



# Trust-Aware Personalized Route Query Using Extreme Learning Machine in Location-Based Social Networks

Yuliang Ma<sup>1</sup> · Ye Yuan<sup>1</sup> · Guoren Wang<sup>2</sup> · Xin Bi<sup>3</sup> · Hongchao Qin<sup>1</sup>

Received: 31 October 2017 / Accepted: 20 September 2018 / Published online: 2 October 2018  
© Springer Science+Business Media, LLC, part of Springer Nature 2018

## Abstract

The task of personalized route query is to find the optimal trip that contains the keywords specified by the query user and satisfies the travel distance constraints. The previous studies mostly focus on collaborative filtering by considering user similarity. Trust is one of the most important factors in decision-making that has been neglected by the existing studies of personalized route query. In this paper, we propose a new type of personalized route query by incorporating trust. We propose a social trust-based optimal trip selection (STOTS) framework for personalized route query. STOTS consists of three key components. The first component predicts social trust based on extreme learning machine (ELM), denoted STP-ELM, that exploits social information and user behavioral patterns as features. In the second component, we propose a novel model to incorporate social trust into personalized route query. Additionally, we propose an index to speed up query processing. In the third component, we propose an optimal route query algorithm called RouteHunter that aims to find an appropriate route satisfying the user-specified constraints. The experiment results show that (1) our social trust prediction approach based on ELM attains superior regression efficiency compared to other traditional methods; (2) our proposed index can efficiently accelerate personalized route query processing; and (3) our route query approach can achieve a better performance than the baseline approach. This paper studies a novel personalized route query incorporating social trust in location-based social networks. We propose a social trust-based optimal trip selection (STOTS) framework that uses ELM to evaluate social trust, applies a ranking model to incorporate social trust, and includes an algorithm to find the required route. Experimental results encouragingly demonstrate the efficiency and effectiveness of our proposed approach.

**Keywords** Extreme learning machine · Personalized route query · Social trust · Location-based social networks

## Introduction

In location-based social networks (LBSNs), users are allowed to provide location data to traditional social networks, fostering several location-based services, e.g., *Foursquare*, *Gowalla*, and *Geolife*. It is very convenient for users to record and share their life experiences with such applications on mobile devices. With advances in location acquisition and wireless communication technology,

location-based social networks are increasingly important in people's everyday life. One of the most important challenges in LBSNs, personalized route query aims to provide an optimal route that passes through the locations specified by the query user and satisfies the travel distance constraints. This topic has recently become popular and has a wide range of applications. In addition, personalized route query has an important application in the field of cognitive computation that helps individuals decide on the optimal route plan.

✉ Ye Yuan  
yuanye@mail.neu.edu.cn

Yuliang Ma  
ylma.neuer@gmail.com

Guoren Wang  
wanggrbit@126.com

Xin Bi  
bixin@bmie.neu.edu.cn

Hongchao Qin  
qhc.neu@gmail.com

<sup>1</sup> School of Computer Science and Engineering, Northeastern University, Shenyang, Liaoning, China

<sup>2</sup> School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China

<sup>3</sup> Sino-Dutch Biomedical and Information Engineering School, Northeastern University, Shenyang, Liaoning, China

Cognitive computation is a multidisciplinary field of research aiming at devising computational models and decision making mechanisms [3, 15, 34, 36, 46]. The objective of cognitive computational models is to endow computer systems with the faculties of knowing, thinking, and feeling.

In the literature on personalized route query, Li et al. [26] proposed the trip planning query (TPQ) to find the shortest path from the starting point to the destination through several venues that can cover the category requirements of the query user. Cao et al. [4] proposed a novel keyword-aware optimal route query (KOR) able to find an optimal route covering a set of user-specified keywords and satisfying a specified budget constraint while optimizing the route's objective score. Zeng et al. [45] studied optimal route searching with the coverage of user preferences. Additionally, most studies have focused on preferred route recommendation [25, 44]. The information provided by an untrusted friend may not influence certain behavioral decisions (personalized route query). We study the personalized route query by incorporating trust that has been neglected by the existing studies. Following the existing work [9], social trust as a probability value of one user relative to another is accepted by the great majority of authors and adopted in our work. Social trust measures the credibility between two persons and has been widely used in providing personalized recommendations [1, 24, 33, 41]. However, most studies focus on social tie recommendations.

In this paper, we propose a new type of personalized route query by incorporating social trust in LBSNs, called *social trust aware personalized route query* (STPRQ). Specifically, given a set of venue categories (such as a restaurant, a gym, and a cafe), STPRQ aims to find a proper route  $R$  from the starting venue to the destination.  $R$  should pass through several venues of the respective categories and be credible and popular in the social circle of the query user. Furthermore, the route should satisfy a road network distance constraint.

As mentioned above, STPRQ is a novel problem that can help individuals make decisions. However, the existing approaches are not suitable for tackling the STPRQ problem. The challenge is threefold. First, the social trust of each pair of users is unknown. Therefore, the first challenge is to evaluate the social trust of each pair of users in the underlying LBSNs. Second, to the best of our knowledge, we are the first to study personalized route query by incorporating social trust. This problem is not trivial. Thus, the second challenge is to incorporate social trust into personalized route query. Third, the existing studies related to personalized route query are unsuitable for addressing the STPRQ problem. A naive solution is to enumerate all possible routes from the starting venue to the destination and subsequently select an appropriate route. Clearly, the

naive solution is time-consuming and inefficient. Therefore, the third challenge is to efficiently search for an appropriate route satisfying the query constraints.

To overcome the above challenges, we design a social trust-based optimal trip selection (STOTS) framework that consists of three key components.

In the first component, we analyze social trust in location-based social networks. Extreme learning machine (ELM) provides a unified learning platform with a widespread type of feature mapping and can be applied in a regression analysis directly [20–22]. Inspired by this platform, we design an approach based on ELM to evaluate social trust effectively. To discover the social trust, we construct features that exploit social information and user behavioral patterns from the following aspects: user profiles, social structure and user behaviors.

In the second component, we design a rank model incorporating social trust into personalized route query that can be leveraged to estimate the credibility of a route. Specifically, we evaluate the popularity score of a venue (a point-of-interest, POI) with respect to a user. Assuming that we know the social trust between users, we can estimate the credibility score of a venue. After that, for a given route, we group venues by category. The sum of the highest ones in each of these groups can be regarded as the credibility score of the route.

In the third component, we propose an advanced search algorithm called RouteHunter to find the preferred route that can satisfy user constraints. RouteHunter adopts a greedy strategy that extends the route from the starting venue. We use a venue ranking scheme that considers the gain from adding a venue into a route. That is, we consider both the category diversity and distance constraint of a route.

Finally, extensive experiments on real-world datasets demonstrate the efficiency and effectiveness of our proposed algorithms.

The remainder of this paper is organized as follows. We begin in “[Related Works](#)”, where we discuss the related studies. We then introduce the preliminaries and problem formulation in “[Preliminaries and Problem Formulation](#)”. We present our framework for the problem of STPRQ in “[Framework](#)” and describe our approach in detail in “[Approach](#)”. We discuss the experiments in “[Experiments](#)” and conclude this paper in “[Conclusion](#)”.

## Related Works

Personalized route query has attracted significant attention in recent years. For instance, Li et al. [26] propose the TPQ query that aims to find the shortest path from the starting point to the destination through several venues that can

satisfy the category requirements of the query user. Cao et al. [4] consider both the user preferences and the budget constraint to find the optimal route. Dai et al. [6] propose a highly expressive personalized route planning query that considers both personalization and the sequence constraint. However, in the existing studies related to personalized route query, the social trust has not received sufficient attention. In this paper, as far as we know, we are the first to study the personalized route query by incorporating social trust. Our proposed problem is different from the problems in the existing studies. Additionally, the previous methods related to personalized route query are not suitable for tackling our proposed problem.

Social trust is a critical factor in decision-making and has recently attracted research interest from many fields, e.g., sociology, psychology, computer science, and cognitive sciences. Studying trust propagation [47], Golback and Hendler [12] propose a trust inference mechanism for assigning trust in Web-based social networks and investigate how trust information can be mined and integrated into applications. Guha et al. [13] develop a formal framework of trust propagation schemes, introducing the formal and computational treatment of distrust propagation. Meanwhile, several studies focus on social trust path selection [27, 39]. Hang et al. [16] propose a social trust path selection method in online social networks, selecting the social trust path with the maximum of propagated trust values as the optimal solution. Liu et al. [27] present a complex social network structure that considers trust information, social relationships and recommendation roles and propose an efficient heuristic algorithm in which multiple foreseen social paths are formed to help social trust path selection.

