Chapter 6

Service selection and recommendation in integrated network environment

Lingyan Zhang[1], Mingzhe Yang[1], Yan Guo[1], and Shangguang Wang[1]

6.1 Introduction

With the rapid development of information technology and network technology, the Internet is becoming more intelligent, personalized, and social, influencing and changing people's way of life. Many heterogeneous networks (mobile network, Internet, Television broadcasting network, etc.) and technology are integrated into an open communication network, that is, the integrated network.

In the context of network integration, a large number of services, provided by service providers, have sprung to meet various needs of service consumers or users. Service is becoming a "collective term" that is also known as "everything as a service." Various services, including online shopping, music download, live streaming video, social networking, and various mobile apps, improve the efficiency of people's work and facilitate people's life. The ubiquitous and openness characters of the integrated network provide users with personalized services better.

However, along with the explosion of services in the integrated network, there have emerged new problems. On the one hand, to the user, with the rapid growth of the type and number of services, the users run into the trouble of information overload, and usually users need to spend a lot of time finding themselves services they need; on the other hand, to the service provider, the process that users browse a large number of irrelevant services will no doubt make consumers submerged in the problem of information overload in the continuous loss. How to achieve fast and reliable selection and recommendation of optimal services for users in integrated network environment has become one of the most challenging issues in the field of service computing [1].

One of the most typical examples is social networking services. Because social networking service always has a huge user groups and users frequently update status, causing social networking services would produce lots of users and redundant

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China

data every day. How to find useful information from these data, and how to provide users with personalized recommendation services, such as friends recommendation and advertising targeting, becomes the focus and keystone of social networking services [2].

In order to let users get information they want accurately, service selection and recommendation systems emerge as the times require. Because it can bring tremendous commercial value and interest, whether in academia area or industry area, service selection and recommendation systems have attracted great attention. In academic area, there have appeared many efficient methods and algorithms of service selection and recommendation, and in the industrial area, service selection and recommendation systems have been widely used in various occasions. With continuous growth in the number of services, selecting and recommending optimal services for users will become more and more important and acute in the field of service computing [3,4].

6.2 Integrated network

The integrated network is an open communication network which integrates many technology and heterogeneous networks (mobile network, Internet, Television broadcasting network, etc.). As network convergence evolves, its characters bring special requirements in service selection and recommendation [5–7].

- Different networks have different user groups, and the integrated network integrates many heterogeneous networks, so the number of users in integrated network is far more than any previous network.
- The ubiquitous of integrated network makes the provision of personalized services in a much convenient and popular way, and more and more user requirements are raised on the integrated network.
- An integrated network needs to satisfy all sorts of service requests of a large number of users, and the service requests of these users generally have the same or similar function and different preferences, it is often the case that too many user requests lead to overloading of the service, which further leads to the decline of the quality of service (QoS).
- The dynamic feature of integrated network will lead to the changing of service execution environment. To ensure the continuity of service, appropriate services should be selected and recommended for their adaptability. And this change process should be automatic, cannot be aware by the users.

6.3 Service selection

6.3.1 Selection problem definition

With the rapidly growing number of available services, users are presented with a choice of functionally similar services. This choice allows user to select services that match other criteria which is often referred to as QoS criteria. This line of work opens

Table 6.1 Major QoS attributes

QoS attributes	Description	Type
Availability	The probability that a service can respond to requests	Positive
Capacity	The limit on the number of requests a service can handle	Positive
Economic	The economic conditions of using the service, such as price	Negative
Throughput	The rate of successful service request completion	Positive
Response time	The delay from the request to getting a response from the service	Negative
Reliability	The likelihood of successfully using a service	Positive
Scalability	Whether the service capacity can increase as needed	Positive
Security	The level and kind of security a service provides	Positive
Stability	The rate of change of service attributes, such as its service interface	Negative

up two fundamental questions: how can these extra attributes be described and how can one select the most appropriate service [8].

These questions should be addressed on both the selection of isolated services and the selection of composite services. In some cases, an individual service cannot satisfy users' requirements, and we need to select and integrate several individual services from multiple service classes to create new value-added composite services [9].

6.3.1.1 Quality of service attributes

In general, nonfunctional extra attributes are defined as QoS attributes. A QoS attribute can be static or dynamic. A static QoS property value has been defined at the time it is described, whereas the dynamic QoS property value requires measuring and updating its value periodically. The QoS value from the service user's perspective can be positive and negative. For example, users expect to buy a service at a low price and expect to call the service in a low response time.

Positive attributes refer to the QoS attributes that the higher the attribute value is, the better the quality is (e.g., reliability and availability);

Negative attributes refer to the QoS attributes that the higher the attribute value is, the worse the quality is (e.g., response time, delay time, and price).

Table 6.1 lists the major QoS attributes of services.

6.3.1.2 Service selection definition

In order to better understand the service selection problem, the related concepts [10] are given below.

Service candidate is a service which can satisfy a user's particular functional demand;
Service class is a set composed by multiple service candidates which have the same functions but different nonfunctional attribute values;
QoS attribute is nonfunctional attribute that defines the performance of a service, such as error rates, bit rate, and throughput;
Utility of a service indicates what degree user-defined constraints will be satisfied;
Service selection is to select the most suitable service candidate(s) to meet the functional and QoS requirements of user from one or more service class.

More formally, the selection problem can be formulated as follows: let CS be the set of the required service classes according to the functional requirements of the user, i.e., $CS = \{S_1, \ldots, S_n\}$; and the composite service cs is integrated by component service s_1, $s_2, \ldots, s_n$; and let $Q(s)$ be the vector of the QoS value of service s, i.e., $Q(s) = \{q_1(s), \ldots, q_m(s)\}$; let u be a utility function that measures the fitness of the composite service cs to user constraints; then, for each individual service s, we want to choose such service that maximizes the utility of the composite service. More formally:

$$\max \sum_{i=1}^{m} c_i \overline{Q_i(s_1, s_2, \ldots, s_n)}, \tag{6.1}$$

where $\overline{Q_1(s_1, s_2, \ldots, s_n)}$ is the normalized value of $Q_i(s_1, s_2, \ldots, s_n)$ which represents the ith QoS attribute value of the composite service integrated by component service $s_1, s_2, \ldots, s_n$, c_i is the weight coefficient corresponding to the ith attribute, and there are $c_i \in [0, 1]$ and $\sum_{i=1}^{m} c_i = 1$.

The constraints are defined by (6.2), which is used to describe the users' personalized requirements.

$$\text{constraint } s \begin{cases} Q_1(s_1, s_2, \ldots, s_n) \leq b_1 \\ \vdots \\ Q_m(s_1, s_2, \ldots, s_n) \leq b_m \end{cases}, \tag{6.2}$$

where b_i is the ith QoS constraint of user.

6.3.2 *Problem induction*

According to the different needs of users under different scenarios, service selection can be converted into different math problems and then designs different models to choose appropriate selection strategies and algorithms.

