6

Explanations in recommender systems

6.1 Introduction

"The digital camera *Profishot* is a must-buy for you because . . ." "In fact, for your requirements as a semiprofessional photographer, you should not use digital cameras of type *Lowcheap* because . . ." Such information is commonly exchanged between a salesperson and a customer during in-store recommendation processes and is usually termed an *explanation* (Brewer et al. 1998).

The concept of explanation is frequently exploited in human communication and reasoning tasks. Consequently, research within artificial intelligence – in particular, into the development of systems that mimic human behavior – has shown great interest in the nature of explanations. Starting with the question, "What is an explanation?", we are confronted with an almost unlimited number of possibilities.

Explanations such as (1) "The car type Jumbo-Family-Van of brand Rising-Sun would be well suited to your family because you have four children and the car has seven seats"; (2) "The light bulb shines because you turned it on"; (3) "I washed the dishes because my brother did it last time"; or simply (4) "You have to do your homework because your dad said so", are examples of explanations depending on circumstances and make the construction of a generic approach for producing explanations difficult. The work of Brewer et al. (1998) distinguishes among functional, causal, intentional, and scientific explanations. Functional explanations (such as explanation 1) deal with the functions of systems. Causal explanations (such as explanation 2) provide causal relationships between events. Intentional explanations (such as explanations 3 and 4) give reasons for human behavior. Scientific explanations are exploited to express relations between the concepts formulated in various scientific fields and are typically based on refutable theories. Unfortunately, there is no accepted

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

unified theory describing the concept of explanation. Consequently, it is unclear how to design a general method for generating explanations.

Facing such fundamental challenges, one might ask why recommender systems should deal with explanations at all. The answer is related to the two parties providing and receiving recommendations. For example, a selling agent may be interested in promoting particular products, whereas a buying agent is concerned about making the right buying decision. Explanations are important pieces of information that can be exploited by both agents throughout the communication process to increase their performance. Different agents will formulate explanations with different intentions – for example, a buying agent looks for bargains and explanations that justify decisions, whereas a selling agent tries to improve profits by providing convincing arguments to the buying agent. We choose to analyze the phenomenon of explanations from a pragmatic viewpoint. Despite the diversity of proposals for characterizing the concept of explanation, almost all sources agree that an explanation is a piece of information exchanged in a communication process. In the context of recommender systems, these pieces of information supplement a recommendation with different aims. Following a pragmatic view, the goals for providing explanations in a recommendation process can be identified as follows (Tintarev 2007, Tintarev and Masthoff 2007):

Transparency. Explanations supporting the transparency of recommendations aim to provide information so the user can comprehend the reasoning used to generate a specific recommendation. In particular, the explanation may provide information as to why one item was preferred over another. For example, consider the case in which you wonder why a film recommender assumes you like Westerns, when in fact you do not. Transparency explanations may indicate, for example, that you purchased country songs and that this information is being exploited for recommending Western films, giving you the chance to change false assumptions.

Validity. Explanations can be generated to allow a user to check the validity of a recommendation. For example, "I recommend this type of car because you have four children and the Jumbo-Family-Van of Rising-Sun has seven seats. Because of the number of children, I cannot recommend the Dinki-coupe of SpeedyGECars, as it has only four seats." The ability to check validity is not necessarily related to transparency. For instance, a neural network may have decided that a product is an almost perfect match to a set of customer requirements. Transparency in the computation process, disclosing how the neural network computed the recommendation, will not help a customer validate the

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

recommendation. However, showing a comparison of the required and offered product features allows the customer to validate the quality of the product recommendation.

Trustworthiness. Following Grabner-Kräuter and Kaluscha (2003), trust building can be viewed as a mechanism for reducing the complexity of human decision making in uncertain situations. Explanations aiming to build trust in recommendations reduce the uncertainty about the quality of a recommendation – for example, "The drug Kural cured thousands of people with your disease; therefore, this drug will also help you."[1]

Persuasiveness. Computing technology (Fogg 1999) is regarded as persuasive if the system is intentionally designed to change a person's attitude or behavior in a predetermined way. In this sense, persuasive explanations for recommendations aim to change the user's buying behavior. For instance, a recommender may intentionally dwell on a product's positive aspects and keep quiet about various negative aspects.

Effectiveness. In the context of recommender systems, the term *effectiveness* refers to the support a user receives for making high-quality decisions. Explanations for improving effectiveness typically help the customer discover his or her preferences and make decisions that maximize satisfaction with respect to the selected product. Effective recommenders help users make better decisions.

Efficiency. In the context of recommender systems, the term *efficiency* refers to a system's ability to support users to reduce the decision-making effort. Thus explanations aiming to increase efficiency typically try to reduce the time needed for decision making. However, a measure for efficiency might also be the perceived cognitive effort, which could be different than efficiency based on the time taken to make the recommendation and select a product.

Satisfaction. Explanations can attempt to improve the overall satisfaction stemming from the use of a recommender system. This aim may not be linked to any other explanation goals, such as persuasiveness. The motivation behind this goal may be manifold – for instance, to increase the customer return rate.

Relevance. Additional information may be required in conversational recommenders. Explanations can be provided to justify why additional information is needed from the user.

[1] The recommender literature usually does not distinguish between trust and confidence.

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

Comprehensibility. Recommenders can never be sure about the knowledge of their users. Explanations targeting comprehension support the user by relating the user's known concepts to the concepts employed by the recommender.

Education. Explanations can aim to educate users to help them better understand the product domain. Deep knowledge about the domain helps customers rethink their preferences and evaluate the pros and cons of different solutions. Eventually, as customers become more informed, they are able to make wiser purchasing decisions.

The aforementioned aims for generating explanations can be interrelated. For example, an explanation generated for improving the transparency of a recommendation can have a positive effect on trust. Conversely, explanations aimed at persuasiveness may result in a loss of trust. Consequently, the first step in designing explanation generators is to define the goals of explanations. To assess the utility of explanations, all effects on the various communication aims of the recommendation process must be crosschecked.