Several recent studies combine recommendation systems with social trust and reputation mechanisms. For instance, Walter et al. [35] propose a recommendation system that combines the concepts of social networking and trust relationships. Jamali and Ester [23] propose a random walk model combining the trust-based and collaborative filtering approaches to recommendations. It can be observed that although many recommendation systems incorporate social trust, such systems mostly focus on item or user recommendations. To the best of our knowledge, no study has considered personalized route query by incorporating social trust.

### Preliminaries and Problem Formulation

In this section, we briefly review the extreme learning machine (ELM), which is used as a starting point of our proposed approach. Subsequently, we formally define the problem of social trust-aware personalized route query.

### Brief Description of Extreme Learning Machine

Extreme learning machine (ELM), developed by Huang et al. [18–22], is based on a generalized single-hidden-layer feedforward network (SLFN). In ELM, the hidden-layer node parameters are mathematically calculated, instead of being iteratively tuned. Thus, a good generalization performance can be provided at speeds thousands of times higher than those of traditional popular learning algorithms of feedforward neural networks [20]. ELM has been widely applied in various fields, such as theoretical research [31, 43], image analysis processing [7, 38], prediction analysis [28, 29], and classification [8, 14, 40].

Given the activation function  $g(x)$  and  $N$  arbitrary distinct samples  $(\mathbf{x}_i, \mathbf{t}_i)$ , where  $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in \mathbf{R}^n$  and  $\mathbf{t}_i = [t_{i1}, t_{i2}, \dots, t_{im}] \in \mathbf{R}^m$ , a standard SLFN with  $L$  hidden nodes is mathematically modeled as

$$f_L(x) = \sum_{i=1}^L \beta_i g(\mathbf{a}_i, b_i, \mathbf{x}) = o_j \tag{1}$$

where  $\mathbf{a}_i$  and  $b_i$  are the learning parameters of hidden nodes,  $\beta_i$  denotes the output weight connecting the  $i$ th hidden node to the output layer, and  $o_j$  is the  $j$ th output vector of the SLFNs.

Usually, an SLFN can approximate these  $N$  samples with zero error. That is, the error of ELM is  $\sum_{j=1}^L \|o_j - t_j\| = 0$ . Thus, there exist  $\mathbf{a}_i, b_i$  and  $\beta_i$  such that

$$\sum_{i=1}^L \beta_i g(\mathbf{a}_i, b_i, \mathbf{x}) = t_j, j = 1, 2, \dots, N. \tag{2}$$

Equation 2 can be written compactly as

$$H\beta = T \tag{3}$$

where

$$H(a_1, a_2, \dots, a_L, b_1, b_2, \dots, b_L, x_1, x_2, \dots, x_L) = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} g(\mathbf{a}_1, b_1, \mathbf{x}_1) & \dots & g(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \dots & \vdots \\ g(\mathbf{a}_1, b_1, \mathbf{x}_N) & \dots & g(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix}_{N \times L} \tag{4}$$

$$\beta = [\beta_1^T, \dots, \beta_L^T]_{m \times L}^T \tag{5}$$

$$T = [t_1^T, \dots, t_N^T]_{m \times N}^T \tag{6}$$

$H$  is the hidden-layer output matrix of the network. The  $i$ th column of  $H$  is the  $i$ th hidden nodes' output vector with respect to inputs  $(x_1, x_2, \dots, x_N)$ , while the  $j$ th row of  $H$  is the output vector of the hidden layer with respect to the  $j$ th input  $x_j$ .

The activation function  $g(x)$  is infinitely differentiable and is usually designed to be the sigmoid function, the Gaussian function or the radial basis function (RBF). In this

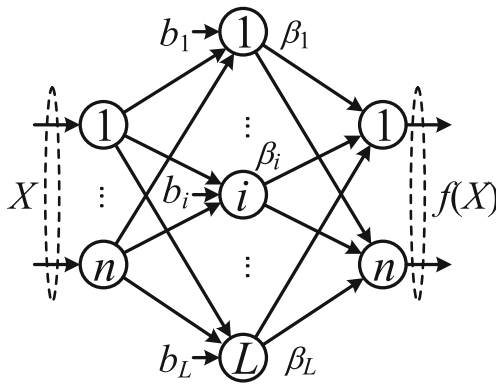


Fig. 1 A single-hidden layer feedforward network

paper, we consider the RBF as the activation function; thus, we obtain

$$g(a_i, b_i, x) = g(b_i \|x - a_i\|), b_i \in R^+ \tag{7}$$

As shown in Fig. 1, the parameters of hidden-layer nodes in ELM, i.e.,  $a_i$  and  $b_i$ , can be randomly selected regardless of the training datasets. The output weight  $\beta$  can be estimated with the matrix computation formula  $\beta = H^\dagger T$ . Therefore, the output function of ELM can be formulated as  $f(x) = h(x)H^\dagger T$ .

**Problem Statement**

**Data Model** As shown in Fig. 2, a location-based social network (LBSN) consists of a *road network* and a *road-social network* (the venues checked-in by users must belong to the road network). We model the *road network* as an undirected graph  $RN(L, E)$ , where  $L$  is the vertex set representing the venues, i.e., POIs, in the road network. Each venue  $l \in L$  has a keyword  $l.\phi$  indicating its category, e.g., restaurant or pub. Each  $e(l_i, l_j) \in E$  is an edge between  $l_i$  and  $l_j$ , indicating that there exists a path between the two venues. Each edge  $e \in E$  is associated with a

positive weight  $w(e)$  showing the road network distance. We consider  $R = (l_1, l_2, l_n)$  to be a route if and only if  $l_i \in L$  and  $(l_i, l_i + 1) \in E$  for every  $i \in 1, 2, \dots, n$ . The venue categories covered by route  $R$  are denoted  $R.\Phi$ . The sum of weights of all edges in a route  $R$  is regarded as the length of  $R$ .

We model the *road-social network* as an undirected graph  $SN(U, C)$ , where  $U$  is the vertex set representing the users in a social network, each  $u_i \in U$  has a check-in list recording the venues the user has visited, and each  $c(u_i, u_j) \in C$  is an edge between  $u_i$  and  $u_j$  indicating that the two users have a relationship, such as friendship or being classmates. As shown in Fig. 2b, the hollow dots are the users of the social network, the solid dots are the locations visited by users of the social network, the solid lines connecting two hollow dots are the social edges indicating that the two users have a relationship, and the blue dotted line connecting a hollow dot and a solid dot means that the user has visited the location, with the number on the dotted line representing the number of visits.

**Query Model** A social trust-aware personalized route query (STPRQ) in LBSNs is described as a tuple  $q = \langle q_u, s, t, \Phi, \Delta \rangle$  where

- $q_u$  is the query user;
- $s$  is the starting venue;
- $t$  is the destination;
- $\Phi$  is a set of venue categories specified by the query user; and
- $\Delta$  is the travel distance constraint.

**Ranking Model** Our goal is to identify a proper route that is credible and popular in the social circle of the query user. Thus, we consider the following aspects to quantify the credibility and popularity of a route.

For the query user, the route popularity reflects the interest score of the route in the user’s social circle. The

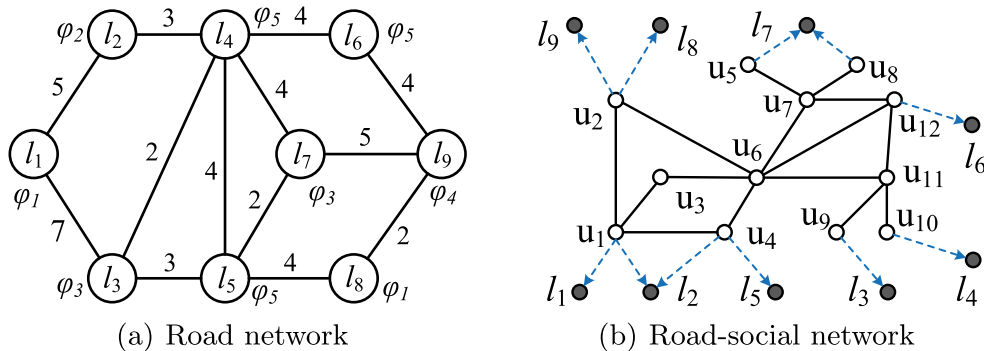


Fig. 2 Graph models of an LBSN

user similarity  $Sim(u, v)$  between users  $u$  and  $v$  can be calculated by the *Jaccard distance*[32]; thus, we obtain

$$Sim(u, v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|} \tag{8}$$

where  $N(v)$  is the neighbor set of user  $v$  in the underlying social network.

