Chapter 3

Recommender Systems – Functional Perspectives

This chapter gives an overlook of the functional aspects of recommender systems. It deals with functional input and output of recommender systems, measurement scales for preference elicitation, information delivery aspects, and recommendation methods.

Input data of recommender systems are described in Section 3.1. Input data can be classified along the dimensions duration, acquisition, originator and origin.

Section 3.2 deals with output data of recommender systems. Besides the recommendations itself, recommender systems may display predictions, text comments and ratings to the user. Further, possible approaches to provide supplementary explanations (i.e. why certain products are recommended) are presented. Finally, the basic flow of input and output data in e-commerce recommendation applications is illustrated.

Section 3.3 examines different statistical measurement scales. It focuses on metric scales for the elicitation of user preferences.

Section 3.4 refers to the information delivery of recommender systems. Push, pull and passive technologies may be used to suggestions, ratings and predic-
tions to the user. Push, pull and passive technologies refer to the extent of the user's initiative to get recommendations.

The chapter concludes with the introduction of different recommendation methods. Personalized and non-personalized recommendation methods and their corresponding advantages and disadvantages are described in detail.

3.1 Input Data of Recommender Systems

This section deals with the functional input data of recommender systems. Recommender systems use input data to generate output in form of suggestions, predictions and ratings. Figure 3.1 illustrates a classification scheme for input data of recommender systems.

![Classification of input data](image)

Figure 3.1: Classification of input data

Depending on duration of the user data storage, persistent data, ephemeral data, or a combination of both may be used for personalized recommendations [MT02]. *Ephemeral data* is used on a per session basis only and is deleted afterwards, whereas persistent input data is stored over different user interaction sessions. Thus, ephemeral personalization can be applied to users, who
3.1. **INPUT DATA OF RECOMMENDER SYSTEMS**

are not authenticated to the e-commerce application. It may be useful when users are new or are reluctant to give personal information to the e-vendor. For instance, the current navigation of an unregistered (i.e. anonymous) user could be used to push recommendations based on that context. *Persistent data* is acquired over different sessions and stored in user profiles permanently. Thus persistent data storage allows improving the user-profile over time and collecting long-term preferences of the users of the e-commerce application.

Acquisition denotes how the input data is gathered from user interaction. *Explicit data* is intentionally submitted by the user to inform the recommender systems about his preferences (e.g. rating items on a nominal scale), whereas *implicit data* stems from monitoring user behavior (e.g. browsing the product catalogue) [SKR01]. In this context data acquisition is related to user awareness. This denotes the extent to which the user is required to give inputs to the recommender system intentionally. Consequently user awareness refers to the user’s state of mind while interacting with the e-commerce application.

The advantage of explicit approaches is that the users know their interest best and are in control of the recommendation process. However, explicit approaches put the effort of adapting recommendations towards the users. Further, the users have to learn to handle the input forms of the recommender system. Thence, complexity is increased from the users’ point of view. Consequently the user-interfaces of recommender systems, which are operated by non-specialists per definition, have to be designed carefully in respect of usability.

The pros of implicit approaches are that no or little effort is put towards the users and that no special knowledge of the user is required. But the user loses control over the recommendation process. Further implicit approaches reduce transparency of recommendations, i.e. the user does not understand how recommendations are generated. Thus it is difficult for the user to develop a coherent cognitive model of the recommender system [Wae04].

*User interrogation* is the most commonly used explicit data acquisition approach. The user is required to fill out forms to describe interests or other relevant parameters (e.g. keywords and attributes of items). User interrogation is often applied to obtain ratings of items the user has already knowledge
CHAPTER 3. RECOMMENDER SYSTEMS – FUNCTIONAL PERSPECTIVES

of. These ratings may be based on an ordinal scale (e.g. "rate this item on a scale from one to five") or on a binary scale (e.g. "do you like this item - yes or no").

Recording user behavior is an typical implicit approach. It does not require the user to intentionally engage in the data acquisition process. A simple approach would be to give recommendations based on the item the user is currently viewing. In e-commerce applications the articles in the virtual shopping-basket, the articles bought in the past or other clickstream-data can be utilized for recommendation purposes. According to studies from Morita and Shinoda as well as Konstan et al. the time a user spends viewing an artefact is a appropriate indicator for the relevance to the user [MS94, KMM+97]. Hence, time spent to view articles can be used as implicit input data for recommendations, although this data may be biased (e.g. the user is interrupted).

Explicit and implicit approaches may be combined. Usually these combined methods use explicit approaches to gain knowledge about the user in the initial phase of the system use and change over to explicit approaches in later phases. For instance reference items can be used to create an initial item space (also referred to as document space because this method was first applied on textual documents [FD92]). The user has to judge the relevancy of these reference items by explicit user interrogation. New items are compared to these reference items and are recommended if the similarity to these reference items, which were rated as relevant, exceeds a certain threshold. The advantage of this method is, that the effort of user interrogations is limited to the initial system use. However, from the user's point of view it is hard to estimate the usefulness in the beginning of the system use. Hence, he or she might not be willing to put effort into judging reference items, when he or she has little knowledge about the advantages of using the system. Additionally ongoing bias of the users interests may occur, if certain areas of interests are not covered by the initial item space [HSS01]. Stereotypic inference is another combined approach. A Stereotype is a collection of attributes that often co-occur in people, i.e stereotypes are typical characteristics of user groups in a given domain [Ric89]. Users are asked to provide personal information by explicit approaches in the initial phase of system use. These data is used to relate the user to a specific stereotype (i.e. default initial profile) [HSS01].
3.1. INPUT DATA OF RECOMMENDER SYSTEMS

This method helps to address the bootstrapping problem (i.e. giving suitable recommendations to new users). Consequently, stereotypes enable the recommender system to make plausible inferences on the basis of a substantially smaller number of observations of the user’s behavior. Over time observations are added to the profile, which may enhance or override default assumptions about the user.

