

A Mobile Tourist Assistance and Recommendation System Based on Complex Networks

Alf-Christian Schering, Martin Duffer
Database Research Group
University of Rostock, Germany
{acsg, md}@informatik.uni-rostock.de

Andreas Finger, Ilvio Bruder
IT Science Center Ruegen
Putbus, Germany
{finger, bruder}@it-science-center.de

ABSTRACT

In recent years, the tourism assistance community is gradually shifting its emphasis to digital, interactive systems. Our project aims for digital tourism assistance by combining mobile guiding and route recommendation based on community networks. We use complex networks to tackle problems, such as deriving optimal routes in conjunction with knowledge deduced from our social tourist network, and giving a respectable prediction of which meta- and multimedia data about touristic objects are needed while the tourist is on tour. This article describes ongoing work on the recommendation system and mobile semantic replication strategies.

Categories and Subject Descriptors

E.1 [DATA STRUCTURES]: Graphs and networks;
H.2.8 [DATABASE MANAGEMENT]: DB applications

General Terms

Algorithms, Design

1. INTRODUCTION AND SCENARIO

Traditional tourism information systems are usually based on simple client-server architectures in conjunction with web applications. Mobile devices generally serve as simple information displays without any substantial own intelligence. The system introduced in this article is based on the premise, that a connection to the service provider is not ubiquitously available while the user is on tour. Hence, this approach focuses on route recommendation based on information given prior to the tour. The scenario, the tourist (also referred to as ‘user’) runs through, is basically subdivided into the following three phases: (1) *tour planning assistance* which takes place prior to the user’s activities, (2) *real-time tour assistance* during the user’s activities, and (3) *post-tour assistance* to help the user giving feedback and organizing digital documents, e.g. digital images, related to the tour.

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

CNIKM’09, November 6, 2009, Hong Kong, China.

Copyright 2009 ACM 978-1-60558-807-0/09/11 ...\$10.00.

1.1 Tour planning assistance

Prior to taking a tour the user is supposed to create a profile to let the tour recommendation system know about his particular interests. Based on this information the system will be able to propose optimized routes which comply with the user’s interests, including touristic objects within the adjacent vicinity of the region the user has chosen to visit. Respective information will be compiled and made available on the user’s mobile client. This enables the user to access information offline while on tour (but does not imply that information generally cannot be accessed on tour), and provides for an adequate information base for users which depend on weak technical equipment (see 4).

1.2 Real-time tour assistance

During his ride or stroll the user is subject to permanent changes within his spatial context. Hence the mobile client continuously checks his position in order to guide the tourist to the right direction, point his attention to the touristic objects on the route and within the adjacent vicinity, which are locally stored in the mobile assistant. This works if the tourist sticks to the route recommendation and schedule. In case the tourist decides to make a detour beyond the informationally covered route vicinity and beyond the schedule, required information will need to be downloaded, on the fly. In order to provide missing data, intelligent techniques, such as context based semantic replication (see 4), are utilized. Content filtering will be applied as well, based on personal profile information and the mobile client’s transmission capabilities. Optimization (query rewriting) will help to avoid redundancies in mobile data storage and data transmission in a poor or expensive web connectivity setting. Inside buildings, e.g. museums, the mobile assistant can interactively provide information about exhibits. In case the planning step prior to the tour has been omitted, the real-time tour assistance will be limited to online recommendations, based on the user’s profile. If a travel destination is not available, the assistant will provide information about touristic objects in the vicinity of the user’s current position.

1.3 Post-tour assistance

After the tour, the user will be able to put his gathered material, such as digital images and routing information back to server and perform desired processing, such as making annotations, creating photo albums, etc. Additionally the tourist can update his user profile and should create recommendations for touristic objects he has visited, to provide for a better tour planning for other users in the community.

