A Framework for Recommending Collections

Jennifer Golbeck
College of Information Studies
University of Maryland, College Park, MD
jgolbeck@umd.edu

Derek L. Hansen
College of Information Studies
University of Maryland, College Park, MD
dlhansen@umd.edu

ABSTRACT

To date, the vast majority of recommender systems research has addressed the problem of recommending individual items that the user will like. Recommending collections of items rather than individual items is an important open space of research in the recommender systems community. In this paper, we present a comprehensive framework for describing and evaluating collections of items. This framework is designed to be domain independent and applicable to any collection recommendation problem. Our framework includes a categorization scheme for describing collections and a list of features upon which a collection can be evaluated. We present a number of examples that showed how these different attribute and evaluation techniques can be combined and applied in a given domain. We then discuss issues relevant to the building of these systems. This includes challenges in obtaining data about users' preferences for collections. We propose methods that include obtaining and analyzing existing collections from websites and developing multi-player online games to generate data about replacements and preferences. In addition, we look at how collection recommenders could be used to assist users either by creating collections from scratch or by assisting users in their own collection creation tasks. We believe this framing of an important problem will lead to new research in the development and evaluation of algorithms for recommending collections in interesting applications and with cross-domain applicability.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Human Factors

Keywords

recommender systems, collections, collection recommender systems

1. INTRODUCTION

To date, nearly all research in recommender systems has focused on recommending individual items that the user will like. This has been successful and is useful in a wide range of domains. Recommending collections of items as a distinct task from recommending individual items also has broad applicability, but has received very little attention in the literature.

There are many reasons to recommend collections of items. In many cases, items in a collection are complementary so that the value of each item is increased when it is combined with other items (e.g., ice-cream, banana, and hot fudge). Even when items are not complementary, recommending a bundle of items can help extract more consumer surplus by essentially sorting individuals into groups with different reservation prices [1]. Information goods such as music, digital books, and software are particularly well suited for combining into collections. Those selling such collections can benefit from the "predictive power of bundling" which under many conditions can lead to increases in sales, efficiency, and profits when compared to selling the items individually [3]. The benefits of bundling items are suggested by the use of Amazon's "Buy Together Today" offers that provide one price for a bundle of items (e.g., two related books).

Many popular online systems thrive on user-generated collections. Collections not only are valuable in themselves, they often provide a meaningful activity that keeps participants coming back. Many major music websites facilitates the creation of user-generated playlists (e.g., imeem, Rhapsody). Tools like iTunes Genius automatically create playlists from music in your iTunes library, as well as recommend related music. They have received mixed reviews. Sites like Playlist.com, Mixpod, and MixTape.me create communities around the creation and sharing of music collections.

Yet, even these sites provide few tools to augment the playlist creation process through recommendations that consider the collection as a whole. In other domains collections are also common: Amazon has Listmania!, Flickr has Galleries, clothings stores such as Marie Claire allow you to create collections of clothing and accessories on a virtual model, recipe sites like AllRecipies have recipe boxes, and Colourlovers.com recommends colors that go well together. Nearly all of these sites currently only support manual creation of collections, often missing out on the opportunity to recommend items

that fit well with partially completed collections.

There are countless varieties of collections that can be recommended. Stock portfolios, playlists, menus, and hobby collections (e.g. Hummel figurines) are just a few diverse examples. The domain of a collection is certainly important for judging its quality. The domain includes the type of item being recommended, the environment in which the collection will be created and used, and by whom the collection will be created and used. However, a general framework that is domain independent has many benefits. It leads to a higher-level understanding of collections and allows recommendation strategies to be shared more easily between domains by identifying strategies and algorithms that work for certain classes of similar collections. For example, a technique for recommending meal plans for diabetics may also be useful in recommending stock portfolios because both are unordered collections treated as a unit that must meet a set of constraints.