A route may pass through multiple venues. Thus, we first describe the measurement of the popularity of a venue  $l_i$  with respect to a user  $v$ , denoted  $p(v, l_i)$ . We have

$$p(v, l_i) = \frac{\sum_{u \in U} Sim(v, u) \cdot count(u, l_i)}{\sum_{u \in U} count(u, l_i)} \tag{9}$$

where  $count(u, l_i)$  represents the number of times  $u$  visited  $l_i$ .

An important measure for social analysis is social similarity. In addition, another important measure is social trust. It is worth noting that social trust is different from social similarity. Specifically, a high social similarity between two users implies that their behavior patterns are similar, even though they may distrust each other. The information provided by an untrusted friend may not influence certain behavioral decisions. In contrast, the information from a trusted friend can be accepted, even if the users' social similarity is very low.

Social trust  $t(u, v)$  is the probability value ( $[0, 1]$ ) between two users  $u$  and  $v$ . For simplicity, social trust is symmetric in this paper. In other words,  $t(u, v) = t(v, u)$ . Evaluation of social trust will be described in “**Social Trust Evaluation**”. Next, we discuss the estimation of the credibility score of a venue for a user. We have

$$c(v, l_i) = \frac{\sum_{u \in U} t(v, u) \cdot p(u, l_i)}{\sum_{u \in U} p(u, l_i)} \tag{10}$$

where  $c(v, l_i)$  denotes the credibility score of  $l_i$  with respect to  $v$ .

Now, we are ready to present the credibility score of route  $R$  that consists of a sequence of venues, i.e.,  $R = \{l_1, l_2, \dots, l_m\}$ . The intuition is that we group the venues in a route by category. Subsequently, we select the groups that match the query categories  $\Phi$ . The highest score in each selected group is added to the credibility score of route  $R$  associated with user  $v$ , denoted  $RS(v, R)$ .

Now, we formally define our social trust-aware personalized route query .

**Problem 1** (Social Trust-aware Personalized Route Query (STPRQ).) Given a road network  $RN(L, E)$ , a corresponding road-social network  $SN(U, C)$ , a query user  $u_q$ , a starting venue  $s$ , a destination  $t$ , a set of keywords  $\Phi$ , and a distance constraint  $\Delta$ , a social trust-aware personalized route query (STPRQ)  $q = \langle u_q, s, t, \Phi, \Delta \rangle$  aims to identify a route  $R$  from  $s$  to  $t$  such that

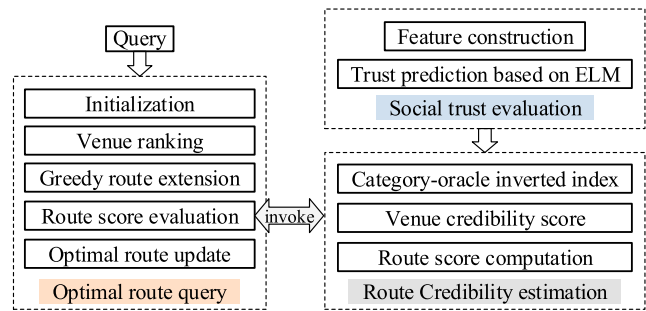


Fig. 3 The framework

1.  $R$  starts from  $s$  and ends at  $t$ , denoted  $R : s \rightsquigarrow t$ ;
2. the length of  $R$  should not exceed  $\Delta$ , i.e.,  $Dist(R) \leq \Delta$ ;
3. the query keyword set  $\Phi$  should be covered by  $R$ , i.e.,  $\Phi \subseteq R.\Phi$ ; and
4. for any other route  $R' : s \rightsquigarrow t$  with  $Dist(R') \leq \Delta$  and  $\Phi \subseteq R'.\Phi$ ,  $RS(u_q, R) \leq RS(u_q, R')$ .

*Example 1* Considering the location-based social networks shown in Fig. 2 as an example, assume an STPRQ  $q = \langle u_1, l_1, l_9, \Phi = \{\varphi_3, \varphi_5\}, 17 \rangle$ , meaning that user  $u_1$  wants to find a route from venue  $l_1$  to  $l_9$ . Additionally, the route must pass through two venue categories, one being  $\varphi_3$  and the other  $\varphi_5$ . The credibility score of the route should be as large as possible, while the length should be no greater than 17.

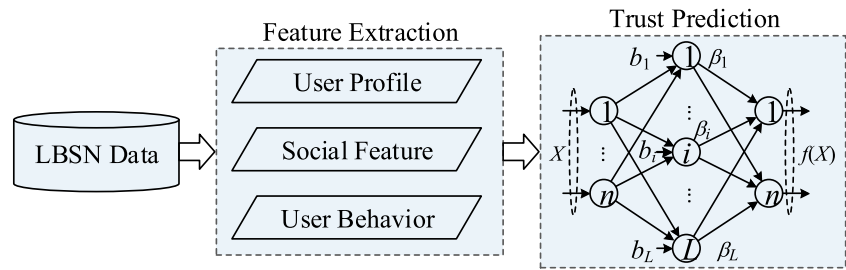
### Framework

When a user proposes a query  $q = \langle u_q, s, t, \Phi, \Delta \rangle$ , our goal is to respond efficiently to the social trust-aware personalized route query. As shown in Fig. 3, we propose a social trust-based optimal trip selection (STOTS) framework that consists of three components: social trust evaluation, route credibility estimation, and optimal route query.

**Social Trust Evaluation** At this stage, we first extract features that exploit social information and user behavioral patterns, including user profiles, social structure, and user behaviors in LBSNs. Subsequently, we propose an approach based on ELM to evaluate social trust between the two users of any social ties in LBSNs. Additionally, we leverage a trust propagation model to evaluate the social trust of any pair of users. It is worth noting that this part can be performed offline.

**Route Credibility Estimation** At this stage, our aim is to estimate the credibility of a route for the query user. As mentioned in “**Problem Statement**”, the route credibility can be obtained by calculating the credibility score of a venue for the query user. Thus, we design a novel index, namely, the category-oracle inverted index that can be used to accelerate the calculation of location credibility.

**Fig. 4** The process of STP-ELM



**Optimal Route Query** When a user proposes a query  $q$ , the venue score can be calculated quickly based on the above two stages; an efficient method will be described to query a proper route in this part. We will illustrate this part in “Optimal Route Query”.

### Approach

In this section, we present the social trust evaluation, detail the estimation of the credibility of a route, and then describe the optimal route query.

### Social Trust Evaluation

Recall that our aim is to discern the social trust value of any two users. There are two cases: in the first, two users are connected by a social edge directly; the second is the opposite case.

**Case 1. Existing direct edge connection** As mentioned above, social trust is a probability value ( $[0,1]$ ) between two users  $u$  and  $v$  implying the creditability between two users. We consider social trust evaluation as a regression analysis problem. To this end, we propose a novel semi-supervised social trust prediction method based on extreme learning machine (ELM), denoted STP-ELM. As shown in Fig. 4, STP-ELM consists of two phases: *feature extraction* and *trust prediction*.

**Feature Extraction** We now describe the extracted features attached to two users connected by a social edge directly. As explained below, various features are summarized from the aspects of user profile, social features, and user behavior.

- 1) User profile. In real life, user profiles have an important influence on the social trust between two users. For instance, two users with the same gender will trust each

other. Specifically, we extract two kinds of features in this class, which are shown in Table 1. Feature  $f_1$  is the absolute value of the difference of two users’ ages. Feature  $f_2$  is the difference of two users’ genders. That is,  $f_2 = 1$  if the two users’ genders are the same; otherwise,  $f_2 = 0$ .

- 2) Social features. In this class, we utilize the *betweenness centrality* [2, 11, 37] that has been proposed to measure the social importance of a user to reflect the social feature. Particularly, a user is likely to be trusted if the user is important in the social network. In the social graph, betweenness centrality measures the fraction of shortest social paths that pass through a given node, averaged over all pairs of node.