A further criterion to classify input data of recommender systems is the originator of the input data, whereby active user input, community input and input from others (e.g. editors, critics) can be distinguished.

*Active user input* refers to the data generated through interactions with the active user (i.e. the user who currently gets recommendations). Active user data typically include:

- Session information (e.g. log-in and log-out times, session-identification numbers, navigational-data)
- Buying behavior (e.g. items in the virtual shopping-basket, items bought or consumed in the past)
- Search behavior (e.g. keywords, queries)
- Transactional information (e.g. forms of payment, account numbers, shipping address)
- Preferences (e.g. expressed preferences, implicit preferences)
- Individual characteristics (e.g. demographic data)

*Community inputs* usually refer to the sum of all active user inputs. Besides those internal data (see below) community inputs may also include external data (e.g. item popularity in form of national best-seller lists). Generally spoken, community inputs comprise of data, which denotes how multiple individuals in the community or the community as a whole perceive attributes of items (e.g. book categories or film genres are derived from the consensus of the broader society) [SKR01].
Text comments are community inputs in form of textual descriptions of users' experiences with single products or services. Text comments may be very useful to enhance the decision making process of the active user. However, the user's effort of processing text comments fairly high, since the user must read this textual information and interpret to what degree these comments contain positive and negative attitudes toward the item.

To ease this procedure, textual comments are often supplemented by scores or ratings of users, which indicate the overall satisfaction with the item. Additionally, these individual ratings can be summarized (e.g. by calculating the arithmetic mean) to get an quick overview of the users' average opinion.

Finally, depending on the source, input data can be classified into internal and external data. External data stem from third parties and may relate to items or users. For instance item-specific external data may be derived from third party electronic product catalogues with categorizations and descriptions of product attributes (e.g. genre and keyword classifications of books or films) [SKR01]. External item popularity (e.g. national best-seller list) is a further example for item-specific external data used for recommendation purposes. Typical user-specific external data stem from market research companies (e.g. general demographic data of online-shoppers) and may also be used in the recommendation process. In contrast to external data internal data is exclusive to the e-commerce vendor. Thus, it is of major importance in regard to competitive advantages. Internal data is often generated automatically by the user's interaction with the e-commerce data (e.g. clickstream-data), but may also be rendered manually (e.g. broad recommendation lists based on editors' manual selections).

3.2 Output Data of Recommender Systems

The outputs of recommender systems are suggestions of items (i.e. products and services). Additionally, the may display ratings, text comments, predictions and explanations.

Suggestions make the user of recommendations systems aware of items that
3.2. OUTPUT DATA OF RECOMMENDER SYSTEMS

the e-vendor considers as useful to the customer. Phrases like "we recommend...", "try this". Other phrases ("additional products", "supreme products") are used to indicate the cross- and up-sell potential of certain items. Recommender systems may suggest either only one item or may display multiple items to the user. When a set of items is recommended by lists, the order of items may be arbitrary, which means that the sequence of items does not reflect any order of preference for the user (e.g. alphabetical). In the other case, the order of items may indicate predictions of the degree of interest to the user (i.e. the first item on the list is the best-fit recommendation).

Predictions are estimates of ratings, the user would give to items. They quantify, how much a user will probably like the recommended item and hence indicate the strength of an recommendation. Predictions may be personalized, which means that they are based on the stored preferences in an individual user-profile. Non-personalized predictions refer to estimates for typical community members [SKR01].

Text comments and ratings constitute further possible output data of recommender systems. Suggestions of items may be supplemented by text comments. Because text comments are not completely machine-understandable, many e-vendors require the user to give an additional numerical rating to indicate the direction of the comment (i.e. pro or against the item). Especially, when the size of the community is large and the number of text reviews is high, the recommender system has to display a selection of text comments, because showing all text comments would lead to information overload. The selection of text comments bears the risk of biasing information (e.g. only positive comments are shown to the user). Hence, accompanying numerical ratings can be used to show an representative selection of comments to the user by choosing a proportional number of positive, neutral and negative comments. Another notion to address the problem of selecting text comments is to apply "meta-ratings". Meta-ratings are ratings about ratings (respectively text comments). This means, that the usefulness and quality of text comments from the community are judged by the community (e.g. "Was this review helpful to you?"). In this case, the most appreciated text comments are displayed first.

In recommender systems explanations can be used to expose the reasoning behind an recommendation. They enhance transparency in the recommenda-
tation process. Thus, they may raise the user's trust in the recommendation process and may also improve the decision-making performance. The benefits of adding explanation capabilities to recommender systems are [Her99]:

- **Justification**: The users get an understanding of the reasoning behind the recommendation. This alleviates the decision of how much confidence to place in a recommendation.

- **User involvement**: User involvement is improved, because explanations allow the user to add his knowledge and inference skills more easily to the recommendation.

- **Education**: The user will better understand, how recommendations are generated as well as strengths and limitations of the system.

- **Acceptance**: Explanations raise the acceptance of the system as a decision aide, because strengths and limitations are better understood and suggestions are justified.

Since recommendation methods range from relative simple to highly complex with large amounts of data and extensive computation (see Section 3.5), the provision of explanations may also vary in terms of complexity. Three possible models for explanations are applicable [Her99]:

- **Data-explorative model**: When this model is applied, the application lets the user explore the data on which recommendations are based. Mathematical processes behind the recommendations are not explained (e.g. because they are too complex for the "average" user). Because some recommendation methods use large amounts of data, initially only a selection of key-data are displayed to the user. Key-data are of significant relevance for the recommendation process. However, the user can navigate to other parts of the data. The data-explorative model allows the user to validate the recommendation by their own personal approaches. For instance amazon.com applies this model. The user may click on a link labeled "Why was I recommended this?" to see the relevant items for the recommendation process as shown in figure 3.2.
3.2. OUTPUT DATA OF RECOMMENDER SYSTEMS

Figure 3.2: Explanations using the data-explorative model

- Process-explorative model: In this case, the recommender system tries to explain the mathematical process on a high level. For example, flowcharts may be used to visualize the process-steps of recommendation process. The user may take a closer look at the individual steps and change the computation by altering parameters.

