusion, we should integrate the two social metrics by making best use of their advantages and bypassing their disadvantages.

In this paper, we specifically exploit the law of universal gravitation to incorporate the centrality and similarity, using the metaphor of mass for node centrality and distance for the similarity between two nodes. The law of universal gravitation states that “every object in the universal attracts every other object with a force that is directly proportional to the product of their masses and inversely proportional to the square of their distance”. Let \( g_{i,j} \) denote the gravitation between nodes \( i \) and \( j \), we have

\[
g_{i,j} = G \frac{C_i C_j}{\text{Sim}(i, j)^2}
\]

where \( G \) is the gravitation constant. Fig. 16 shows the distribution of gravitation between nodes. We see that there exist a few node pairs that have relatively big gravitation. These nodes can be used to relay content for the destinations. Together with the factor of personality discussed above, we give the Hotent metric as follows.

\[
\text{Hotent}_{i,j} = \text{Per}_i \ast g_{i,j}.
\]

**Hotent routing**: We outline the Hotent routing in Algorithm 2, which shows the communication process between node \( i \) and node \( j \). Take node \( i \) as an example. When it encounters node \( j \), for any content \( C \) that node \( i \) carries, if its destination \( d \)
is node $j$, node $i$ delivers it to node $j$ and removes the content from its buffer. Otherwise, if node $j$ does not hold the content, they swap their own Hotent utility. If $\text{Hotent}_{i,d}$ is smaller than $\text{Hotent}_{j,d}$, node $i$ delivers the content to node $j$ and removes it from $i$'s buffer,\(^2\) i.e., Hotent takes a single copy scheme.

**Algorithm 2** Hotent Algorithm, pseudo-code of node $i$

1. upon meeting up node $j$ do
2. for any content $C$ in $i$'s queue do
3. if $d == j$ then
4. deliverMsg($C$)
5. remove($C$) [deliver and remove the content]
6. else if $C \notin j$ then
7. $i \leftarrow \text{Hotent}_{i,d}$
8. isForwarding($C$) [make forwarding decision]
9. end if
10. end for
11. isForwarding($C$) [compare the Hotent metric]
12. if $\text{Hotent}_{i,d} < \text{Hotent}_{j,d}$ then
13. forwardingMsg($C$)
14. remove($C$)
15. end if

### 4.5. Theory analysis

In this paper, we select the top $k$ grids as hotspots. In real-world, people may spend a few time to visit some ordinary grids. This situation will impact the information we obtain. We next discuss the lost information encoded in these ordinary grids.

The influence on the entropy of the user $i$: Note that there exist $K - k$ ordinary grids and the user $i$ visits them with cumulative probability $w^o_r$, where $o_r$ denotes the id of the $r$th ordinary grid and $w^o_r$ denotes the weight of that grid influenced by the user $i$ (see Section 3.2). Let $\sum_{r \in [1, K-k]} w^o_r = \varepsilon$. We have the following theorem.

**Theorem 2** (The Upper Bound on Lost Information in the Entropy of the User $i$). Let $H(X_i)$ denote the lost information of user $i$ encoded in the ordinary grids. $H(X_i)$ can be bounded by

$$H(X_i) \leq \varepsilon \log((K - k)/\varepsilon).$$

**Proof.** From the definition of entropy, we have $H(X_i) = -\sum_{r \leq K-k} w^o_r \log(w^o_r)$. According to the maximum entropy theory, $H(X_i)$ reaches the maximum when each $w^o_r (\forall r \leq K-k)$ holds the same value $\varepsilon/(K - k)$. We have

$$H(X_i) \leq -\sum_{r \leq K-k} \varepsilon/(K - k) \log(\varepsilon/(K - k))$$

$$= \varepsilon \log((K - k)/\varepsilon). \quad \Box$$

The influence on the relative entropy: Let $R_i$ denote the lost information of the relative entropy of the $X_i$ with respect to $Y$. Let $H(Y)$ denote the lost entropy of $Y$. Suppose $\sum_{r \in [1, K-k]} w^r = \rho$. We have