Let  $SN(U, C)$  be the graph associated with a road-social network, given three nodes  $u, v, w \in U$ . Brandes et al. in [2] introduces the pair-dependency of  $u$  and  $w$  on  $v$ , demonstrated as Eq. 5:

$$\delta(v) = \frac{\sigma_{uw}(v)}{\sigma_{uw}} \tag{11}$$

where  $\sigma_{uw}$  is the number of shortest social paths from  $u$  to  $w$ , while  $\sigma_{uw}(v)$  is the cardinality of the subset of social paths from  $u$  to  $w$  passing through  $v$ .

The *betweenness centrality* of node  $v$  is obtained as the sum of the pair-dependency values of each pair of nodes on  $v$ . To eliminate the direct computation of all sums, Brandes introduces the dependence of vertex  $u$  on  $v$  as:

$$\delta_{u\bullet}(v) = \sum_{w \in U} \delta_{uw}(v) \tag{12}$$

Thus, the betweenness centrality  $B$  of node  $v$  is obtained by summing contributions from all source nodes:

$$B(v) = \sum_{u \in U} \delta_{u\bullet}(v) \tag{13}$$

**Table 1** Features in user profile class

Feature	Description	Instance
$f_1$	The difference of two users’ ages	$u.age=16, v.age=40, f_1 = 24$
$f_2$	The difference of two users’ genders	$u.gender=f, v.gender=m, f_2 = 0$

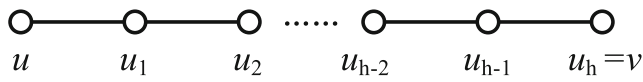


Fig. 5 The process of trust propagation

It is worth noting that we can preprocess the shortest path for every pair of vertices. Subsequently, the betweenness of every node can be computed according to the preprocessed path.

Recall that our aim is to extract features with respect to two users connected by a social edge directly. For a given edge  $e(u, v)$ , we have  $f_3 = B(u)$  and  $f_4 = B(v)$ .

- 3) User behaviors in LBSNs. To predict the score of social trust, we rely on basic intuition. That is, two users who often visit certain venues together in real life may have a high social trust. In most LBSNs, such venue-visiting behavior is denoted as a check-in. Therefore, we gather statistics of common check-ins (generated by the two endpoints of an edge) as the features of this social edge, such as the time of the generated check-ins. In addition, the attributes of venues, such as categories, will be also extracted as features of corresponding edges.

We extract the *five* most visited venues from the common check-in lists of two users. That is, we have *five* features ( $f_5$ - $f_9$ ) in this class. Each feature  $f_i$  can be formulated as a vector  $f_i = \langle C_{imes}, V_{cat} \rangle$ , where  $5 \leq i \leq 9$ ,  $C_{imes}$  denotes the common check-in times, and  $V_{cat}$  denotes the category of the visited venue.

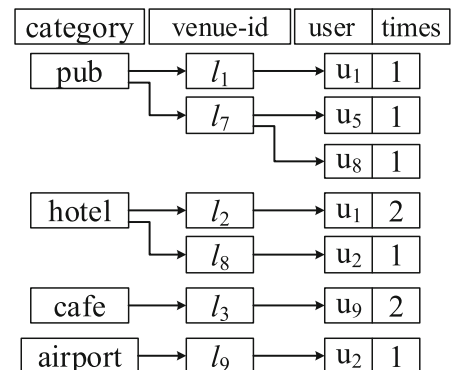
**Trust Prediction** After the feature construction, STP-ELM uses the above three classes of features to learn ELM-based social trust regression models. As mentioned above, STP-ELM leverages the ensemble ELM mechanism to further improve the accuracy of social trust evaluation.

Specifically, STP-ELM learns  $k$  regression models. We obtain partial edges with trust scores between the two endpoints via a crowdsourcing strategy and use these edges as the training data. That is, we use the three kinds of features as input and manually annotate the social trust score of two users with a direct social edge connection.

Fig. 6 An example of the category-oracle inverted index

user	venue-id	category
$u_1$	$l_1$	pub
$u_1$	$l_2$	hotel
$u_9$	$l_3$	cafe
$u_5$	$l_7$	pub
$u_8$	$l_7$	pub
$u_2$	$l_8$	hotel
$u_2$	$l_9$	airport
$u_1$	$l_2$	hotel
$u_9$	$l_3$	cafe
...	...	...

(a) Example of check-in data



(b) Inverted index

Finally, the social trust score of two users is the average value of the social trust value learned from the ensemble of  $k$  ELM-based regression models.

**Case 2. No Direct Edge Connection** We leverage the learned social trust score in Case 1 to infer the social trust value of two users without a direct edge connection. Following the existing work [42], social trust can be propagated through the network. That is, two users are more likely to trust each other if they have a common trusted peer.

As shown in Fig. 5, for any two users ( $u$  and  $v$ ) without a direct edge connection in the underlying social network, we obtain the shortest social path between  $u$  and  $v$ , denoted  $path(u, v) = \{(u = u_0, u_1), (u_1, u_2), \dots, (u_{h-1}, u_h = v)\}$ . Given the social trust value  $t(u_i, u_{i+1})$  of two users in each edge of this path, which would be learned in Case 1, we obtain

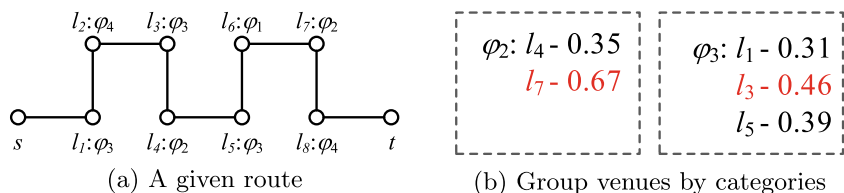
$$t(u, v) = \prod_{(u_i, u_{i+1}) \in path(u, v)} t(u_i, u_{i+1}). \tag{14}$$

**Route Credibility Estimation**

As mentioned in “Problem Statement”, Eqs. 9 and 10 show how to calculate the popularity score and the credibility score, respectively, of a venue for a user, respectively. One important factor is the number of check-ins of a user at a venue. However, the number of social network users increases rapidly, as does the scale of check-in data. It is time-consuming to obtain the check-in information for a venue and the category of a venue by searching the check-in data directly. Inspired by the study [48], we design a *category-oracle inverted index* to quickly retrieve the visiting times of a venue by a user, which can be used to accelerate the processing of route credibility estimation.

Each check-in record has at least three items: the check-in user, the venue id, and the category of the venue. Figure 6a is an example of check-in data in location-based social networks of Fig. 2. For instance, user  $u_1$  has visited venue  $l_1$  once and  $l_2$  twice, while the categories of the two venues are pub and hotel, respectively.

**Fig. 7** An example of route credibility score evaluation



The venues are classified by their categories; each category contains several venues in the road network. The counts of times each venue was visited by various users are summarized as one of the terms of our *category-oracle inverted index*.

The structure of the *category-oracle inverted index* is shown in Fig. 6b. To summarize, a category-oracle inverted index has four primary items:

- Category. The first item in our index is the venue category, which can be used to obtain the venue category of a check-in record directly.
- Venue id. Venue id is the only identifier of a venue, which can be used to obtain more information about the venue.
- Check-in user. This item indicates the user in a check-in record.
- Check-in times. In real life, a venue may be visited by the same user multiple times. The check-in times can reveal the degree of interest of a user in the venue. Thus, we use this item to record the number of times a venue was visited by the same user.

Considering the check-in data in Fig. 6a as an example, there are four categories: pub, hotel, cafe, and airport. Figure 6b shows the category-oracle inverted index, which can illustrate the check-in information intuitively. For instance, there are two hotels, i.e.,  $l_2$  and  $l_8$ , where  $l_2$  has been visited by user  $u_1$  twice and  $l_8$  has been visited by user  $u_2$  once.

A personalized route consists of a series of venues, whose category set should cover the query category set. The intuition for estimating the credibility of a route is that we first group venues in a route by category and subsequently select the groups that match the query category set  $\Phi$ . The

sum of the highest credibility scores of venues in each of these selected groups can be regarded as the credibility score of a route. Here, we first describe the calculation of the credibility score of a venue and subsequently discuss the estimation of the route credibility score.

**Computing the Credibility Score of a Venue** As mentioned in “Social Trust Evaluation”, the social trust between two users  $t(u, v)$  has already been evaluated. The process of calculating the credibility score between a user and a venue is described in Algorithm 1. We first compute the social similarity of any user pair by Eq. 8 and subsequently calculate the popularity score between each user and  $l_i$ . It is worth noting that the popularity score reflects the interest score of a user in venue  $l_i$ . By incorporating the social trust, we compute the credibility score of  $l_i$  with respect to  $v$  by Eq. 10, which incorporates the interest score and the social trust.