6.3.2.1 Multiple attribute decision making problem

In the service selection problem, it requires consideration of specific preferences and constraints by a service requester as well as evaluation of combinations of different QoS constraints. Hence, QoS-based service selection is basically the problem of multiple attribute decision making.

Although regular QoS attributes are listed in Table 6.1, it remains some issues on selection of services. First, the perception on QoS of services is distinct between the user and provider. Second, service requestors may have varying preferences for the QoS attributes depending on the situation.

In a real service selection scene, because of the complexity of QoS attributes, it is difficult for users to provide the weights or the precise value of QoS attributes to express their preferences; besides, it is difficult to standardize the QoS attribute values to the unified metric space and accurately assess a comprehensive QoS value of service. Therefore, the service selection problem can be converted to a multiple attribute decision-making problem, a problem with the multiple and conflicting attributes.

6.3.2.2 Integer linear programming problem

Service selection problem for composite services becomes a decision problem to maximize the fitness of the constraints of service requesters. Actually, any decision problem in which the decision variables must assume nonfractional or discrete values can be classified into an integer optimization problem. And one way of solving this optimization problem is to model it as an integer linear programming (ILP) problem. In the ILP approach, decision variables are integers representing whether a particular service is selected for composite services. To define an ILP problem, three inputs should be provided: a set of decision variables, an objective function, and a set of constraints.

Decision variables refer to service candidates from service classes.
Objective function is to reflect some benefit or utility of service requesters. It is defined as weighted sum of QoS attributes that are favorable to the requesters.
Constrains refer to the functional and QoS requirements of the service requester.

However, the computational complexity and cost of the ILP solution will increase exponentially with the growth of the size of services. And it requires the objective function and constraint conditions are linear, which limits the usefulness of the algorithm in a certain extent.

6.3.2.3 Knapsack problem

In order to deal with the computational complexity and cost in large-scale service selection, service selection for a composite service can be modeled as a multidimension multichoice knapsack problem (MMKP), which is known to be an NP-hard problem in the strong sense. This means that for large problems, it is unlikely that an optimal solution can be found given a reasonable amount of computational effort.

In certain cases, in the process of service selection, there is no need to search all candidate services to find global optimal solution. Looking for near-optimal service solution can ensure the needs of users, and can greatly reduce the time cost as well.

MMKP is about taking items into a knapsack with a limited capability. Thus, a selection must be performed to identify the optimal subset. Service selection problem can be mapped into a knapsack problem, when each item represents a candidate service, the profit is on behalf of the utility of service, the weight represents the service quality attribute, and the capacity of backpack is on behalf of the QoS constraint. The objective of MMKP is to select exactly one service candidate from each service class to be included in the knapsack within the functional and QoS constraint while maximizing the total profit.

6.3.2.4 Combinatorial optimization problem

At first, researchers only focus on a certain indicator in the process of service selection, such as to minimize response time or maximize service utility, so this kind of problem is often depicted as single objective optimization problem. With the diverse needs of users and service providers, the concerned indexes and attributes gradually increase, the optimize target transformed from a single index into multi indexes, so the service selection turned into a multiobjective optimization problem, i.e., combinatorial optimization problem.

The combinatorial optimization is often difficult to have an optimal solution meeting all the objective functions. For example, it may be impractical if you want to minimize the response time while minimize the system cost, because the low response time means the high service level, and it is likely to pay more fee for a high level of service. So the methods adopted to solve the single-objective optimization problems are difficult to directly be applied to combinatorial problems.

The most common thinking is that transform the multiobjective optimization problem into the single-objective optimization problem. The linear weighted sum method and the ε-constraint method are the most widely used methods. In addition, the Pareto model is also used to the multiobjective optimization problem of service selection.

6.3.3 Service selection algorithm

Traditional service selection algorithms mainly contain exhaustive algorithm and greedy algorithm. However, the complexity of exhaustive algorithm seeking the optimal solution is too high, and greedy algorithm cannot necessarily obtain global optimal solution, the effectively approximate solution has been put forward to attain near-optimal solution. In general, service selection algorithms include optimization algorithms, heuristic algorithms, and fusion algorithms.

Optimization algorithms: The optimization algorithms can guarantee the optimality of the solution, and the specific optimal algorithms applied to the service selection problem mainly include the exhaustive algorithm and graph algorithm.

Heuristic algorithms: Relative to the optimization algorithm, a heuristic algorithm can be defined as a constructed algorithm based on intuition or experience. The specific heuristic algorithms applied to the service selection problem mainly include genetic algorithm (GA), particle swarm optimization algorithm (PSO), and ant colony optimization algorithm (ACO).

Fusion algorithms: Although heuristic algorithm can solve the feasible solution of the optimization problem under the acceptable costs (such as computing time), the deviation between the feasible solution and the optimal solution cannot be expected necessarily in advance. Fusion algorithms can combine the advantages of various algorithms and improve the performance and efficiency of the optimization.

6.3.3.1 Optimization algorithm

Optimization algorithms, which try to find the maximum values of utility functions, have been widely applied in service selection.

Exhaustive algorithm is a straightforward one by computing the QoS value for each possible execution plan and selecting the best one from them. It lists all possible execution plans by exhaustive searching and compares their QoS values. It can always produce the optimal solution but is time and memory consuming, and the complexity depends on the size of the solution space. Assuming that there are *n* service classes and *m* service candidates for each class, the total number of execution plans is m^n, making this approach impractical. So it is only suitable when the number of classes and the candidates of each class are all small.

Greedy algorithm is an algorithm that follows the problem solving heuristic of making the locally optimal candidate services at each class with the hope of finding a global optimal service. In service selection, a greedy strategy does not in general produce a global optimal solution, but nonetheless a greedy algorithm may yield locally optimal solutions that approximate a global optimal solution in a reasonable time.

Graph algorithm [11] is used to solve the abstracted graph theory problems of service selection. For example, the service selection problem can be transformed into the shortest path problem with functional and QoS constraints. First, build the directed graph corresponding to the composite service, in which the graph nodes represent the candidate services, and the QoS parameters are represented by edges. The optimal path obtained by traversing the graph edge satisfies all the constraint conditions and has the best utility, and it corresponds to the optimal solution of the original service selection problem for the composite service. Using the graph algorithm often needs access the topology information of service deployment, so the access to information before optimization may cost a lot of time.

6.3.3.2 Genetic algorithm

GA [12,13] solves the formulated optimization problem by using the idea of Darwinian evolution. As a powerful search algorithm based on natural selection and population genetics, GA represents the problem solving process by the survival process of the fittest chromosome, through the iterative evolution of population, including selection, crossover, and mutation operation, combines populations to produce the next generation individuals, gradually evolves toward the optimized population, and ultimately obtains the chromosome most adapted to the environment, which is the optimal solution or near-optimal solution of problem.

