Chapter 22
Aggregation of Preferences in Recommender Systems

Gleb Beliakov, Tomasa Calvo and Simon James

Abstract This chapter gives an overview of aggregation functions toward their use in recommender systems. Simple aggregation functions such as the arithmetic mean are often employed to aggregate user features, item ratings, measures of similarity, etc., however many other aggregation functions exist which could deliver increased accuracy and flexibility to many systems. We provide definitions of some important families and properties, sophisticated methods of construction, and various examples of aggregation functions in the domain of recommender systems.

22.1 Introduction

Aggregation of preferences, criteria or similarities happens at various stages in recommender systems. Typically such aggregation is done by using either the arithmetic mean or maximum/minimum functions. Many other aggregation functions which would deliver flexibility and adaptability towards more relevant recommendations are often overlooked. In this chapter we will review the basics of aggregation functions and their properties, and present the most important families, including generalized means, Choquet and Sugeno integrals, ordered weighted averaging, triangular norms and conorms, as well as bipolar aggregation functions. Such functions can model various interactions between the inputs, conjunctive, disjunctive and mixed behavior. Following, we present different methods of construction of aggregation functions, based either on analytical formulas, algorithms, or empirical data. We discuss how parameters of aggregation functions can be fitted to observed

Gleb Beliakov and Simon James
School of Information Technology, Deakin University
221 Burwood Hwy, Burwood 3125, Australia, e-mail: gleb@deakin.edu.au,sjames@deakin.edu.au

Tomasa Calvo
Departamento de Ciencias de la Computación, Universidad de Alcalá
28871-Alcalá de Henares (Madrid), Spain. e-mail: tomasa.calvo@uah.es

data, while preserving these essential properties. By replacing the arithmetic mean with more sophisticated, adaptable functions, by canceling out redundancies in the inputs, one can improve the quality of automatic recommendations, and tailor recommender systems to specific domains.

22.2 Types of Aggregation in Recommender Systems

In general, recommender systems (RS) guide users to *items of interest* selected from vast databases of electronic objects and information. The orientation toward the presentation of personalized item-subsets distinguishes RS conceptually from similar processes such as internet filtering, with the RS drawing on a number of user-specific justifications in order to generate individualized recommendations. Since their inception, the use of RS has expanded rapidly with existing applications that recommend movies [34], web-pages [5], news articles [36], medical treatments [14, 31], music and other products [32, 40].

Clearly, the justifications used to recommend an item will depend on the specific application and the way data is collected and used by the system. Recommendations based on justifications concerning item features can be broadly classified as *content-based* (CB), whereas recommendations that utilize user similarity are referred to as *collaborative* (CF) [1, 2]. It is useful to further identify demographic (DF), utility- (UB) and knowledge-based (KB) methods [16] as distinct from the usual perception of CB recommendation as anything that uses item-item similarity. The more recent literature has been characterized by a focus on hybrid systems (HS), which combine two or more of these approaches.

Collaborative methods use the item preferences or ratings of similar users as justification for recommendation. This type of RS has been successful for e-commerce sites like *Amazon.com* [32] where interest is better inferred through similar taste than vague or subjective item descriptions. Consider a periphery genre like *Indie* music, which is defined loosely by its separation from the mainstream. As the genre encompasses a broad range of styles, Indie artists may have little in common besides their fans.[1] Aggregation functions (usually the simple or weighted average) are often employed to aggregate the ratings or preferences of similar users, however they can also be used to determine user similarity and help define *neighborhoods* (see also Chapter 4 of this book).

Content-based filtering methods form justifications by matching item-features to user profiles. For instance, a news recommender may build a profile for each user that consists of *keywords* and the interest in an unseen news item can be predicted by the number of keywords in the story that correspond to those in the user's profile. The way aggregation functions are used (and whether they are used) for content-based methods depends on the nature of the profile that is given to each user and

[1] It is interesting that Indie music fans, who thrive on a lack of conformity to pop culture and consumerism, have become an easy target-market for e-commerce cites that utilize collaborative RS. This is discussed in a recent literary article [26].

the description of items. We consider their use in item score computation, similarity computation and the construction of profiles.

Demographic filtering techniques assign each user to a demographic class based on their user profiles. Each demographic class has an associated user archetype or user stereotype that is then used to form justifications for recommendation. Rather than item history, user similarity here is more likely to be calculated from personal information and hence may be of lower dimension than most collaborative techniques. This makes nearest-neighbor or other classification and clustering tools particularly useful.

Rather than build long-term models, *utility-based* recommenders match items to the current needs of the users, taking into account their general tendencies and preferences. For instance, a user may be looking for a particular book, and it is known from past behavior that old hardback editions are preferred even if it takes longer to ship them. As is the case with content-based filtering, items can be described in the system by their features and, more specifically, the utility associated with each of those features. Aggregation can then be performed as it is with content-based filtering, although the user profiles and system information may differ.

Knowledge-based recommenders use background knowledge about associated and similar items to infer the needs of the user and how they can best be met. Knowledge-based methods will then draw not only on typical measures of similarity like correlation, but also on feature similarities that will interest the user. For instance, when a user indicates that he liked *A Good Year*, a KB recommender system might know that this film could be associated with either *A Beautiful Mind* (which also stars Russell Crowe) or *Jeux d'Enfants* (which also stars Marion Cotillard). Since the user has shown a preference for French films in the past, the system will assume that the user liked *A Good Year* because it featured Marion Cotillard, and recommend accordingly. It is pointed out in [16] that KB recommenders often draw on case-based reasoning approaches.

Hybrid recommender systems are employed to overcome the inherent drawbacks of each recommendation method. Burke [16] distinguishes weighted, mixed, switching, feature combination, cascade, feature augmentation and meta-level HS. Aggregation functions may be involved in the hybridization process - e.g. to combine different recommender scores in weighted HS or the features in feature combination HS. On the other hand, some of these hybrid methods are particularly useful in improving the performance of aggregation functions used at different stages. For instance, cascade methods use one filtering technique to reduce the size of the dataset, while feature augmentation HS might use one method to reduce its dimension. Similarity measures used for CF could be based on the similarity between user-specific aggregation functions (e.g. the similarity between weights and parameters) constructed in UB and CB frameworks. Similar meta-level HS are described in [16]. The switching criteria in switching HS could be based to some degree on aggregation functions, however here, as with mixed HS, their use is less likely.

Aggregation functions take multiple inputs and merge them into a single representative output. Simple examples of aggregation functions include the arithmetic mean, median, maximum and minimum. The use of more complicated and expressive functions in RS would usually be motivated by the desire for more accurate recommendations, however in some circumstances aggregation functions might provide a practical alternative to other data processing methods. In the following subsections we will investigate the role of aggregation functions within different types of recommender system, indicating where they can be and have been applied.

22.2.1 Aggregation of Preferences in CF

Given a user u and a neighborhood of similar users $U_k = \{u_1, ..., u_k\}$, the preference of u for an unseen item d_i can be predicted by aggregating the scores given by U_k. We will denote the predicted degree of interest, rating or preference by $R(u, d_i)$.

$$R(u,d_i) = \sum_{j=1}^{k} sim(u,u_j)R(u_j,d_i) \tag{22.1}$$

The function can be interpreted as a weighted arithmetic mean (WAM) where similarities between the user and similar users $sim(u,u_j) = w_j$ are the weights and $R(u_j,d_i) = x_j$ are the inputs to be aggregated. Provided $w_j,x_j \geq 0$, the function $R(u,d_i)$ is an aggregation function. Whilst the WAM is simply interpreted, satisfies many useful properties and is computationally inexpensive, other aggregation functions including power means (which can be non-linear) or the Choquet integral (which accounts for correlated inputs) may give a more accurate prediction of the users' ratings.

22.2.2 Aggregation of Features in CB and UB Recommendation

Where the profile is representable as a vector of feature preferences, $P_u = (p_1, ..., p_n)$, items can then be described in terms of the degree to which they satisfy these features, i.e. $d_i = (x_1, ..., x_n)$. Here, a value of $x_j = 1$ indicates that the preference p_j is completely satisfied by the item. P_u could also be a vector of keywords, in which case $x_j = 1$ might simply mean that the keyword p_j is mentioned once. The overall rating $R(u,d_i)$ of an item is then determined by aggregating the x_j,

$$R(u,d_i) = f(x_1, ..., x_n). \tag{22.2}$$

Eq. (22.2) is an aggregation function provided the function satisfies certain boundary conditions and is monotone with respect to increases in x_j. The $R(u,d_i)$ scores can be used to provide a ranking of unseen items, which can then be recommended. If the RS allows only one item to be shown, the how and why of this score evaluation becomes paramount. If the user is only likely to buy/view items when all of their preferences are satisfied, a *conjunctive* function like the *minimum* should be used. On the other hand, if some of the preferences are unlikely to be satisfied simultaneously, e.g. the user is interested in drama and horror films, an *averaging* or *disjunctive* function might be more reliable. We present many examples of these broad classes of aggregation functions in Section 22.3.