**Algorithm 1** Computing the credibility score between user  $v$  and venue  $l_i$

```

Input: LBSNs:  $RN(L, E)$ ,  $SN(U, C)$ , a user  $v \in U$ , a venue  $l_i \in L$ 
Output: The credibility score between  $v$  and  $l_i$ :  $c(v, l_i)$ 
1 for each user pair of any social edge  $(v, u) \in C$  do
2   Compute the user similarity of the user pair by Eq. 8;
3 for each user  $u \in U$  do
4   Compute the popularity score between  $u$  and  $l_i$  by Eq. 9;
5 Compute the credibility score  $c(v, l_j)$  integrating the social trust by Eq. 10;
6 Return the credibility score  $c(v, l_j)$ ;
    
```

**Table 2** Main notations

Notations	Description
$q : u, l_s, l_d, \Phi, \Delta$	the query submit by a user
$Dis^*$	the distance from $l_s$ to $l_d$ in the best feasible solution until now
$Dis$	the distance from $l_s$ to current location
$S$	the keywords set covered by the path $l_s$ to current location
$T$	a path represent by the location matching the category in query $q$
$T^*$	the path of the best feasible solution
$RS^*$	the route score in the best feasible solution until now
$RS(u, T)$	the route score of the path T about $u$
$Q$	a min-priority queue



**Computing the Route Score** In our query model described in “**Problem Statement**”,  $\Phi$  denotes the set of categories specified by the query user. For a route  $R = \{l_1, \dots, l_m\}$ , we denote the category covered by  $R$  as  $RC = \{l_1.c, \dots, l_m.c\}$ . We prune the route  $R$  without computing the route score if  $\Phi \not\subseteq RC$ . That is, we only compute the route score if the route can cover the query category set; thus, in the remainder of this subsection, we discuss this case by default.

Algorithm 2 illustrates the procedure of estimating the route credibility score. First, we group the venues in a route by category. For each group, if the category of venues belongs to  $\Phi$ , we calculate the credibility score of each venue in this group with respect to user  $v$ , and the highest credibility score in this group is added to the route score.

---

**Algorithm 2** Computing the Route Score( $v, R$ )

---

**Input:** A user  $v$ , a route  $R = \{l_1, \dots, l_m\}$ , the query categories set  $\Phi$

**Output:** the score of a route for user  $u$ ,  $RS(v, R)$

- 1 Initialize the route score  $RS(v, R) = 0$ ;
- 2 Group the venues in  $R$  by category  
 $Gr = \{gr_1, \dots, gr_n\}$ ;
- 3 **for** each group  $gr_i \in Gr$  **do**
- 4     **if** the category of this group  $gr_i.c \in \Phi$  **then**
- 5         **for** each venue  $l_i \in gr_i$  **do**
- 6             Calculate credibility score  $c(v, l_i)$  using  
Algorithm 1;
- 7              $RS(v, R) = RS(v, R) + \max_{l_i \in gr_i}(v, l_i)$ ;
- 8 **Return**  $RS(v, R)$ ;

---

Example 2 shows a running instance of how Algorithm 2 operates.

*Example 2* Suppose that the venue category set  $\Phi$  is  $\{\varphi_2, \varphi_3\}$  specified by user  $v$  in an STPRQ. As shown in Fig. 7a, a sample route  $R$  from  $s$  to  $t$  is  $\{l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8\}$ . The category set covered by  $R$  is  $RC = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ . Next, we group the venues in  $R$  by category. The groups with the categories belonging to  $\Phi$  are selected. Subsequently, for each venue in the selected groups, we calculate the credibility score. We sum the highest venue credibility scores in each selected group to obtain the credibility score of the route. As shown in Fig. 7b, assuming that the venue credibility score of each venue in the  $\varphi_2$  and  $\varphi_3$  groups has already been calculated, we obtain the credibility score  $RS(v, R) = 0.67 + 0.46 = 1.13$ .

## Optimal Route Query

As the route score for the query user can be computed and a category-oracle inverted index has been constructed, we are now ready to present our proposed optimal route query algorithm called RouteHunter. The notations used throughout “**Optimal Route Query**” are listed in Table 2.

The primary idea of RouteHunter is that we evaluate the credibility score of route  $R$  if and only if the venue categories covered by  $R$  can contain the query category set  $\Phi$ . Furthermore, for a given route  $R$ , we only need to calculate the credibility score of a venue if its category belongs to  $\Phi$ . That is, assuming that the query user arrives at venue  $l_i$  from the starting venue, we calculate the credibility score of  $l_i$  for the user if and only if the category of  $l_i$  matches one of the categories specified by the user, i.e.,  $l_i.\varphi \in \Phi$ .

In the route extension phase, we traverse the neighborhoods of a venue; a neighbor  $l_j$  can be regarded as a candidate only if the sum of the distance from the start location  $s$  to the current location  $l_i$ ,  $\omega(e(l_i, l_j))$ , and the shortest distance from  $l_j$  to the destination  $l_d$  is less than the distance constraint  $\Delta$ . If the sum exceeds  $\Delta$ , the route from  $s$  to the neighbor  $l_j$  does not satisfy the requirement of the query user. By applying this approach, many venues can be pruned. It is worth noting that we precompute the shortest distance between each venue and  $s$  (and  $t$ ) in the underlying road network.

As shown in Algorithm 3, for a query  $q = \langle q_u, s, t, \Phi, \Delta \rangle$ , RouteHunter first initializes the parameters (lines 1 and 2) and assigns  $s$  as the first element of a min-priority queue  $Q$  (line 3) used to maintain the candidate venues satisfying the distance constraint. Next, we select the optimal venue by the **Q.Ranking** function in line 5 that ranks venues in the following order: we denote  $l_k < l_t$  if and only if  $|p(l_s \rightarrow l_k).\varphi| > |p(l_s \rightarrow l_t).\varphi|$  **or**  $[|p(l_s \rightarrow l_k).\varphi| = |p(l_s \rightarrow l_t).\varphi| \text{ and } RS(u, p(l_s \rightarrow l_k)) \geq RS(u, p(l_s \rightarrow l_t))]$  **or**  $[|p(l_s \rightarrow l_k).\varphi| = |p(l_s \rightarrow l_t).\varphi| \text{ and } RS(u, p(l_s \rightarrow l_k)) = RS(u, p(l_s \rightarrow l_t)) \text{ and } Dist(p(l_s \rightarrow l_k)) < Dist(p(l_s \rightarrow l_t))]$ . Subsequently, we extend the selected venue by applying the category cover strategy, i.e., we prefer to extend the venue with category belonging to the query category set (lines 6–8). In each extension, we compare the distance lower bound of the current partial route to the query distance constraint (lines 15–19). When the destination  $t$  emerges in a route  $T$ , we check whether this route satisfies the query category constraint and the distance constraint and subsequently evaluate the route score and update the optimal route obtained thus far (lines 9–14).

**Algorithm 3** RouteHunter

```

Input: A query  $q = \langle q_u, s, t, \Phi, \Delta \rangle$ 
Output: A best route  $R$ 
1 Initialize a min-priority queue  $Q$ ;
2  $Dis^* = \infty, Dis = 0, S = NULL, RS^* = 0, RS = 0,$ 
    $T^* = \emptyset, T = \emptyset;$ 
3  $Q.enqueue(s);$ 
4 while  $Q$  is not empty do
5    $l_i = \mathbf{Q.Ranking}();$ 
6   if  $l_i.\varphi \in \Phi$  then
7      $S = S \cup l_i.\varphi;$ 
8      $T = T \cup l_i;$ 
9     if  $t \in T$  and  $\Phi \subseteq S$  and  $Dis < \Delta$  then
10      Evaluate the route score  $(q_u, T)$  by
        invoking Algorithm 2;
11      if  $RS(T) > RS^*$  then
12         $RS^* = RS(T);$ 
13         $T^* = T;$ 
14        Continue;
15   for each edge  $(l_i, l_j)$  do
16      $Dis = Dis + dist(l_i, l_j);$ 
17     if  $Dis + \delta(l_i, t) > \Delta$  then
18       Continue;
19      $Q.enqueue(l_j);$ 
20 if  $T^* \neq \emptyset$  then
21   Return  $R = T^*;$ 

```

*Example 3* Considering the location-based social networks in Fig. 2 as an example, the category-oracle inverted index are shown in Fig. 6. Assuming a query  $q = \langle u_1, l_1, l_9, \Phi = \{\varphi_3, \varphi_5\}, 15 \rangle$ , the detailed steps of RouteHunter are shown in Fig. 8. We first extend the starting location  $l_1$ ; locations  $l_2$  and  $l_3$  are the neighbors of  $l_1$ . For each neighbor  $l_i$ , we examine the sum of  $w(l_1, l_i)$  and the shortest distance from  $l_i$  to the destination  $l_t$  that we have obtained using the well-known Floyd algorithm [10]. If the sum exceeds the distance constraint  $\Delta$ , we can prune this partial route from  $l_1$  to  $l_i$ , as the lower bound of such a partial route with respect to distance has violated constraint  $\Delta$ .