$$R_i \leq \rho \log((K - k)/\rho).$$

**Proof.** According to the definition of the relative entropy, we have

$$R_i = \sum_{r \leq K-k} w^o_r \log(w^o_r/w^r)$$

$$= \sum_{r \leq K-k} (w^o_r \log(w^o_r) - w^o_r \log(w^r))$$

$$= -H(X_i) + H(X_i, Y)$$

$$\leq H(X_i) + H(Y) - H(X_i) = H(Y)$$

$$\leq \rho \log((K - k)/\rho)$$

where $H(X_i, Y)$ denotes the cross entropy of $X_i$ and $Y$. \(\Box\)

\(^2\) Note that if node $i$ never meets a node with higher Hotent, node $i$ carries the content until it encounters the destination or the content expires.
The influence on the inverse symmetrized entropy: Let $S_{i,j}$ denote the lost information of the inverse symmetrized entropy of $X_i$ over $X_j$. Suppose $\sum_{r \in [1, k-k]} w_{p_j}^{o_r} = \delta$. We have

$$S_{i,j} \leq \varepsilon \log((K - k)/\varepsilon) + \delta \log((K - k)/\delta).$$

(17)

Proof.

$$S_{i,j} = \sum_{r \leq k-k} (w_{p_j}^{o_r} \log(w_{p_j}^{o_r} / w_{p_i}^{o_r}) + w_{p_i}^{o_r} \log(w_{p_i}^{o_r} / w_{p_j}^{o_r}))$$

$$= -H(X_i) + H(X_i, X_j) - H(X_j) + H(X_j, X_i)$$

$$\leq H(X_j) + H(X_i).$$

Based on Theorem 1, we have Eq. (17).

From the results of Eqs. (15)–(17), we have a high confidence that the influence of the ordinary grids on the Hotent performance is little when $\varepsilon$, $\rho$ and $\delta \to 0$. This is always true since people spend most of their time in the landmarks in real life [48].

5. Performance evaluation

5.1. Simulation setup

In this paper, we make a performance evaluation through the aforementioned two real data-sets KAIST and NCSU. In the experiment, the first 1/10 of both traces were used as the initialization period to collect enough stay points, so as to identify and swap hotspot of nodes. After that, each source sends a message/content to a randomly chosen destination and altogether 1000 messages are generated. The communication range is 250 m, a typical value of WiFi, the simulation results are the average over 50 runs for statistical confidence.

We compare three routing protocols (i) SimBet [16]: The state-of-the-art work for social based forwarding scheme; (ii) PeopleRank [25]3: The most recent work that used PageRank idea to estimate node centrality; (3) Hotent: Our newly proposed algorithm. We evaluate their performance taking the following criteria into account.

Cumulative packet delivery ratio (CPDR): This criterion represents the delivery performance of the network in terms of the number of successfully received messages over the totally sent messages. We evaluate the delivery performance of the three algorithms under different message TTLs.

Mean delivery delay: Although delay is tolerant in OSN, a low end-to-end delay is still desirable as long delay means more system resources are occupied for longer periods.

Average number of hops per message: The least hop does not mean the shortest delay in opportunistic social networks, since it is measured as the successful forwarding times of a message until the destination receives it. Whereas, we still try to minimize this criterion due to the two aspects of considerations, the channel interference and battery power. Minimizing the number of hops helps to reduce the probability of channel interference and the consumption of battery power.

5.2. Performance analysis

In this section, we present the benefits of Hotent.

Cumulative Packet Delivery Ratio (CPDR): Fig. 17 illustrates the performance of packet delivery ratio under different message TTLs. We can see that Hotent improves the CPDR under the two scenarios. Compared to SimBet and PeopleRank, it increases the delivery rate by about 20% and 35% at KAIST, and 10% and 15% at NCSU, respectively. This is mainly because Hotent incorporates the personality of nodes. Nodes with high personality regularly wander among the hotspots, which increases the probability to meet the destination.