In situations where it is practical to calculate item-item similarity, content-based filtering could also be facilitated using methods that mirror those in collaborative filtering [2]. In this case, a user profile might consist of all or a subset of previously rated/purchased items, $D = \{d_1,...,d_q\}$, and a measure of similarity is calculated between the unseen item d_i and those in D,

$$R(u,d_i) = \sum_{j=1,(j\neq i)}^{q} sim(d_i,d_j)R(u,d_j). \tag{22.3}$$

In this case, content-based methods can benefit from the use of aggregation functions in determining item similarity and item neighborhoods as in Section 22.2.4.

22.2.3 Profile Construction for CB, UB

More sophisticated systems will assign a weight w_j to each of the preferences in P_u. To enhance the online-experience, many recommenders opt to learn the preferences (and weights) from online behavior, rather than ask the user to state them explicitly. The features of previously rated or purchased items can be aggregated to give an overall score for each preference. Given a preference p_j, let x_{ij} be the degree to which item d_i satisfies p_j, then the score $w(p_j)$ will be

$$w(p_j) = f(x_{1j},...,x_{nj}). \tag{22.4}$$

Once all the preferences are determined, these $w(p_j)$ can be used to determine w_j for use in calculations such as Eq. (22.2).

22.2.4 Item and User Similarity and Neighborhood Formation

The behavior and accuracy of recommendation when using (22.1) will be largely dependent on how similarity (the weighting vector) is determined. The similarity between one user and another can be measured in terms of items previously rated or bought, or may be calculated based on known features associated with each user - e.g. the age, location and interests of a user may be known. The most commonly used measures of similarity, i.e. the weights in Eq. (22.1), are based on the cosine calculation [39] and Pearson's correlation coefficient [36]. Recently, other similarity measures have emerged such as fuzzy distance [4] and other recommender-specific metrics [18, 3], based on the distribution of user ratings (see also Chapter 4 of this book).

Eq. (22.1) can also be considered within the framework of a *k-nearest-neighbors* (kNN) approach. Aggregation functions have been used to enhance the accuracy and efficiency of nearest-neighbor rules, with the OWA and Choquet integral providing the framework to model decaying weights and neighbor interaction [45, 12]. In the nearest-neighbor setting, similarity is tantamount to multi-dimensional proximity or distance. Euclidean distance was considered for measuring similarity for recommenders that use both ratings and personal information as inputs in [42]. Euclidean distance is just one type of metric, and may not capture the concept of distance well - for instance, where the data dimensions are correlated to some degree or even incommensurable. Metrics defined with the help of certain aggregation functions, including the OWA operator and Choquet integral, have been investigated in [41, 13] and could potentially prove useful for measuring similarity in some RS.

If we regard each value $sim(u,u_j)$ in (22.1) as a weight rather than a similarity, we can keep in mind that the problem of weight identification for various aggregation functions has been studied extensively. One method is to learn the weights from a data subset by using least-squares fitting techniques. For instance, given a set of mutually rated items $D = \{d_1,...,d_q\}$, the weights of a WAM can be fitted using the following program:

$$\text{minimize} \sum_{i=1}^{q} \left(R(u,d_i) - \sum_{j=1}^{k} w_j R(u_j,d_i) \right)^2$$

$$\text{s.t.} \qquad w_j \geq 0, \qquad \forall j$$

$$\sum_{j=1}^{k} w_j = 1.$$

What is actually being determined is the vector of weights $\mathbf{w} = (w_1,...,w_k)$ that minimizes the residual errors. Each weight is then the importance of a given user u_j in accurately predicting $R(u,d_i)$. Non-linear functions such as the weighted geometric mean can also be fitted in this way. Such algorithms are relatively efficient in terms of computation time, and could be calculated either offline or in real-time depending on the RS and size of the database.

Alternatively, aggregation functions can be used to combine differing measures of similarity. Given a number of similarity measures $sim_1(u,u_1)$, $sim_2(u,u_1)$ etc., an overall measure of similarity can be obtained[2]. This type of aggregated similarity was used in [20] for the recommendation of movies. In this example, cosine and correlation scores were combined using the product, which is a non-linear and conjunctive aggregation function.

[2] For similarity based on multiple criteria, see Chapter 24 of this book.

22.2.5 Connectives in Case-Based Reasoning for RS

The approach of many researchers in the fuzzy sets community has been to frame the recommendation problem in terms of case-based reasoning [23] where aggregation functions can be used as connectives . This results in rules of the form,

$$\textbf{If } d_{i1} \textit{ is } A_1 \textbf{ AND } d_{i2} \textit{ is } A_2 \textbf{ OR } \dots \ d_{in} \textit{ is } A_n \textbf{ THEN ...} \tag{22.5}$$

$x_1, x_2, \dots, x_n$ denote the degrees of satisfaction of the rule predicates d_{i1} is A_1, etc., and aggregation functions are used to replace the AND and OR operations. For instance, a user whose profile indicates a preference for comedies and action films might have a recommendation rule "IF the film is a comedy *OR* an action THEN recommend it." [3] Each genre can be represented as a fuzzy set with fuzzy connectives used to aggregate the degrees of satisfaction. The OR- and AND-type behavior are usually modeled by disjunctive and conjunctive aggregation functions respectively. In recommender systems, it has been shown that the property of noble reinforcement is desirable [44, 9]. This property allows many *strong* justifications to result in a *very strong* recommendation, or a number of *weak* justifications to reduce the recommendation if desired.

Functions that model (22.5) can be used to match items to profiles or queries in CB, UB and KB. In some demographic RS, items will be generically recommended to everyone in a given class, making the classification process the primary task of the RS. It may be desirable to classify users by the degree to which they satisfy a number of stereotypes, and in turn describe items in terms of their interest to each of these. For instance, a personal loan with an interest-free period could be very attractive to graduating students and somewhat attractive to new mothers, but of no interest to someone recently married. A user could partially satisfy each of these archetypes, requiring the system to aggregate the interest values in each demographic. This leads to rules similar to (22.5). "IF the item is interesting to students OR interesting to mothers THEN it will be interesting to user *u*" or "IF user *u* is unmarried AND either a student OR mother, THEN recommend the item".

22.2.6 Weighted Hybrid Systems

Given a number of recommendation scores obtained by using different methods, e.g. $R_{CF}(u,d_i)$, $R_{CB}(u,d_i)$, etc., an overall score can be obtained using

$$R(u,d_i) = f(R_{CF}(u,d_i), R_{CB}(u,d_i), \dots) \tag{22.6}$$

with f an aggregation function. The P-Tango system [19] uses a linear combination of collaborative and content-based scores to make its recommendations, and adjusts

[3] We note here also that such rules could be used in any RS to decide *when* to recommend items, e.g. "IF user is inactive THEN recommend *something*".

the weight according to the inferred user preferences. Aggregation of two or more methods can be performed using a number of functions with different properties and behavior. The use of non-linear or more complicated functions would enable some recommenders to fine-tune the ranking process, creating less irrelevant and more accurate predictions.

22.3 Review of Aggregation Functions

The purpose of aggregation functions is to combine inputs that are typically interpreted as degrees of membership in fuzzy sets, degrees of preference, strength of evidence, or support of a hypothesis, and so on. In this section, we provide preliminary definitions and properties before giving an introduction to some well known families.

22.3.1 Definitions and Properties

We will consider aggregation functions defined on the unit interval $f : [0,1]^n \to [0,1]$, however other choices are possible. The input value 0 is interpreted as no membership, no preference, no evidence, no satisfaction, etc., and naturally, an aggregation of n 0s should yield 0. Similarly, the value 1 is interpreted as full membership (strongest preference, evidence), and an aggregation of 1s should naturally yield 1.

Aggregation functions also require monotonicity in each argument, where an increase to any input cannot result in a decrease in the overall score.

Definition 22.1 (Aggregation function). An aggregation function is a function of $n > 1$ arguments that maps the (n-dimensional) unit cube onto the unit interval $f : [0,1]^n \to [0,1]$, with the properties

(i) $f(\underbrace{0,0,\ldots,0}_{n-times}) = 0$ and $f(\underbrace{1,1,\ldots,1}_{n-times}) = 1$.

(ii) $\mathbf{x} \le \mathbf{y}$ implies $f(\mathbf{x}) \le f(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in [0,1]^n$.

For some applications, the inputs may have a varying number of components (for instance, some values can be missing). Particularly in the case of automated systems, it may be desirable to utilize functions defined for $n = 2,3,\ldots$ arguments with the same underlying property in order to give consistent aggregation results. Functions satisfying the following definition [33] may then be worth considering.

Definition 22.2 (Extended aggregation function). An extended aggregation function is a mapping

$$F : \bigcup_{n\in\{1,2,\ldots\}} [0,1]^n \to [0,1],$$

such that the restriction of this mapping to the domain $[0,1]^n$ for a fixed n is an n-ary aggregation function f, with the convention $F(x) = x$ for $n = 1$.

Aggregation functions are classed depending on their overall behavior in relation to the inputs [17, 21, 22]. In some cases we require high inputs to compensate for low inputs, or that inputs may average each other. In other situations, it may make more sense that high scores reinforce each other and low inputs are essentially discarded.

Definition 22.3 (Classes). An aggregation function $f : [0,1]^n \to [0,1]$ is:

Averaging if it is bounded by $\min(\mathbf{x}) \leq f(\mathbf{x}) \leq \max(\mathbf{x})$;
Conjunctive if it is bounded by $f(\mathbf{x}) \leq \min(\mathbf{x})$;
Disjunctive if it is bounded by $f(\mathbf{x}) \geq \max(\mathbf{x})$;
Mixed otherwise.