In this example, as the distance lower bound for both partial routes from  $l_1$  to  $l_2$  (or  $l_3$ ) satisfies the distance constraint, we put both of them into min-priority queue  $Q$ ,

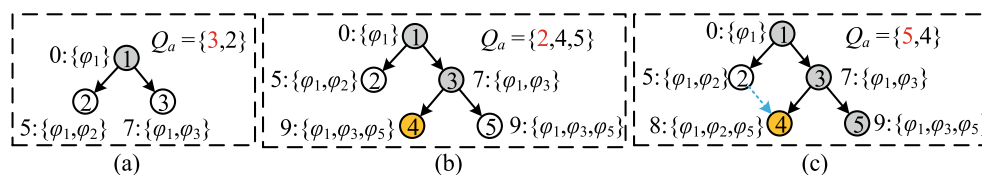
**Table 3** Accuracy evaluation for EN-SVM and STP-ELM

Dataset	Algorithm	Training accuracy	Testing accuracy
Gowalla	EN-SVM	0.668	0.674
	STP-ELM	0.693	0.701
Brightkite	EN-SVM	0.726	0.733
	STP-ELM	0.745	0.758
Foursquare	EN-SVM	0.731	0.752
	STP-ELM	0.738	0.750

calculate the interest score for all locations in  $Q$  (we rank them by the Rank function). After each iteration, we record the categories covered by the respective partial routes. As the ranking order of  $l_3$  is superior to that of  $l_2$ , we extend  $l_3$  in the second iteration. The locations  $l_4$  and  $l_5$  can be reached via  $l_3$ ; as the distance of the partial route  $\{l_1, l_3, l_4\}$  is 9 and the shortest distance from  $l_4$  to the destination  $l_9$  is 8, the sum is 17, exceeding the distance constraint 15; hence, this partial route would be pruned. In Fig. 8b,  $l_4$  is marked by a yellow color. In contrast, we put  $l_5$  into the priority queue  $Q$ , calculate the interest score and rank those locations in  $Q$ . After that, we check the categories covered by the least partial route to determine whether these categories contain the categories specified by the query user, i.e.,  $\Phi = \{\varphi_3, \varphi_5\}$ . If the answer is positive, this means that a feasible solution has been obtained. For instance, in Fig. 8c, location  $l_5$  meets the constraint of the lower bound being less than 15, and the categories covered by the partial route contain  $\Phi = \{\varphi_3, \varphi_5\}$ . Subsequently, we continue to extend the location with the greatest priority in  $Q$  until  $Q$  becomes empty. In this example, we extend location  $l_2$  to  $l_4$ , checking that the distance of the partial route  $\{l_1, l_2, l_4\}$  is 8, and the distance lower bound of this partial route exceeds the constraint 15; hence, we terminate the RouteHunter algorithm. The partial route  $\{l_1, l_3, l_5\}$  merged with the shortest path from  $l_5$  to  $l_9$  is the query answer to be returned.

**Experiments**

In this section, we experimentally evaluate the effectiveness and efficiency of our proposed algorithms for the STPRQ problem. We perform a series of sensitivity tests



**Fig. 8** An example of how RouteHunter operates

**Table 4** Performance evaluation for EN-SVM and STP-ELM

Dataset	Algorithm	Training time (s)	Testing time (s)
Gowalla	EN-SVM	7.6325	7.3047
	STP-ELM	1.2464	1.0358
Brightkite	EN-SVM	5.8351	5.1563
	STP-ELM	1.0193	0.9836
Foursquare	EN-SVM	6.2356	5.9734
	STP-ELM	0.8460	0.8231

to study the impact of query parameters with real social graph datasets. In the following, we first describe the experimental settings and subsequently analyze the experimental results.

## Experimental Settings

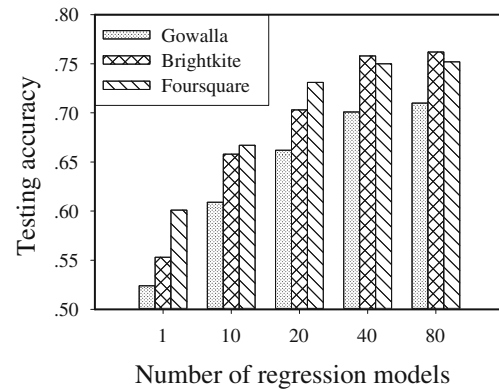
**Dataset** We use three real-world datasets, namely, Gowalla, Brightkite and Foursquare. Table 2 provides the details.

- Gowalla. The Gowalla dataset, available at the Stanford Large Network Dataset Collection<sup>1</sup>, contains data generated worldwide from February 2009 to October 2010. This dataset contains 6,442,892 check-ins generated by 196,591 users at 1,280,969 venues. Additionally, there are 950,327 social edges and 1,356,399 road network edges.
- Brightkite. The Brightkite dataset is available at the Stanford Large Network Dataset Collection<sup>3</sup>. Brightkite was once a location-based social networking service provider with users sharing their locations by checking-in. This dataset contains 4,491,143 check-ins generated by 58,228 users at 772,789 venues. Additionally, there are 214,078 social edges and 1,025,364 road network edges.
- Foursquare. We crawl the Foursquare dataset via the Foursquare API<sup>2</sup> from November 2014 to January 2016. This dataset has 76,503 users and 1,531,357 social edges. For each user, we collect the check-ins during this period of time. Each check-in is a pair of user and venue. In total, we obtained 299,995 venues located in Singapore and 969,549 check-ins. Each venue has its category type description, such as museum, bar, and hotel. In addition, we leverage the OpenStreetMap<sup>3</sup> to obtain a road network of Singapore. It is worth noting that we regard the venues as the vertices of the road network. Thus, the number of road network edges is 416,723.

<sup>1</sup><http://snap.stanford.edu/data/index.html>

<sup>2</sup><https://developer.foursquare.com/>

<sup>3</sup><http://www.openstreetmap.org>

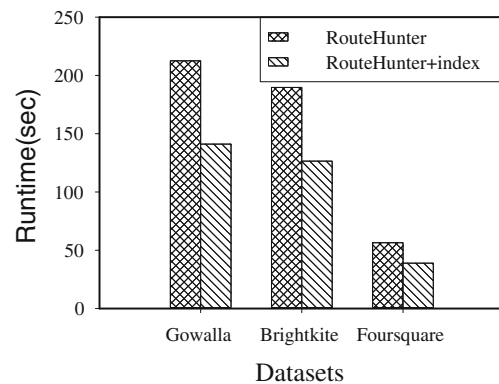


**Fig. 9** Testing accuracy vs. the number of regression models

**Comparative Analysis** In this paper, we conduct three sets of experiments. First, we test the performance of our proposed social trust regression analysis method based on extreme learning machine. Second, we test the performance of our proposed index, including the effectiveness of our index, the index construction time and the index space cost. Third, we provide an overall evaluation of our proposed algorithms, including the route query precision analysis and the effect of parameter  $\Phi$ .

## Experiment 1: Evaluation of Social Trust Regression Analysis

As discussed in “**Social Trust Evaluation**”, a plurality voting method using multiple regression models is used to increase the social trust prediction accuracy. To test the performance of STP-ELM (illustrated in “**Social Trust Evaluation**”), we consider ensemble SVM (support vector machine) [5, 17], denoted EN-SVM, as the comparison algorithm, which uses multiple SVM regression models to obtain a better predictive performance. We set 40 as the default value of the number of regression models in both EN-SVM and STP-ELM.