Some top-n collection recommenders have focused on the quality of the set of recommended items as a whole collection, particularly with respect to diversity of items [2, 4, 20, 12, 19]. We discuss this work further below. However, there are many aspects of collection quality to be considered beyond what has been treated in top-n recommenders so far. In our previous work [7], we introduced some preliminary notions of collection recommendation. We focused on a certain subset of collections, of which mix tapes served as the canonical example. In this paper, we present a more comprehensive framework for describing collections by type and feature. We also present a full set of attributes by which the quality of a collection can be measured independent of its domain. Finally, we discuss challenges to collecting data needed to support the creation and evaluation of collection recommender systems and describe how the framework can be used in different ways to assist the user in building collections.

2. A FRAMEWORK FOR COLLECTIONS

A framework for collections must describe all the features of a collection that allow different types of collections to be compared and contrasted independent of their domain. These features do not address the quality of a collection; they simply describe it. We introduce four features in the framework: collection type - unit or selection, ordered or unordered, finite or infinite, and constrained or unconstrained. We also present six different attributes that can be measured to determine the quality of a collection: individual item ratings, order interaction effects, item co-occurrence effects, size, diversity, coverage, and balance.

2.1 Attributes of Collections

2.1.1 Collection Type: Unit or Selection

Some collections are treated as a single unit to be used as and evaluated as a single item (e.g. an outfit for a person to wear). Other collections are designed to be drawn from (e.g. a library of books). Here, we elaborate more on this distinction.

Unit Collections. Many collections are used as a single item. When all items in a collection are used together, as though the collection is its own item, we call this a *unit collection*.

For example, a mixtape is made up of a series of songs, but the quality of the collection can be evaluated separately from the songs themselves. A mix tape that randomly pulls together songs from completely unrelated genres – some classical songs, some gangsta rap, some death metal, and easy listening – and presents them in no particular order may be considered to be of lower quality than a tape that has a theme to tie all of its songs together, with carefully selected ordering to provide smooth transitions between songs, and with a diverse yet compatible set of songs that are enjoyable both individually and in relation to one another. Stock portfolios, family meals, edited volumes, and the collection of readings in a syllabus are other examples.

Selection Collections. In contrast to a unit collection are collections that are not designed to be used all at once, rather they exist as a set from which the user can draw a subset of items when needed. We call these *selection collections*.

A library is an example of a selection collection. We can evaluate the quality of the collection as a whole by considering how well it meets the needs it was set up to address. A library of cookbooks, for example, is not designed to be used as a unit where every cookbook is used at once. Rather, individual books are selected out of the collection and used when needed. Music libraries, wardrobes, and menus are other examples.

2.1.2 Ordered or Unordered

The order of items in a collection may be important or not. In the mix tape example above, ordering is very important to making a good tape. In other collections, order does not make sense, such as a stock portfolio or a collection of accessories for an outfit. Finally, depending on the domain, order may sometimes be important for a collection and not other times. A cookbook is a collection of recipes. One could argue that the order in which recipes appear in the cookbook is not important since the book is not read consecutively, but rather accessed at arbitrary points. On the other hand, if the book is treated as a unit, the ordering of items may tell a story or otherwise improve the experience of using the book, so it may matter.

2.1.3 Constrained or Unconstrained

For certain collections, there are constraints that must be met in order for the collection to be useful. A stock portfolio must be within a certain range of risk. As a more complicated example, a medical diet must have a certain number of calories, nutrients, and a balance of protein, carbohydrates, and fat. Certain foods may also be excluded. When recommending a collection with constraints, each item must be evaluated with respect to these requirements before it is added to the collection.

Note that some combinations of constraints can lead to computationally intractable problems. If, for example, the user

is designing a daily meal plan and needs to select a set of foods that has at least 1,500 calories and cannot exceed 1,700 calories, where foods must come from a balance of categories (starches, vegetables, proteins) and also achieve the recommended daily allowances of a set of nutrients, the problem quickly begins to look like a variant of the Knapsack Problem which is NP-Complete [5]. Recommender systems are not designed to search for optimal solutions; they find preferences. Thus, when putting constraints on collections it is important to consider if a recommender system is the appropriate technique for selecting items. Recommenders should be used when constraints are simple and user preferences are important, not when finding items that meet the constraints is the difficult problem.