- Argumentative model: In this model, the explanation facility of the recommender systems works as an agent that uses logical argument techniques to support a conclusion. The system makes claims at multiple stages. The user can challenge the inference and data of each claim. In this model the amount of data processed by the user is minimized.
Figure 3.3: Flow of input and output data

Figure 3.3 summarizes the basic flow of input and output data in e-commerce recommendation applications and denotes the software components of a typical recommender system.

As illustrated in the figure, an e-store component is in charge of the information delivery to the user by applying push or pull technologies. The component also forwards the interaction data to the user model builder, which constructs a long-term and/or short-term user profile and stores the user profile(s) in a database.

The user profile stored in the database is employed by the recommender component. This component generates the suggestions, predictions, and explanations and summarizes ratings by applying recommendation methods (e.g. collabora-
3.3 Measurement Scales for Preference Elicitation

Preferences of users are the most important data for recommendation systems. They are generally used as input data but – as mentioned in Section 3.2 – may also be displayed as an output in form of predictions. To measure or indicate these preferences, different statistical measurement scales can be applied. Measurement scales can be categorized into nonmetric (qualitative) and metric (quantitative) scales [HATB98, BEPW03].

Nonmetric scales include nominal scales, binary scales and ordinal scales. Nominal scales are classifications of qualitative attributes, characteristics or properties (e.g. gender, color). Binary scales are a sub-type of nominal scales with exact two possible occurrences of an attribute (e.g. yes or no, male or female, zero or one). Nominal and binary scales are the scales with the lowest level of measurement precision. Arithmetical operations can not be applied to nominal and binary scales, but it is possible to calculate the absolute and relative frequency of an attribute. With ordinal scales variables can be ordered or ranked, i.e. attributes can be compared by “greater than” or “less than” relationships. The ranking of variables is relative. However, it is not possible to determine the distance between two occurrences of a variable. Similar to nominal scales it is not possible to use any arithmetic operation. However, additional to absolute and relative frequency, quantile and median can be calculated.

Interval scales and ratio scales are both metric scales, which refer to quantitative measurable attributes (e.g. amount of time, size of an object, temperature). Metric scales have constant units of measurement, i.e. the distances between two adjacent points are equal on any part of the scale [HATB98]. Interval scales have arbitrary zero points (e.g. temperature in Fahrenheit or
Celsius). Possible arithmetical operations for transformations of the scale are addition or subtraction. Feasible statistical operations are (amongst others) to calculate the mean value and standard deviation. Interval scales are widely used for measuring preferences explicitly (e.g. "rate this item on a scale from one to five"). These scores and ratings are regularly assumed to be based on interval scales. However, strictly speaking, ratings rest upon ordinal scales, because it can not be assumed, that equal distances between two adjacent points on the scale are given on any part of the scale. In spite of this, ratings are predominantly treated as interval scales (e.g. building the mean value of all user ratings) [BEPW03]. In contrast to interval scales, ratio scales have an absolute zero point (e.g. weight, length, speed). They represent the highest form of measurement precision and all arithmetical operations are allowed. In the context of recommender systems, ratio scales are preferably used when preferences are surveyed by means of implicit data acquisition methods (e.g. time spend viewing an item).

### 3.4 Information Delivery

The output of recommender systems (i.e. suggestions, ratings, text comments and predictions) may be transferred to the user by push, pull and passive information delivery techniques.

**Push technologies** refer to methods, where the suggestions are given to the user without requiring the users' initiative, i.e. the recommender systems initiates the communication process [MGL97]. A distinctive example for push communication is the use of e-mails to send recommendations to users on a regular basis (e.g. fixed time schedule). This has the advantage of giving recommendations to users without requiring them to interact with the e-commerce application. They can be understood as an promotional activity to invite users to return to the e-commerce vendor. However, if the user is not satisfied with the recommendations (e.g. due to lack of personalization) he or she might consider the e-mails mentioned in the example above as spam.

**Passive technologies** denote information delivery, which supplements the presentation of recommendations to the normal use (i.e. "the natural context")
of the e-commerce application [SKR01]. Hence, the might be understood as a sub-class of push technologies. For instance, recommendations are displayed based on the item the user is currently browsing. Another example for passive delivery is the presentation of supplemental goods or special shipping options during the ordering process. At this time the user may be very receptive to the vendors idea of up- and cross-selling. A possible disadvantage of passive recommendations is that the user might not recognize them as recommendations [SKR01].

In contrast pull technologies require the user to take initiative to get recommendations. In e-commerce applications these is usually achieved by clicking on a link (e.g. “your recommendations”). Pull technologies are usually perceived as unobtrusive, because no recommendations are displayed unless the user wants them to see.

### 3.5 Recommendation Methods

This Section focuses on specific recommendation methods. Recommendation methods can be classified according to the degree of personalization. Methods for non-personalized recommendations do not refer to individual user profiles. Thus they give identical recommendations to different users. Methods for personalized recommendation refer to individual user profiles, which may be based on persistent or ephemeral data. Consequently they offer recommendations adapted to the individual user.

Figure 3.4 gives an overview of varying degrees of personalization of recommendations regarding (1) the target of recommendations, (2) the typical recommendation method(s) applied, (3) the characteristical data acquirement method, and (4) the deployment of user-profiles.