As noted earlier, explanations are used in a communication process. Therefore, the suitability of an explanation depends on the goals of both the explanation sender and the receiver. As a consequence, the quality of explanations can be improved by modeling the receiving agent. For instance, to make explanations comprehensible, it is necessary to have information about the knowledge level of the receiver. Generally, the better the model of a receiving agent is, the more effective the arguments generated by a persuasive sending agent will be.

Furthermore, what is regarded as a valid (or good) explanation depends on the communication process itself, a fact that was neglected in early explanation generation attempts and that may lead to spurious explanations, as will be shown later. To summarize, the following factors influence the generation of explanations by a recommender agent communicating with an agent receiving recommendations:

- The piece of information to be explained.
- The goals of the agent in providing (receiving) an explanation.
- The model of the receiving agent, including its behavior and knowledge.
- The state of communication – the exchanged information, including provided recommendations.

How explanation goals can be achieved depends on the recommendation method employed. In particular, if knowledge is available as to how a conclusion was derived, then this knowledge can be exploited for various explanation goals – for example, to increase trust by providing a logically correct

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

argument. However, if such knowledge is not available, then trust building must be supported by other means, such as by referring the receiver to previous high-quality recommendations.

In the next section, we explore the standard explanation approaches for various recommendation methods, such as constraint-based recommenders, case-based recommenders, and recommenders based on collaborative filtering.

6.2 Explanations in constraint-based recommenders

The generation of explanations has a long history in expert systems (Shortliffe 1974). Constraint-based recommenders can be seen as a descendant of expert systems in general, and therefore can draw on established methods. However, we show how these methods have to be extended to work properly for constraint-based recommenders.

For example, in the area of sales support systems (Jannach 2004), constraint-based software applications have been developed to help customers find the right product (configuration) for their needs in domains as diverse as financial products and digital cameras. Constraint-based methods have become a key technology for the implementation of knowledge-based recommender systems because they offer enough expressive power to represent the relevant knowledge. In addition, extensive research has provided a huge library of concepts and algorithms for efficiently solving the various reasoning tasks.

In such applications a solution represents a product or a service a customer can purchase. Explanations are generated to give feedback about the process used to derive the conclusions – for example, "Because you, as a customer, told us that simple handling of a car is important to you, we included a special sensor system in our offer that will help you to park your car easily." Such explanations are exploited by customers in various ways – for instance, to increase confidence in a solution or to facilitate trade-off decisions (Felfernig et al. 2004).

In this section we focus on answering two typical questions a customer is likely to pose. In the case that the recommendation process requires input from the customer, the customer could ask why this information is needed. This corresponds to the classical *why-explanations* of expert systems (Buchanan and Shortliffe 1984). Conversely, when a recommender proposes a set of solutions (e.g., products) then a customer might ask for an explanation why a proposed solution would be advantageous for him or her. Traditionally, this type of explanation is called a *how-explanation* (Buchanan and Shortliffe 1984) because classical expert systems exploited information as to how a conclusion was

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

deduced – for example, the sequence of rules activated in the decision-making process.

We now examine methods for answering these two types of questions by exploiting the reasoning methods of constraint-based systems. We start with an introductory example and subsequently elaborate on the basic principles.[2]

6.2.1 Example

Consider an example from the car domain. Assume two different packages are available for a car; a business package and a recreation package. The customer can decide if he or she wants one, both, or neither of these packages.

The recreation package includes a coupling device for towing trailers and a video camera on the rear of the car, which allows the driver to ascertain the distance to an obstacle behind the car. This camera supports the customer-oriented product function *easy parking*. Customers may not be interested in the technical details a priori but may request functions for describing the wanted product features – for example, customers may not be interested in how the function *easy parking* is implemented, but they are interested that the car gives assistance for parking. Furthermore, functions are exploited to characterize products in an abstract way to justify consumer confidence in a solution – for instance, a sales rep can argue that a car supports easy parking because it includes a video camera at the rear. Functions of products are used to describe them in a more abstract way, which allows the hiding of technical details. Communicating with nontechnical customers requires this abstraction.

The business package includes a sensor system in the back bumper, which also supports easy parking. However, the sensor system is incompatible with the recreation package for technical reasons (the coupling device of the recreation package prevents the sensor system from being mounted). From the customer's point of view, the video camera and the sensor system provide the same functionality. Therefore, if the customer orders the business package and the recreation package, the car includes the video camera, which implements the *easy parking* function. In this configuration, the sensors are not only forbidden (because they are incompatible with the coupling device), but also dispensable. In addition, the business package includes a radio with a GSM telephone (GSM radio), which supports hands-free mobile communication.

[2] Parts are reprinted from Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), G. Friedrich, Elimination of Spurious Explanations, pp. 813–817. Copyright (2004), with permission from IOS Press.

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

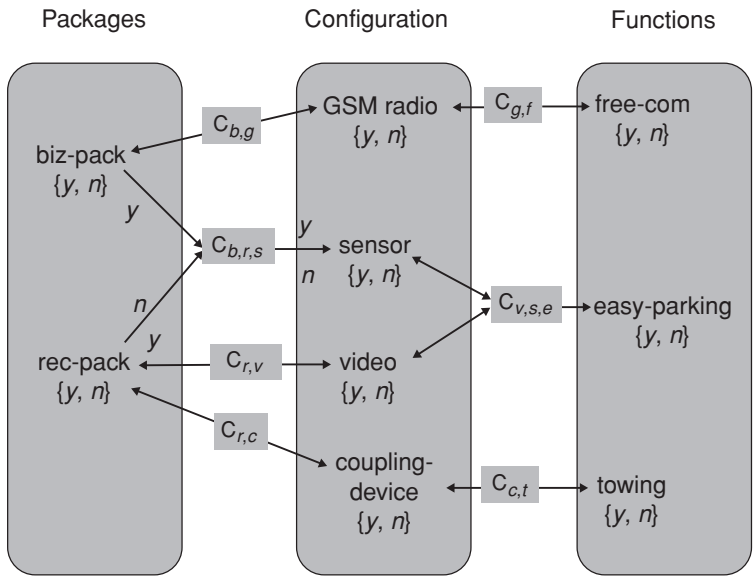

Figure 6.1. Constraint network of car example.