2.1.4 Finite or Infinite

While no collection is truly infinite, we borrow the concept of finite and infinite horizons from game theory. In an iterated game, a finite horizon describes when the players know how many times the game will be played. An infinite horizon describes when the game is played such that the players do not know when it will end, or it is played so many times that the end seems so far away that the players treat it as though it will continue indefinitely. Translating this to collections, some collections have a fixed size that is small enough where users can consider the whole collection at once; we call these finite collections. Other collections are designed to be ever-increasing in size and these are what we call infinite collections. It is not that the set of items that make up the collection is infinite, it is that they are cycled through continuously without end.

To understand the distinction between a finite and infinite collection consider a music playlist of classical music. The playlist itself could be considered a finite collection that stands on its own. Now consider a system that continuously samples from the classical music playlist, which can be thought of as a "seed" collection. It may randomly select songs from the playlist or it may select songs to play next based on rules such as "don't play the same song twice in a row" or "play songs from different time periods." No matter the case, the entire music stream would be considered an infinite collection and it could be judged independently of the underlying collection of songs from which it pulls.

Most of the examples mentioned so far – cookbooks, mix tapes, stock portfolios – are finite. A radio station or an ongoing meal plan for an individual are examples of infinite collections.

2.2 Valuing Collections

To create systems that recommend collections of items, we must have a method of scoring a collection to determine if one collection is better than another. In this section we describe a set of measures that can be used to evaluate collections. Note that some of these evaluation methods will not apply to all type-feature combinations of collections.

2.2.1 Individual Item Values

In current research that recommends sets of items, the individual item value has been the primary – and often the only — concern. When creating a collection, including items that the user will like will certainly make it better. Previous work has shown this for mix tapes, where collections with many highly rated songs outperformed those with many poorly rated songs [16]. Thus, collection recommenders should consider the user preferences for individual items. Evaluating the quality of a given item for a user is where the bulk of existing recommender systems research has focused [8]. Existing techniques can be used for this part of the evaluation.

Note that the tolerance for lower value items may be higher in a selection collection than in a unit collection because not every item in a selection collection need be used. Having a song that I don't particularly care for in my iTunes library (selection collection) doesn't lower the value of the entire collection as much as it would if it were included in a mix tape of 10 songs designed to be played straight through (item collection).

2.2.2 Order Interaction

In ordered collections, ordering can impact the quality of the collection in several ways. Absolute placement of an item can be important; some items may work better in a given position. For example, an overview article may fit best at the beginning of an edited collection of articles rather than in the middle or at the end. Similarly, some songs may work well as the first or last song in a mix tape. Or, songs with certain characteristics (e.g., "favorite" songs) may work best as first songs, a hypothesis our mix tape experiment discussed below. Note that this type of absolute placement order effect does not apply to infinite collections. It only applies to selection collections inasmuch as the absolute ordering helps with the selection process itself (e.g., items are listed alphabetically or sortable by other characteristics).

Relative placement of items to one another can also be important when ordering items in a collection. Two items may go very well together in a particular order, while other pairs may clash when placed in sequence. The relative ordering effects are typically independent of their absolute placement in the collection. For this reason they are applicable to infinite collections as well as finite collections. If two songs clash with one another, this is likely true if they are part of a mix tape (finite collection) or a radio station play sequence (infinite collection). Or, songs with certain characteristics (e.g., favorite songs) may work best as first songs, as demonstrated in earlier work [7].

2.2.3 Item Co-Occurrence Effects

Regardless of whether a collection is ordered, the interaction of items within it can affect its quality. Some items work well together and others do not. These co-occurrence effects are one of the most important factors in the success or failure of many collections.

It can be a complex task to evaluate co-occurrence effects. Even two items that both have high individual item ratings may not work well together. Someone might like chocolate and also like pickles, but not the two together. This is a rather intuitive effect when considering pairs, but gets more complicated when considering the quality of larger sets of items such as a triple.