The implementation of GA in application of the service selection problem incorporates three basic steps so that the algorithm is formulated for the specific application: the presentation of individual according to the specific service selection, the formulation of the fitness function, crossover and mutation operators. Each chromosome represents a possible service combination which consists of candidate services selected from different service classes. Each gene corresponds to a candidate service, and the value of the ith genes is the serial number of selected candidate services in the ith service class. The determination of fitness function is according to the QoS value of each candidate service. Crossover operation is to exchange the value of some genes in two different service combinations by a crossover method, and to produce two new chromosomes. Mutation operation is to replace the candidate service by another corresponding available candidate service.

6.3.3.3 Particle swam optimization algorithm

PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a population of candidate solutions (i.e., particles) and moving these particles around in the search space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position, but is also guided toward the best known positions in the search space, and the positions are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

Using PSO algorithm to solve service selection problem [14,15], the key is to establish a mapping relationship between the particle position vector and combination plan, the expression of particle position, the determination of velocity formula, and the evaluation of fitness function. Each particle represents a candidate solution, which is a complex service composed of several candidate services in a certain order. The position of each particle represents the serial number of candidate services selected from each service class. The QoS attribute values of composition service represented by a particle is the particle's fitness, the fitness function in accordance with the specific composition service example. The update formula of speed and position depends on the coding scheme of service selection which is the presentation of particle position.

However, PSO exhibits some disadvantages: it sometimes is easy to be trapped in local optima (it is often called premature convergence), and the convergence rate decreased considerably in the later period of evolution.

6.3.3.4 Ant colony optimization algorithm

ACO is a simulated evolutionary algorithm which is aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. The basic principle of ACO comes from the observation of entomologists is as follows: the ants can release a kind of gas (pheromone) on the path they have walked when they finding food. The pheromone can be known by other ants and influences their behavior. As more and more ants walk the same path, the density of pheromone will be bigger and bigger, which improves the selected probability of this path. And the attractiveness of this path will be bigger and bigger too. The behavior of releasing the pheromone is the foundation of positive feedback mechanism of ant colony algorithm. The basic thinking of ant colony algorithm is as follows: at the certain point, an ant chooses the next path, then, the path many ants have walked before has the bigger probability to be selected again, and the more pheromone means the shorter path length, which means a better answer.

The selection problem for composite services can be transformed into a problem seeking the QoS optimal path from a starting point to the target point. Suppose that there is a composite service which combined with n services with different functions, the QoS-aware problem of the complex service selection is to select n services from n services classes, and each class has n_i independent services which have the similar or the same function. So the number of the possible selection plan is $n_1 \times n_2 \times \cdots \times n_n$.

The essence of process of service selection based on the ACO algorithm [16] is to find a shortest path from the initiative point to the end point, and the ant must travel each class only once. The key is to solve how to determine state transition rule according to the QoS parameters in a composite service, update methods of pheromone and assessment of the fitness function, etc.

6.3.3.5 Fusion algorithm

Each algorithm has its advantages and disadvantages, and a single algorithm will not be able to solve the more complicated problem with larger scale. Fusion algorithms which can combine the advantages of various algorithms are used to improve the performance and efficiency of service selection in more complex scenarios.

Many fusion algorithms [17] are proposed to solve service selection problem. For example, on the basis of redefining the location, speed, add/subtraction and multiplication of PSO, combining the crossover and mutation operations in the GA, a QoS scheduling method based on hybrid particle swarm is designed to select the best candidate services satisfying users' QoS requirements. Scalable QoS computation method [1] based on heuristic algorithm is designed, which decomposes a services selection problem into several sub optimization problems. Compared with the ILP method, it can find nearly optimal solution more quickly. A reliable random QoS-aware service selection algorithm is designed based on the Markov Decision Process. The methods based on skyline reduce a large number of redundant candidate services, are more effectively for service selection.

6.3.4 Service selection summary

There has been much research done on service selection over the past several years that have inducted service selection problem into Multiple Attribute Decision Problem, Integer Programming Problem, Knapsack Problem, Combinatorial Optimization Problem, and so on. And many optimization algorithms, heuristic algorithms, and fusion algorithms have been proposed to solve the above problems. A classification and summary of the service selection research is presented in Table 6.2.

It can be concluded that most approaches contribute specific aspects to the overall picture of service selection, which requires methods for expressing user requirements, expressing service offerings, and also the actual service selection method. With the increasing availability of services as a solution to application integration, the QoS parameters offered by services are becoming the chief priority for service providers and their service users. Due to the agile and dynamic nature of the Internet, providing the suitable QoS for various services is really a challenging task. In addition to this, important aspects that need addressing are powerful mechanisms to capture user requirements that are both user friendly and also expressive enough to capture large numbers of preferences and the logical relationships between preferences. One aspect that falls into this area is the measuring of weights, that is, users' fuzzy view on QoS parameters has to be modeled and weighted in universal manner. Also, in the process of capturing the needs of users, their preference of data, research has to show interest and capability to automatically capture this, to reduce the burden on the user part, and to react to changes in circumstances automatically [18,19].

6.4 Service recommendation

Nowadays, lots of applications are available for recommending books and products in online shopping websites, friends and hot topics in social networks, music and movies

Table 6.2 Classification and summary of service selection research

Selection approach		Techniques	Papers
Optimization algorithm	Exhaustive algorithm	Weighted multistage graph	Gao *et al.* (2006) [20]
	Greedy algorithm	Relational algebra	Ding *et al.* (2009) [21]
	Graph algorithm	Multigrain clustering	Schröpfer *et al.* (2007) [11]
	ILP algorithm	Fuzzy synthetic evaluation	Ardagna and Pernici (2007) [22]
		Mixed integer programming cloud computing	Wang *et al.* (2011) [23] Wang *et al.* (2015) [24]
Heuristic algorithm	GA	Global optimization	Mardukhi *et al.* (2013) [12]
		Local optimization	Ma and Zhang (2008) [13]
		Hybrid GA	Tang and Ai (2010) [25]
	PSO	Fuzzy logic	Wang *et al.* (2013) [15]
		Skyline operator	
	ACO	Swarm intelligence	Wang *et al.* (2010) [16]
	Fusion algorithms	Service clustering	Guidara *et al.* (2015) [1]
		Constraints decomposition	
		Neural network	

in video websites. Moreover, some of the vendors have incorporated recommendation capabilities into their commerce servers. With the increasing amounts of applications that help users deal with information overload, service recommendation is playing an increasingly significant role [26,27].

At the same time, users have more independence and options when choosing services including movie, music, and video. And they can choose services depending on their own personal preferences. With movies as an example, they can be tagged with fiction, action, war, comedy, etc., or classified as director and main actor. When watching movies, users always make decisions according to these tags. Based on user's history records and film scores, service recommendation system can dig up active users' preferences and find out users with similar interests, then, we can recommend similar users' movies which have never been watched by the active user.