This domain can be modeled as a constraint satisfaction problem using the following variables and constraints: The example constraint network is depicted in Figure 6.1. The set of variables $\mathcal{V}$ is {*biz-pack*, *rec-pack*, *GSM-radio*, *sensor*, *video*, *coupling-device*, *free-com*, *easy-parking*, *towing*}. Unary constraints define the domains of the variables. For simplicity, it is assumed that for each variable the domain is $\{y, n\}$.

Further constraints are specified by the following tables:

$c_{r,v}$[3]: If *rec-pack* is chosen, then *video* must also be included (and vice versa).

$c_{b,r,s}$: If *biz-pack* is chosen and *rec-pack* is not chosen, then *sensor* is included. *rec-pack* and *sensor* are incompatible.

$c_{r,v}$:

rec-pack	*video*
y	y
n	n

$c_{b,r,s}$:

biz-pack	*rec-pack*	*sensor*
y	y	n
y	n	y
n	y	n
n	n	n
n	n	y

[3] Variables are abbreviated by their first letter(s).

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

$c_{v,s,e}$: If *video* or *sensor* is included, then *easy-parking* is supported (and vice versa).

$c_{v,s,e}$:		
video	*sensor*	*easy-parking*
n	n	n
y	n	y
n	y	y
y	y	y

The constraint connecting the variables *biz-pack* and *GSM-radio* is called $c_{b,g}$. $c_{g,f}$ connects *GSM-radio* and *free-com*. The constraint connecting the variables *rec-pack* and *coupling-device* is called $c_{r,c} \cdot c_{c,t}$ connects *coupling-device* and *towing*. The tables for these four constraints are identical to the table of $c_{r,v}$.

To find out which packages are appropriate for a given customer, assume that the recommender system asks the customer if he or she wants the car to be able to tow trailers. In return, the customer may ask for the motivation behind this question. In our example, the answer would be: "If you want the car to be able to tow trailers, then the recreation package is appropriate for you." Likewise, the customer is asked if he or she wants hands-free mobile communication.

Assume that the customer wants both towing and hands-free communication. Consequently, the car would come with the business package and the recreation package, including both a video camera and a GSM radio. Functions supported by such a car would include easy parking, hands-free mobile communication, and towing. More formally, if the customer sets $\{$*free-com* $= y$, *towing* $= y\}$, then the solution to the constraints $C = \{c_{r,v}, c_{b,r,s}, c_{v,s,e}, c_{b,g}, c_{g,f}, c_{r,c}, c_{c,t}\}$ representing the configured car would be to assign $\{$*video* $= y$, *sensor* $= n$, *GSM-radio* $= y$, *coupling-device* $= y$, *easy-parking* $= y$, *free-com* $= y$, *towing* $= y$, *biz-pack* $= y$, *rec-pack* $= y\}$.

Assume that this solution is presented to the customer. If the customer asks which choices led to the parking capabilities of the specific configured car, clearly the following answer must be provided: easy parking is supported because the car comes with a video camera. This video camera is included because it is included in the recreation package. The recreation package was chosen because the car should support towing. The business package is not suitable because the sensor system cannot be included.

In the following sections we present concepts and algorithms that are able to compute such explanations automatically. In particular, those parts of the user input and knowledge base must be identified that can be used to explain the

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

features of a given solution. We also review the standard approach for generating explanations and introduce the concept of well-founded explanations.

6.2.2 Generating explanations by abduction

Abduction is the widely accepted concept for generating explanations (Console et al. 1991, Junker 2004). The basic idea of these proposals is to use entailment ($\models$) to explain the outputs of a problem-solving process. Following Junker (2004), the approach of Friedrich (2004) is based on the concept of constraint satisfaction.

More formally, a constraint satisfaction problem (CSP) $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ (Junker 2004) is defined by a set of variables $\mathcal{V}$, a set of constraints $\mathcal{C}$, and a global domain $\mathcal{D}$. Each constraint has the form $c(x_i, \ldots, x_j)$, where $x_i, \ldots, x_j$ are n variables in $\mathcal{V}$ and c is an n-ary constraint predicate. Each n-ary constraint predicate has an associated n-ary relation $R(c) \subseteq \mathcal{D}^n$. A mapping $v : \mathcal{V} \rightarrow \mathcal{D}$ of variables to values represented by a set of values associated to variables $\{(x_k = v_{x_k}) | x_k \in \mathcal{V} \wedge v_{x_k} = v(x_k)\}$ satisfies a constraint $c(x_i, \ldots, x_j)$ if and only if (iff) $(v_{x_i}, \ldots, v_{x_j}) \in R(c)$. Such a mapping v is a solution of the CSP iff it satisfies all constraints in $\mathcal{C}$.

A set of constraints C is satisfiable iff the CSP with variables $V(C)$ and constraints C has a solution. A CSP $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ is trivially satisfied if $\mathcal{V}$ or $\mathcal{C}$ is empty. A mapping $v : \mathcal{V} \rightarrow \mathcal{D}$ is a solution of $(C, \mathcal{V}, \mathcal{D})$ iff $C \cup \{(x_k = v_{x_k}) | x_k \in V(C) \wedge v_{x_k} = v(x_k)\}$ is satisfied. Consequently, finding a solution to a CSP can be mapped to the problem of checking the consistency of a set of constraints.

Entailment is defined as usual:

Definition 1. (Junker) *A constraint ϕ is a (logical) consequence of a set of constraints C with variables $V(C)$ iff all solutions of the CSP with variables $V(C \cup \phi)$ and constraints C also satisfy the constraint ϕ. We write $C \models \phi$ in this case.*

The following standard definition of an abductive explanation is based on entailment:

Definition 2. *Let $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ be a consistent CSP, and ϕ a constraint.*

A subset C of $\mathcal{C}$ is called an explanation for ϕ in $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ iff $C \models \phi$. C is a minimal explanation for ϕ in $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ iff no proper subset of C is an explanation for ϕ in $(\mathcal{C}, \mathcal{V}, \mathcal{D})$.

For the computation of minimal explanations, QuickXPlain Junker (2004) can be applied, as introduced in Chapter 4. The required reasoning services,

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

such as checking whether a set of constraints is satisfiable, generating a solution, and proving entailment, are supported by standard constraint satisfaction problem solvers. Given a set of constraints C, which are a minimal explanation for ϕ, then usually these constraints may be translated to natural language by exploiting the natural language descriptions of constraints C and ϕ and the values of their variables. This method is termed the *canned text* approach and was originally applied in MYCIN-like expert systems (Buchanan and Shortliffe 1984).