For example, chocolate bars and graham crackers are a fine

combination; marshmallows and chocolate bars are also; and marshmallows and graham crackers are as well. None of these pairs are poor but neither are they exceptional. However, the combination of all three into a smore makes a much beloved snack for many people. The combination of all three items is better than would be indicated by looking at the three pairs. On the other hand, three items that are very good pairwise can make a bad triple. Consider building a research team of two professors and one graduate student. The professors may work well together, and each may work well with the student. However, all three may have trouble working together. The presence of a student may bring out some tension between the faculty members about who is in control, and the student may have trouble balancing work or contradictory instructions from them.

Similar scenarios can be made moving up from groups of three to four, and so on. While it is certainly useful to look at the compatibility of groups of two or even three items, this approach quickly becomes computationally difficult, requiring $O(n^k)$ comparisons for groups of size k.

Co-occurrence effects are most relevant to unit collections, where each item is directly tied to other items in a whole. However they may apply to selection collections. For example, a music selection collection that includes many music genres (e.g., rap, country, and gospel) may lose credibility and be valued less by those who strongly dislike one of those genres, even if they would not select songs of that genre when using the collection.

2.2.4 Size

For finite collections, the number of items in the collection may be important. Collections can be too big or too small, depending on the domain or purpose. Consider a collection of accessories for an outfit. Even if all of them work well together, there still may be too many for the collection to be considered good. On the other hand, a mix tape with only three songs would often be considered too short. Selection collections typically benefit from an increase in size since larger collections mean more options from which to choose. However, even selection collections can grow too large, making selections too challenging or time intensive.

2.2.5 Item Distribution

For collections to be successful, they may need items to be distributed in certain ways. For example, having diversity among recommended items has been shown to be important. Similarly, in some domains it is important to have items that cover a set of sub-categories and/or have a proper balance across those sub-categories. In these latter cases, the value derives not from the items simply being different from another, but from the fact that there are different categories represented and the distribution of items over categories is appropriate to the domain. These three ideas are distinct, but obviously strongly interrelated. To emphasize these differences we discuss each of these in separate sections, recognizing that their relationship is a tight one.

Item Diversity. Diversity of recommended items is one feature of collections that has been addressed by a handful of researchers in recent years, although it deserves much more attention. Although not discussed in the context of collections, researchers have recognized problems with many existing recommender systems which suggest the top-N items (e.g., Amazon's list of the 5 most related books) [2, 4, 20, 12]. The problem they have identified is that the items often lack sufficient diversity, recommending items that the person already knows or recommending items that are too similar (e.g., songs all by the same artist). Even though the items each have a high probability of being liked, they fail to satisfy the user's desire to be exposed to new material. Researchers have recognized that to overcome this problem they must consider the top-N recommended items as a "portfolio" rather than individual items [2].

Some authors have developed algorithms that recognize the need to balance and diversify recommendation lists in order to reflect a user's complete array of interests. For example, Zeigler et al. have considered the entire top-N "portfolio" in the context of recommending books [20]. They develop a "topic diversification" algorithm that balances accuracy of suggestions with an individual's full range of interests using existing hierarchal book classifications. They also develop a metric for measuring intra-list similarity that is generic enough to refer to different kinds of item features such as genre, author, timeframe. Their metric is designed for a case where order is not important since rearranging positions of recommendations in a top-N list does not affect the list's intra-list similarity metric.

Their user study found that item-based algorithms benefited from a small boost in diversity, while user-based algorithms did not [20]. Zhang and Hurley develop a general approach to considering diverse subsets of items (e.g., top-N lists) by considering the problem as the optimization of an objective function under constraints of a certain type [19]. They also develop an objective measure of diversity which only requires that there is a measure of the dissimilarity between each pair of items. Finally, diversity of a set is addressed in [13] in the context of news aggregators where users vote on articles. The approaches that these papers have developed could be applied to collections in addition to top-N lists.

Coverage. Diversity deals with having items that are different from one another. These differences may be based on the attributes of the items themselves or on the categories or genres into which the items fall. Increased diversity will have more items that are in different categories or have different attributes, or the magnitude of the difference between items will increase. Coverage, on the other hand, is interested only in the categories in which recommended items are found. Furthermore, coverage measures which categories are covered by the recommendation.