**Fig. 10** Effectiveness of the index

**Table 5** Index construction time and memory cost

Dataset	Time cost (s)	Memory cost (MB)
Gowalla	876.7	1135.2
Brightkite	627.4	977.6
Foursquare	239.5	640.9

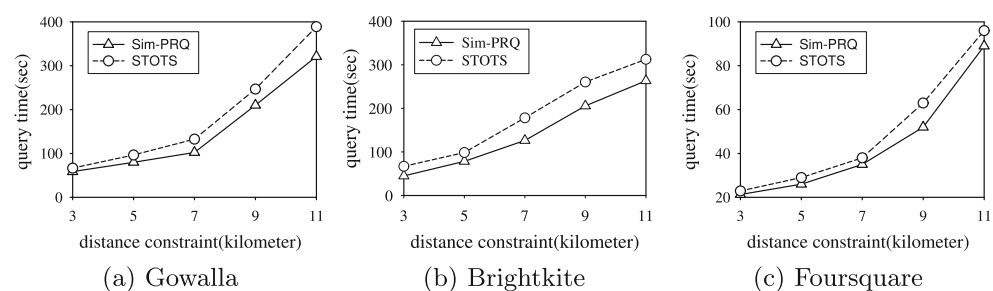
Table 3 shows the accuracy rate of EN-SVM and STP-ELM applied to the three datasets. Table 4 shows the efficiency of EN-SVM and STP-ELM. The above two sets of experiments show that our proposed STP-ELM algorithm can obtain a comparable accuracy yet needs less training time and testing time than EN-SVM.

In Fig. 9, the effect of the number of used regression models is investigated. The testing accuracy varies similarly for the three datasets. It can be observed that the accuracy of the STP-ELM algorithm increases with the number of regression models. The reason is that the accuracy of social trust regression analysis can be improved by increasing the number of regression models. For the number of regression models ranging from 40 to 80, the accuracy remains relatively stable. Therefore, we set 40 as the default value of the number of regression models in the following experiments.

## Experiment 2: Evaluation of the Index

In this subsection, we first test the effectiveness of our proposed index. As mentioned in “[Optimal Route Query](#)”, our optimal route query algorithm, RouteHunter, leverages the constraints specified by the query user to reduce the search space, and the heuristic extends from the starting venue. We consider RouteHunter without our index as the baseline and denote by “RouteHunter+index” the algorithm version with the index described in “[Route Credibility Estimation](#)”.

In this set of experiments, we set the distance constraint  $\Delta$  to be 7, randomly select the starting and ending venues, and assign the size of the venue category set to be 3. As shown in Fig. 10, the average runtime of 50 queries for algorithm “RouteHunter+index” is less than that of the route query algorithm without the index (RouteHunter). It can be concluded that our proposed index can speed up the query processing.

**Fig. 11** Query time

Next, we examine the index construction time and index space cost. The results are shown in Table 5. It can be observed that the index construction for the Gowalla datasets needs much more time and space than that for the Foursquare datasets. The reason is that our proposed index is a venue category-aware inverted file implying that each term in our index is a check-in record. The index construction time and space cost are proportional to the size of check-in data.

## Experiment 3: Performance of the Query Algorithm

In this paper, to the best of our knowledge, we are the first to study the personalized route query by incorporating social trust. Our proposed problem is different from the problems in the existing studies. Additionally, the previous methods related to personalized route query are not suitable for tackling our proposed problem. Thus, we implement a baseline algorithm that only considers social similarity instead of social trust, called similarity-based personalized route query (*Sim-PRQ*). It is worth noting that our proposed approach, *STOTS*, not only considers the user similarity but also incorporates the social trust into personalized route query. By comparing it to *Sim-PRQ*, we will evaluate the performance of *STOTS* in this subsection.

**Efficiency of Methods** In this set of experiments, we compare our *STOTS* approach to the baseline method (*Sim-PRQ*) by considering the run time under various values of the travel distance constraint  $\Delta$ . Note that both methods are able to obtain the optimal solution. Figure 11 shows that the run time of both methods increases quickly with  $\Delta$ . The reason is that a query with a larger travel distance constraint  $\Delta$  is likely to have more potential routes.

**Effectiveness of Methods** Note that any route can be represented by a sequence of venues (POIs). From user history check-in data, we extract successive check-in behaviors that can provide venue sequences. We consider such venue sequences as route test sequences. For all datasets, we extract 1000 test sequences with the length of six. For each venue sequence with length six, the size of the category set in a generated query is *four* after subtracting

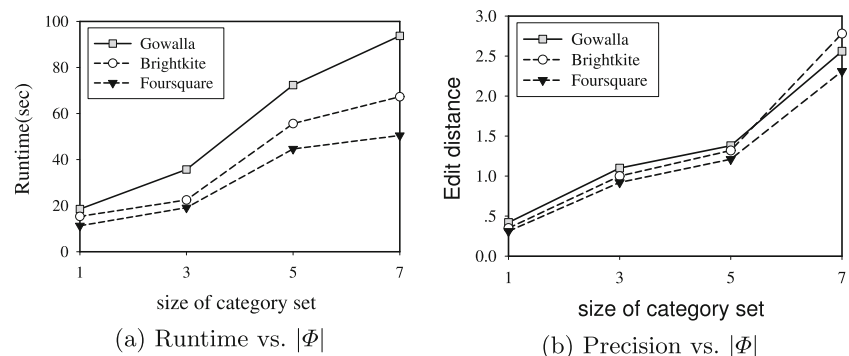
**Table 6** Average edit distance

Datasets	Approach	Edit distance
Gowalla	Sim-PRQ	1.78
	STOTS	1.21
Brightkite	Sim-PRQ	1.73
	STOTS	1.06
Foursquare	Sim-PRQ	1.16
	STOTS	1.03

the starting venue and the destination. We measure the difference between the routes generated by our algorithms and the test sequences to evaluate the route query accuracy. Thus, the edit distance is applied as the evaluation metric, as it measures the distance between two sequences in terms of the minimum number of edit operations required to transform one sequence into the other [25, 30].

Table 6 shows the performance of the two compared approaches, i.e., Sim-PRQ and STOTS, for all three datasets. The results show that our proposed approach STOTS attains a higher accuracy (i.e., a lower edit distance) than Sim-PRQ. The reason is that STOTS not only considers the user similarity but also incorporates the social trust into personalized route query. In contrast, the Sim-PRQ approach only considers the user similarity.

**Effect of Parameter  $\Phi$**  In this set of experiments, we study the accuracy and the run time of our STOTS algorithm under various sizes of the query category set  $\Phi$ . Figure 12 shows the results. In our experience, the size of the category set specified by query users is usually small. We set  $|\Phi|$  to values of 1, 3, 5, and 7. We observe that the run time of the STOTS algorithm increases quickly with  $|\Phi|$ . The reason is that a query with a larger  $|\Phi|$  will cause our algorithm to converge more slowly. We also observe that the edit distance increases with  $|\Phi|$ , indicating that the accuracy decreases with the increase in  $|\Phi|$ . A larger  $|\Phi|$  will introduce more inexact factors into the query result, reducing the accuracy.

**Fig. 12** Effect of parameter  $|\Phi|$ 

## Conclusion

Personalized route query is an important topic due to the growing popularity of location-based social networks and related applications on mobile devices. However, the existing studies did not adequately explore the trust between users in the context of personalized route query. In this paper, we propose a novel personalized route query considering the factor of social trust. We note three major challenges in solving such personalized route query incorporating social trust (STPRQ), namely, evaluating social trust, incorporating social trust, and finding the required route. Therefore, we propose the social trust-based optimal trip selection (STOTS) framework that uses extreme learning machine (ELM) to evaluate social trust, applies a rank model to incorporate social trust, and includes an algorithm to find the required route. Experimental results on a real-world dataset encouragingly demonstrate the efficiency and effectiveness of our proposed approach.

In our future work, we plan to study the evaluation of social trust in location-based social networks. Another direction of our future research is to seek other approximate algorithms for solving this new problem.

**Funding Information** This research is partially funded by the National Natural Science Foundation of China (Grant Nos. 61572119, 61622202, 61732003, 61729201, 61702086, and U1401256) and the Fundamental Research Funds for the Central Universities of China (Grant Nos. N150402005 and N171904007).

## Compliance with Ethical Standards

**Conflict of interest** The authors declare that they have no conflicts of interest.

**Informed Consent** Informed consent was obtained from all individual participants.

**Human and Animal Rights** This article does not contain any studies involving human participants and/or animals by any of the authors.