Assume that the recommender asks questions to deduce which packages are appropriate for the customer because cars are manufactured according to selected packages. After a question is posed, the customer may ask for its reason. We consider the case in which the recommender asks whether the feature *towing* is required. In this example, the explanation for $\mathit{rec\text{-}pack} = y$ is $\{\mathit{towing} = y, c_{c,t}, c_{r,c}\} \models \mathit{rec\text{-}pack} = y$. User inputs are considered to be additional constraints. This explanation serves as a justification for why a customer should provide information about the need to tow trailers. In particular, if towing is required, then by constraint $c_{c,t}$ it can be deduced that a coupling device is needed and by $c_{r,c}$ it may be deduced that the car must come with the recreation package option.

To sum up, minimal explanations can be exploited to give reasons for questions asked by the recommender. If a question is posed to deduce the product characteristics necessary to recommend products, then we can compute minimal explanations for these product characteristics as justifications for the questions posed by the recommender.

Now, assume that a solution (i.e., the recommended car) is presented to the customer and advertised to support not only towing and hands-free communication, but also easy parking (remember, the customer requirements were towing and hands-free communication). Suppose the customer wants to know why the car supports easy parking and which input leads to the inclusion of this function. If we follow strictly the definition of abductive explanations (Definition 2), then there are two minimal explanations (arguments) for $\mathit{easy\text{-}parking} = y$:

EXP1: $\{\mathit{towing} = y, c_{c,t}, c_{r,c}, c_{r,v}, c_{v,s,e}\} \models \mathit{easy\text{-}parking} = y$, which is intended. If the customer wants the feature *towing*, then the car needs a coupling device. The need for a coupling device implies that the car will be fitted with the recreation package. The recreation package includes a video camera at the rear, which supports easy parking. Hence requiring towing includes a video camera, which supports easy parking as a side effect. However, there is a second minimal explanation:

EXP2: $\{\mathit{free\text{-}com} = y, c_{g,f}, c_{b,g}, c_{b,r,s}, c_{r,v}, c_{v,s,e}\} \models \mathit{easy\text{-}parking} = y$. This explanation states that hands-free communication implies the inclusion of a

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

GSM telephone, which in turn implies the selection of the business package. The business package then causes either the sensor or the video camera to be included, depending on the choice regarding the recreation package. In both variants of such a car, easy parking is supported.

The original solution (for which we are generating explanations) includes a video camera. Clearly, the second *abductive* explanation is not correct with respect to the original solution, as easy parking is provided by a video camera and the video camera is included in the recreation package and not in the business package. An explanation of the consequences of user inputs must be based on the solution implied by these user inputs. The question is how such spurious explanations can be avoided.

6.2.3 Analysis and outline of well-founded explanations

In this section we elaborate on the reasoning of Friedrich (2004) and outline the basic ideas for eliminating spurious explanations. More specifically, the concept of projection as defined in database relational algebra is applied. Let constraint c have variables $x_i, \ldots, x_j$ (and possibly others). The projection of constraint c on $x_i, \ldots, x_j$ (written as $c\{x_i, \ldots, x_j\}$) is a constraint with variables derived from the variables of c by removing all variables not mentioned in $x_i, \ldots, x_j$, and the allowed-tuples of c are defined by a relation $R(c\{x_i, \ldots, x_j\})$ consisting of all tuples $(v_{x_i}, \ldots, v_{x_j})$ such that a tuple appears in $R(c)$ with $x_i = v_{x_i}, \ldots, x_j = v_{x_j}$. A constraint with no variables is trivially satisfied.

Subsequently, the solutions of the original problem and those of the two minimal explanations EXP1 and EXP2 are compared. The only solution of the original CSP based on the set of all constraints C and all user inputs *towing* $= y$ and *free-com* $= y$ is:

tow	fre	eas	rec	biz	vid	sen	gsm	cou
y	y	y	y	y	y	n	y	y

For the explanation of *easy-parking* $= y$ where EXP1 is used (i.e., the user input *towing* $= y$ and $c_{c,t}, c_{r,c}, c_{r,v}, c_{v,s,e}$), the solutions implied for the original variables $\mathcal{V}$ are

solution	tow	fre	eas	rec	biz	vid	sen	gsm	cou
1	y	y	y	y	y	y	n	y	y
2	y	n	y	y	n	y	n	n	y

In both solutions *easy-parking* is y. Solution 1 is identical to the solution of the original CSP. However, solution 2 differs in the variables {*biz-pack*,

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

GSM-radio, *free-com*} from the original CSP. One might argue that an explanation that possibly exploits variable values that are out of the scope of the original solution might lead to a spurious explanation. However, to derive *easy-parking* $= y$, we need only the constraints $c_{c,t}, c_{r,c}, c_{r,v}, c_{v,s,e}$. Consequently, variables {*biz-pack*, *GSM-radio*, *free-com*} are superfluous in the derivation of *easy-parking* $= y$. In addition, not all variables in $c_{r,v}, c_{v,s,e}$ are necessary for the derivation. If we analyze $c_{v,s,e}$ then we recognize that setting *video* to y implies *easy-parking* $= y$, regardless of the value of *sensor*. Consequently, the relevant variables in our case are *towing*, *coupling-device*, *rec-pack*, *video*, and *easy-parking*. The solutions of $\{towing = y, c_{c,t}, c_{r,c}, c_{r,v}, c_{v,s,e}\}$ and the solutions of the original CSP projected on these relevant variables are identical.