6.4.1 Recommendation scenarios

Service recommendation systems are applied to various occasions, especially in commercial vocation. In our daily life, there are lots of recommendation examples and scenarios. Examples of such applications include recommending products at Amazon.com [28], movies by MovieLens [29], and news at VERSIFI Technologies. Following is a list of several common service recommendation scenarios.

6.4.1.1 Commodity recommendation in e-commerce site

Commodity recommendation in e-commerce site is one of the most widely used applications. Most of these websites recommend relevant items to users depending on users' browsing history. As an international e-commerce company, Amazon has always been an active participant and a user of recommendation system. The most important applications are personalized and relevant items' recommendation lists in Amazon's recommendation system.

6.4.1.2 Advertisement targeting and recommendation

When browsing webpages, we always see all kinds of advertisements which are elaborately designed to draw users' attention. In addition to advertisement content, manifestation pattern, the most important factor is increasing the accuracy of advertising, which means precisely carrying advertisements to users with purchasing demand. In fact, the precisely advertisement targeting is the personalized product recommending to user and the advertisement is the output product of recommendation system. YouTube has a variety of targeting options, such as age, gender, location, interest, and others, that help advertisers reach the right customer for their business.

6.4.1.3 Location based service recommendation

Through Global Position System (GPS), Wi-Fi, and base station signal, users' location information can be acquired in real time. Location-based service recommendation is widely used in modern life. For example, when a user stays in a strange place, as long as opening the map of mobile phone, recommendation system can immediately provide restaurants suitable for user's taste and hotels meeting user's consumption level by mobile phone location.

6.4.2 Recommendation problem definition

Service recommendation problem can be reduced to a problem of estimating ratings for the services that have not been seen by a user. Intuitively, this estimation is usually based on the ratings given by the user to other services and on some other information. Once we can estimate ratings for the yet unrated services, we can recommend to the user the service with the highest estimated rating.

More formally, the recommendation problem can be formulated as follows: let C be the set of all users and let S be the set of service candidates that can be recommended; let u be a utility function that measures the usefulness of service s to user c, i.e., $u : C \times S \to R$, where R is a totally ordered set; then, for each user c, we want to choose such service that maximizes the user's utility. More formally:

$$\forall c \in C, \mathrm{s}'_c = \arg\max_{s \in S} u(c, s) \tag{6.3}$$

In recommendation systems, the utility of a service is usually represented by a rating, which indicates how a particular user liked a particular service, for example, John Doe gave the movie "Harry Potter" the rating of 7 (out of 10). Each element of the user space C can be defined with a profile that includes various user characteristics, such as age, gender, income, and marital status. Similarly, each element of the service space S is defined with a profile that includes various service characteristics, such as title, genre, director, year of release, and leading actors.

6.4.3 Recommendation systems and techniques

Much research over the past decade has focused on developing service recommendation systems and techniques from disciplines such as human-computer interaction, statistics, machine learning, and information retrieval. These methods are often classified into broad categories according to recommendation approach, as well as to algorithmic technique.

Recommendation systems [30] are usually classified based on recommendation approach as:

Content-based recommendation [31]: Content-based methods analyze the common features among the services a user has already rated highly. Only the services similar to user's past preferences are then recommended;
Collaborative recommendation [32]: Collaborative methods recommend services to the user that people with similar tastes and preferences have liked in the past;
Hybrid recommendation: It is a combination of collaborative and content-based methods to avoid certain limitations of content-based and collaborative systems.

Recommendation systems methods can be classified based on the algorithmic technique as:

Heuristic-based technique: Heuristic-based techniques make rating predictions based on the entire collection of previously rated services by the users;
Model-based technique: Model-based techniques use previous transactions to learn a model (usually using some machine-learning [33] or statistical technique) that is then used for making recommendations.

6.4.3.1 Content-based recommendation with heuristic-based technique

In content-based recommendation methods, the utility $u(c, s)$ of service s for user c is estimated based on the utilities $u(c, s_i)$ assigned by user c to services $s_i \in S$ that are "similar" to service s. For example, in a movie recommendation application, in order to recommend movies to the target user, the content-based recommendation system tries to understand the commonalities among the movies user c has rated highly in the past (specific actors, directors, subject matter, etc.). Then, only the movies that have a high degree of similarity to whatever user's preferences would get recommended.

More formally, let $ServiceP(s)$ be a service profile, that is, a set of attributes characterizing service s. It is usually computed by extracting a set of features from service s and is used to determine appropriateness of the service for recommendation purposes. Let $UserP(c)$ be the profile of user c containing preferences of the user. These profiles are obtained by analyzing the content of the services previously seen and rated by the user. Thus, the utility function $u(c, s)$ is usually defined as:

$$u(c, s) = score(UserP(c), ServiceP(s)). \tag{6.4}$$

Note, both service profile $ServiceP(s)$ and user profile $UserP(c)$ are constructed using keyword analysis techniques from information retrieval. And one of the best-known measures for specifying keyword weights in Information Retrieval is the term frequency/inverse document frequency (TF-IDF) measure. By using the TF-IDF measure method, both *UserP*(c) and *ServiceP*(s) can be represented as vectors $\boldsymbol{w}_c$ and $\boldsymbol{w}_s$ of keyword weights. Thus, utility function $u(c, s)$ is usually represented in information retrieval literature by some scoring heuristic defined in terms of vectors $\boldsymbol{w}_c$ and $\boldsymbol{w}_s$, such as cosine similarity measure:

$$u(c,s) = \cos(\boldsymbol{w}_c, \boldsymbol{w}_s) = \frac{\boldsymbol{w}_c \cdot \boldsymbol{w}_s}{\|\boldsymbol{w}_c\|_2 \times \|\boldsymbol{w}_s\|_2} = \frac{\sum_{i=1}^{K} w_{i,c} w_{i,s}}{\sqrt{\sum_{i=1}^{K} w_{i,c}^2}\sqrt{\sum_{i=1}^{K} w_{i,x}^2}} \tag{6.5}$$

where K is the total number of keywords in the system.

6.4.3.2 Content-based recommendation with model-based technique

Model-based techniques differ from information retrieval-based approaches in that they calculate utility predictions based not on a heuristic formula, but rather are based on a model learned from the underlying data using statistical learning and machine learning techniques [34].

For example, based on a set of services that are rated as "relevant" or "irrelevant" by the user, we can use the naïve Bayesian classifier to classify unrated services. More specifically, the naïve Bayesian classifier is used to estimate the following probability that service s_j belongs to a certain group G_i (e.g., relevant or irrelevant) given the set of keywords, $k_{1,j}, \ldots, k_{n,j}$, on that service:

$$P(G_i | k_{1,j} \,\&\ldots\&\, k_{n,j}). \tag{6.6}$$

Moreover, we can assume that keywords are independent and, therefore, the above probability is proportional to:

$$P(G_i) \prod_x P(k_{x,j} | G_i). \tag{6.7}$$

While the keyword independence assumption does not necessarily apply in many applications, naïve Bayesian classifiers still produce high classification accuracy. Furthermore, both $P(k_{x,j}|G_i)$ and $P(G_i)$ can be estimated from the underlying training data. Therefore, for each service s_j, the probability $P(G_i|k_{1,j} \,\&\ldots\&\, k_{n,j})$ is computed for each group G_i, and service s_j is assigned to the group G_i having the highest probability.