2. THE TOURIST ASSISTANCE SYSTEM

The requirements for our tourist assistance system are based on the usage scenarios described in section 1. The two most important and likewise interesting requirements are the need of a **recommendation subsystem for routes and intelligent data allocation techniques**.

The recommendation subsystem is the main part of our tourist assistance system and is therefore in the focus of this paper (see 3). The data allocation techniques we use, directly rely upon the recommendation subsystem. Section 4 will supply you with an idea of our replication strategy.

3. RECOMMENDATION SUBSYSTEM

A vital part of our project is the recommendation subsystem, which is mainly used in the sub scenario *tour planning assistance* (see 1.1) and marginally in the sub scenario *real-time tour assistance* (see 1.2) as well. The goal is always to create a recommendation for the user, which fits the following requirements:

- must include all touristically relevant information available in the data pool (can be taken for granted)
- must take into consideration the travel localities and times which are unique each time a user employs the recommendation system
- must account for the users' personal preferences
- must evaluate and regard feedback, the user has already given for past activities
- feedback given by other users of the system must be embraced in the recommendation process as well.

These requirements must be appropriately reflected in the data model, which is aimed to provide for integrated connectivity between all kinds of information, particularly touristic objects' meta information (see 3.1) and user profiles (see 3.2). The metadata structures of touristic objects and user data satisfy the requirements of complex networks: Touristic and user data are highly interconnected in a non-trivial way. A simple graph, in which a node has only a few edges connecting it to a couple of adjacent nodes, is not adequate for our touristic data model. A structure, such as lattice, is neither adequate to describe the relationships between these data, since there is no regularity present [1]. Touristic and user profile data are real world data and therefore are interconnected in the same complex way as in social networks. They can grow very large if applied to large areas, or even world-wide. Since the number of edges between the nodes is neither subject to any limitation nor the power set of the graph's nodes nor purely regular in any other sense, the touristic objects and the user profile graph can be classified as complex networks. Data structures, we use to manage all touristically relevant information, hereinafter referred to as *touristic data*, are described in section 3.1. Information describing the users of the system (see 3.2), is hereinafter referred to as *user profile data*.

3.1 A complex touristic data network

Touristic data consist of a great deal of touristic objects and an even huger amount of connections in between them. Touristic objects include items that could be interesting for tourists and hence are subject to be part of a route recommendation. In general these include all kinds of *points of*

interest such as buildings, sights, recreational facilities, civic and natural places and sites, events, etc.

In order to meet the requirements of the recommendation subsystem, various properties, touristic objects have in common, are attached to the touristic object nodes:

- location information, incl. address data and geographic coordinates,
- time information, esp. used for events and other time-dependent touristic objects,
- a type property and type-dependent information,
- user feedback-based evaluation information, and
- a web-connectivity property, used to describe the opportunity to acquire an internet connection to access data needed for tour assistance (see 1.2 and 4).

These properties play an important role within the *edge weight normalization phase* (see 3.3.4). Because of their flexibility, the recommendation system supports a great variety of property types.

The formal representation of touristic data is a weighted complex graph. Its nodes are the touristic objects, its edges are the connections between the nodes, which describe the relationships between their respective end-points. These relationships originate from two sources: initial deduction and evaluation processing.

Initial deduction is the second step in the touristic data network creation process. After importing the initial set of touristic objects, including their static properties (e.g. geospatial information), obvious connections (edges) can be deduced automatically by content based analysis of the touristic object nodes. For this purpose we draw on geospatial and temporal information and use clustering to find correlations which are most likely relevant in terms of user preference matching (e.g. someone interested in water sports could be offered certain beaches as well as surfing contest events). The edge creation algorithm is very generous to make sure an edge exists between nodes if there is only a slight chance it could be useful later on. Eventually the amount of generosity will probably be a trade-off between quality and performance.