For the explanation of *easy-parking* $= y$ where EXP2 is used (i.e., the user input *free-com* $= y$ and the constraints $c_{g,f}, c_{b,g}, c_{b,r,s}, c_{r,v}, c_{v,s,e}$), the solutions implied for the original variables $\mathcal{V}$ are

solution	tow	fre	eas	rec	biz	vid	sen	gsm	cou
1	y	y	y	y	y	y	n	y	y
2	n	y	y	n	y	n	y	y	n

Because only $c_{g,f}, c_{b,g}, c_{b,r,s}, c_{r,v}$, and $c_{v,s,e}$ are needed for the explanation, the variables not included in these constraints are irrelevant for this explanation. All other variables in these five constraints (i.e., *free-com*, *GSM-radio*, *biz-pack*, *rec-pack*, *video*, *sensor*, *easy-parking*) are needed. For example, if the variable *video* is deleted, then $\{$*free-com* $= y, c_{g,f}, c_{b,g}, c_{b,r,s}, c_{r,v}\{r\}, c_{v,s,e}\{s, e\}\} \not\models$ *easy-parking* $= y$ because there are solutions in which *easy-parking* $= n$.

The solutions with respect to these relevant variables are

solution	fre	eas	rec	biz	vid	sen	gsm
1	y	y	y	y	y	n	y
2	y	y	n	y	n	y	y

The explanation for *easy-parking* $= y$ must show that *easy-parking* $= y$ is contained in both solutions (in both logical models). In particular, the user input *free-com* $= y$ implies *biz-pack* $= y$. The constraints $c_{b,r,s}$, and *biz-pack* $= y$ imply either *sensor* $= y$ or *rec-pack* $= y$. *rec-pack* $= y$ implies *video* $= y$. In both cases, *easy-parking* $= y$ is implied by $c_{v,s,e}$. This explanation uses variable assignments that are not contained in the original solution, such as the car being supplied with video and not sensor equipment. Such an explanation is considered to be spurious because the reasoning includes a scenario (also called a possible world) that is apparently not possible given the current settings.

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

The principal idea of well-founded explanations is that an explanation C for a constraint ϕ and for a specific solution $\int$ must imply ϕ (as required for an abductive explanation); additionally, possible solutions of the explanation must be consistent with the specific solution $\int$ (with respect to the relevant variables).

6.2.4 Well-founded explanations

The definitions of *explanation* presented in this section provide more concise explanations compared with previous approaches (Junker 2004). Friedrich (2004) not only considers the relevance of constraints but also investigates the relevance of variables. The goal is to compute a minimal explanation consisting of the constraints and variables needed to deduce a certain property; these variables are exploited to construct an *understandable* chain of argument for the user (Ardissono et al. 2003). To introduce the following concepts, the projection operation on sets of constraints needs to be specified. $C\{V\}$ is defined by applying the projection on $V \subseteq \mathcal{V}$ to all $c \in C$ – that is, $C\{V\} = \{c\{V \cap V(c)\} | c \in C\}$. $V(c)$ are the variables of c.

Definition 3. *Let* $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ *be a satisfiable CSP,* ϕ *a constraint.*

A-tuple (C, V) *where* $C \subseteq \mathcal{C}$ *and* $V \subseteq \mathcal{V}$ *is an explanation for* ϕ *in* $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ *iff* $C\{V\} \models \phi$.

(C, V) *is a minimal explanation for* ϕ *in* $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ *iff for all* $C' \subset C$ *and all* $V' \subset V$ *it holds that neither* (C', V) *nor* (C, V') *nor* (C', V') *is an explanation for* ϕ *in* $(\mathcal{C}, \mathcal{V}, \mathcal{D})$.

($\{towing = y,\ c_{c,t},\ c_{r,c},\ c_{r,v},\ c_{v,s,e}\}$, $\{towing,\ coupling\text{-}device,\ rec\text{-}pack,\ video,\ easy\text{-}parking\}$) is a minimal explanation for $easy\text{-}parking = y$.

For the computation of minimal explanations, the following monotonicity property is employed.

Remark 1. *If* $C\{V\} \not\models \phi$ *then for all* $V' \subseteq V$ *it holds that* $C\{V'\} \not\models \phi$. *The same applies for deleting constraints. However, it could be the case that* (C', V) *and* (C, V') *are minimal explanations for* ϕ *in* $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ *and* $C' \subset C$ *and* $V' \subset V$.

A CSP solver to find a solution for the user is employed. Such a solution is described by a set of *solution-relevant* variables S which consists of all or a subset of variables of the CSP. Friedrich (2004) makes the reasonable assumption that sufficient information has been provided by the user (or about the user) such that the CSP unambiguously defines the values of variables S. More formally, $\int = \{(x_k = v_{x_k}) | x_k \in S \wedge (\mathcal{C} \models x_k = v_{x_k})\}$. For example, in the

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

presented car configuration case, the user has to provide enough information so a car is well defined. Information gathering is the task of an elicitation process (Pu et al. 2003, Ardissono et al. 2003). The approach of Friedrich (2004) deals with the generation of explanations for the properties of a (possible) solution. Consequently, a user is free to explore various solutions and can ask for an explanation of the relation between user decisions and properties of a specific solution.

Subsequently, the projection $\int\{V\}$ of a solution $\int$ on variables V is defined as $\{(x_k = v_{x_k}) | x_k \in V \wedge (x_k = v_{x_k}) \in \int\}$.

The definition of well-founded explanations for a property ϕ with respect to a solution $\int$ is based on the following idea. First, an explanation (C, V) for ϕ – that is, $C\{V\} \models \phi$ – must show that every solution (also known as a *logical model* or sometimes as a *possible world model*) of the set of constraints $C\{V\}$ is a solution (a model) of ϕ. This follows the standard definition of abductive explanations. Second, if the explanation $C\{V\}$ permits some possible world models (i.e., solutions of $C\{V\}$) with value assignments of solution-relevant variables S *other* than those assigned in $\int$ (i.e., the solution of the unreduced set of constraints $\mathcal{C}$), then it follows that the explanation of ϕ is based on possible world models that are in conflict with the original solution $\int$ (which was presented to the user). Therefore it must be ensured that every possible world (solution) of $C\{V\}$ is consistent with the variable assignment of $\int$.

Definition 4. *Let $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ be a satisfiable CSP, $\int$ the solution of $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ for the solution relevant variables S, (C, V) an explanation for ϕ.*

A-tuple (C, V) is a WF (WF) explanation for ϕ with respect to $\int$ iff every solution $s\{S\}$ of $(C\{V\}, V, \mathcal{D})$ is a part of $\int$ (i.e., $s\{S\} \subseteq \int$).