The class of aggregation function to be used depends on how the inputs of the recommender system are interpreted and how sensitive or broad an output is desired. When aggregating recommendation scores in CF, the use of averaging functions ensures that the predicted interest in an item is representative of the central tendency of the scores. On the other hand, the semantics of some mixed aggregation functions makes their use appealing. For instance, MYCIN [14] is a classical expert system used to diagnose and treat rare blood diseases and utilizes a mixed aggregation function so that inputs of only high scores reinforce each other, while scores below a given threshold are penalized.

There are several studied properties that can be satisfied by aggregation functions, making them useful in certain situations. We provide definitions for those that are frequently referred to in the literature.

Definition 22.4. [Properties] An aggregation function $f : [0,1]^n \to [0,1]$ is:

Idempotent if for every $t \in [0,1]$ the output is $f(t,t\ldots,t) = t$;
Symmetric if its value does not depend on the permutation of the arguments, i.e.,$f(x_1,x_2,\ldots,x_n) = f(x_{P(1)},x_{P(2)},\ldots,x_{P(n)})$ for every $\mathbf{x}$ and every permutation $P = (P(1),P(2),\ldots,P(n))$ of $(1,2\ldots,n)$;
Associative if, for $f : [0,1]^2 \to [0,1]$, $f(f(x_1,x_2),x_3) = f(x_1,f(x_2,x_3))$ holds for all x_1,x_2,x_3;
Shift-invariant if for all $\lambda \in [-1,1]$ and for all $\mathbf{x} = (x_1,\ldots,x_n)$, $f(x_1+\lambda,\ldots,x_n+\lambda) = f(\mathbf{x})+\lambda$ whenever $(x_1+\lambda,\ldots,x_n+\lambda) \in [0,1]^n$ and $f(\mathbf{x})+\lambda \in [0,1]$;
Homogeneous if for all $\lambda \in [0,1]$ and for all $\mathbf{x} = (x_1,\ldots,x_n)$, $f(\lambda x_1,\ldots,\lambda x_n) = \lambda f(\mathbf{x})$;
Strictly monotone if $\mathbf{x} \leq \mathbf{y}$ but $\mathbf{x} \neq \mathbf{y}$ implies $f(\mathbf{x}) < f(\mathbf{y})$;
Lipschitz continuous if there is a positive number M, such that for any two inputs $\mathbf{x},\mathbf{y} \in [0,1]^n$, $|f(\mathbf{x}) - f(\mathbf{y})| \leq Md(\mathbf{x},\mathbf{y})$, where $d(\mathbf{x},\mathbf{y})$ is a distance between $\mathbf{x}$ and $\mathbf{y}$. The smallest such number M is called the Lipschitz constant of f.
Has *neutral elements* if there is a value $e \in [0,1]$ such that $f(e,\ldots,e,t,e,\ldots,e) = t$ for every $t \in [0,1]$ in any position.
Has *absorbing elements* if there is a value $a \in [0,1]$ such that $f(x_1,\ldots,x_{j-1},a,x_{j+1},\ldots,x_n) = a$ for any $\mathbf{x}$ with $x_j = a$.

22.3.1.1 Practical Considerations in RS

We will discuss some of the implications of each of these properties with some examples before providing the formal definitions of many important and extensively studied aggregation functions.

Idempotency All averaging aggregation functions, including the means, OWA and Choquet integral defined in Section 22.3.2, are idempotent[4]. The usual interpretation of this property is toward a representation of consensus amongst the inputs. However in some RS applications, e.g. when aggregating ratings in CF, the relative ranking of items is of more concern than the commensurability of input/output interpretations.

Example 22.1. The geometric mean $G(x,y) = \sqrt{xy}$ is idempotent, whereas The product $T_P(x,y) = xy$ is not. For any two objects, $d_1 = (x_1, y_1)$ and $d_2 = (x_2, y_2)$, however it follows that $G(d_1) > G(d_2)$ implies $T_P(d_1) > T_P(d_2)$.

Example 22.2. Let $d_1 = (0.5, 0.5), d_2 = (0.2, 0.8)$. Using the geometric mean and the product to aggregate gives $G(d_1) = 0.5, G(d_2) = 0.4, T_P(d_1) = 0.25, T_P(d_2) = 0.16$. If d_i are item scores in CF, it might be better to interpret the outputs as the predicted ratings for user u, in which case we use G. If d_i are items described by the degree to which they satisfy two of the user preferences in UB filtering, the overall utility might be better indicated by T_P since we want most of the preferences satisfied.

Symmetry Symmetry is often used to denote equal importance with regard to the inputs. Weighted and non-weighted quasi-arithmetic means can be used depending on the situation. Although the ordered weighted averaging function (OWA) is defined with respect to a weighting vector, the inputs are pre-sorted into non-increasing order, hence it is symmetric regardless of $\mathbf{w}$.

Example 22.3. A collaborative RS considers an item rated by three similar users $d_i = (0.2, 0.7, 0.5)$. We consider using the weighting vector $\mathbf{w} = (0.6, 0.3, 0.1)$ with either an OWA function or a weighted arithmetic mean (WAM). In the case of the WAM, the weights suggest that user u_1 is very

[4] Idempotency and averaging behavior are equivalent for aggregation functions due to the monotonicity requirement. This property is sometimes referred to as unanimity since the output agrees with each input when the inputs are unanimous.

similar to user u, and further that $sim(u,u_1) > sim(u,u_2) > sim(u,u_3)$. The aggregated score in this case would be $R(u,d_i) = WAM(d_i) = 0.6(0.2) + 0.3(0.7) + 0.1(0.5) = 0.38$ since u_1 didn't particularly like the item. If using the OWA, one interpretation of the weights suggests that user u will like the item if one or two similar users liked it, no matter which of the similar users it is. This gives $R(u,d_i) = OWA(d_i) = 0.6(0.7) + 0.3(0.5) + 0.1(0.2) = 0.59$.

Associativity Associativity is a useful property for automatic computation as it allows functions to be defined recursively for any dimension. This is potentially useful for collaborative RS where data sparsity is a problem. The same function could be used to evaluate one item rated by 10 similar users, and another rated by 1000 similar users. T-norms and t-conorms, uninorms and nullnorms are associative, however the quasi-arithmetic means are not.

Example 22.4. A collaborative RS uses personal information to determine similarity between users (i.e. the values do not need to be reassessed every time a new item is rated). Rather than store an $items \times users$ matrix for each user, the system uses a uninorm $U(x,y)$ to aggregate the similar user ratings and stores a single vector of aggregated item scores $\mathbf{d} = (U(d_i),...,U(d_n))$. When a new item score x_{ij} is added, the system aggregates $U(U(d_i),x_{ij})$ and stores this instead of $U(d_i)$. The advantage here is that neither the previous scores nor the number of times the item is rated is required in order to update the predicted rating.

Shift-invariance and Homogeneity The main advantage of shift-invariant and homogeneous functions is that translating or dilating the domain of consideration will not affect relative orderings of aggregated inputs. The weighted arithmetic mean, OWA and Choquet integral are all shift invariant, so it makes no difference whether inputs are considered on [0,100] or [1,7], as long as the inputs are commensurable.

Strict monotonicity Strict monotonicity is desired in applications where the number of items to be shown to the user is limited. Weighted arithmetic means and OWA functions are strictly monotone when $w_j > 0, \forall j$, while geometric and harmonic means are strict for $\mathbf{x} \in]0,1]^n$. Aggregation functions which are not strict, the *maximum* function for instance, could not distinguish between an item $d_1 = (0.3, 0.8)$ and another $d_2 = (0.8, 0.8)$.

Example 22.5. A holiday recommendation site uses a utility-based RS where the Łukasiewicz t-conorm $S_L(x,y) = \min(x+y,1)$ is used to aggregate item features. It is able to show the user every item $S_L(d_i) = 1$ by notifications through e-mail. It doesn't matter that $d_1 = (0.3, 0.8)$ and $d_2 = (0.8, 0.8)$, since both of them are predicted to completely satisfy the user's needs.

Lipschitz continuity Continuity, in general, ensures that small input inaccuracies cannot result in drastic changes in output. Such a property is especially important in RS where the inputs, whether item descriptions or user ratings, are likely to be inexact. Some functions only violate this property on a small portion of the domain (See Example 22.6). As long as this is taken into account when the RS considers the recommendation scores, the function might still be suitable.

Example 22.6. The geometric mean $G(x,y) = \sqrt{xy}$ fails the Lipschitz property since the rate-of-change is unbounded when one of the inputs is close to zero. On the other hand, the harmonic mean, given by $H(x,y) = \frac{2xy}{x+y}$ (in the two-variate case) is Lipschitz continuous with Lipschitz constant $M = 2$.

Neutral and absorbent elements Absorbent elements could be useful in RS to ensure that certain items always or never get recommended. For example, a UB recommender could remove every item from consideration which has any features that score zero, or definitely recommend items which completely satisfy one of the user preferences. T-norms and t-conorms each have absorbent elements. Incorporating functions with neutral elements into a recommender system that aggregates user ratings (in either a CF or CB framework) allows values to be specified which will not affect recommendation scores. A movie that is liked by many people, for instance, would usually have its overall approval rating reduced by someone who was indifferent toward it but still required to rate it. If a neutral value exists it will not influence the aggregated score.

