**MODULES DESCRIPTION:**

1. **SYSTEM MODEL**
2. **LIFE STYLE MODELING**
3. **ACTIVITIES RECOGNITION**
4. **LIFE STYLE EXTRACTION USING LDA**
5. **FRIEND MATCHING GRAPH**
6. **QUERY AND FRIEND RECOMMENDATION**
7. **FEEDBACK CONTROL**

**SYSTEM MODEL**

We give a high-level overview of the Friend book system. The system architecture of Friend book which adopts a client-server mode where each client is a smart phone carried by a user and the servers are data centers or clouds. On the client side, each smart phone can record data of its user, perform real-time activity recognition and report the generated life documents to the servers. It is worth noting that an offline data collection and training phase is needed to build an appropriate activity classifier for real-time activity recognition on smart phones. We spent three months on collecting raw data of eight volunteers for building a large training data set. As each user typically generates around 50 MB of raw data each day, we choose My SQL as our low level data storage platform and Hadoop Map Reduce as our computation infrastructure. After the activity classifier is built, it will be distributed to each user’s smart phone and then activity recognition can be performed in real-time manner. As a user continually uses Friend book, he/she will accumulate more and more activities in his/her life documents, based on which, we can discover his/her life styles using probabilistic topic model. On the server side, seven modules are designed to fulfill the task of friend recommendation. The data collection module collects life documents from users’ smart phones. The life styles of users are extracted by the life style analysis module with the probabilistic topic model. Then the life style indexing module puts the life styles of users into the database in the format of (life-style, user) instead of (user, lifestyle). A friend-matching graph can be constructed accordingly by the friend-matching graph construction module to represent the similarity relationship between users’ life styles. The impacts of users are then calculated based on the friend-matching graph by the user impact ranking module. The user query module takes a user’s query and sends a ranked list of potential friends to the user as response. The system also allows users to give feedback of the recommendation results which can be processed by the feedback control module. With this module, the accuracy of friend recommendation can be improved.

**LIFE STYLE MODELING:**

Life styles and activities are reflections of daily lives at two different levels where daily lives can be treated as a mixture of life styles and life styles as a mixture of activities. This is analogous to the treatment of documents as ensemble of topics and topics as ensemble of words. By taking advantage of recent developments in the field of text mining, we model the daily lives of users as life documents, the life styles as topics, and the activities as words. Given “documents”, the probabilistic topic model could discover the probabilities of underlying “topics”. Therefore, we adopt the probabilistic topic model to discover the probabilities of hidden “life styles” from the “life documents”. In probabilistic topic models, the frequency of vocabulary is particularly important, as different frequency of words denotes their information entropy variances. Following this observation, we propose the “bag-of-activity” model to replace the original sequences of activities recognized based on the raw data with their probability distributions. Thereafter, each user has a bag-of-activity representation of his/her life document, which comprises a mixture of activity words.

**ACTIVITIES RECOGNITION:**

We need to first classify or recognize the activities of users. Life styles are usually reflected as a mixture of motion activities with different occurrence probability. Therefore, two motion sensors, accelerometer and gyroscope, are used to infer users’ motion activities. Generally speaking, there are two mainstream approaches: supervised learning and unsupervised learning. For both approaches, mature techniques have been developed and tested. In practice, the number of activities involved in the analysis is unpredictable and it is difficult to collect a large set of ground truth data for each activity, which makes supervised learning algorithms unsuitable for our system. Therefore, we use unsupervised learning approaches to recognize activities. Here, we adopt the popular K-means clustering algorithm [9] to group data into clusters, where each cluster represents an activity. Note that activity recognition is not the main concern of our paper. Other more complicated clustering algorithms can certainly be used. We choose K-means for its simplicity and effectiveness.

**LIFE STYLE EXTRACTION USING LDA:**

We use the Expectation-Maximization (EM) method to solve the LDA decomposition, where the E-step is used to estimate the free variational Dirichlet parameter g and multinomial parameter F in the standard LDA model and the Mstep is used to maximize the log likelihood of the activities under these parameters. After the EM algorithm converges, we are able to calculate the decomposed activity-topic matrix. Readers are referred to for more details of the LDA algorithm and alternative decomposition approaches. It is worth noting that the matrix decomposition process can be implemented more efficiently through incremental iteration. That is, when a user’s life document changes or a new user’s life document is uploaded to the system, Friend book can calculate the new life style vectors for each user based on previously derived life style vectors and the new life document.

**FRIEND MATCHING GRAPH:**

To characterize relations among users, in this section, we propose the friend-matching graph to represent the similarity between their life styles and how they influence other people in the graph. In particular, we use the link weight between two users to represent the similarity of their life styles. Based on the friend-matching graph, we can obtain a user’s affinity reflecting how likely this user will be chosen as another user’s friend in the network.

**QUERY AND FRIEND RECOMMENDATION:**

Before a user initiates a request, he/she should have accumulated enough activities in his/her life documents for efficient life styles analysis. The period for collecting data usually takes at least one day. Longer time would be expected if the user wants to get more satisfied friend recommendation results. After receiving a user’s request (e.g., life documents), the server would extract the user’s life style vector, and based on which recommend friends to the user.

**FEEDBACK CONTROL:**

To support performance optimization at runtime, we also integrate a feedback control mechanism into Friend book. After the server generates a reply in response to a query, the feedback mechanism allows us to measure the satisfaction of users, by providing a user interface that allows the user to rate the friend list. Let \_r denote the impact ranking vector calculated from the feedback of users.