A collection may have high item diversity but poor coverage. For example, a cookbook may include a wide range of recipes of several types, suggesting high item diversity. However, it may not include any desserts or side dishes, suggesting poor coverage, particularly if it were a general purpose cookbook. Conversely, a cookbook with good coverage of all of the types of dishes may lack enough diversity of recipes and ingredients to make it valuable

The needed coverage will depend on the domain and intended use. While diversity and coverage are closely related, they are certainly distinct concepts measure different interactions between items in a set.

Balance. Balance is closely related to coverage. While coverage addresses if there are or are not items recommended in a specific category (essentially a binary measure), balance describes the distribution of items in categories. Balance can be applied with or without any category coverage requirements. Simply having a "good" proportion of recommended items among categories may be sufficient to make a good collection.

In the cookbook example above, all categories may be covered, but the balance may be poor. If, for example, the book was marketed as a general cookbook but 90% of the recipes were for main dishes featuring chicken, it would not be well balanced for its type.

3. EXAMPLE COLLECTIONS

There are countless types of collections with different features, requirements, and domains. We present several example collections to illustrate some of the different possibilities, see how they relate to our framework, and review related research that has been done in the space of collection recommendation. Table 3 shows even more examples and how they fit into the framework.

3.1 Family Dinner

As opposed to a multi-course meal, a family dinner is one where all dishes are served at once. The meal usually includes a main course and several side dishes, often including a vegetable and a starch. Depending on the number of people and the occasion, there may be a large number of options (e.g. American Thanksgiving dinner which often includes 6 or more side dishes) or only one choice for each category (one main course, one starch, and one vegetable dish).

This menu is not ordered. Thus, there are no ordering effects in the menu, but other interaction effects are present. Ideally, each menu item would be enjoyable by itself, and the combinations of items work well pairwise and overall. A menu with tacos as a main course would probably not serve cranberry sauce as a side dish. The size of the meal also matters. For two people, a dinner with 12 different dishes is likely to be considered to have too many items, whereas a meal with only two dishes may be completely appropriate.

The items in the meal are usually expected to provide some coverage of different categories (e.g. a main course and side dish). Among these dishes, there must be proper balance. Many people would consider a dinner for four with four loaves of garlic bread and one small piece of lasagna to share among all four people improperly balanced, even though it covers the main course and side dish categories. However, diversity of items beyond coverage and balances is sometimes but not always a requirement.

As just one example, a meal of fried chicken, french fries, and a biscuit has very little variety diversity relative to what is possible in a meal; two of the three items are fried, two are starches, and everything is similarly flavored and textured. While not the healthiest option, many people would consider this a tasty dinner and a good combination of items. Thus, while variety has its place, it is not always an important component of a single meal. Generally, these meals are not constrained, but if the domain is shifted to one of dieting or where there are medical conditions to be considered, constraints on many aspects of the meal could arise.

3.2 Collectible Card Games

Collectible Card Games, like Magic the Gathering, are games where players build decks of cards from their collections, and play a game with at least one other player using those cards. Thus, the overall collection of cards is a selection collection, since individual items are chosen from it to be used in a particular game. The quality of a collection is generally judged by its size, diversity, coverage, and balance.

With more cards, the player has more options in creating a deck. Thus, larger collections are almost always better. Games have different categories of cards, and having a proper balance among those categories, covering all the categories in some way, and having a wide range of cards from common to rare and across categories is important. Interestingly, though, the user's preference for individual cards does not generally impact the quality of a collection.

For example, in Magic the Gathering a large proportion of the cards - roughly 1/3rd - are common cards called "lands". These are necessary in this proportion for game play, but the value of an individual land card is extremely low. Common, low-valued cards of other types are also necessary to have well represented in the collection because they are needed in most decks for the player to be effective. Generally, individual cards that are rare and highly valued cannot be used extensively in a game deck because of the way the game is played, and this means they are also a small part of the overall collection. Thus, in this example, individual item values are not important to the value of the collection. Diversity, coverage, and balance, on the other hand, are critical.