(C, V) is a minimal WF (MWF) explanation for ϕ with respect to $\int$ iff for all $C' \subset C$ and for all $V' \subset V$ it holds that neither (C', V) nor (C, V') nor (C', V') is a WF explanation for ϕ in $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ with respect to $\int$.

Remark 2. *Let $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ be a satisfiable CSP, (C, V) an explanation for ϕ and $\int$ the solution of $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ for the solution relevant variables S.*

(a) An explanation (C, V) is a WF explanation for $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ with respect to $\int$ iff $C\{V\} \models \int\{V\}$.
(b) If $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ is satisfiable and $\mathcal{C} \models \phi$ then there exists a WF explanation for ϕ.
(c) It could be the case that, for a satisfiable $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ and a ϕ such that $\mathcal{C} \models \phi$ and $\int$ the solution of $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ for S, no minimal explanation of ϕ exists that is also WF.

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

By applying Definitions 3 and 4 and Remark 2, the subsequent corollary follows immediately, which characterizes well-founded explanations based on logical entailment:

Corollary 1. *Let* $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ *be a satisfiable CSP and* $\int$ *the solution of* $(\mathcal{C}, \mathcal{V}, \mathcal{D})$ *for the solution relevant variables* S.

A-tuple (C, V) *where* $C \subseteq \mathcal{C}$ *and* $V \subseteq \mathcal{V}$ *is a WF explanation for* ϕ *with respect to* $\int$ *iff* $C\{V\} \models \phi \wedge \int\{V\}$.

Let a car be characterized by the solution-relevant variables *coupling-device*, *video*, *sensor*, and *GSM-radio*, which describe the configuration requested by the customer. ($\{towing = y, c_{c,t}, c_{r,c}, c_{r,v}, c_{v,s,e}\}$, {*towing*, *coupling-device*, *rec-pack*, *video*, *easy-parking*}) is a MWF explanation for *easy-parking* $= y$ with respect to the solution (car configuration) *coupling-device* $= y$, *video* $= y$, *sensor* $= n$, *GSM-radio* $= y$. It entails *easy-parking* $= y$ as well as *coupling-device* $= y$ and *video* $= y$.

($\{free\text{-}com = y, c_{g,f}, c_{b,g}, c_{b,r,s}, c_{r,v}, c_{v,s,e}\}$, {*free-com*, *GSM-radio*, *biz-pack*, *rec-pack*, *video*, *sensor*, *easy-parking*}) is a minimal explanation for *easy-parking* $= y$, but it is not WF because it does not entail *video* $= y$, *sensor* $= n$.

The computation of MWF explanations is described by Friedrich (2004). The basic idea follows Corollary 1. First the variables and then the constraints are minimized so each further deletion of a constraint or variable will result in the loss of entailment of ϕ or $\int\{V\}$.

In this section we have shown how an improved version of abductive reasoning can be used to compute explanations in constraint-based recommenders. In the next section, we review state-of-the-art explanation approaches for case-based recommenders, which are typically classified as knowledge-based recommenders. However, because this type of knowledge relies on similarity functions instead of logical descriptions, abduction cannot be exploited for explanation generation.

6.3 Explanations in case-based recommenders

The generation of solutions in case-based recommenders is realized by identifying the products that best fit a customer's query. Each item of a product database corresponds to a case. Typically a customer query puts constraints on the attributes of products – for example, a customer is interested only in digital cameras that cost less than a certain amount of money. In particular, given a query Q about a subset A_Q of attributes A of a case (product) description, the

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

Table 6.1. *Example product assortment: digital cameras.*

id	price	mpix	opt-zoom	LCD-size	movies	sound	waterproof
p_1	148	8.0	4×	2.5	no	no	yes
p_2	182	8.0	5×	2.7	yes	yes	no
p_3	189	8.0	10×	2.5	yes	yes	no
p_4	196	10.0	12×	2.7	yes	no	yes
p_5	151	7.1	3×	3.0	yes	yes	no
p_6	199	9.0	3×	3.0	yes	yes	no
p_7	259	10.0	3×	3.0	yes	yes	no
p_8	278	9.1	10×	3.0	yes	yes	yes

similarity of a case C to Q is typically defined (see McSherry 2005) as

$$sim(C, Q) = \sum_{a \in A_Q} w_a \, sim_a(C, Q) \tag{6.1}$$

The function $sim_a(C, Q)$ describes the similarity of the attribute values of the query Q and the case C for the attribute a. This similarity is weighted by w_a, expressing the importance of the attribute to the customer.

A recommendation set is composed of all cases C that have a maximal similarity to the query Q. Usually this recommendation set is presented directly to the customer, who may subsequently request an explanation as to why a product is recommended or why the requirements elicitation conversation must be continued. The typical approach used to answer a why-question in case-based recommenders is to compare the presented case with the customer requirements and to highlight which constraints are fulfilled and which are not (McSherry 2003b).

For example, if a customer is interested in digital cameras with a price less than €150, then p_1 is recommended out of the products depicted in Table 6.1. Asking *why* results in the explanation that the price of camera p_1 satisfies the customer constraint. The definition of similarities between attributes and requirements depends on the utility they provide for a customer. McSherry (2003b) distinguishes between more-is-better, less-is-better, and nominal attributes. Depending on the type of attribute, the answer to a why-question could be refined by stating how suitable the attribute value of an item is compared with the required one. Furthermore, the weights of the attributes can be incorporated into the answers – for instance, if the customer requires a price less than €160 and LCD size of more than 2.4 inches, where LCD size is weighted much more than price, then p_5 is recommended. The answer to a why-question can reflect this by stating, "p_5 is recommended because the price is €9 less and

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

the LCD size is 0.6 inches greater than requested. Furthermore, emphasis was placed on LCD size." Of course, a graphical representation of the similarity between values and weights makes such recommendations easier to comprehend.

Basically, the similarity function can be viewed as a utility function over the set of possible cases. Consequently, more elaborated utility functions can be applied. Carenini and Moore (2001) expressed the utility function using an additive multiattribute value function based on the multiattribute utility theory. This function is exploited to generate customer-tailored arguments (explanations) as to why a specific product is advantageous for a customer. A statistical evaluation showed that arguments that are tailored to the preferences of customers are more effective than nontailored arguments or no arguments at all. Effectiveness in this context means that the user accepts the recommended product as a good choice.