During the evaluation process (see 1.3) user experiences are made available to the community by adding new edges to the touristic data graph, or by modifying existing edges. New edges are added if the user had spontaneously altered his tour and the resulting edge did not yet exist, what in reality is most likely to happen on each tour. The edges in the touristic data graph are therefor weighted by a **cardinality** value, which is simply the total number of times an edge has actually appeared in any user's tour.

Credibility in contrast is a volatile, weighted, normalized edge attribute (see 3.3.4), which is derived from different heterogeneous attribute values of the participating nodes or functions of the latter. It describes how much the nodes fit to each other. Generally it is possible to have an edge between each pair of nodes. Both cardinality and credibility, appropriately combined based on user preferences, together determine appropriate weights for edges that will be consulted during compilation of other routes later on.

3.2 A complex user profile network

A network of tourists, in our case: users of a tourist information system, is a social network. [8] and [2] have already investigated the usage of complex networks in the area

of music recommendation. Our complex network describes tourists that are connected to each other because of a similar traveling history, i.e. they have visited the same point of interest in the past. This is very important for the *collaborative path finding strategy* (see 3.3). Furthermore properties of a tourist, respectively his interests, determine a certain degree of similarity. In a first step of recommending touristic objects, we deduce a *joint user profile* (see 3.3.2) which provides a basis for weighting potential touristic items of interest by the tourist’s interests. With this approach we also face the *new user* problem and the *over-specialization* problem described in [5]. The *new user* problem represents the lack of sufficient initial information which can prevent reasonable recommendations. Discovering preferences that are not explicitly known to the user (*over-specialization*) even if a detailed user profile exists, is a second problem in touristic recommendation systems and is therefore another convincing argument for using social networks in touristic information systems.

The formal representation of user profile data is a weighted complex graph, where nodes represent tourists and edges represent connections between the nodes. The weight of an edge between two tourists represents the number of identical visited touristic objects.

3.3 The collaborative path finding strategy

It is the goal of the collaborative path finding strategy, shown in figure 1, to use all information available in our system’s data structures, to come up with a tour recommendation, that best fits the user’s preferences. The strategy, which is pursued to eventually devise a tour recommendation for the user, consists of the following five steps:

1. Community analysis
2. Joint user profile vector
3. Filtering nodes
4. Edge weight normalization
5. Path finding

3.3.1 Community analysis

A vital feature of our System is the utilization of knowledge associated with the user community. Within the *community analysis* phase we employ clustering techniques to find like-minded users, which have similar interests and less differing individual-related data relative to the tourist, who is using the system to generate a recommendation for the best route to travel around in a particular area. Every user’s profile can be considered as a vector of preferences, which describes its owner. Components suitable for clustering are e.g. individual-related information such as age or number of children. Depending on μ (mainstream (dis-)favor), a real number between 0 and 1, which describes the affinity the user attaches to the community’s ‘opinion’, the information gathered from other user profiles are comprised to create the *joint user profile vector* accordingly.

3.3.2 Joint user profile vector

Let $\vec{P}_1, \dots, \vec{P}_n$ be the profile vectors of other users which reside in the same cluster as the user’s own profile vector \vec{P}_U , the joint user profile vector \vec{P}_J is calculated as follows:

$$\vec{P}_J = (1 - \mu)\vec{P}_U + \frac{\mu}{n}(\vec{P}_1 + \dots + \vec{P}_n)$$