3.3 Music Libraries and Playlists

One space of collection recommendation that has received significant attention in the literature is playlist generation. These systems build lists of songs for users based on their known preferences. However, much of this research focuses on building a list of songs where each song is evaluated individually; little attention is paid to the quality of the collection as a whole with focus on interaction effects, co-occurrence relationships, order effects, etc.

Consider an individual who has an iTunes Music Library of a few hundred songs. The library itself can be considered a collection, one that is typically a finite, selection collection where order is not particularly important (except perhaps to help locate a song). Constraints on the collection may include hard drive space and cost. Note that we could use the music library as the seed for an infinite collection that continuously played music from the library (e.g., in random order).

Although talking about music libraries can be useful in some contexts, users typically consider individual playlists - collec-

Table 1: A table of collection types with indications of the value measures that may apply to them. Note that these are intended as examples but there may be cases for a given type of collection where a different mix of measures would be used.

	Features				Value Measures						
Collection	**Type**	**Finite**	**Ordered**	**Constrained**	**Individual Items**	**Order Interaction**	**Co-Occurrence**	**Size**	**Diversity**	**Coverage**	**Balance**
Stock Portfolio	Unit	X		X	X			X	X	X	X
Mix Tape	Unit	X	X		X	X	X	X	X		
Playlist	Unit		X		X	X	X		X	X	X
Family Dinner	Unit	X			X		X	X		X	X
Fashion Runway Collection	Unit	X	X	X	X	X	X	X	X	X	X
Collectible Card Games	Selection	X						X	X	X	X
Medical Meal Plan	Selection			X	X				X	X	X
Cookbook	Selection	X			X				X	X	X
Radio Station	Selection		X		X	X	X		X	X	X
Board of Directors	Unit	X	X	X	X		X	X	X		

tions of songs that are pulled from a personal music library (or larger music database) into some coherent collection. Playlists can be hand-crafted or automatically generated. Indeed, automatic playlist generation via systems such as iTunes Genius, The Filter, and MusicIP are already popular. In these tools, users typically provide a seed song and the system automatically creates a list of related songs from the user's library, often using content-based approaches that measure the similarity of songs based on various dimensions (e.g., rhythm, artist, genre) (e.g., [15]).

These automatic playlist generators don't typically pay attention to order, simply showing the top-N similar songs, perhaps with a few dissimilar songs thrown in at the end of the list to enhance diversity (e.g., [11]). A few novel systems such as PATS try to balance a desire for coherence (i.e., similarity of songs) and variation (diversity of songs) by assuring that the same song is not recommended multiple times [16]. Their approach was successful in that PATS-generated playlists outperformed randomly assembled playlists [16]. A user study of an automatic playlist generator running on a mobile device showed that there was significant interest in such tools and that there is a need to group or spread out songs that are overly similar (e.g., from the same artist) [11], suggesting that relative order effects are important.

As with the iTunes Music Library example, playlists can be used as seeds for infinite collections that cycle through the songs in the playlist, as for example occurs when songs from a playlist are selected and played as background music at a party. The way in which songs from the playlist are cycled through may take into consideration order effects, diversity, coverage, and balance, or it could be completely random.

Playlists also highlight how it is possible to conflate several types of collections. Note that the quality of the music library and the quality of a playlist created from that library are related, but different. The music library should be evaluated as its own collection. Since the music collection serves at least in part as a selection collection from which playlists can be created, the music library should be fairly large, diverse, and have good balance and coverage. If the library is only used for a specific genre (e.g. classical music) it should still have all those attributes within the given genre. The playlists created from this library obviously depend on the collection of items available, but are judged on other criteria. This will include the diversity of songs selected, the order interaction as one song flows to the next, its coverage and balance, and the quality of the individual items.

3.4 Mixtapes

Unlike playlists, which often serve as seeds for infinite collections that can continue forever, mixtapes are always finite collections, usually with fewer than 20 songs. This difference allows for consideration of absolute placement in the ordering (e.g. which song goes first or last), and farther reaching interaction effects as we judge the flow of songs over the whole collection rather than within a sliding window.