22.3.2 Aggregation Families

22.3.2.1 Quasi-Arithmetic Means

The family of weighted quasi-arithmetic means generalizes the power mean, which in turn includes other classical means such as the arithmetic and geometric mean as special cases (see [15] for an overview of means).

Definition 22.5 (Weighted quasi-arithmetic means). For a given strictly monotone and continuous function $g : [0,1] \to [-\infty, +\infty]$, called a generating function or generator, and a weighting vector $\mathbf{w} = (w_1, ..., w_n)$, the weighted quasi-arithmetic mean is the function

$$M_{\mathbf{w},g}(\mathbf{x}) = g^{-1}\left(\sum_{i=1}^{n} w_j g(x_j)\right), \qquad (22.7)$$

where $\sum w_j = 1$ and $w_j \geq 0 \; \forall \; j$.

Special cases include:

Arithmetic means	$WAM_{\mathbf{w}} = \sum_{j=1}^{n} w_j x_j,$	$g(t) = t;$
Geometric means	$G_{\mathbf{w}} = \prod_{j=1}^{n} x_j^{w_j},$	$g(t) = log(t);$
Harmonic means	$H_{\mathbf{w}} = \left(\sum_{j=1}^{n} \frac{w_j}{x_j}\right)^{-1},$	$g(t) = \frac{1}{t};$
Power means	$M_{\mathbf{w},[r]} = \left(\sum_{j=1}^{n} w_j x_j^r\right)^{\frac{1}{r}},$	$g(t) = t^r.$

The term *mean* is usually used to imply averaging behavior. Quasi-arithmetic means defined with respect to a weighting vector with all $w_j = \frac{1}{n}$ are symmetric, and asymmetric otherwise. Usually the weight allocated to a particular input is indicative of the importance of that particular input. All power means (including $WAM_{\mathbf{w}}, G_{\mathbf{w}}$ and $H_{\mathbf{w}}$) are idempotent, homogeneous and strictly monotone on the open interval $]0,1[^n$, however only the weighted arithmetic mean is shift-invariant. The geometric mean is not Lipschitz continuous[5].

22.3.2.2 OWA Functions

Ordered weighted averaging functions (OWA) are also averaging aggregation functions, which associate a weight not with a particular input, but rather with its relative value or order compared to others. They have been introduced by Yager [43] and have become very popular in the fuzzy sets community.

Definition 22.6 (OWA). Given a weighting vector $\mathbf{w}$, the OWA function is

$$OWA_{\mathbf{w}}(\mathbf{x}) = \sum_{j=1}^{n} w_j x_{(j)},$$

where the $(.)$ notation denotes the components of $\mathbf{x}$ being arranged in non-increasing order $x_{(1)} \geq x_{(2)} \geq \ldots \geq x_{(n)}$.

Special cases of the OWA operator, depending on the weighting vector $\mathbf{w}$ include:

[5] The Lipschitz property for quasi-arithmetic means and other generated aggregation functions is explored in [11].

Arithmetic mean where all the weights are equal, i.e. all $w_j = \frac{1}{n}$;
Maximum function for $\mathbf{w} = (1, 0, ..., 0)$;
Minimum function for $\mathbf{w} = (0, ..., 0, 1)$;
Median function for $w_j = 0$ for all $j \neq m$, $w_m = 1$ if $n = 2m + 1$ is odd, and $w_j = 0$ for all $j \neq m, m+1$, $w_m = w_{m+1} = 0.5$ if $n = 2m$ is even.

The OWA function is a piecewise linear idempotent aggregation function. It is symmetric, homogeneous, shift-invariant, Lipschitz continuous and strictly monotone if $w_j > 0, \forall\, j$.

22.3.2.3 Choquet and Sugeno integrals

Referred to as fuzzy integrals, the Choquet integral and the Sugeno integral are averaging aggregation functions defined with respect to a fuzzy measure. They are useful for modeling interactions between the input variables x_j.

Definition 22.7 (Fuzzy measure). Let $\mathcal{N} = \{1, 2, \ldots, n\}$. A discrete fuzzy measure is a set function[6] $v : 2^{\mathcal{N}} \to [0, 1]$ which is monotonic (i.e. $v(A) \leq v(B)$ whenever $A \subseteq B$) and satisfies $v(\emptyset) = 0, v(\mathcal{N}) = 1$. Given any two sets $A, B \subseteq \mathcal{N}$, fuzzy measures are said to be:

Additive where $v(A \cup B) = v(A) + v(B)$, for $v(A \cap B) = \emptyset$;
Symmetric where $|A| = |B| \to v(A) = v(B)$;
Submodular if $v(A \cup B) - v(A \cap B) \leq v(A) + v(B)$;
Supermodular if $v(A \cup B) - v(A \cap B) \geq v(A) + v(B)$;
Subadditive if $v(A \cup B) \leq v(A) + v(B)$ whenever $A \cap B = \emptyset$;
Superadditive if $v(A \cup B) \geq v(A) + v(B)$ whenever $A \cap B = \emptyset$;
Decomposable if $v(A \cup B) = f(v(A), v(B))$ whenever $A \cap B = \emptyset$, for a given function $f : [0, 1]^2 \to [0, 1]$;
Sugeno (λ-fuzzy measure) if v is *decomposable* with $f = v(A) + v(B) + \lambda v(A)v(B)$, $\lambda \in]-1, \infty[$.

The behavior of the Sugeno and Choquet integral depends on the values and properties of the associated fuzzy measure. The fuzzy measure used to define the Choquet integral can be interpreted as a weight allocation, not merely to individual inputs but rather to each subset of inputs. It may be that there are redundancies among the inputs, or that certain inputs complement each other.

Definition 22.8 (Choquet integral). The discrete Choquet integral with respect to a fuzzy measure v is given by

$$C_v(\mathbf{x}) = \sum_{j=1}^{n} x_{(j)} [v(\{k | x_k \geq x_{(j)}\}) - v(\{k | x_k \geq x_{(j+1)}\})], \tag{22.8}$$

[6] A set function is a function whose domain consists of all possible subsets of $\mathcal{N}$. For example, for $n = 3$, a set function is specified by $2^3 = 8$ values at $v(\emptyset)$, $v(\{1\})$, $v(\{2\})$, $v(\{3\})$, $v(\{1,2\})$, $v(\{1,3\})$, $v(\{2,3\})$, $v(\{1,2,3\})$.

where $(.)$ in this case denotes the components of $\mathbf{x}$ being arranged in non-decreasing order such that $(x_{(1)} \le x_{(2)} \le \cdots \le x_{(n)})$ (note that this is opposite to OWA).

Special cases of the Choquet integral include weighted arithmetic means and the OWA function where the fuzzy measure is additive or symmetric respectively. Submodular fuzzy measures result in Choquet integrals which are concave, the upshot of which is that increases to lower inputs affect the function more than increases to higher inputs. Conversely, supermodular fuzzy measures result in convex functions. Choquet integrals are idempotent, homogeneous, shift-invariant and strictly monotone where $A \subsetneq B \to v(A) < v(B)$. Where the fuzzy measure is symmetric, the function will obviously satisfy the symmetry property.

The Choquet integral has been predominantly used for numerical inputs, the Sugeno integral defined below is useful where the inputs are ordinal. It also uses fuzzy measures for its definition.

Definition 22.9 (Sugeno integral). The Sugeno integral with respect to a fuzzy measure v is given by

$$S_v(\mathbf{x}) = \max_{j=1,\dots,n} \min\{x_{(j)}, v(H_j)\}, \tag{22.9}$$

where $(.)$ denotes a non-decreasing permutation of the inputs such that $(x_{(1)} \le x_{(2)} \le \cdots \le x_{(n)})$ (the same as with the Choquet integral), and $H_j = \{(j), \dots, (n)\}$.

Certain indices have been introduced in order to better understand the behavior of the Choquet and Sugeno integrals. In particular, the Shapley value gives an indication of the overall importance of a given input, while the interaction index between two inputs shows to what extent they are redundant or complimentary.

Definition 22.10 (Shapley value).
Let v be a fuzzy measure. The Shapley index for every $i \in \mathcal{N}$ is

$$\phi(i) = \sum_{A \subseteq \mathcal{N} \setminus \{i\}} \frac{(n - |A| - 1)!|A|!}{n!} [v(A \cup \{i\}) - v(A)].$$

The Shapley value is the vector $\phi(v) = (\phi(1), \dots, \phi(n))$.

Definition 22.11 (Interaction index). Let v be a fuzzy measure. The interaction index for every pair $i, j \in \mathcal{N}$ is

$$I_{ij} = \sum_{A \subseteq \mathcal{N} \setminus \{i,j\}} \frac{(n - |A| - 2)!|A|!}{(n-1)!} [v(A \cup \{i,j\}) - v(A \cup \{i\}) - v(A \cup \{j\}) + v(A)].$$

Where the interaction index is negative, there is some redundancy between the two inputs. Where it is positive, the inputs complement each other to some degree and their weight together is worth more than their combined individual weights.