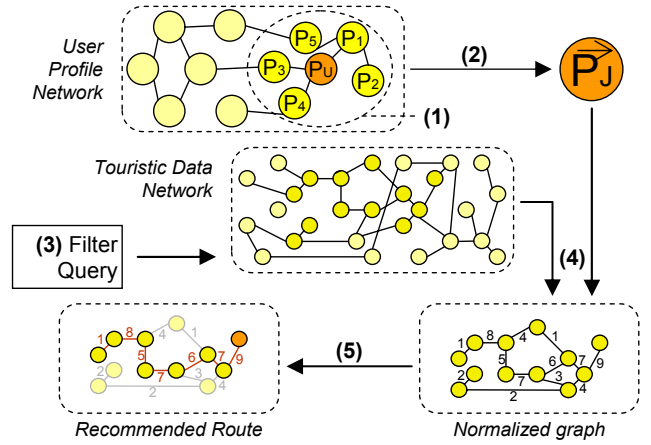


Figure 1: Collaborative path finding strategy

The components of the vector \vec{P}_J are combinations of the vectors $\vec{P}_U, \vec{P}_1, \dots, \vec{P}_n$, e.g. a union of keywords in the component *interest*. The joint user profile vector is now a derivative of the user’s own profile vector. The better the user’s own profile vector is, the better works the community analysis and the better is the joint user profile vector, which plays an important role in the *edge weight normalization* phase, when the joint user profile vector is applied to the filtered touristic data graph.

3.3.3 Filtering nodes

To quickly limit the mere size of the touristic data graph $G = (V, E)$, a simple filtering f_s based on route characteristics, such as travel region and time, is applied to it:

$$G' = f_s(G)$$

Let $G' = (V', E')$, then G' is an induced sub graph of G with $E' = \{uv : uv \in E \text{ and } u, v \in V'\}$. The function f_s annotates the nodes of G' with special scalars. If the resulting scalar of a node is less than a specified threshold the respective node is eliminated from V . All ‘dangling’ edges in E are removed as well. The resulting sub graph contains a drastically reduced graph, which contains all touristic objects that could potentially be interesting for the user, but is small enough though, to be handed over to the more complex *edge weight normalization* phase. Since the resulting graph is not necessarily connected, missing edges will be created in the same way as described in section 3.1 (initial edge creation).

3.3.4 Edge weight normalization

At this point we have two structures: the filtered, cardinality-weighted touristic data graph $G' = (V', E')$ (as described in 3.1) and the joint user profile vector \vec{P}_J .

Let $u, v \in V'$, $uv \in E'$ and $\vec{P}_J = \begin{pmatrix} \mu \\ x_1 \\ \dots \\ x_n \end{pmatrix}$, then $\forall uv$ an atomic edge weight w_{uv} is calculated as follows:

$$w_{uv} = f(u, v, \vec{P}_J) = \frac{(2\mu - 1) \cdot \delta_c(\text{card}_{uv}) + x_1\delta_1(u_1, v_1) + \dots + x_n\delta_n(u_n, v_n)}{n + 1}$$

The symbols $\delta_1, \dots, \delta_n$ represent the individual distance

functions (the credibility part of the formula specified above) used to compare the corresponding property values of the nodes connected by an edge. The resulting normalized values are then multiplied with the respective components of the joint user profile vector. δ_c in conjunction with μ determines a normalized value, representing the user's affinity to mainstream recommendations, based on community feedback (cardinality).

Example: Spatial distance: Both nodes u_i and v_i have a location property (e.g. geographic coordinates). δ_i calculates the reciprocal of the distance between them (shorter distances are preferred). This value is then multiplied with x_i , which represents the importance the user attaches to this particular property distance.

The function f combines all individual weights with the cardinality of the edge (see 3.1) into a single atomic edge weight w .

After \vec{P}_J has been applied to G' ($G'' = \vec{P}_J \cdot G'$) the edges of G'' are weighted with only a single scalar, which represents the user-specific, situation-unique correlation value between each pair of touristic object nodes, which can be utilized as basis for a route recommendation.

3.3.5 Path finding

Devising a tour recommendation from the graph G'' is now boiled down to a path finding problem: In order to create an optimal route between points of interest starting at a certain point, we have to find a route within the graph, which has the highest sum of all weights. This route-finding problem is a kind of the traveling salesman problem, and especially the traveling salesman selection problem [4]. With some further restrictions, it is possible to solve the problem in an appropriate time. An acceptable route that can be recommended to a tourist, needs among other restrictions to meet conditions such as target duration. Target duration could be e.g. ten hours for a day trip. Assuming the mean dwell time to visit one touristic object is about one hour, the recommended route would cover at most ten touristic objects.