As in the case of content-based approaches, the main difference between model-based techniques and heuristic-based techniques is that the model-based techniques calculate utility (rating) predictions based not on some ad hoc heuristic rules, but rather based on a *model* learned from the underlying data using statistical and machine learning techniques.

6.4.3.3 Collaborative recommendation with heuristic technique

Collaborative recommendation systems try to predict the utility of services for a particular user based on the services previously rated by other users. More formally, the utility $u(c, s)$ of services s for user c is estimated based on the utilities $u(c_j, s)$ assigned to services s by those users $c_j \in C$ who are "similar" to user c. For example, in a movie recommendation application, in order to recommend movies to user c, the collaborative recommendation system tries to find the “peers” of user c, that is, other users that have similar tastes in movies (rate the same movies similarly). Then, only the movies that are most favored by the “peers” of user c would get recommended.

Collaborative recommendation with heuristic-based techniques [35] makes rating predictions based on the entire collection of previously rated services by users. That is, the value of the unknown rating $r_{c,s}$ for user c and service s is usually computed as an aggregate of the ratings of the N most similar users for the same services:

$$r_{c,s} = \operatorname*{aggr}_{c'=\hat{C}} r_{c',s}, \tag{6.8}$$

where $\hat{C}$ denotes the set of N users that are the most similar to user c and have rated service s. The most common aggregation approach is to use the weighted sum as:

$$r_{c,s} = \bar{r}_c + k \sum_{c' \in \hat{C}} \text{sim}(c, c') \times (r_{c',s} - \bar{r}_{c'}), \tag{6.9}$$

where k serves as a normalizing factor and is usually selected as $k = 1/\sum_{c' \in \hat{C}} |\text{sim}(c, c')|$, $\bar{r}_c$ is the average rating of user c, and $\text{sim}(c, c')$ is the similarity between user c and user c'.

In most of these approaches, the similarity between two users is based on their ratings of services that both users have rated. The two most popular approaches are correlation-based and cosine-based approaches. The Pearson correlation coefficient is shown as follows:

$$\text{sim}(x, y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)(r_{y,s} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)^2 \sum_{s \in S_{xy}} (r_{y,s} - \bar{r}_y)^2}}, \tag{6.10}$$

where $S_{x,y}$ is the set of all services correlated by both user x and user y.

6.4.3.4 Collaborative recommendation with model-based technique

Collaborative recommendation with model-based techniques makes rating predictions based on a model learn from the collection of ratings. For example, we can use a simple probabilistic model to implement collaborative filtering, where the unknown ratings of services are calculated as:

$$r_{c,s} = E(r_{c,s}) = \sum_{i=0}^{n} i \times \Pr(r_{c,s} = i | r_{c,s'}, s' \in S_c), \tag{6.11}$$

where $r_{c,s}$ is the rating value for user c and service s, and it is assumed that rating values are integers between 0 and n, and the probability expression is the probability that user c will give a particular rating to service$s$ given that user’s ratings of the previously rated services.

To estimate this probability, there are two alternative probabilistic models: cluster models and Bayesian networks. In the first model, like-minded users are clustered into classes. Given the user’s class membership, the user ratings are assumed to be independent, that is, the model structure is that of a naïve Bayesian model. The number of classes and the parameters of the model are learned from the data. The second model represents each service in the domain as a node in a Bayesian network, where the states of each node correspond to the possible rating values for each service. Both the structure of the network and the conditional probabilities are learned from the data. One limitation of this model is that each user can be clustered into a single cluster, whereas some recommendation applications may benefit from the ability to cluster users into several categories at once. For example, in a book recommendation application, a user may be interested in one topic for work purposes and a completely different topic for leisure.

Several other model-based collaborative recommendation approaches [36] have been proposed in the literature. Statistical model, Bayesian model, probabilistic relational models, linear regression models, and maximum entropy models have been developed to gain more ideal performance for collaborative recommendation.

6.4.3.5 Hybrid recommendation

Several service recommendation systems use a hybrid approach by combining collaborative and content-based methods, which helps to avoid certain limitations of content-based and collaborative systems. Different ways to combine collaborative and content-based methods into a hybrid recommendation system can be classified as follows:

Combining separate recommendations [37]
One way to build hybrid recommendation systems is to implement separate collaborative and content-based systems. Then we can have two different scenarios. First, we can combine the ratings obtained from individual recommendation systems into one final recommendation using either a linear combination of ratings or a voting scheme [38]. Alternatively, we can use one of the individual recommendations, at any given moment choosing to use the one that is "better" than others based on some recommendation "quality" metric.

Adding content-based characteristics to collaborative models
Several hybrid recommendation systems, including Fab and the "collaboration via content" approach, are based on traditional collaborative techniques but also maintain the content-based profiles for each user. These content-based profiles, and not the commonly rated services, are then used to calculate the similarity between two users. This type of methods can overcome some sparsity-related problems of a purely collaborative approach, and users can be recommended with the service not only when this service is rated highly by users with similar profiles, but also when this service scores highly against the user's profile.

Adding collaborative characteristics to content-based models
The most popular approach in adding collaborative characteristics to content-based models is using some dimensionality reduction technique on a group of content-based profiles. For example, we can use latent semantic indexing to create a collaborative view of a collection of user profiles, where user profiles are represented by term vectors, resulting in a performance improvement compared to the pure content-based approach.

Developing a unifying recommendation model [39]
Many researchers have followed this approach in recent years, such as single rule-based classifiers based on content-based and collaborative characteristics, unified probabilistic methods based on the probabilistic latent semantic analysis, Bayesian mixed-effects regression models based on Markov chain Monte Carlo methods, and so on.

Hybrid recommendation systems can also be augmented by knowledge-based techniques in order to improve recommendation accuracy and to address some of the limitations (e.g., new user, new service problems) of traditional recommendation systems. For example, we can use some domain knowledge about restaurants, cuisines, and foods to recommend restaurants to its users. The main drawback of knowledge-based systems is the need for knowledge acquisition—a well-known bottleneck for many artificial intelligence applications. However, knowledge-based recommendation systems have been developed for application domains where domain knowledge is readily available in some structured machine-readable form, for example, as an ontology. For example, Quickstep and Foxtrot systems use research paper topic ontology to recommend online research articles to the users.

6.4.3.6 Mobile recommendation systems

Mobile recommendation system is the hot spot of the current research. With the increasing ubiquity of Internet-accessing smart phones, it is now possible to offer personalized, context-sensitive service recommendations. This is particularly a difficult area of research as mobile data are complex, heterogeneous, and noisy, and these require spatial and temporal autocorrelation. Mobile recommendation systems have to deal with its validation and generality problems.