**General recommendations** are suggestions that are given to all users of a recommender system. Typical recommendation methods are statistical summarization (e.g. Top sellers of all customers of an e-commerce application) and manual selection. Usually no user-specific information is necessary to give this kind of recommendations. As a consequence a user profile is not deployed.
CHAPTER 3. RECOMMENDER SYSTEMS – FUNCTIONAL PERSPECTIVES

<table>
<thead>
<tr>
<th>Type of Recommendation</th>
<th>Target</th>
<th>Typical Recommendation Method</th>
<th>Typical Data Acquisition</th>
<th>Application of User-Profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personalized Recommendation (Persistent, Long-Term Perspective)</td>
<td>Individual User</td>
<td>Information Filtering</td>
<td>Explicit and Implicit</td>
<td>Yes (persistent)</td>
</tr>
<tr>
<td>Personalized Recommendation (Ephemeral, Short-Term Perspective)</td>
<td>Individual User</td>
<td>Association Rules based on Items in Shopping Basket</td>
<td>Implicit</td>
<td>Yes (ephemeral)</td>
</tr>
<tr>
<td>Group-Specific Recommendation</td>
<td>Group of User</td>
<td>Statistical Summarization</td>
<td>Explicit</td>
<td>Yes (persistent)</td>
</tr>
<tr>
<td>General Recommendation</td>
<td>All Users</td>
<td>Manual Selection (e.g. Editors’ Picks) Statistical Summarization (e.g. Top-Sellers)</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>

Figure 3.4: Degrees of personalization

Group-specific recommendations are tailored towards a group of users. Usually statistical summarization is applied to generate recommendations for each group. Data acquisition usually takes place by explicit user interrogation (e.g. by offering fields of interest the user can specify, asking for demographic data). If the number of groups the users are segmented into is small, manual selection is also a possible alternative. Personalized recommendations with a short-term perspective are suggestions adapted to the individual user. However, a persistent personalization approach is not pursued. This is suitable, if a authentication of the user is not possible or desired. In e-commerce applications short-term personalized recommendations are often based on items in the virtual shopping basket. Based on this items, complementary items may be recommended to increase cross-sales. Product association rules may be used for this purpose. Ephemeral personalization regularly uses user-profiles to store user-related information. Albeit the profile may be discarded after the user quits the interaction session. Personalized recommendations with a long-term perspective are also adapted to the individual. Information filtering methods in conjunction with persistent user-profiles are typically used to achieve long-term personalization.

Figure 3.5 shows a categorization of recommendation methods based on the personalization criterion. Personalized and non-personalized recommendation methods as well as their advantages and disadvantages are discussed in detail in the following sections.
3.5. RECOMMENDATION METHODS

Figure 3.5: Classification of recommendation methods

3.5.1 Non-Personalized Recommendation Methods

Non-personalized methods do not adapt recommendations to the user. Hence, all users get identical recommendations. Non-personalized recommendation methods generally require little (statistical summarization) or no (manual selection) computational power. In regard to privacy these methods are less problematic, because mapping tastes, preferences, individual characteristics etc. to individual users is not necessary for the recommendation process.

Manual selection refers to the creation of lists of items to recommend by editors, critics, artists and other experts. These lists reflect the personal interests, tastes, preferences and objectives of these specialists and are made available to the community. These lists are regularly supplemented by text comments for the individual items to get a better understanding of the recommendations. This method does not require any machine computation at all. Manual selection is a traditional form of providing recommendations and has been used by magazines, newspapers etc. for a long time. By nature, manual recommendations are prone to bias, because they rely on a single persons preferences [SKR01]. However, because they are based on the opinion of experts they may offer deep insights to the items, especially when recommendations are accompanied by high quality text comments. Some e-stores encourage “normal” customers and community members respectively to create manual recommendation lists (e.g. “Listmania Lists” at Amazon.com). Links to spe-
specific customer generated lists may be displayed while browsing the product catalogue, if the current article is a part of these lists.

Statistical summarization denotes the aggregation of community opinions and community popularity. Typical examples of these summarizations are the number of community members, who like or purchase an item or the arithmetic mean of community ratings. A more complex method is to use association rules for recommendation purposes. Association rules may be applied on the shopping basket data (i.e. items purchased on a per-transaction basis) of e-stores [AIS93, SVA97]. A typical example for an association rules would be the finding, that 80 per cent of people, who bought the book “The Last Juror” by John Grisham also bought the “The Da Vinci Code” by Dan Brown. Association rules consist of three elements: (1) the antecedent (in this example “The Last Juror”) (2) the consequent (“The Da Vinci Code”) and (3) the confidence factor (“80 per cent”), which expresses the strength of the rule.

Table 3.1 shows a simple example of a customer-item matrix for basket data. The columns include different items, the rows contain the customers. A checkmark indicates, that a certain customer has bought the item.

<table>
<thead>
<tr>
<th>Item A</th>
<th>Item B</th>
<th>Item C</th>
<th>Item D</th>
<th>Item E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer A</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Customer B</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Customer C</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Customer D</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Customer E</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

The customer-item matrix is transformed into an item-item matrix as shown in Table 3.2 by summing up the individual purchase entries. The result of this transformation is always a symmetric matrix (i.e. entries are symmetric with respect to the main diagonal). In this case the figures in the cells show the absolute number of customers who bought a particular item. For example if a customer browses item E, item A (matrix value: 2) would be recommended in the first place followed by item C (value: 1) and D (value: 1).

Product association rules are generally non-personalized (e.g. every customer,
who browses item E will be recommended item A) but can simply be extended to a low level of personalization by using ephemeral navigation patterns (click-streams). In this case the values in the corresponding lines may be aggregated. For example if a customer has viewed item A and is currently browsing to item E he or she will be displayed item C (aggregated value: 3) as a recommendation in the first place. Additionally item D (aggregated value: 1) may be recommended. More complex personalized recommendation methods are explained in the following Section.

3.5.2 Personalized Recommendation Methods

This Section deals with methods for generating personalized recommendations. Personalized recommendations are adapted to the individual users on the basis of knowledge about their preferences and behavior [AT05]. In the following sections the personalization process is illustrated. A general synopsis of information filtering methods is given, characteristics of information filtering methods are described and information filtering is compared to information retrieval. Finally, collaborative filtering, attribute-based filtering, and rules-based filtering are discussed in detail.