Depending on the recommended route the information needed for the guided tour data can now be allocated on the tourist's mobile client, as explained in section 4.

4. DATA ALLOCATION

An important aspect of our tourist assistance system is beneath tour recommendation (see 3.) data allocation on mobile clients.

Systems as described in e.g. [6] use permanent internet connections and web-based applications to display their content on mobile clients. However, a permanent internet connection can not be guaranteed and besides, is cost-intensive and speed-limited. To provide information offline, our system's clients are equipped with mobile databases. Required data are transferred to the mobile client predictively. Although all three phases (see 1) make demands on data allocation, we want to focus only on replication at this point.

Replication

Replication techniques copy explicit portions of server data to mobile clients and thus allow to work offline [3]. There are two architectures depending on the mobile DBMS: client/server und middleware. A Client/server architecture, as we use it (see figure 2), provides direct access to server

data. While offline the local DBMS takes over. The level of autonomy is determined by its performance and the amount of locally stored data.

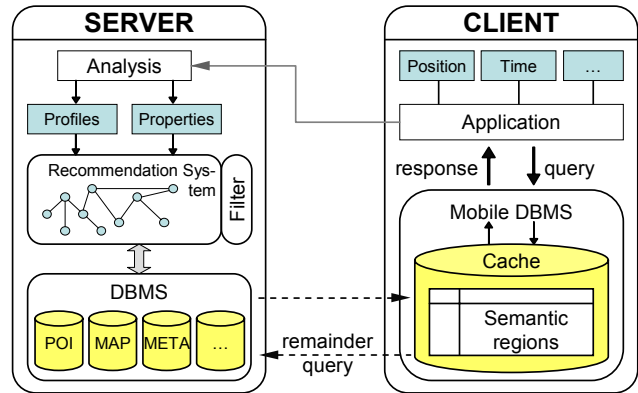


Figure 2: Client/Server Architecture, Semantic Replication

Semantic replication will be object of further investigation, because the optimization potential in our system appears to be there, due to the high complexity between *semantic regions*, *probe query*, and *remainder query* [7].

5. CONCLUSION

This paper describes a strategy for combining mobile tourist assistance and recommendation with tourist community management on top of complex networks. As a result, lists of touristic objects are created, which serve on the one hand as guiding recommendations for tourists and on the other hand as predictions of data that need to be made available on mobile clients with limited internet connection and storage capabilities. As usual in ongoing work, the next step will be an evaluation of the system within a real usage environment together with our industrial partner.

6. REFERENCES

- [1] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph Classes - A Survey*. SIAM, 1999.
- [2] J Buldú, P Cano, M Koppenberger, J Almindral, and S Boccaletti. The complex network of musical tastes. *New Journal of Physics*, 2007.
- [3] C Gollmick. Using Replication Proxy Servers for Scalable Mobile Database Access. In *EDBT PhD Workshop*, 2002.
- [4] A. Hamacher and C. Moll. The Euclidian Traveling Salesman Selection Problem. *Proc. of the Symposium on Operations Research*, 1995.
- [5] A Hinze and S Junmanee. Advanced Recommendation Models for Mobile Tourist Information. *OTM*, 2006.
- [6] G Pospischil, M Umlauf, and E Michlmayr. Designing LoL@, a Mobile Tourist Guide for UMTS. In *Mobile HCI '02*, 2002.
- [7] Q Ren, M Dunham, and V Kumar. Semantic caching and query processing. *IEEE Trans. on Knowl. and Data Eng.*, 15(1), 2003.
- [8] M Zanin, P Cano, J Buldú, and O Celma. Complex networks in recommendation systems. *Proc. of the 2nd WSEAS*, 2008.