As discussed previously, the requirements of a customer might be too specific to be fulfilled by any product. In this circumstance, why-explanations provide information about the violated constraints (McSherry 2003b). For example, if the customer requires a price less than €150 and a movie function, then no product depicted in Table 6.1 fulfills these requirements. However, p_1 and p_5 can be considered as most similar products for a given similarity function, although one of the user requirements is not satisfied. A why-explanation for p_1 would be, "p_1 is within your price range but does not include your movie requirement." The techniques presented in Chapter 4 can be used to generate minimal sets of customer requirements that explain why no products fit and, moreover, to propose minimal changes to the set of requirements such that matching products exist.

Conversational recommender systems help the customer to find the right product by supporting a communication process. Within this process various explanations can be provided. In ExpertClerk (Shimazu 2002), three sample products are shown to a shopper while their positive and negative characteristics are explained. Alternatives are then compared to the best product for a customer query – for example, "Camera p_4 is more expensive than p_1, but p_4 has a sensor with greater resolution (*mpix*)."

The concept of conversational recommenders is further elaborated on by McSherry (2005), while discussing the Top Case recommender, where a customer is queried about his or her requirements until a single best-matching product is identified, regardless of any additional requirements that may be posed by the customer. For example, if a customer requires a camera to be waterproof and to support movies and sound, but price and sensor resolution are of little importance, then p_8 is recommended regardless of any additional requirements about other attributes such as *LCD-size*, *opt-zoom*, *mpix*, and *price*. The systems described by McSherry (2005) take this into account by providing

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

explanations of the form, "Case X differs from your query only in *attribute-set-1* and is the best case no matter what *attribute-set-2* you prefer." In this template, Case X is the recommended case, *attribute-set-1* contains the attributes that differ from the user's query, and *attribute-set-2* is the set of attributes for which preferred values have not been elicited because they are of no consequence to the recommendation.

Furthermore, Top Case can explain why a customer should answer a question by considering two situations. First, questions are asked to eliminate inappropriate products. Second, a question is asked such that a single product is confirmed as the final recommendation. A target case is always shown to the customer along with a number of competing (but similar) cases during requirement elicitation. The target case is selected randomly from the cases that are maximally similar to the customer requirements.

The explanation template used in the first situation (McSherry 2005) is:

> If $a = v$ this will increase the similarity of Case X from S_1 to S_2 {and eliminate N cases [including Cases $X_1, X_2, \ldots X_r$]}
>
> where:

- a is the attribute whose preferred value the user is asked to specify,
- v is the value of a in the target case,
- Case X is the target case,
- S_1 is the similarity of the target case to the current query,
- S_2 is the similarity of the target case that will result if the preferred value of a is v,
- N is the number of cases that will be eliminated if the preferred value of a is v, and
- Cases $X_1, X_2, \ldots, X_r$ are the cases that the user was shown in the previous recommendation cycle that will be eliminated if the preferred value of a is v.

The part of the template enclosed in curly brackets is shown only if cases are eliminated. The part enclosed in square brackets is used only if shown cases of the previous recommendations are eliminated.

If a question can confirm the target case then following template is used:

> If $a = v$ this will confirm Case X as the recommended case.

where a, v, and Case X are defined as previously.

These approaches assume that a customer can formulate requirements about the attributes of products. However, customers may have only a very rough idea about the product they want to purchase. In particular, they may not have fixed preferences if compromises must be made. For example, if the preferred product

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

is too expensive, then it is not clear which features should be removed. Consequently, help is required while exploring the product space. More generally, the transparency goal applies not only to the reasoning process but also to the case space itself (Sørmo et al. 2005). Transparency of suitable cases and the associated tradeoffs help the customer understand the options available and to reduce uncertainty about the quality of the decision. Tweaking critiquing, as presented in Chapter 4, provides a means exploring the case space effectively (Reilly et al. 2005b). This work was further improved by Pu and Chen (2007), who suggest a new method for ranking alternative products. The idea is to compare potential gains and losses (the "exchange rate") to the target candidate in order to rank alternatives. Moreover, Pu and Chen (2007) show that interfaces based on compound critiques help improve trust in recommender systems. Compared with explanations that merely explain the differences between alternatives and the target case, compound critiques improve the perceived competence and cognitive effort, as well as the intention to return to the interface.

The generation of explanations for case-based recommenders assumes that knowledge about products can be related to customer requirements. However, collaborative filtering recommenders must follow different principles because they generate suggestions by exploiting the product ratings of other customers.

6.4 Explanations in collaborative filtering recommenders

In contrast to the case with knowledge-based recommenders, explicit recommendation knowledge is not available if the collaborative filtering (CF) approach is applied. Consequently, recommendations based on CF cannot provide arguments as to why a product is appropriate for a customer or why a product does not meet a customer's requirements. The basic idea of CF is to mimic the human word-of-mouth recommendation process. Therefore, one approach for implementing explanations in CF recommenders is to give a comprehensible account of how this word-of-mouth approach works. Clearly, this approach aims to increase the transparency of recommendations but also has side effects regarding persuasion, which we subsequently discuss.

In the following we highlight key data and steps in the generation of CF outcomes and discuss their relevance to the construction of explanations. On a highly abstract level there are three basic steps that characterize the operation of CF, as presented in Chapter 2:

- Customers rate products.
- The CF locates customers with similar ratings (i.e., tastes), called neighbors.
- Products that are not rated by a customer are rated by combining the ratings of the customer's neighbors.

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

In concrete CF systems, these steps are usually implemented by sophisticated algorithms. In theory, the execution of these algorithms can serve as a profound explanation of the outcome. However, such an explanation tends to contribute to the confusion of customers and has negative impact on accepting the results of a recommender, as we will report later on. The art of designing effective explanation methods for CF is to provide the right abstraction level to customers.