Providing personalized recommendations constitutes an iterative process that is shown in Figure 3.6 and includes the following four stages [AT05]:

1. Define goals and evaluate appropriate personalization approaches: Personalization initiatives should be tied to discrete and quantifiable business goals (e.g. increase cross-sales by 10 per cent). Depending on this
goals and the general condition (e.g. customer base, characteristics of the offered products and services) appropriate personalization approaches have to be evaluated. The pros and cons of the specific approaches (i.e. information filtering methods) are described below.

2. Understand the consumer: This is achieved by collecting comprehensive information about consumers and converting it into knowledge that may be used for personalized recommendation purposes. This information is stored in the user profiles.

3. Deliver personalized recommendations: Based on the data collected in Step 2 the most relevant products and services have to be delivered to the consumer by applying appropriate information filtering methods. As discussed in Section 3.4 push, pull, and passive delivery may be chosen.
3.5. RECOMMENDATION METHODS

4. Measure impact of personalization: The last step includes the measurement of the impact of personalization and adequate responses by adjusting the personalization strategy. Measuring personalization impact serves as a feedback for possible improvements of the whole process. This feedback may help to decide whether to collect additional data, build better user profiles, develop better recommendation algorithms or improve the information delivery and presentation [AT05].

3.5.2.1 Synopsis of Information Filtering Methods

Recommender systems apply information filtering methods to deliver personalized recommendations. Information filtering systems share the following characteristics [BC92]:

- Information filtering systems are designed for unstructured and semi-structured data instead of structured data. Structured data conform to a certain format and are “well-defined”. Well-defined denotes that the meaning of data is defined unambiguously in a mathematical or logical way through axioms. A typical example for structured data would be a record set in a relational database with simple data types. Unstructured and semi-structured refer to data which have high complexity but no or much less well-defined meaning. A typical example for semi-structured data would be an e-mail, which has structured and well-defined header fields but an unstructured body.

- Information filtering is primarily applied on large amounts of textual information, but may also deal with other unstructured data like images, audio and video.

- Filtering is based on individual or group profiles. Profiles ideally represent the long-term interests and preferences of the individual or group.

- Information filtering may either remove irrelevant information (i.e. “leave things out”) or may select relevant data (i.e. “selecting things from a larger set of possibilities” from an incoming stream of data [MGT+87]. In the first case the user is presented the data, which is left after the
filtering process (e.g. junk mail filter). In the latter the user sees only the extracted data (e.g. “your recommendations” at amazon.com).

However, these characteristics are not exclusive to information filtering systems and are also valid for information retrieval systems, which makes it necessary to further distinguish information filtering from information retrieval along the following aspects [BC92, HSS01]:

- Frequency of use: Information filtering systems are designed for a repeated and continuous application by the user, who has long-term goals or interests. In contrast information retrieval systems are primarily characterized by an ad-hoc use of an one-time user with an one-time information need.

- Representation of information needs: In information filtering systems the users' needs in respect of information are represented by user profiles. Information retrieval systems apply queries instead of user-profiles as a representation for the information needs.

- Dynamics of data source: Information filtering is predominantly used on dynamic data streams, where irrelevant information is removed or relevant information is selected from that data stream. Information retrieval is applied on relatively static databases, where relevant information is selected.

- Timeliness: For information filtering up-to-dateness of the relevant information is of major importance (which is reflected by the dynamic nature of the data source). In information retrieval, timeliness is not that essential.

- Heterogeneity of users: Information filtering systems deal with undefined, highly heterogeneous user communities in various domains (e.g. entertainment). Information retrieval systems operate predominantly in environments with homogenous and well defined user groups in specific domains (e.g. science and technology).
3.5. RECOMMENDATION METHODS

- Privacy: Because information filtering systems apply user profiles which may contain sensitive personal data, it is highly concerned with privacy issues, which are mostly of no interest to information retrieval systems.

3.5.2.2 Human Approaches towards Information Filtering

Based on organizational studies, Malone et al. identified three basic filtering approaches for persons: cognitive, social and economic filtering [MGT+87]. These concepts of human approaches towards information filtering are incorporated into information filtering systems. The characteristics of these approaches are [MGT+87, HSS01]:

- Cognitive Filtering: Cognitive filtering refers to the attributes, contents and characteristics of an information object. This means that the person, who filters uses the information object characteristics (e.g. content of an e-mail header, title of a book) to judge the relevance. For instance, if a researcher looks for the specific keywords “call for papers” and a title of a conference in received e-mails to get an overview of relevant conferences, he or she employs the cognitive filtering approach. Because cognitive processes are generally attributed to humans, the term cognitive filtering is seldom used in the context of recommender systems. Thus, in literature the terms “attribute-based filtering”, “content-based” and “feature-based filtering” are used to describe techniques, that mimic this filtering approach in information systems.

- Social Filtering: According to this approach a person uses his social network for filtering purposes. It works by supporting the personal and organizational relationships of individuals in a community. If a person considers to give a high-priority to an e-mail, because it is sent from his supervisor, he or she uses social filtering. In recommender systems the idea of collaborative filtering is based on a social filtering approach.

- Economic Filtering: By using economic filtering a person employs cost-benefits assessments and explicit or implicit pricing mechanisms on information objects. Cost versus value decisions are taken to decide whether or not to process an information object. If a person decides to read
CHAPTER 3. RECOMMENDER SYSTEMS – FUNCTIONAL PERSPECTIVES

the executive summary instead of the whole report, because his or her workload is high, economic filtering is utilized.

3.5.2.3 Collaborative Filtering

Collaborative filtering is an approach which applies similarities between users' tastes and preferences for recommendation purposes. The basic idea behind collaborative filtering approaches is that the active user will be recommended items, which other users liked in the past (user-to-user correlation) [SKR01].