22.3.2.4 T-Norms and T-Conorms

The prototypical examples of conjunctive and disjunctive aggregation functions are so-called triangular norms and conorms respectively (t-norms and t-conorms) [28]. Given any t-norm $T : [0,1]^2 \to [0,1]$, there is a dual function which is a t-conorm S, with

$$S(x,y) = 1 - T(1-x, 1-y)$$

and vice-versa. T-norms and t-conorms are hence often studied in parallel, as many properties concerning S can be determined from T. Triangular norms are associative, symmetric with the neutral element $e = 1$, whereas triangular conorms are associative, symmetric and have the neutral element $e = 0$. The definitions of the four basic t-norms and t-conorms are provided below.

Definition 22.12 (The four basic t-norms). The two-variate cases for the four basic t-norms are given by

$$\begin{array}{ll}
\textit{Minimum} & T_{min}(x,y) = \min(x,y); \\
\textit{Product} & T_P(x,y) = xy; \\
\textit{Łukasiewicz t-norm} & T_L(x,y) = \max(x+y-1,0); \\
\textit{Drastic Product} & T_D(x,y) = \begin{cases} 0, & \text{if } (x,y) \in [0,1[^2, \\ \min(x,y) & \text{otherwise.} \end{cases}.
\end{array}$$

Definition 22.13 (The four basic t-conorms). The two-variate cases for the four basic t-conorms are given by

$$\begin{array}{ll}
\textit{Maximum} & S_{max}(x,y) = \max(x,y); \\
\textit{Probabilistic Sum} & S_P(x,y) = x+y-xy; \\
\textit{Łukasiewicz t-conorm} & S_L(x,y) = \min(x+y,1); \\
\textit{Drastic Product} & S_D(x,y) = \begin{cases} 1, & \text{if } (x,y) \in]0,1]^2, \\ \max(x,y) & \text{otherwise.} \end{cases}.
\end{array}$$

There are families of parameterized t-norms and t-conorms that include the above as special or limiting cases. These families are defined with respect to generating functions and are known as Archimedean t-norms.

Definition 22.14 (Archimedean t-norm). A t-norm is called Archimedean if for each $(a,b) \in]0,1[^2$ there is an $n = \{1,2,...\}$ with $T(\overbrace{a,...,a}^{n-times}) < b$.

For t-conorms, the inequality is reversed, i.e. the t-conorm $S > b$. Continuous Archimedean t-norms can be expressed by use of their generators as

$$T(x_1,...,x_n) = g^{(-1)}(g(x_1) + ... + g(x_n)),$$

where $g : [0,1] \to [0,\infty]$ with $g(1) = 0$ is a continuous, strictly decreasing function and $g^{(-1)}$ is the pseudo inverse of g, i.e.,

$$g^{(-1)}(x) = g^{-1}(\min(g(1), \max(g(0), x))).$$

Archimedean families include Schweizer-Sklar, Hamacher, Frank, Yager, Dombi, Aczel-Alsina, Mayor-Torrens and Weber-Sugeno t-norms and t-conorms.

22.3.2.5 Nullnorms and Uninorms

In some situations, it may be required that high input values reinforce each other whereas low values pull the overall output down. In other words, the aggregation function has to be disjunctive for high values, conjunctive for low values, and perhaps averaging if some values are high and some are low. This is typically the case when high values are interpreted as "positive" information, and low values as "negative" information.

In other situations, it may be that aggregation of both high and low values moves the output towards some intermediate value. Thus certain aggregation functions need to be conjunctive, disjunctive or averaging in different parts of their domain.

Uninorms and nullnorms are typical examples of such aggregation functions, but there are many others. We provide the definitions below.

Definition 22.15 (Nullnorm). A nullnorm is a bivariate aggregation function $V : [0,1]^2 \to [0,1]$ which is associative, symmetric, such that there exists an element a belonging to the open interval $]0,1[$ verifying

$$\forall t \in [0,a],\quad V(t,0) = t,$$
$$\forall t \in [a,1],\quad V(t,1) = t.$$

Definition 22.16 (Uninorm). A uninorm is a bivariate aggregation function $U : [0,1]^2 \to [0,1]$ which is associative, symmetric and has a neutral element e belonging to the open interval $]0,\ 1[$.

Some uninorms can be built from generating functions in a similar way to quasi-arithmetic means and Archimedean t-norms. These are called representable uninorms.

Definition 22.17 (Representable uninorm). Let $u : [0,1] \to [-\infty,+\infty]$ be a strictly increasing bijection verifying $g(0) = -\infty, g(1) = +\infty$ such that $g(e) = 0$ for some $e \in]0,1[$.

- The function given by

$$U(x,y) = \begin{cases} g^{-1}(g(x)+g(y)), & \text{if } (x,y) \in [0,1]^2 \backslash \{(0,1),(1,0)\}, \\ 0, & \text{otherwise,} \end{cases}$$

  is a conjunctive uninorm with the neutral element e, known as a *conjunctive representable uninorm.*
- The function given by

$$U(x,y) = \begin{cases} g^{-1}(g(x)+g(y)), & \text{if } (x,y) \in [0,1]^2 \backslash \{(0,1),(1,0)\}, \\ 1, & \text{otherwise,} \end{cases}$$

is a disjunctive uninorm with the neutral element e, known as a *disjunctive representable uninorm.*

The $3-\Pi$ function is an example of a representable uninorm [46]. It uses a generating function $g(x) = \ln(\frac{x}{1-x})$ and is used by the expert system PROSPECTOR [24] for combining uncertainty factors.

$$f(\mathbf{x}) = \frac{\prod_{i=1}^{n} x_i}{\prod_{i=1}^{n} x_i + \prod_{i=1}^{n} (1-x_i)},$$

with the convention $\frac{0}{0} = 0$. It is conjunctive on $[0, \frac{1}{2}]^n$, disjunctive on $[\frac{1}{2}, 1]^n$ and averaging elsewhere. It is associative, with the neutral element $e = \frac{1}{2}$, and discontinuous on the boundaries of $[0,1]^n$.

22.4 Construction of Aggregation Functions

There are infinitely many aggregation functions. The question is how to choose the most suitable aggregation function for a specific application. Sometimes one function may suffice for all components of the application, at other times a different type of aggregation may be employed at various stages. The following considerations should be helpful.

22.4.1 Data Collection and Preprocessing

The type of data, and how it is collected affects the way it can be aggregated to form justifications. If users could thoughtfully provide accurate scores on a consistent scale for each item, or numerical descriptions of themselves with their preferences expressed to a degree of certainty, an RS could quite comfortably make some relevant recommendations. Of course, the aesthetic preference is usually to limit the explicit information required from the user and hence enhance the interactive experience. We will briefly consider the different types of data that systems are able to obtain and how this might affect the suitability of certain aggregation functions.

Ordinal Data CF recommenders that ask for explicit ratings information will usually do so on a finite ordinal scale - e.g. {1 = *didn't like it!*,..., 5 = *loved it!*}. On the other hand, it may be possible to convert user actions into ordinal values as part of their profile - e.g. {*regularly views, sometimes views*, etc.}. Where there is an ordinal scale, these values can be turned into numbers and aggregated.

For non-homogeneous functions and those which lack the shift-invariance property, it will be necessary to express these ordinal values on the unit interval. The coarseness of the aggregated values may make the difference between, say, the weighted arithmetic mean and the geometric mean negligible. Examples of aggregation functions particularly suitable for the aggregation of ordinal data are the Sugeno integral and the induced OWA.

- The Sugeno integral S_v (Def. 22.9), is a function which is able to process ordinal data and take into account interactions. It is necessary for the fuzzy measure values to be on the same ordinal scale as the input values. The Sugeno integral is capable of modeling median-type functions as well as minimum and maximum functions, and has the advantage of expressing outputs as ordinal values.
- The induced OWA function [45] is capable of modeling nearest-neighbor approaches even if the similarity is expressed as ordinal values, although it does require the ratings to be expressed numerically.

Numerical Data Where a system is capable of representing user inputs or actions as numerical data, it is useful to take into account whether these values are accurate, whether they are commensurate, and whether they are independent. Functions such as the geometric mean have a higher rate of change when input values are high than the arithmetic mean. This can help provide granularity to the outputs, however it also means that errors on this portion of the domain will influence the recommendation accuracy. In CF, two users might have similar preferences however one may consistently overrate items. In these cases, it might make sense to standardize the ratings before aggregating so the values between users are comparable. The use of the WAM implies independence between inputs, however other averaging functions, especially the Choquet integral, can express interaction and correlation either among certain inputs or relative scores.

Categorical Data In some cases, the use of categorical data may make it impractical to use aggregation functions. If there is no order between categories, it is meaningless to take the *average* or *maximum*, and other techniques may be useful for establishing similarity between users etc. It may be possible to transform the categorical data, for example, by the degree to which it contributes towards a certain archetype in DF.

There could however, be variations: some components of the vectors associated with d_i could be missing - e.g. ratings in CF, or the inputs $d_i = (x_1, ..., x_n)$ may have varying dimension by construction. In other cases, the uncertainty associated with some of the inputs or outputs could prescribe a range of values - e.g., the interval $[6, 8]$ as a rating for a film. Associative or generating functions are capable of aggregating inputs of varying dimension with some consistency in terms of the properties, while either transformations of the data or interval-valued functions can be employed in the latter case.