In previous work [7], we ran experiments with users, asking them to create mix tapes of 10 songs from a set of 15 possible songs. Subjects were also asked what factors they thought were important in making a good mixtape. Our results showed that subjects included songs they liked more often (individual item values), that the first song on the mixes was rated significantly higher than songs in other positions (order interaction effects), and certain songs appeared together much more often than expected while others were never used together (co-occurence effects). In the open responses, 70% of subjects said that there should be a theme

to a mixtape (co-occurrence effects on a larger scale than pairwise interactions) and 2/3rds of subjects said that the order of songs is important. These quantitative and qualitative results show that apart from the individual songs that make them up, mixtapes have value as collections and that certain features can make one mix better than another.

4. RECOMMENDER SYSTEMS FOR COLLECTIONS

Once this background data is available and the type and attributes of the collection have been identified, there are many ways a collection recommender system can be used. While item recommender systems generally suggest one item or a set of items from which the user can choose, collection recommenders have more possibilities. They can suggest whole collections, assist users in their creation of collections, and help improve existing collections by offering additions, removals, and replacements according to constraints or the user's preferences.

Certainly, recommending entire collections from scratch is important and useful. There are many domains where fully automated collection generation is desired. For example, if a user is at the gym with her MP3 player, she may not have time to create a playlist from scratch. In this case, a system that automatically chooses and orders songs with little to no user intervention is desirable. Users with little to no knowledge of the stock market may have no preferences about individual stocks, and so after specifying constraints for the portfolio, a system that automatically selects investments would be useful. In fact, this latter example is similar to the way people invest when choosing a fund; they do not focus on the individual items but rather select an existing collection with attributes that best meet their desires for risk, return, etc.

On the other hand, there are also many cases where users do not want fully automated recommendations of collections. Rather, they would prefer a system that helps them in their own collection creation. One domain where this has been studied is in playlist generation. Users have complained that automatic playlist generators remove the fun of creating playlists and do not provide enough possibilities for customizing playlists [11]. One approach to overcome this problem is to create a semi-automatic playlist generator such as SatisFly that augments the creation of playlists by recommending songs that fit various specified constraints [17]. This general approach leads to questions not just in collection recommendation but also in designing appropriate user interfaces and social practices around the use of these system. These recommender systems that augment collection creation will need to walk a fine line between suggesting content while still facilitating exploration and autonomy.

With proper background knowledge, these recommenders can also be built into existing systems. For example, a person with a Hummel figurine collection may search eBay for new items. A collection recommender could work on top of eBay, searching available items and ranking those which would add the most value to the existing collection. Similarly, a recipe website that allowed users to input the dishes they planned to serve could suggest other recipes to fill out the meal with compatible items. Making changes to existing collections could also be a useful application of these algorithms. Someone may have a recipe and want a substitution for an item.

For example, someone who does not like asparagus may ask the system to recommend a replacement for a stir fry, and the system could look at its underlying data and suggest snow peas as a substitute. More generally, systems could allow users to increase the level of diversity in a collection along a sliding scale or highlight items that may be problematic when placed together.

Indeed, optimizing any feature of collections - diversity, individual item preference, etc. - by adding, removing, or changing items are all valid and useful techniques for recommender systems in this area.

Although many collections are used by an individual, other collections are used by many people. These shared collections are a particularly interesting area of future research. Indeed, group recommender systems that balance the preferences of multiple individuals to recommend items are an active area of research [14, 9]. Issues such as diversity, item co-occurrence interaction effects, coverage, and balance within collections seem particularly important within a group context.

Finally, it is worth noting that it is possible that for some types of collections it will simply not be possible to produce a recommender algorithm that takes into account all the value measures that apply to the collection. The data space may simply be too sparse, even in the most well used systems. The interaction of items in a collection and the connection between those interaction effects and personal taste may also be too complex for a recommender system to address. As algorithms for these systems and data collection mechanisms are developed, the limitations will become clearer.

5. CONCLUSIONS

Recommending collections of items rather than individual items is an important open space of research in the recommender systems community. In this paper, we presented a comprehensive framework for describing and evaluating collections independent of their domain. Collection types include unit or selection collections, ordered and unordered, finite and infinite, and constrained or unconstrained.