The term collaborative filtering was first used in literature by Goldstein et al. [GNOT92]. This paper describes “Tapestry”, a document filtering system developed at the Xerox Paolo Alto Research Center, which used collaborative filtering to reduce information overload. Tapestry enabled the user to annotate documents (e.g. e-mails, NetNews articles) with text comments and ratings (explicit approach) but also used implicit feedback (e.g. reply to an e-mail as an indicator for relevance) for recommendation purposes. The tapestry system suffered from two problems. Firstly, a small number of users used the system. Because of the absence of a critical mass of users most of the documents were not annotated and hence could not be used for recommendations. Secondly, Tapestry required the user to describe the filtering needs by a complex SQL-like language. This was a hindrance for users to operate the system [ME95]. Other early implementations of collaborative filtering systems were Grouplens, Ringo and Video Recommender.

In literature, the distinction between active and passive collaborative filtering systems can be found [ME95, Run00]. In active systems users actively recommend items to other users (push communication). Active collaborative filtering closely mimics the common practice that people recommend interesting items to other people of their social network (e.g. friends or colleagues). Active collaborative filtering systems support this process by providing information systems as communication tools. Active collaborative filtering requires the user to know interests and preferences of other users. Hence, active systems are of limited scalability. Because of this shortcoming, e-commerce applications regularly apply passive systems for recommendation purposes. In passive systems,
3.5. RECOMMENDATION METHODS

the user does not actively recommend items to other users. A direct communication between the users is not necessary. Passive collaborative filtering uses automated information systems in which people provide recommendations as inputs. These inputs are aggregated and directed to appropriate recipients by the system automatically [SV99]. Consequently, passive systems are also referred to as automated collaborative filtering systems.

Table 3.3 illustrates the basic idea of (passive) collaborative filtering based on a simple example. It shows a sample user–item matrix, in which preferences are measured on a binary scale. A "+" indicates that a user liked the item. A "-" means that the user does not like the item. An empty cell indicates, that the user has not rated the item (missing value).

| Table 3.3: Collaborative filtering: example of a user–item matrix |
|---------------------|-----|-----|-----|-----|-----|
|                     | Item A | Item B | Item C | Item D | Item E |
| User A              | +     | +     | -     | +     |        |
| User B              | +     | +     | +     | -     | +     |
| User C              | -     | +     | +     | +     |        |
| User D              | -     | +     | +     | +     |        |
| User E              | +     | +     | +     | -     |        |

Let's assume recommendations are given to user E. User E is very similar to user B, because both liked item A and item C and disliked item D. Because user B also liked item E, item E will be recommended to user E in the first place. User A is less close to user E (both liked item A, and disliked item D). Hence, item B could be recommended to customer E additionally. Between user E, user C and user D are no similarities at all. Consequently preferences of user C and user D are not used to give recommendations for user E.

Different statistical methods or machine–learning techniques are applied to calculate the similarity between users. Memory–based techniques directly compare users against each others (similar to the example above). They operate over the entire user–item matrix using statistical methods to perform similarity measures between the users. Correlation–based approaches use the Pearson correlation coefficient ("correlation–based") to determine the similarity between users [RIS+94, SM95, Paz99]. Other memory–based methods use the cosine ("cosine–based") [BHK98, SKKR00] to calculate the proximity between
users. In contrast *model-based approaches* use the users' historical rating data to derive a model. This model is used to make predictions, how the individual users will like certain items. Various *machine-learning techniques* – including Bayesian networks [BHK98], neural networks and latent semantic indexing [FD92] – are used to generate recommendations [Bur02]. However, the latter two techniques typically do not rely on user-ratings solely. Additionally they include attributes of the items (i.e. text documents) in the recommendation process. Hence, they can not be regarded as "pure collaborative filtering systems".

The typical *application domain* of recommender systems based on collaborative filtering is to suggest items, whose central characteristics and qualities can not properly measured with "objective" criteria (e.g. books, movies, music) [Run00]. Hence, this items are highly subject to personal taste and preferences.

In order to give reasonable recommendations, correlations of preferences have to exist between users and items. This means that certain groups of users with similar preferences for certain groups of items are given. Collaborative filtering requires a *sufficient number of users* ("critical mass") and an adequate number of known preferences (i.e. ratings of items) stored in user-profiles to give reasonable recommendations. Because collaborative filtering is based on ratings of a community, it employs human judgement. Thus, it enables the exchange of human knowledge between a large number of people without the requirement of knowing each other personally. This makes collaborative filtering a very powerful approach for recommendations.

In contrast to attribute-based filtering (see Section 3.5.2.4) collaborative filtering systems can give recommendations for items, which have no "objective" commonalities in terms of attributes with items the user liked in the past. This may lead to very innovative recommendations from the users' perspective. In fact, the recommendations are founded on relationships between the users of the recommender system, hence similarities between item characteristics are not necessary. For instance, a collaborative filtering systems may recommend a book to the active user because of his past ratings of music or movies. This would be hard to achieve with attribute-based or rules-based systems, because music and books generally have different attributes (an exception would be if
3.5. RECOMMENDATION METHODS

A songwriter also works as a book-author; in this case an attribute-based system could recommend books written by the songwriter, because they share the attribute "author".

Collaborative filtering approaches are subject to some limitations [BS97, Run00, SKKR00]. The new user problem refers to the challenge of giving accurate recommendations to new users. Because the preferences of new users are unknown it is impossible to make appropriate recommendations. An approach to address this problem is to use non-personalized recommendation methods (e.g. manual selection and association rules) until sufficient preferences are gathered from the user.

The new item problem reflects the hindrance to make recommendations for items, which have not been rated by the community. This is usually the case, when new items are added to the database. Because pure collaborative filtering systems solely use community ratings instead of item attributes for the recommendation process, new items can not be recommended [AT03]. Possible solutions are to use non-personalized methods or to combine collaborative filtering with attribute-based filtering ("hybrid approaches") [Bur02, SPUP02]. However the later requires that the object can be reasonably described by objective criteria. These approaches may be accompanied by incentive programs to get ratings for new items (e.g. to offer vouchers for users who write text comments and add ratings to items, which have not been previously rated).