22.4.2 Desired Properties, Semantics and Interpretation

The first step in choosing an aggregation function once the data structure is known is usually to decide which class of either averaging, conjunctive, disjunctive or mixed is desired. As discussed in Section 22.3.1.1, sometimes it will be more important to have a function which sorts items into order of preference than one which gives easily interpreted outputs. We consider four functions whose semantics can be used to decide which class of function is required:

Minimum (conjunctive) The minimum uses the minimum input as its output. This means the function can only return a high output if all the inputs are high. Such aggregation is useful for certain KB or UB systems using Eq. (22.5) or even CB where it is desired that all the inputs be satisfied. Functions such as the product (T_P) have an accumulative effect for any output which is not perfect, so might be less useful than the min when the dimension is high.

Maximum (disjunctive) Whereas the minimum models AND-like aggregation, disjunctive functions model OR. This type of aggregation results in outputs which are equal to or greater than the highest input. This is useful in KB, UB or CB as well if there are multiple preferences or criteria and one good score is enough justification for recommendation. Consider Example 22.7.

Example 22.7. A user of a CB news recommender has the keywords {*Haruki Murakami, X-Men, bushfires, mathematics*, *Jupiter orbit*} associated with her profile. It is unlikely that any one news story will be highly relevant to all or even any few of these keywords, so the RS uses disjunctive aggregation as a basis for recommendation.

Arithmetic Mean (averaging) When aggregating user ratings in CF or item features in CB it is reasonable to assume that although scores will vary, if enough inputs are used, the output will be reliable. We do not want the recommendations to be severely affected by an isolated user that is unsatisfied with every item he purchases, or a single feature among twenty or so that is completely satisfied.

Uninorm (mixed) In cases where different behavior is required on different parts of the domain, a mixed aggregation function may be required. This can be as straightforward as deciding that only values with *all* high inputs should be high, or it could be that the bounded behavior affects the accuracy of the function. The use of a uninorm, for instance, allows high values to push the score up and low values push the score down. An item with consistently high scores would be preferred to one with mostly high scores but one or two low ones.

Certain properties of aggregation functions might also make them appealing. Table 22.1 lists the main aggregation functions we have presented and whether they always, or under certain circumstances, satisfy the properties detailed in Section 22.4.

Table 22.1: Aggregation Functions and Properties

Property	$WAM_{\mathbf{w}}$	$G_{\mathbf{w}}$	$H_{\mathbf{w}}$	$M_{\mathbf{w},[r]}$	C_v	S_v	$OWA_{\mathbf{w}}$	max	min	T_P	T_L	U	V
idempotent	♦	♦	♦	♦	♦	♦	♦	♦	♦				
symmetric	◊	◊	◊	◊	◊	◊	♦	♦	♦	♦	♦	♦	♦
asymmetric	◊	◊	◊	◊	◊	◊							
associative								♦	♦	♦	♦	♦	♦
strictly monotone	◊			◊	◊	◊	◊						
shift-invariance	♦			◊	♦	♦	♦	♦	♦				
homogeneous	♦	♦	♦	♦	♦	♦	♦	♦	♦				
Lipschitz continuous	♦		♦	◊	♦	♦	♦	♦	♦	♦	♦		△
neutral elements							◊	♦	♦	♦	♦	♦	
absorbent elements		♦	♦				◊	♦	♦	♦	♦		

♦ = always ◊ = depending on weights △ = depends on T,S used

22.4.3 Complexity and the Understanding of Function Behavior

In some cases, simple functions such as the WAM will be adequate to meet the goals of recommendation, with potential improvements to the RS lying in other directions. Due to its properties, the WAM is quite a robust and versatile function. It is not biased towards high or low scores, it does not accumulate the effects of errors, it is computationally inexpensive and its common use makes it well understood and easily interpreted. We present the power mean and Choquet integral as two example alternatives whose properties might make them more appropriate in certain situations.

The power mean The power mean is a parameterized function, capable of expressing functions that graduate from the minimum to the maximum including the WAM. This makes it immediately useful when fitting techniques are at our disposal, since we can use the one process to identify any number of functions as the best candidate. Consider the harmonic mean $M_{\mathbf{w},[-1]}$ and the quadratic mean $M_{\mathbf{w},[2]}$. The harmonic mean cannot give an output greater than zero if even one of the inputs is zero. This has the nice interpretation of only allowing items to be considered that at least partially satisfy every criteria, however it is not conjunctive, so still gives a score somewhere between the highest and lowest inputs. The harmonic mean is also concave and its output is equal to or less than the WAM for any choice of d_i. This allows less compensation for low inputs, so items must satisfy more of the criteria overall to rate highly. On the other hand, the quadratic power mean tends more towards high scores, favoring items that have a few very high scores which compensate more for low-scoring features or ratings.

The Choquet integral As with the power mean, the Choquet integral is capable of expressing functions ranging between the minimum and maximum. The use of the Choquet integral is most interesting in asymmetric situations where there tends to be some correlation. For example, in a KB recommender, sometimes preferences will be contradictory while at other times one implies the other. In the case of Entree [16], it is noted that users might demonstrate a preference for *inexpensive* and *nice* restaurants. Since usually some trade-off is involved, a restaurant that does satisfy these criteria should be especially rewarded when it comes to recommendation. In the case of CB movie recommendation, it could be that a user likes *Johnny Depp* and *Tim Burton*. As there is a high frequency of films which are directed by Tim Burton that also star Johnny Depp, it might not make sense to *double-count* these features. The Choquet integral can account for a combination of these situations, since a weight is allocated to each subset of criteria. The subset of "stars Depp AND is directed by Burton" would be allocated less weight than the sum of its parts, while inexpensive and nice restaurants in the KB example would be allocated more.

Of course, sometimes the structure of the data might be difficult to understand and interpret towards the use of a particular function. In these cases, it might be worthwhile to check the accuracy of a number of functions on a subset of the data. A comparison of the minimum, maximum, arithmetic mean and harmonic mean could suggest much about which functions will be useful.

22.4.4 Weight and Parameter Determination

The determination of weights for use in ratings aggregation for CF is often understood in terms of the similarity between users and neighborhood formation. Weights in CB and UB are a measure of the importance of each feature to the user, while the weights in weighted HS are indicative of the reliability of each component in recommendation. Weights can be selected using predetermined measures like cosine, or might be decided in advance by the RS designers - e.g. we decide to weight the similar users with a decreasing weighting vector $\mathbf{w} = (0.4, 0.3, 0.2, 0.1)$. Some systems adjust weights incrementally according to implicit or explicit feedback concerning the quality of recommendation, for instance in the hybrid RS, P-Tango [19]. In Section 22.5, programming methods are discussed for determining weights from available data-sets.

22.5 Sophisticated Aggregation Procedures in Recommender Systems: Tailoring for Specific Applications

We consider the fitting problem in terms of a CF recommender, however it is also possible to fit weights in CB and UB recommender systems provided the system has access to input and output values so that the strength of fit can affirm the suitability of the weights or parameters. Fitting can be accomplished by means of interpolation or approximation. In the case of interpolation, the aim is to fit the specified output values exactly (in the case of aggregation functions, the pairs $((0,0,\ldots,0),0)$ and $((1,1,\ldots,1),1)$ should always be interpolated). In the case of RS, the data will normally contain some errors or degree of approximation, and therefore it may not be appropriate to interpolate the inaccurate values. In this case our aim is to stay close to the desired outputs without actually matching them. This is the approximation problem.

The selection of an aggregation function can be stated formally as follows:

Let us have a number of mathematical properties $P_1, P_2, \ldots$ and the data $D = \{(\mathbf{x}_k, y_k)\}_{k=1}^K$. Choose an aggregation function f consistent with $P_1, P_2, \ldots$, and satisfying $f(\mathbf{x}_k) \approx y_k, k = 1, \ldots, K$.

We can also vary the problem to accommodate a fitting to intervals, i.e. we require $f(\mathbf{x}_k) \in [\underline{y}_k, \bar{y}_k]$. How these values are specified will depend on the application. In some cases it may be possible to fit the function exactly without violating any of the desired properties, however most of the time we merely want to minimize the error of approximation.

Mathematically, the satisfaction of approximate equalities $f(\mathbf{x}_k) \approx y_k$ can be translated into the following minimization problem.

$$\begin{aligned} &\text{minimize } ||\mathbf{r}|| \\ &\text{subject to } f \text{ satisfies } \mathcal{P}_1, \mathcal{P}_2, \ldots, \end{aligned} \tag{22.10}$$

where $||\mathbf{r}||$ is the norm of the residuals, i.e., $\mathbf{r} \in R^K$ is the vector of the differences between the predicted and observed values $r_k = f(\mathbf{x}_k) - y_k$. There are many ways to choose the norm, and the most popular are the least squares norm

$$||\mathbf{r}||_2 = \left(\sum_{k=1}^{K} r_k^2 \right)^{1/2},$$

and the least absolute deviation norm

$$||\mathbf{r}||_1 = \sum_{k=1}^{K} |r_k|,$$

or their weighted analogues if some of the y_k are considered less reliable than others.