Customers rate products. Transparency for this step of the collaborating filtering process is supported by presenting the customer information about the source of the ratings. Some recommenders are clear about the exploited ratings; for example, in the film domain, a system could consider ratings only when the customer explicitly rated a film, neglecting ratings based on other observations. However, some recommenders explore additional information by observing the customer and drawing conclusions based on these observations. For example, if a customer is observed to order mostly science fiction literature, then a movie recommender can exploit this information to hypothesize that the user would rate science fiction films highly. Although explaining this functionality might be an asset, in most cases this could lead to a severe drawback. For example, someone who is not interested in science fiction buys some science fiction books as presents. CF might conclude that the buyer likes this type of literature and starts recommending it. To change this behavior, it is important for the customer to understand the basis of recommendations.

Furthermore, as customers we may be interested in knowing how influential some of our ratings are. For example, if we were to wonder about the frequency of recommendations about science fiction films, we might discover that we had unjustifiably good ratings for some of these films. In addition, the diversity of rated products is essential. For example, if the majority of films rated positively by a customer are science fiction movies, then CF could be misled because the nearest neighbors would tend to be science fiction fans. Informing the customer about the diversity of the rated products gives important feedback about the quality of the customer's ratings.

Locate customers (neighbors) with similar tastes. In the second step, CF exploits the most similar customers to predict customer ratings. Clearly, the quality of this step is tightly linked to the similarity function used to locate customer ratings. From the customer's point of view, the question is whether this similarity function is acceptable – that is whether the selected neighbors have similar tastes. For example, a Beatles fan might not accept pop music ratings derived from other customers who do not like the Beatles themselves, although their ratings on other types of music are almost identical. Consequently,

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

providing information about potential neighbors helps the customer to assess the quality of a prediction and to specify which neighbors are acceptable for predicting ratings.

Compute ratings by combining ratings of neighbors. In the third step, CF combines the ratings of a customer's neighbors. This is usually implemented by a weighted average. The reliability of this calculation is usually ascertained by considering the number and the variance of individual ratings. A large number of ratings of neighbors with a low variance can be regarded as more reliable than a few ratings that strongly disagree. Consequently, providing this information to customers may help them assess the quality of the recommendation, improving their confidence in the system.

In line with these considerations, Herlocker et al. (2000) examined various implementations of explanation interfaces in the domain of the "MovieLens" system. Twenty-one variants were evaluated. In particular, potential customers were asked, on a scale of 1 to 7, how likely they would be to go to see a recommended movie after a recommendation for this movie was presented and explained by one of the twenty-one different explanation approaches. In this comparison, Herlocker et al. (2000) also included the base case in which no additional explanation data were presented. In addition to the base case, an explanation interface was designed that just output the past performance of the recommendation system – for instance, "MovieLens has provided accurate preductions for you 80% of the time in the past".

The results of the study by Herlocker et al. (2000) were

- The best-performing explanation interfaces are based on the ratings of neighbors, as shown in Figures 6.2 and 6.3. In these cases similar neighbors liked the recommended film, and this was comprehensibly presented. The histogram performed better than the table.
- Recommenders using the simple statement about the past performance of MovieLens were the second best performer.
- Content-related arguments mentioning the similarity to other highly rated films or a favorite actor or actress were among the best performers.
- Poorly designed explanation interfaces decreased the willingness of customers to follow the recommendation, even compared with the base case.
- Too much information has negative effects; poor performance was achieved by enriching the data presented in histograms (Figure 6.2) with information about the proximity of neighbors.
- Interestingly, supporting recommendations with ratings from domain authorities, such as movie critics, did not increase acceptance.

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

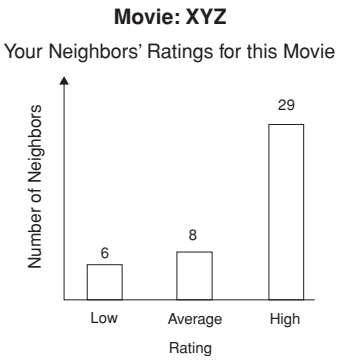

Figure 6.2. Histogram of neighbors' ratings.

The study by Herlocker et al. (2000) showed that customers appreciate explanations. In addition, the study analyzed the correctness of decisions by asking the participants if the movie was worth seeing and how they would rate it after seeing the recommended movie. Interestingly, there was no observable difference between the base case (no explanation given) and the cases of the customers who received explanations. Summing up, explanations in the described setting helped persuade customers to watch certain movies but they did not improve the effectiveness of decisions.

A further line of research (O'Donovan and Smyth 2005) deals with trust and competence in the area of CF recommenders. The basic idea is to distinguish between producers and consumers of recommendations and to assess the quality of the information provided by producers. In such scenarios, the consumer of a recommendation is the customer; conversely, the neighbors of a customer are the producers. The quality of a producer's ratings is measured by

Movie: XYZ

Personalized Prediction: ****

Your Neighbors' Ratings for This Movie

Rating	Number of Neighbors
★	2
★★	4
★★★	8
★★★★	20
★★★★★	9

Figure 6.3. Table of neighbors' ratings (Herlocker et al. 2000).

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press

the difference between the producer's ratings and those of consumers. These quality ratings offer additional explanation capabilities that help the customer assess recommendations. Possible explanation forms include "Product Jumbo-Family-Van was recommended by 20 users with a success rate of over 90% in more than 100 cases".

6.5 Summary

As outlined at the beginning of this chapter, there are many types of explanations and various goals that an explanation can achieve. Which type of explanation can be generated depends greatly on the recommender approach applied. A single implementation may, however, contribute to different explanation goals. For example, providing explanations could aim to improve both transparency and persuasiveness of the recommendation process. Great care must be taken to design explanation interfaces to achieve the planned effects.

Various forms of explanations for different approaches were described in this chapter. Because the development of recommender techniques is a highly active research area, many new proposals for supplementary explanation methods can be expected. However, explanations and their effects are an important field in their own right. From psychology it is known that customers do not have a stable utility function. Explanations may be used to shape the wishes and desires of customers but are a double-edged sword. On one hand, explanations can help the customer to make wise buying decisions; on the other hand, explanations can be abused to push a customer in a direction that is advantageous solely for the seller. As a result, a deep understanding of explanations and their effects on customers is of great interest.

https://doi.org/10.1017/CBO9780511763113.008 Published online by Cambridge University Press