Rating sparsity means that the number of given ratings is usually very small compared to the number of items, which may be recommended. This may occur, when the number of users is too small (absence of critical mass of users), when the underlying database of items is rapidly changing or when the users are "too similar" (i.e. all users like and rate the same small set of items). These phenomena lead to a high number of "missing values" [Run00] in the user-item matrix and consequently reduce the quantity ("reduced coverage" because products with no ratings can not be recommended) and quality of recommendations. To address this problem hybrid-approaches may be used. For instance, if the user-profile includes demographic data (e.g. gender, age, education), this information may be used to find similar users not solely based on similar ratings of items but also on demographic compliance ("demographic filtering") [Paz99].
CHAPTER 3. RECOMMENDER SYSTEMS – FUNCTIONAL PERSPECTIVES

The unusual user refers to a user, whose tastes are very different from the rest of the population. Hence, it is impossible to find any “nearest neighbors” (i.e. like-minded users) to derive recommendations from their ratings. Hence, the quality of recommendations for this kind of user are poor.

Collaborative filtering systems perform complex mathematical operations over large amounts of data. For the user it is hard to understand, why a certain item is recommended. This is called the “Black-Box problem”. A possibility to enhance transparency of the recommendation process is to display explanations (see Section 3.2).

Scalability problems may arise when collaborative filtering methods are used, because with this technique computation grows with the number of users and the number of items. In e-commerce applications these systems are challenged with millions of users and items. Consequently serious scalability problems may occur [SKKR00]. This is especially the case when memory-based algorithms are used. As mentioned above, memory-based algorithms operate over the entire database (which contains the user–item matrix) to give recommendations [BHK98]. Hence, this algorithms are prone to scalability problems. In contrast model-based approaches use the database to estimate parameters of a model in advance. This model is used to give recommendations to individual users after the calculation of the model parameters. Thus, it is not necessary to access the whole database while giving recommendations to the user. Consequently, model-based approaches outperform memory-based algorithms but may show a lack of accuracy, especially when the database is frequently changing [BHK98].

Collaborative filtering systems disregard product attributes for recommendation purposes, even when they are of high relevance. Pure collaborative filtering systems are not reasonably applicable, when the “objective” criteria of the recommended items are dominating the user’s preferences. For instance in the application domain of personal computers, objective attributes (e.g. performance data) have a strong influence on the buyers decision making process. The impact of subjective criteria (like the user’s brand affinity) on the buying decision may still be given, but is usually of less importance. Consequently, the quality of recommendations based on collaborative filtering techniques may be considered as poor, because the user’s requirements regarding these objec-
tive attributes are not taken into consideration. In addition, associations of items based on similarities between item characteristics can not be discovered by collaborative filtering systems. For example, a user likes films directed by Robert Rodriguez. A collaborative approach can not recommend all movies, music or books by Robert Rodriguez, because the attributes (e.g. "directed by", "composed by" and "written by") and the corresponding relationships are not modelled in pure collaborative filtering systems.

3.5.2.4 Attribute–Based Filtering

*Attribute–based filtering* is an filtering technique, which uses similarities between items for recommendations. This fundamental assumption is, that a user will like items similar to the ones he or she liked in the past [BS97].

In attribute–based filtering systems, the interest of a user is determined by the associated features of items. Hence the term "feature–based" approaches is also used for such systems [Run00]. Because the basic idea of this method is an outgrowth of information filtering research and was initially applied on textual documents, the term "content–based filtering" is a further term found in literature to describe such systems [BS97, Bur02, HSS01]. Strictly speaking, content–based approaches are a subclass of attribute–based filtering systems, where the application domain is textual documents. These documents are described by a restricted number of attributes of the content (e.g. characteristic words) [SPK00].

Similar to collaborative filtering, attribute–based filtering approaches employ a long–term user–model to learn and store user–preferences. In contrast to collaborative filtering, the interests of the user are not determined by comparing the similarity of the user to other users. Instead the interests of the user are derived from the attributes of the items, the user has already rated. Hence, attribute–based filtering systems generate recommendations based on a user–profile built up by analyzing the attributes of items which the user has rated in the past [BS97].

When designing a content–based filtering system two problems have to be addressed [Paz99]:

...
1. The representation of the items to recommend: This refers to the selection of relevant characteristics of the items to recommend. Depending on the application domain, this could be a fairly straightforward or rather complex task. For instance, when applied to automobiles the representation might focus on key-characteristics (i.e. specifications like horsepower, transmission, fuel economy etc.). In other domains (recommendation of textual documents, i.e. "content-based filtering") finding the right representation of items is more complex. For instance, the hybrid-recommender system Fab [BS97] uses the 100 most “important” words to represent documents, which are recommended to the users. The determination of the importance is determined by a weighting measure. For instance Fab uses the term “frequency/inverse document frequency measure” (TF-IDF) [Sal89] to gather the most informative keywords of web-pages.

2. The employment of a classification algorithm on user-profiles: A classification algorithm is used to estimate the degree of interest in the item. The user-profile contains ratings based on the classification scheme developed in Step 1. These ratings may be surveyed explicitly or implicitly. In literature a variety of classifications algorithms are used based on different statistical or machine learning methods (e.g. cosine similarity measures, Bayesian classifiers, clustering, decision trees, and artificial networks) [PB97].

Table 3.4 illustrates a representation scheme as described above in conjunction with a user profile. In this simple and fictional example books on e-commerce are represented by four keywords. A checkmark indicates that the term corresponding term occurs in the description of the book. A “+” in the column of “User A” means, that the user was interested in the book. A “-” indicates, that the user was not interested in the book. Because Book E and F are unknown to the user, they can be used for recommendation purposes. For example Book E might not be of interest, because in the past the user was not interested in books, which dealt with E-Branding. However, he or she might be interested in “Book F”, because it covers topics the user is interested in.