Mean Delivery Delay (MDD): Looking at the mean delivery delay (Fig. 18). We notice that the Hotent still shows a competitive performance in MDD, remind that it delivers more messages than the SimBet and PeopleRank. Compared to PeopleRank, it reduces the delivery delay by about $3 \times$ at KAIST, and still has a 20% reduction at NCSU, the very sparse scenario. Compared to SimBet, Hotent almost achieves the same delivery delay with in a big TTL traffic scenario as shown in Fig. 18(a), and reduces the latency by about 50% at NCSU. This is mainly because SimBet uses a growing time window technology to aggregate the past contacts between nodes. Hence, the adjacent matrices that nodes carry will quickly become identical in a relatively dense scenario such as the KAIST. As a result, the heterogeneity of the nodes cannot be well reflected [49], which makes SimBet tend to randomly forward the messages. The random walk feature benefits the network performance in a well-connected scenario [50], but it may fail to work in a sparse one.

Average Number of Hops per Message: Fig. 19 illustrates the average number of hops per message. Hotent metric outperforms the other two schemes as well. For example, at KAIST, the average number of hops per message achieved by Hotent

---

3 To have a fair comparison with SimBet and Hotent, we make a slight modification of the PeopleRank protocol. We operate it as a single-copy scheme rather than a multi-copy one.
5.3. Accuracy of different centrality metrics

This section focuses on studying the accuracy of different centrality metrics. We first calculate node centrality based on the methods of hotspot entropy (Eq. (8)), degree (Eq. (4)) and betweenness centrality (Eq. (6)), respectively. We then select
Finally, we analyze the impact of removing central nodes on content diffusion speed. Using this method, we try to indirectly verify which technology can identify the truly influential nodes. Fig. 20 shows the experimental results. We can see that the central nodes computed by hotspot entropy have the largest impact on content diffusion, removing them leads almost to 30% increase in content diffusion delay at KAIST, and 20% at NCSU. This phenomenon confirms the accuracy of hotspot-entropy metric. Another interesting phenomenon is that the classical betweenness centrality does not show the expected results, though it has the highest time complexity. This is mainly because the betweenness centrality needs to collect the shortest paths in advance, and the shortest path for a source–destination pair is time-variant. Furthermore, computing the shortest path is time-consuming in opportunistic scenarios. Both of them make the traditional betweenness metric stale.

5.4. Accuracy of different similarity metrics

We now evaluate the accuracy of different similarity metrics. We exploit the inverse symmetrized entropy (ISE) and cosine angle separation used in MobySpace [12] to compute node similarity, respectively. We then use the two similarity metrics to select the relays and analyze their performance gain in terms of mean delivery delay and normalized cost (the average copies per message/the number of nodes). Fig. 21 shows the evolution of the mean delivery delay in seconds over time for different similarity metrics. ISE performs better when the message TTL is short. For the Cos metric, the average delay improves slightly when we increase the message TTL. Fig. 22 presents the normalized cost of the metrics. ISE significantly improves the cost compared to the Cos metric. It reduces the forwarding cost by almost $7 \times$ at KAIST and $4 \times$ at NCSU, respectively. Both the two results demonstrate that the ISE metric achieves a better trade off between delivery delay and cost since it can accurately identify the relationship among nodes.
6. Conclusion

In this paper, we present a novel routing metric, called Hotent, to share contents in opportunistic social networks. We exploit hotspot entropy to design utility function. We first use the relative entropy between the public hotspots and the personal hotspots to evaluate the centrality of nodes. Then we utilize the inverse symmetrized entropy of the personal hotspots of two nodes to compute the similarity between them. Third, we exploit the law of universal gravitation to integrate the two social metrics. Besides, we use the entropy of personal hotspots of a node to characterize its personality. Trace-driven simulation results show that Hotent largely outperforms other solutions, especially in terms of packet delivery ratio and the average number of hops per message.

Acknowledgments

This work is supported by the Natural Science Fund of China under Grant No. 61332005, the Funds for Creative Research Groups of China under Grant No. 61121001, the Cosponsored Project of Beijing Committee of Education, the Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20120005130002.

References