One example of a mobile recommendation system is one that offers potentially profitable driving routes for taxi drivers in a city. This system [40] takes as input data in the form of GPS traces of the routes that taxi drivers took while working, which include location (latitude and longitude), time stamps, and operational status (with or without passengers). It then recommends a list of pickup points along a route that will lead to optimal occupancy times and profits. This type of system is obviously location-dependent, and as it must operate on a handheld or embedded device, the computation and energy requirements must remain low.

Another example of mobile recommendation is the system developed for professional users. This system takes as input data the GPS traces of the user and his agenda to suggest him suitable information depending on his situation and interests. The system uses machine learning techniques and reasoning process in order to adapt dynamically the mobile recommendation system to the evolution of the user's interest.

Mobile recommendation systems have also been successfully built using the Web of Data as a source for structured information. A good example of such system is SMARTMUSEUM [41]. The system uses semantic modeling, information retrieval, and machine learning techniques in order to recommend services matching user's interest, even when the evidence of user's interests is initially vague and based on heterogeneous information.

6.4.4 Service recommendation summary

There has been much research done on service recommendation over the past several years that have used a broad range of statistical, machine learning, information retrieval, and other techniques and that significantly advanced the state of the art in comparison to early recommendation systems that utilized collaborative- and content-based heuristics. However, both the content-based recommendation and collaborative-based recommendation have their own limitations.

6.4.4.1 Limitation of content-based recommendation

New user problem
A user has to rate a sufficient number of services before the system can really understand the user's tastes and preferences and present the user with reliable recommendations. Therefore, a new user, having very few ratings of services, would not be able to get accurate recommendations.

Limited content analysis
Content-based techniques are limited by the features that are explicitly associated with the objects these systems recommend. Therefore, in order to have a sufficient set of features, the content must either be in a form that can be parsed automatically by a computer, or the features should be assigned to services manually. While information retrieval techniques work well in extracting features from text documents, some other domains have an inherent problem with automatic feature extraction. Moreover, it is often not practical to assign attributes by hand. Another problem is that, if two different services are represented by the same set of features, they are indistinguishable in content-based recommendation systems.

Overspecialization
When the system can only recommend services that score highly against a user's profile, the user is limited to being recommended services similar to those already rated, and it cannot recommend services that are different from anything the user has seen before. Therefore, some content-based recommendation systems filter out services not only if they are too different from user's preferences, but also if they are too similar to something the user has seen before. In summary, the *diversity* of recommendations is often a desirable feature in recommendation systems. Ideally, the user should be presented with a range of options and not with a homogeneous set of alternatives.

6.4.4.2 Limitation of collaborative recommendation

New user problem
It is the same problem as with content-based systems. In order to make accurate recommendations, the system must first learn the user's preferences from the ratings that the user makes. Several techniques have been proposed to address this problem.

New service problem
New services are added regularly to recommendation systems. Collaborative systems rely solely on users' preferences to make recommendations. Therefore, until the new

service is rated by a substantial number of users, the recommendation system would not be able to recommend it.

Sparsity
In any recommendation system, the number of ratings already obtained is usually very small compared to the number of ratings that need to be predicted. Effective prediction of ratings from a small number of examples is important [42,43]. Also, the success of the collaborative recommendation system depends on the availability of a critical mass of users.

6.4.4.3 Summary of the recommendation systems research

As discussed above, recommendation systems can be categorized as being (a) content-based, collaborative-based or hybrid-based recommendation approach, and (b) heuristic-based or model-based recommendation techniques used for the rating estimation. We use these two orthogonal dimensions to classify the recommendation systems research as shown in Table 6.3.

6.5 Evaluation index

There are several major indexes that are commonly considered when comparing different selection or recommendation approaches. As different applications have different needs, the designer of the system must decide on the important index to measure the concrete application at hand, and some of the indexes can be traded off. Therefore, when suggesting a method that improves one of the indexes, it should evaluate the influence of user experience affected by changes in this index either through a user study or through online experimentation.

6.5.1 User preference

When we need to choose one out of a set of candidate services, an obvious option is to run a user study (within subjects) and ask the participants to choose one of the candidates. The service with the largest number of votes is the ideal one to recommend to users. And we need to further weight the vote by the importance of the user and provide nonbinary answers for the preference question in the user study.

6.5.2 Accuracy

A basic assumption in a selection/recommendation system is that a system providing more accurate selections/predictions will be preferred by the user. The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) is the most popular metric used in evaluating accuracy of predicted ratings. The RMSD represents the sample standard deviation of the differences between predicted values and actual values. The RMSD serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. At the same time, we could measure the diversity of a selection/recommendation list based on the sum, average, min, or max distance between service pairs, or measure the value of adding each

Table 6.3 Classification of service recommendation system research

Recommendation approach		Techniques	Papers
Content-based	Heuristic-based method	TF-IDF	Zhu *et al.* (2013) [31]
		KNN algorithm	Deldjoo *et al.* (2016) [44]
		Clustering	
	Model-based method	Bayesian classifiers clustering	Liu *et al.* (2013) [33]
		Artificial neural networks	Meehan *et al.* (2013) [34]
Collaborative	Memory-based method	KNN algorithm	Kim *et al.* (2010) [32]
		Nearest neighbor clustering	Chen *et al.* (2015) [35]
		Graph theory	Park *et al.* (2015) [45]
	Model-based method	Bayesian networks	Breese *et al.* (1998) [36]
		Artificial neural networks	Wei *et al.* (2016) [46]
		Linear regression	Nilashi *et al.* (2015) [47]
		Markov process	
Hybrid	Feature combining	Bayesian clustering	Sattari *et al.* (2015) [37]
	Recommendation result combining	Linear combination	Dooms *et al.* (2015) [48]
	Unique models	Probabilistic model	He *et al.* (2016) [39]
		Maximum entropy	

service to the selection/recommendation list as the new service's diversity from the services already in the list.

6.5.3 Coverage

The term coverage can refer to several distinct properties of the system as follows:

Service space coverage, which refers to the proportion of services that the recommendation system can recommend; this is often referred to as catalog coverage; the measure of catalog coverage may be the percentage of all services that can ever be recommended or the sales diversity which measures how unequally different services are chosen by users when a particular recommendation system is used;

User space coverage, which refers to the proportion of users or user interactions for which the system can recommend services; coverage can be measured by the richness of the user profile required to make a recommendation. For example, in the collaborative filtering case, this could be measured as the number of services that a user must rate before receiving recommendations;

Cold start, which concerns the issue that the system cannot make any recommendation for users when the system has not yet gathered sufficient information; a more generic way is to consider the "coldness" of a service using either the amount of time it exists in the system or the amount of data gathered for it.