Consider Example 22.8.[7]

[7] All examples in this section utilize the software packages *aotool* and *fmtools* [8]

Example 22.8. In a CF recommending application we want to use five similar users to predict the ratings of new objects for a given user. At hand we have a data set of many items previously rated by the user and the five similar users or neighbors $\{(d_i, R(u,d_i))\}_{i=1}^{10}$ where $d_i = (R(u_1,d_i),...,R(u_5,d_i))$ denotes the ratings given by each of the neighbors $u_1,...,u_5$ to a past item d_i, and the $R(u,d_i)$ are the user's actual ratings. I.e. $d_i = \mathbf{x}_k, R(u,d_i) = y_k$ from above. Table 22.2 shows an example data set with two items rated by the neighbors which the user is yet to rate and could be recommended. We want to define a weighted arithmetic mean using the least squares approach that assigns a weight w_i to each user. So we have

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{1} 0 \left(\sum_{j=1}^{5} w_j R(u_j,d_i) - R(u,d_i) \right)^2 \\ \text{subject to} \quad & \sum_{j=1}^{5} w_j = 1, \\ & w_1,\ldots,w_5 \geq 0. \end{aligned}$$

This is a quadratic programming problem, which is solved by a number of standard methods. In the current example one resulting model allocates the weights $\mathbf{w} =< 0.27, 0.07, 0.06, 0.19, 0.41 >$ with recommendation scores of 4.7 and 7.9 for the unrated items. The maximum difference between observed and predicted ratings is 2.45 with an average of 0.98. If we had instead used the cosine calculation to define the weights, we would have $\mathbf{w} =< 0.19, 0.24, 0.23, 0.18, 0.17 >$ and recommendation scores of 5.6 and 7.1. The accuracy is similar for this method, with maximum error 2.48 and average error 1.6. Interestingly u_5 was least similar using this measure, but most important when accurately predicting the ratings for u.

Table 22.2: Example dataset for mutually rated items in CF

	Items $i = 1..10$ rated by user and neighbors										Unrated	
User ratings $R(u,d_i)$	6	4	6	8	10	5	7	7	5	5	?	?
Neighbor ratings												
$R(u_1,d_i)$	4	4	4	8	10	3	7	5	3	3	4	7
$R(u_2,d_i)$	6	0	6	4	6	1	3	3	1	5	8	7
$R(u_3,d_i)$	3	1	8	5	7	2	4	4	2	2	7	5
$R(u_4,d_i)$	6	5	6	8	8	6	5	5	3	5	3	8
$R(u_5,d_i)$	6	4	6	7	8	1	5	8	5	8	5	9

As mentioned, if the number of items to be recommended is limited, the ranking, rather than the accuracy of prediction becomes crucial (see also [27]). In situations where it makes sense, the ranking of the outputs can be preserved with $f(R(u_1,d_k),...,R(u_n,d_k)) \leq f(R(u_1,d_l),...,R(u_n,d_l))$ if $R(u,d_k) \leq R(u,d_l)$ for all pairs k,l added as an extra constraint. In CF, imposing this condition weights the similar users higher who have rankings that better reflect the user's. This is useful when we know that some users might tend to overrate or underrate items, but will be consistent in terms of the items they prefer.

The approximation problem thus far described may turn out to be a general nonlinear optimization problem, or a problem from a special class. Some optimization problems utilize a convex objective function or variant of this, in which case the difficulty is not so much in this step, but rather in defining the constraints. Fitting the Choquet integral, for instance has an exponential number of constraints which need to be defined. Many problems, however can be specified as linear or quadratic programming problems, which have been extensively studied with many solution techniques available. Example 22.9 uses the same dataset (Table 22.2) with the Choquet integral as the desired function. In practice, it is preferable to have a much larger data set for the Choquet integral given that it is defined at 2^n points (so ideally, the number of data for fitting should be well above this). This ensures that the resulting function is not too specialized.

Example 22.9. (Continued from Example 22.8)... The system designers decide that they would prefer to use a Choquet integral to predict the unknown ratings. To make the fitting process less susceptible to outliers, they decide to use the least absolute deviation norm and express the optimization process as the following:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{5} |C_v(d_i) - R(u,d_i)| \\ \text{subject to} \quad & v(A) - v(B) \geq 0, \text{ for all } B \subseteq A, \\ & v(A) \geq 0, \forall A \subset \mathcal{N}, v(\emptyset) = 0, v(\mathcal{N}) = 1. \end{aligned}$$

This results in a Choquet integral defined by a fuzzy measure with the following values

$$v(\{1\}) = 1, v(\{2\}) = 0.33, v(\{3\}) = 0, v(\{4\}) = v(\{5\}) = 0.67$$

$$v(\{2,3\}) = 0.33, v(\{2,4\}) = v(\{3,4\}) = v(\{3,5\}) = v(\{2,3,4\}) = 0.67$$

$$v(A) = 1 \text{ for all other subsets.}$$

The Shapley values provide a good indication of the influence of each of the neighbors, and are given as

$$\phi_1 = 0.39, \phi_2 = 0.11, \phi_3 = 0, \phi_4 = 0.22, \phi_5 = 0.28$$

. As with the weighted arithmetic mean, the values suggest that neighbors 1, 4 and 5 are perhaps more similar to the given user. We also note the interaction indices for pairs, given as

$$I_{12} = I_{24} = I_{45} = -0.17, I_{14} = -0.33, I_{15} = -0.5$$

$$I_{ij} = 0 \text{ for all other pairs.}$$

This shows the redundancy between some of the neighbors. In particular, neighbors 1 and 5 are very similar. The maximum error in this case is 1.6 and the average error is 0.6, with resulting recommendations 6.0 and 8.7. Because of the substitutive variables, the function behaves similar to a maximum function. We see the high score given for the latter item, mainly due to the high ratings given by neighbors 4 and 5.

The families of aggregation functions defined in Section 22.3.2 are convenient to use when trying to understand and interpret the results. The weights and parameters have a tangible meaning and fitting these functions essentially involves finding the best values for each parameter to maximize the reliability of the RS.

In other situations however, the interpretation side of things may not be as important: we just want to predict the unknown ratings reliably and automatically. There are many non-parametric methods for building aggregation functions, which do not have the advantage of system interpretation, however can be constructed automatically and fit the data closely. One "black-box" type method is to build a general aggregation operator piecewise from the data. We can ensure that monotonicity and boundary conditions are specified by smoothing the data and ensuring these properties hold for each individual segment. We consider here, the construction of spline based aggregation functions [10].

Monotone tensor product splines are defined as

$$f_B(x_1,...,x_n) = \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \cdots \sum_{j_n=1}^{J_n} c_{j_1 j_2 \ldots j_n} B_{j_1}(x_1) B_{j_2}(x_2) \ldots B_{j_n}(x_n).$$

If it is desired the built function belong to a particular class or hold certain properties, additional constraints can be added when fitting. In particular, we can ensure monotonicity holds by expressing linear conditions on the coefficients $c_{j_1 j_2 \cdots j_n}$. The fitting of this function to data involves sparse matrices, their size increasing with the number of basis functions in respect to each variable and exponentially with n. We give an example of this fitting process in the Example 22.10.

Example 22.10. (Continued from Examples 22.8-22.9). It is not necessary in our application that the weighting of similar users be known. We simply want automatically built functions that can predict the ratings of unseen items. We decide that we still desire the properties of monotonicity and idempotency to ensure reliable outputs, and build a general aggregation operator represented by tensor product splines. The following quadratic programming problem is used:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{5} (f_B(d_i) - R(u,d_i))^2 \\ \text{subject to} \quad & \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \dots \sum_{j_n=1}^{J_n} c_{j_1 j_2 \dots j_n} \geq 0, \\ & f_B(0,...,0) = 0, f_B(1,...,1) = 0. \end{aligned}$$

Idempotency is also ensured by imposing a number of interpolation conditions such that $f_B(t_i,...,t_i) = t_i$. These conditions must be chosen in a certain way (see [6, 7]). The fitted non-parametric function gives resulting recommendation scores for the unrated items of 4.2 and 8.1 so it seems that the latter item should be suggested to the user.

Clearly it is the choice of system designers of whether to use non-parametric or parametric methods, and how complex an aggregation function should be used. Recommender systems usually require timely decisions and deal with large data sets, so a compromise between expressibility and simplicity is usually sought.

22.6 Conclusions

The purpose of this chapter has been to present the state of the art in aggregation functions and introduce established families of these functions that have properties useful for the purposes of recommendation. This has included means defined with various weights, Choquet integrals defined with respect to fuzzy measures, t-norms/t-conorms which can be built from generators, and representable uninorms. Many of the current methods used in recommender systems involve constructing weighted arithmetic means where weights are determined by varying measures of similarity, however in many cases the accuracy and flexibility of functions could be improved with only slight increases to complexity. We have provided a number of illustrative examples of the different ways in which aggregation functions can be applied to recommendation processes including ratings aggregation, feature combination, similarity and neighborhood formation and component combination in weighted hybrid systems. We also referred to some current software tools which can be used to fit these functions to data (see also [29, 25]) when we are trying to find weights, similarity or the parameters used that best model the dataset.