## References

1. Adwereboamah J, Hufstедler S. Predicting social trust with binary logistic regression. *Research in Higher Education Journal*. 2015;27:6.
2. Brandes U. A faster algorithm for betweenness centrality. *J Math Sociol*. 2001;25(2):163–177.
3. Cao K, Wang G, Han D, Ning J, Zhang X. Classification of uncertain data streams based on extreme learning machine. *Cogn Comput*. 2015;7(1):150–160.
4. Cao X, Chen L, Cong G, Xiao X. Keyword-aware optimal route search. *Proceedings of the VLDB Endowment*. 2012;5(11):1136–1147.
5. Cortes C, Vapnik V. Support-vector networks. *Mach Learn*. 1995;20(3):273–297.
6. Dai J, Liu C, Xu J, Ding Z. On personalized and sequenced route planning. *World Wide Web*. 2016;19(4):679–705.
7. Deng C, Wang S, Li Z, Huang GB, Lin W. Content-insensitive blind image blurriness assessment using weibull statistics and sparse extreme learning machine. *IEEE Trans Syst Man Cybern: Syst*. 2017.
8. Duan L, Bao M, Cui S, Qiao Y, Miao J. Motor imagery eeg classification based on kernel hierarchical extreme learning machine. *Cogn Comput*. 2017;9(6):758–765.
9. Falcone R, Castelfranchi C. Social trust: a cognitive approach. In: *Trust and deception in virtual societies*, p. 55–90. Springer; 2001.
10. Floyd RW. Algorithm 97: shortest path. *Commun ACM*. 1962;5(6):345.
11. Freeman LC. A set of measures of centrality based on betweenness. *Sociometry*. 1977;40(1):35–41.
12. Golbeck J, Hendler J. Inferring binary trust relationships in web-based social networks. *ACM Trans Internet Technol (TOIT)*. 2006;6(4):497–529.
13. Guha R, Kumar R, Raghavan P, Tomkins A. Propagation of trust and distrust. In: *Proceedings of the 13th international conference on World Wide Web*. p. 403–412. ACM; 2004.
14. Guo T, Zhang L, Tan X. Neuron pruning-based discriminative extreme learning machine for pattern classification. *Cogn Comput*. 2017;9(4):581–595.
15. Gutierrez-Garcia JO, Lopez-Neri E. Cognitive computing: a brief survey and open research challenges. In: *International conference on applied computing and information technology/ international conference on computational science and intelligence*. p. 328–333; 2015.
16. Hang CW, Wang Y, Singh MP. Operators for propagating trust and their evaluation in social networks. In: *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. p. 1025–1032. International Foundation for Autonomous Agents and Multiagent Systems; 2009.
17. Heinemann J, Kramer O. Precise wind power prediction with SVM ensemble regression springer international publishing. 2014.
18. Huang GB, Chen L, Siew CK, et al. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans Neural Networks*. 2006;17(4):879–892.
19. Huang GB, Siew CK. Extreme learning machine: Rbf network case. In: *Control, automation, robotics and vision conference, 2004. ICARCV 2004 8th*. vol. 2, p. 1029–1036. IEEE; 2004.
20. Huang GB, Wang DH, Lan Y. Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*. 2011;2(2):107–122.
21. Huang GB, Zhu QY, Siew CK. Extreme learning machine: a new learning scheme of feedforward neural networks. In: *IEEE international joint conference on Neural networks, 2004. Proceedings, 2004*. vol. 2, p. 985–990. IEEE; 2004.
22. Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications. *Neurocomputing*. 2006;70(1):489–501.
23. Jamali M, Ester M. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. p. 397–406. ACM; 2009.
24. Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks. In: *ACM Conference on recommender systems*, p. 135–142; 2010.
25. Kurashima T, Iwata T, Irie G, Fujimura K. Travel route recommendation using geotags in photo sharing sites. In: *Proceedings of the 19th ACM international conference on Information and knowledge management*, p. 579–588. ACM; 2010.
26. Li F, Cheng D, Hadjieleftheriou M, Kollios G, Teng SH. On trip planning queries in spatial databases. In: *International symposium on spatial and temporal databases*, p. 273–290. Springer; 2005.
27. Liu G, Wang Y, Orgun MA, Liu H. Discovering trust networks for the selection of trustworthy service providers in complex contextual social networks. In: *2012 IEEE 19th international conference on web services (ICWS)*. p. 384–391. IEEE; 2012.
28. Liu N, Sakamoto JT, Cao J, Koh ZX, Ho AFW, Lin Z, Ong MEH. Ensemble-based risk scoring with extreme learning machine for prediction of adverse cardiac events. *Cogn Comput*. 2017;9(4):545–554.
29. Mao W, Jiang M, Wang J, Li Y. Online extreme learning machine with hybrid sampling strategy for sequential imbalanced data. *Cogn Comput*. 2017;9(6):780–800.
30. Navarro G. A guided tour to approximate string matching. *ACM Comput. Surv*. 2001;33(1):31–88.
31. Oneto L, Bisio F, Cambria E, Anguita D. Slt-based elm for big social data analysis. *Cogn Comput*. 2017;9(2):259–274.
32. Scott J. *Social network analysis*. Sage. 2012.
33. Tang M, Xu Y, Liu J, Zheng Z, Liu X. Trust-aware service recommendation via exploiting social networks. In: *IEEE International conference on services computing*, p. 376–383; 2013.
34. Taylor JG. Cognitive computation. *Cogn Comput*. 2009;1(1):4–16.
35. Walter FE, Battiston S, Schweitzer F. A model of a trust-based recommendation system on a social network. *Auton Agent Multi-Agent Syst*. 2008;16(1):57–74.
36. Wang B, Zhu R, Luo S, Yang X, Wang G. H-mrst: a novel framework for supporting probability degree range query using extreme learning machine. *Cogn Comput*. 2017;9(1):68–80.
37. Wang H, Hernandez JM, Van MP. Betweenness centrality in a weighted network. *Phys Rev E Stat Nonlin Soft Matter Phys*. 2008;77(4 Pt 2):046105.
38. Wang S, Deng C, Lin W, Huang GB, Zhao B. Nmf-based image quality assessment using extreme learning machine. *IEEE transactions on cybernetics*. 2017;47(1):232–243.
39. Wang Y, Varadarajan V. Role-based recommendation and trust evaluation. In: *The 9th IEEE international conference on E-commerce technology and the 4th IEEE international conference on enterprise computing, e-commerce, and e-services, 2007. CEC/EEE 2007*, p. 278–288. IEEE; 2007.
40. Wong PK, Gao XH, Wong KI, Vong CM. An analytical study on reasoning of extreme learning machine for classification from its inductive bias. *Cogn Comput*. 2016;8(4):746–756.

41. Wu H, Yue K, Pei Y, Li B, Zhao Y, Dong F. Collaborative topic regression with social trust ensemble for recommendation in social media systems. *Knowl-Based Syst.* 2016;97(C):111–122.
42. Wu J, Xiong R, Chiclana F. Uninorm trust propagation and aggregation methods for group decision making in social network with four tuple information. *Knowl-Based Syst.* 2016;96:29–39.
43. Xu L, Ding S, Xu X, Zhang N. Self-adaptive extreme learning machine optimized by rough set theory and affinity propagation clustering. *Cogn Comput.* 2016;8(4):720–728.
44. Xu Y, Hu T, Li Y. A travel route recommendation algorithm with personal preference. In: 2016 12th international conference on natural computation, fuzzy systems and knowledge discovery (ICNC-FSKD), p. 390–396. IEEE; 2016.
45. Zeng Y, Chen X, Cao X, Qin S, Cavazza M, Xiang Y. Optimal route search with the coverage of users' preferences. In: *IJCAI*, p. 2118–2124; 2015.
46. Zhang Z, Zhao X, Wang G. FE-ELM: a new friend recommendation model with extreme learning machine. *Cogn Comput.* 2017;9(3):1–12.
47. Zheng X, Wang Y, Orgun MA, Zhong Y, Liu G, et al. Trust prediction with propagation and similarity regularization. In: *AAAI*, p. 237–244; 2014.
48. Zhu R, Wang B, Yang X, Zheng B, Wang G. Sap: Improving continuous top-k queries over streaming data. *IEEE Trans Knowl Data Eng.* 2017;29(6):1310–1328.