6.5.4 Confidence and trust

Confidence refers to the system's trust in its selection or recommendations. The most common measurement of confidence is the probability that the predicted value is indeed true, or the interval around the predicted value [49,50]. For example, a recommendation may accurately rate a movie as a four-star movie with probability 0.85. The most general method of confidence is to provide a complete distribution over possible outcomes. Trust refers to the user's trust in the selection/recommendation system. The most common method for evaluating user trust on the recommendation system is by asking users whether the system recommendations are reasonable in a user study. In an online test, one could associate the number of recommendations followed with the trust in the recommendation, assuming that higher trust in the recommendation would lead to more recommendations being used. Alternatively, we could also assume that trust in the system is correlated with repeated users.

6.5.5 Robustness and privacy

Robustness is the stability of the recommendation in the presence of fake information, typically inserted on purpose in order to influence the recommendations. Such attempts to influence the recommendation are typically called attacks. It is more useful to estimate the cost of influencing a recommendation, which is typically measured by the amount of injected information. We can simulate a set of attacks by introducing fake information into the system data set, empirically measuring average cost of a successful attack. Privacy refers to the right that the private messages of users are not infringed. It is important for most users that their preferences stay private, that is, no third party can use the recommendation system to learn something about the preferences of a specific user. It is generally considered inappropriate for a service recommendation system to disclose private information even for a single user.

6.6 Conclusion

This chapter introduces the background and characteristics of service selection and recommendation in integrated network. Then it details the scenarios and definitions of service selection and recommendation, analyzes various algorithms, techniques, and strategies of service selection and recommendation, and finally summarizes limitations and evaluation indexes for service selection and recommendation approaches.

References

[1] Guidara I., Guermouche N., Chaari T., Tazi S. 'Heuristic based time-aware service selection approach'. *Proceedings of International Conference on Web Services*; New York, USA, Jun 2015 (New York, IEEE, 2015), pp. 65–72.

[2] Ma Y., Wang S.G., Yang F.C., Chang R.N. 'Predicting QoS values via multi-dimensional QoS data for web service recommendations'. *Proceedings of*

IEEE International Conference on Web Services; New York, USA, Jun 2015 (New York, IEEE, 2015), pp. 249–256.

[3] Guo Y., Wang S.G. 'Skyline service selection based on QoS prediction'. *Proceedings of IEEE International Conference on Cluster Computing*; Taipei, Taiwan, Sep 2016 (New York, IEEE, 2016), pp. 150–151.

[4] Wang S.G., Hsu C-H., Liang Z.J., Sun Q.B., Yang F.C. 'Multi-user web service selection based on multi-QoS prediction'. *Information Systems Frontiers*. 2014;16(1): 143–152.

[5] Wang S.G., Zhou A., Hsu C.-H., Xiao X.Y., Yang F.C. 'Provision of data-intensive services through energy- and QoS-aware virtual machine placement in national cloud data centers'. *IEEE Transactions on Emerging Topics in Computing*. 2016;4(2): 290–300

[6] Wang S.G., Lei T., Zhang L.Y., Hsu C.-H., Yang F.C. 'Offloading mobile data traffic for QoS-aware service provision in vehicular cyber-physical systems'. *Future Generation Computer Systems*. 2016;61: 118–127

[7] Wang S.G., Sun Q.B., Zou H, Yang F.C. 'Web service selection based on adaptive decomposition of global QoS constraints in ubiquitous environment'. *Journal of Internet Technology*. 2011;12(5): 757–768.

[8] Yu H.Q., Reiff-Marganiec S. 'Non-functional property based service selection: a survey and classification of approaches'. *Proceedings of Non-Functional Properties and Service Level Agreements in SOC Workshop*; Dublin, Ireland, Nov 2008 (New York, IEEE, 2008), pp. 12–14.

[9] Sun Q.B., Wang S.G., Zou H, Yang F.C. 'QSSA: a QoS-aware service selection approach'. *International Journal of Web and Grid Services*. 2011;7(2): 147–169.

[10] Wang S.G., Zheng Z.B., Sun Q.B., Zou H, Yang F.C. 'Reliable web service selection via QoS uncertainty computing'. *International Journal of Web and Grid Services*. 2011;7(4): 410–426.

[11] Schröpfer C., Binshtok M., Shimony S.E., *et al.* 'Introducing preferences over NFPs into service selection in SOA'. *Proceedings of International Conference on Service-Oriented Computing*. 2007; Vienna, Austria, Sep 2007 (Springer, Berlin, 2009), pp. 68–79.

[12] Mardukhi F., Nemat Bakhsh N., Zamanifar K., Barati A. 'QoS decomposition for service composition using genetic algorithm'. *Applied Soft Computing*. 2013;13(7): 3409–3421.

[13] Ma Y., Zhang C. 'Quick convergence of genetic algorithm for QoS-driven web service selection'. *Computer Networks*. 2008;52(5): 1093–1104.

[14] Li F., Huang Y. 'An web service selection optimization method based on particle swarm optimization'. *Proceedings of International Conference on Computer Design and Applications*; Qinhuangdao, China, Jun 2010 (New York, IEEE, 2010), pp. V2–V477. AQ1

[15] Wang S.G., Sun Q.B, Zou H., Yang F.C. 'Particle swarm optimization with skyline operator for fast cloud-based web service composition'. *Mobile Networks and Applications*. 2013;18(1): 116–121.

[16] Wang R., Ma L., Chen Y. 'The research of Web service selection based on the ant colony algorithm'. *Proceedings of International Conference on Artificial Intelligence and Computational Intelligence*; Sanya, China, Oct 2010 (New York, IEEE, 2010), pp. 551–555.

[17] Yang Z., Shang C., Liu Q. Zhao C. 'A dynamic web services composition algorithm based on the combination of ant colony algorithm and genetic algorithm'. *Journal of Computational Information Systems*. 2010;6(8): 2617–2622.

[18] Wang S.G., Sun Q.B., Yang F.C. 'Towards web service selection based on QoS estimation'. *International Journal of Web and Grid Services*. 2010;6(4): 424–443.

[19] Sun L., Wang S.G., Li J.L., Sun Q.B., Yang F.C. 'QoS uncertainty filtering for fast and reliable web service selection'. *Proceedings of IEEE International Conference on Web Services*; Alaska, USA, Jun 2014 (New York, IEEE, 2014), pp. 550–557.

[20] Gao Y., Na J., Zhang B., Yang L., Gong Q. 'Optimal web services selection using dynamic programming'. *Proceedings of IEEE Symposium on Computers and Communications*; Sardinia, Italy, Jun 2006 (Washington, IEEE, 2006), pp. 365–370.

[21] Ding K., Deng B., Zhang X., Ge L. 'Optimization of service selection algorithm for complex event processing in enterprise service bus platform'. *Proceedings of International Conference on Computer Science & Education*; Nanning, China, Jul 2009 (New York, IEEE, 2009), pp. 582–586.

[22] Ardagna D., Pernici B. 'Adaptive service composition in flexible processes'. *IEEE Transactions on Software Engineering*. 2007;33(6): 369–384.