The research in aggregation functions is extensive with a number of important results, some of which have been explored with the application to recommender systems in mind. As only an introduction has been provided here, we recommend the recent books listed under *Further Reading* which provide details of many aggregation methods.

22.7 Further Reading

- Alsina, C., Frank, M.J. and Schweizer, B.: Associative Functions: Triangular Norms And Copulas. World Scientific, Singapore (2006)
- Beliakov, G., Pradera, A. and Calvo, T.: Aggregation Functions: A guide for practitioners. Springer, Heidelberg, Berlin, New York (2007)
- Calvo, T. and Mayor, G. and Mesiar, R.: Aggregation Operators: New Trends and Applications. Physica-Verlag, Heidelberg, New York (2002)
- Grabisch, M., Marichal, J.-L. Mesiar, R. and Pap, E.: Aggregation Functions. Cambridge University Press, Encyclopedia of Mathematics and its Applications, No 127, Cambridge (2009)
- Klement, E.P., Mesiar, R. and Pap, E.: Triangular Norms. Kluwer, Dordrecht, (2000)
- Torra, V. and Narukawa, Y.: Modeling Decisions. Information Fusion and Aggregation Operators. Springer, Berlin, Heidelberg (2007)

Acknowledgements

T. Calvo wishes to acknowledge her support from the projects MTM2006-08322 and PR2007-0193 from Ministerio de Educación y Ciencia, Spain and the EU project 143423-2008-LLP-ES-KA3-KA3MP.

References

1. Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A.: Incorporating contextual information in recommender systems using a multi-dimensional approach, ACM Transactions on information systems, **23**(1), 103–145 (2005)
2. Adomavicius, G. and Kwon, Y.: New Recommendation Techniques for Multicriteria Rating Systems. IEEE Intelligent Systems. **22**(3), 48–55 (2007)
3. Ahn, H.J.: A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Sciences. **178**, 37–51 (2008)
4. Al-Shamri, M.Y.H. and Bharadwaj, K.K.: Fuzzy-genetic approach to recommender systems based on a novel hybrid user model. Expert Systems with Applications, **35**, 1386–1399 (2008)
5. Balabanovic, M. and Shoham, Y.: Fab: Content-Based, Collaborative Recommendation. Comm. ACM. **40**(3), 66-72 (1997)
6. Beliakov, G.: Monotone approximation of aggregation operators using least squares splines. Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems. **10**, 659–676 (2002)
7. Beliakov, G.: How to build aggregation operators from data? Int. J. Intelligent Systems. **18**, 903–923 (2003)
8. Beliakov, G.: FMTools package, version 1.0, http://www.deakin.edu.au/∼gleb/aotool.html, (2007)
9. Beliakov, G. and Calvo, T.: Construction of Aggregation Operators With Noble Reinforcement. IEEE Transactions on Fuzzy Systems. **15**(6), 1209–1218 (2007)
10. Beliakov, G., Pradera, A. and Calvo, T.: Aggregation Functions: A guide for practitioners. Springer, Heidelberg, Berlin, New York (2007)
11. Beliakov, G., Calvo, T. and James, S: On Lipschitz properties of generated aggregation functions. Fuzzy Sets and Systems, 161, 1437-1447 (2010)
12. Beliakov, G. and James, S: Using Choquet Integrals for kNN Approximation and Classification. In Gary G. Feng (ed.), 2008 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008), 1311-1317 (2008)
13. Bolton, J., Gader, P. and Wilson, J.N.: Discrete Choquet Integral as a Distance Metric. IEEE Trans. on Fuzzy Systems, **16**(4), 1107–1110 (2008)
14. Buchanan, B. and Shortliffe, E.: Rule-based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley, Reading, MA (1984)
15. Bullen, P.S.: Handbook of Means and Their Inequalities. Kluwer, Dordrecht (2003)
16. Burke, R.: Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-adapted interaction, **12**(4), 331–370 (2002)
17. Calvo, T., Kolesárová, A., Komorníková, M. and Mesiar, R.: Aggregation operators: properties, classes and construction methods. In : Calvo, T., Mayor, G. and Mesiar, R. (eds.) Aggregation Operators. New Trends and Applications, pp. 3–104. Physica-Verlag, Heidelberg, New York (2002)
18. Chen, Y.-L. and Cheng, L.-C.: A novel collaborative filtering approach for recommending ranked items. Expert Systems with Applications. **34**, 2396-2405 (2008)
19. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D. and Sartin, M.: Combining Content-Based and Collaborative Filters in an Online Newspaper. In Proceedings of SIGIR 99 Workshop on Recommender Systems: Algorithms and Evaluation, Berkeley, CA (1999)
20. Campos, L.M.d., Fernández-Luna, J.M. and Huete, J.F.: A collaborative recommender system based on probabilistic inference from fuzzy observations. Fuzzy Sets and Systems. **159**, 1554–1576 (2008)
21. Dubois, D. and Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
22. Dubois, D. and Prade, H.: Fundamentals of Fuzzy Sets. Kluwer, Boston (2000)
23. Dubois, D., Hllermeier, E., and Prade, H.: Fuzzy methods for case-based recommendation and decision support. J. Intell. Inform. Systems. **27**(2), 95-115 (2006)

24. Duda, R. Hart, P. and Nilsson, N.: Subjective Bayesian methods for rule-based inference systems. In Proc. Nat. Comput. Conf. (AFIPS), volume 45, 1075–1082 (1976)
25. Grabisch, M., Kojadinovic, I., and Meyer, P.: A review of methods for capacity identification in Choquet integral based multi-attribute utility theory: Applications of the Kappalab R package. European Journal of Operational Research. **186**, 766–785 (2008)
26. Hibbert, R.: What is Indie Rock? Popular Music and Society. **28**(1), 55-77 (2005)
27. Kaymak, U. and van Nauta Lemke, H.R.: Selecting an aggregation operator for fuzzy decision making. In 3rd IEEE Intl. Conf. on Fuzzy Systems, volume 2, 1418–1422 (1994)
28. Klement, E.P., Mesiar, R. and Pap, E.: Triangular Norms. Kluwer, Dordrecht, (2000)
29. Kojadinovic, I. and Grabisch, M.: Non additive measure and integral manipulation functions, *R* package version 0.2, http://www.polytech.univ-nantes.fr/kappalab, (2005)
30. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., and Riedl, J.: GroupLens: applying collaborative filtering to Usenet news, Communications of the ACM, **40**(3), 77–87 (1997)
31. Krawczyk, H., Knopa, R., Lipczynska, K., and Lipczynski, M.: Web-based Endoscopy Recommender System - ERS. In International Conference on Parallel Computing in Electrical Engineering (PARELEC'00), Quebec, Canada, 257–261 (2000)
32. Linden, G., Smith, B., and York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, **7**(1), 76–80 (2003)
33. Mayor, G., and Calvo, T.: Extended aggregation functions. In IFSA'97, volume 1, Prague, 281–285 (1997)
34. Miller, B.N., Albert, I., Lam, S.K., Konstan, J.A., and Riedl, J.: MovieLens unplugged: experiences with an occasionally connected recommender system. In Proceedings of the 8th international ACM conference on Intelligent user intervaces, Miami, USA, 263–266 (2003)
35. Riedl, J. and Dourish, P.: Introduction to the special section on recommender systems, ACM Transactions on Computer-Human Interaction, **12**(9), 371–373 (2005)
36. Resnick, P., Iakovou, N., Sushak, M., Bergstrom, P., and Riedl, J.: GroupLens: An open architecture for collaborative filtering of Netnews. In Proceedings of ACM conference on computer supported cooperative work, Chapel Hill, NC, 175–186 (1994)
37. Rokach, L., Maimon, O., and Arbel, R.: Selective voting-getting more for less in sensor fusion, International Journal of Pattern Recognition and Artificial Intelligence 20 (3), pp. 329–350 (2006)
38. Rokach, L. and Maimon, O., Theory and applications of attribute decomposition, IEEE International Conference on Data Mining, IEEE Computer Society Press, pp. 473–480 (2001).
39. Salton, G. and McGill, M.: Introduction to Modern Information retrieval. McGraw Hill, New York (1983)
40. Schafer, J.B., Konstan, J.A., and Riedl, J.: E-commerce recommendation applications, Data Mining and Knowledge Discover, **5**, 115–153 (2001)
41. Santini, S. and Jain, R.: Similarity Measures. IEEE Transactions on Pattern Analysis and Machine Intelligence, **21**(9), 871–883 (1999)
42. Ujjin, S., and Bentley, P.: Using evolutionary to learn user preferences. In: Tan, K., Lim, M., Yao, X. and Wang, L. (eds.). Recent advances in simulated evolution and learning, pp. 20-40. World Scientific Publishing (2004)
43. Yager, R.: On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Trans. on Systems, Man and Cybernetics. **18**, 183–190 (1988)
44. Yager, R.: Noble Reinforcement in Disjunctive Aggregation Operators. IEEE Transactions on Fuzzy Systems. **11(6)** 754–767 (2003)
45. Yager, R. R. and Filev, D. P.: Induced ordered weighted averaging operators. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, **20**(2), 141–150 (1999)
46. Yager, R. and Rybalov. A.: Uninorm aggregation operators. Fuzzy Sets and Systems. **80**, 111–120 (1996)