Applying attribute-based filtering requires two preconditions: (1) The items can be described by ”objective” criteria and (2) there must be a significant
3.5. RECOMMENDATION METHODS

### Table 3.4: Attribute–based filtering

<table>
<thead>
<tr>
<th>Last Mile Logistics</th>
<th>E–Branding</th>
<th>E–CRM</th>
<th>Business Models</th>
<th>User A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book A</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Book B</td>
<td>✓</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Book C</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Book D</td>
<td>✓</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Book E</td>
<td>✓</td>
<td></td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>Book F</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>?</td>
</tr>
</tbody>
</table>

coherence between these criteria and the global preferences of the users of attribute–based recommender systems [Run00].

Consequently, attribute–based filtering systems are well suited for domains where subjective tastes are not dominating the selection process and judgments are merely based on “hard–facts” (e.g. technical products). A typical example are digital cameras, which can be described with technical data. However, subjective criteria (e.g. design, brand–attitude) might still play a considerable role in the purchase decision process. Attribute–based filtering systems have limitations in the incorporation of these subjective criteria.

In contrast to collaborative filtering, attribute–based filtering methods do not depend on ratings of other users than the active user. Hence, attribute–based filtering systems are faster applicable than collaborative filtering systems, because building a “critical mass” of users is not crucial for the deployment of attribute–based systems [Run00].

A further advantage of is the structured representation of the attributes of items. Consequently, these meta–data could be used for purposes that go beyond attribute–based filtering. For example, the search for specific attributes is easy to implement (e.g. “show all books written by Umberto Eco”). Rules–based filtering approaches (see Section 3.5.2.5) can further be applied, when structured meta–data of items are already existent.

However, attribute–based systems are prone to some limitations. Limited content analysis refers to the fact, that attribute–based systems are limited by the attributes that are explicitly linked to the items these systems recom-
mend [AT03]. Depending on the domain, these features can be extracted automatically or have to be assigned by hand. As mentioned above, information retrieval offers a variety of methods to extract features of textual documents automatically. However, in other domains (e.g. multimedia-data) automatic feature extraction is much more complicated [BS97]. The assignment of attributes by hand is a time consuming task, which is often not practical due to limitations of resources [SM95]. Depending on the application domain a further problem with limited content analysis may be that two items with the same associated attributes may be indistinguishable. This may be of no concern when the two items are equivalent (e.g. technical products with the same specifications). However, if attribute-based filtering systems are applied on textual documents ("content-based filtering") a problem might occur. Textual documents are usually represented by the "most important keywords". Consequently, a well-written article can not be distinguished from a bad one, if the same terms are used [SM95].

*Over-specialization* is a further shortcoming of attribute-based systems. Because this kind of filtering system can only recommend items that score highly against the active user's profile, the user is limited to get recommendations of items that are similar to those already rated [BS97]. Consequently, the recommendations of attribute-based systems may not appear as "innovative" to the user compared to recommendations based on collaborative-filtering algorithms. In some fields of application, items that are to similar should not be recommended (e.g. articles in different newspapers, which describe the same event). Hence in some cases it may be sound to filter out items which are too similar to the ones the user has rated or seen before additionally [AT03].

Similar to collaborative filtering, attribute-based filtering systems also face the *new user problem*. If the number of ratings in the user-profile is insufficient, the system is not able to give accurate and reliable recommendations.

### 3.5.2.5 Rules-Based Filtering

Rules-based filtering is an approach that employs *business-rules* for recommendations. In this context, rules describe on-line behavioral activities of the
users [AT01]. In general, rules-based approaches can be designed stereotypical or personalized [KSS03]. In stereotype rule-based filtering approaches, the individual user is assigned to a group of similar users. For filtering purposes, the identical set of rules is used on each member of the group. In contrast personalized rules-based filtering systems apply an individual set of rules for each user [KSS03]. Consequently, the degree of personalization is higher with the latter approach.

Rules-based approaches are widely used in the field of expert systems for knowledge representation purposes [Jac98]. Generally rules may be described in the following form: IF {predicate} THEN {result}. In personalized rule-based filtering approaches a user profile contains a set of rules, that expresses the preferences of an individual user [KSS03]. For instance: IF {book.abstract contains "Macroeconomics" and book.year_of_publication not less than 1995} then {user_relevancy = "very high"}.

The main task of rule-based approaches is to discover suitable rules. In general, finding appropriate rules is accomplished with human experts (e.g. a marketing manager). Consequently, the effort for employing rules-based approaches tends to be higher compared to collaborative and attribute-based filtering methods due to the involvement of human expertise.

Figure 3.7 shows a structured approach towards the rule discovery process. In order to get "truly" personalized recommendations, rule discovery methods are applied to the data of every single user. The process of discovering rules could be divided in two phases: (1) data mining and (2) validation of the rules [AT01].

In the first step, data mining methods are applied on the user data to generate a large set of rules. Many of these rules are trivial, spurious and not relevant in the given application domain [AT01]. Hence step two, i.e. rule validation, is an important issue with this approach to get high-quality recommendations. Because of the sheer number of rules and users in e-commerce applications it is impossible to validate each rule for an individual customer by a domain expert. Consequently, rule validation is not performed separately for each user, but for all users at once by applying rule validation operators. Because there are many similar or identical rules across different users, validation effort can
be significantly reduced. In the end, the accepted rules form the profile of the individual users. For a detailed description of this process see [AT01].

One of the major drawbacks of rules-based filtering is the relative static nature of this approach. In contrast to collaborative filtering, changes in the taste of the user-population is reflected over the time due to the permanent rating of items by the users. However, in rules-based approaches the rules stay the same until a new discover and validation process is initiated. Because this process uses human expert knowledge, the effort of updating the rules is much higher compared to the "automatic" collaborative-filtering approach.