[23] Wang S., Zheng Z., Sun Q., Zou H., Yang F. 'Cloud model for service selection'. *Proceedings of IEEE International Conference on Computer Communications Workshops*; Shanghai, China, Apr 2011 (New York, IEEE, 2011), pp. 666–671.

[24] Wang S., Sun L., Sun Q., Li X., Yang F. 'Efficient service selection in mobile information systems'. *Mobile Information Systems*. 2015;2015(1): 1–10.

[25] Tang M., Ai L. 'A hybrid genetic algorithm for the optimal constrained web service selection problem in web service composition'. *Proceedings of IEEE Congress on Evolutionary Computation*; Barcelona, Spain, Jul 2010 (New York, IEEE, 2010), pp. 1–8.

[26] Ma Y., Xin X, Wang S.G., Li J.L., Sun Q.B., Yang F.C. 'QoS evaluation for web service recommendation'. *China Communications*. 2015;12(4): 151–160

[27] Wang S.G., Ma Y., Cheng B., Yang F.C., Chang R.N. 'Multi-dimensional QoS prediction for service recommendations'. *IEEE Transaction on Services Computing*. 2016;99: 1–1 AQ2

[28] Su X.Y., Khoshgoftaar T.M. 'A survey of collaborative filtering techniques'. *Advances in Artificial Intelligence*. 2009;2009(12): 4. AQ3

[29] SG12 I. 'Definition of quality of experience'. *TD 109rev2 (PLEN/12)*; Geneva, Switzerland, 2007. pp. 16–25.

[30] Ricci F. 'Introduction to recommender systems handbook', in Rokach L. (ed.). *Recommender System Handbook*. New York: Springer; 2011. pp. 1–35.

[31] Zhu Q., Shyu M.L., Wang H. 'Videotopic: content-based video recommendation using a topic model'. *Proceedings of International Conference on Multimedia*; Anaheim, CA, USA, Dec 2013 (New York, IEEE, 2013), pp. 219–222.

[32] Kim H.N., Ji A.T., Ha I., Jo G.S. 'Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation'. *Electronic Commerce Research and Applications*. 2010;9(1): 73–83.

[33] Liu J., Wu C., Liu W. 'Bayesian probabilistic matrix factorization with social relations and item contents for recommendation'. *Decision Support Systems*. 2013;55(3): 838–850.

[34] Meehan K., Lunney T., Curran K., McCaughey A. 'Context-aware intelligent recommendation system for tourism'. *Proceedings of International Conference on Pervasive Computing and Communications Workshops*; San Diego, CA, USA, Mar 2013 (New York, IEEE, 2013), pp. 328–331.

[35] Chen M.H., Teng C.H., Chang P.C. 'Applying artificial immune systems to collaborative filtering for movie recommendation'. *Advanced Engineering Informatics*. 2015;29(4): 830–839.

[36] Breese J.S., Heckerman D., Kadie C. 'Empirical analysis of predictive algorithms for collaborative filtering'. *Proceedings of Conference on Uncertainty in Artificial Intelligence*; Madison, WI, USA, Jul 1998 (San Francisco, Morgan Kaufmann Publishers, 1998), pp. 43–52.

[37] Sattari M., Toroslu I.H., Karagoz P., Symeonidis P., Manolopoulos Y. 'Extended feature combination model for recommendations in location-based mobile services'. *Knowledge and Information Systems*. 2015;44(3): 629–661.

[38] Claypool M., Gokhale A., Miranda T., Murnikov P., Netes D., Sartin M. 'Combining content-based and collaborative filters in an online newspaper'. *Proceedings of ACM SIGIR Workshop on Recommender Systems*; Berkeley, CA, USA, Aug 1999 (New York, ACM, 1999), pp. 1–8.

[39] He P., Yuan H., Chen J., Zhao C. 'An effective and scalable algorithm for hybrid recommendation based on learning to rank'. *Proceedings of International Congress on Signal and Information Processing, Networking and Computers*; Beijing, China, Oct 2015 (Boca Raton, CRC Press, 2016), p. 59.

[40] Ge Y., Xiong H., Tuzhilin A., Xiao K., Gruteser M., Pazzani M. 'An energy-efficient mobile recommender system'. *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; Washington, DC, USA, Jul 2010 (New York, ACM, 2010), pp. 899–908.

[41] Ruotsalo T., Haav K., Stoyanov A., *et al.* 'SMARTMUSEUM: a mobile recommender system for the Web of data'. *Web Semantics*. 2013;20(2): 50–67.

[42] Ma Y., Wang S.G., Hung P.C.K., Hsu C.-H., Sun Q.B., Yang F.C. 'A highly accurate prediction algorithm for unknown web service QoS value'. *IEEE Transactions on Services Computing*. 2016;9(4): 511–523.

[43] Sun Q.B., Wang L.B., Wang S.G., Ma Y., Hsu C.-H. 'QoS prediction for web service in mobile internet environment'. *New Review of Hypermedia and Multimedia*. 2016;22(3): 207–222.

[44] Deldjoo Y., Elahi M., Cremonesi P., Garzotto F., Piazzolla P., Quadrana M. ‘Content-based video recommendation system based on stylistic visual features’. *Journal on Data Semantics*. 2016;5(2): 99–113.

[45] Park Y., Park S., Jung W., Lee S.G. ‘Reversed CF: a fast collaborative filtering algorithm using a k-nearest neighbor graph’. *Expert Systems with Applications*. 2015;42(8): 4022–4028.

[46] Wei J., He J., Chen K., Zhou Y., Tang Z. ‘Collaborative filtering and deep learning based recommendation system for cold start items’. *Expert Systems with Applications*. 2016;69(1): 29–39.

[47] Nilashi M., Jannach D., Bin Ibrahim O., Ithnin N. ‘Clustering-and regression-based multi-criteria collaborative filtering with incremental updates’. *Information Sciences*. 2015;293(293): 235–250.

[48] Dooms S., De Pessemier T., Martens L. ‘Online optimization for user-specific hybrid recommender systems’. *Multimedia Tools and Applications.* 2015;74(24): 11297–11329.

[49] Wang L.B., Sun Q.B., Wang S.G., Ma Y., Xu J.L., Li J.L. ‘Web service QoS prediction approach in mobile internet environments’. *Proceedings of IEEE International Conference on Data Mining*; Shenzhen, China, Dec 2014 (New York, IEEE, 2015), pp. 1239–1241

[50] Ma Y., Wang S.G., Sun Q.B., Zou H, Yang F.C. ‘Predicting unknown QoS value with QoS-prophet’. *Proceedings Demo & Poster Track of ACM/IFIP/USENIX International Middleware Conference*; Beijing, China, Dec 2013 (New York, ACM, 2013), pp. 1–2.

Chapter 6
Service selection and recommendation in integrated network environment

Author Queries

AQ1	Please check the page range in Reference [14], and amend if necessary.
AQ2	Please check the volume number and page range in Reference [27], and amend if necessary.
AQ3	Please provide last page for Reference [28], if available.