The quality of these collections is judged based on the value of the individual items, order interaction (on ordered collections), co-occurrence effects, size, diversity, coverage, and balance. We presented a number of examples that showed how these different attribute and evaluation techniques could be combined and applied in a given domain.

Work that looks at more diverse types of collections will provide many valuable insights into collection recommendation generally as well as to the specific domain. There is also independent research to be done in the data collection techniques. The games research described in [10, 6, 18] has been successful in gathering data for individual item collection, and projects that extend this research to collections would be interesting and relatively straightforward to conduct. Our framework helps in this area particularly because

once a data collection technique is developed for a particular problem, it should be immediately and directly applicable to problems with the same framework attributes and valuation methods.

In addition, collection recommender systems can support a variety of different applications: automatic collection creation, augmented collection development, and item selection. These techniques will all require usability research in addition to development of the algorithms themselves.

6. REFERENCES

[1] W. Adams and J. Yellen. Commodity bundling and the burden of monopoly. *The Quarterly Journal of Economics*, pages 475–498, 1976.

[2] K. Ali and W. van Stam. Tivo: making show recommendations using a distributed collaborative filtering architecture. In *Proceedings of the 2004 ACM Conference on Knowledge Discovery and Data Mining*, pages 394–401, New York, NY, USA, 2004. ACM.

[3] Y. Bakos and E. Brynjolfsson. Bundling information goods: Pricing, profits, and efficiency. *Management Science*, pages 1613–1630, 1999.

[4] K. Bradley and B. Smyth. Improving recommendation diversity. In *Proceedings of AAAI'01: The Sixteenth International Conference on Artificial Intelligence*, 2001.

[5] M. Garey, D. Johnson, et al. *Computers and Intractability: A Guide to the Theory of NP-completeness*. wh freeman San Francisco, 1979.

[6] S. Hacker and L. von Ahn. Matchin: eliciting user preferences with an online game. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1207–1216, New York, NY, USA, 2009. ACM.

[7] D. Hansen and J. Golbeck. Mixing it up: Recommending collections of items. In *CHI '09: Proceedings of the SIGCHI conference on Human factors in computing systems*, 2009.

[8] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.

[9] A. Jameson. More than the sum of its members: Challenges for group recommender systems. In *Proceedings of the working conference on Advanced visual interfaces*, pages 48–54. ACM New York, NY, USA, 2004.

[10] E. Law and L. von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1197–1206, New York, NY, USA, 2009. ACM.

[11] A. Lehtiniemi and J. Seppänen. Evaluation of automatic mobile playlist generator. In *Mobility '07: Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, pages 452–459, New York, NY, USA, 2007. ACM.

[12] S. M. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 1097–1101, New York, NY, USA, 2006. ACM.

[13] S. Munson, D. X. Zhou, and P. Resnick. Sidelines: An algorithm for increasing diversity in news and opinion aggregators. In *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media*, 2009.

[14] M. OConnor, D. Cosley, J. Konstan, and J. Riedl. PolyLens: A recommender system for groups of users. In *Proceedings of the European Conference on Computer-Supported Cooperative Work*, pages 199–218, 2001.

[15] E. Pampalk, A. Flexer, and G. Widmer. Hierarchical organization and description of music collections at the artist level. In *Proceedings of the 2005 European Conference on Digital Libraries*, pages 37–48, 2005.

[16] S. Pauws and B. Eggen. PATS: Realization and user evaluation of an automatic playlist generator. In *Proceedings of the 3rd International Conference on Music Information Retrieval*, pages 222–230, 2002.

[17] S. Pauws and S. van de Wijdeven. Evaluation of a new interactive playlist generation concept. In *ISMIR International Conference on Music Information Retrieval*, pages 638–643, 2005.

[18] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, New York, NY, USA, 2004. ACM.

[19] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130, New York, NY, USA, 2008. ACM.

[20] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the FIfteenth World Wide Web Conference*, pages 22–32, New York, NY, USA, 2005. ACM.