Building Recommender Systems using a Knowledge Base of Product Semantics

Rayid Ghani and Andrew Fano

Accenture Technology Labs
161 N Clark St, Chicago, IL 60601
{Rayid.Ghani, Andrew.E.Fano}@accenture.com

Abstract. Online retailers have access to large amounts of transactional data but current recommender systems tend to be short-sighted in nature and usually focus on the narrow problem of pushing a set of closely related products that try to satisfy the user's current need. Most e-commerce recommender systems analyze a large amount of transactional data without actually having any idea of what the items in the transactions mean or what they say about the customers who purchased or browsed those items. In this paper, we present a case study of a system that recommends items based on a custom-built knowledge base that consists of products and associated semantic attributes. Our system first extracts semantic features that characterize the domain of interest, apparel products in our case, using text learning techniques and populates a knowledge base with these products and features. The recommender system analyzes descriptions of products that the user browses or buys and automatically infers these semantic attributes to build a model of the user. This abstraction allows us to not only recommend other items in the same class of products that "match" the user model but also gives us the ability to understand the customer's "tastes" and recommend items across categories for which traditional collaborative filtering and content-based systems are unsuitable. Our approach also allows us to "explain" the recommendations in terms of qualitative features which, we believe, enhances the user experience and helps build the user's confidence in the recommendations.

1 Introduction

Recommender Systems are being increasingly used by online retailers such as Amazon.com, CDNow, Reel.com to suggest products and provide customers with information to help them find the product they want quickly. Such systems enable the retailers to conduct a virtual dialogue with every customer and respond to the customer's every click appropriately. Unfortunately, most systems currently in use have a very narrow focus of capturing transaction data and just recommending a closely related product instead of actually trying to understand the customer. For example, a store may know that a particular customer bought or browsed a blouse. They may also know past apparel views or purchases this customer has made and for each of the purchases they may know the SKU, date,

time, price paid, color and brand. While this data is useful, there is a lot of information not being captured that would facilitate insight into the customer and enable a variety of applications. For example, is the blouse conservative or flashy? How trendy is it? Is it casual or formal? Sporty? How strong is the appeal of the brand?

Such features, to a certain extent, constitute a model of the domain that can account for a person's taste in clothing. The problem, of course, is that these features tend not to be readily available in data warehouses or other databases that are accessible to retailers or manufacturers. We set out to see if we could extract these features by examining the marketing messages associated with a given product. One way of viewing the problem is to try to bridge the gap between the information formally represented in a retailers back-end system and the information implicitly contained in texts of marketing materials. We believe that product databases augmented with the semantic features that account for customer preferences in a particular product domain will enable a variety of CRM and strategic applications. To demonstrate the utility of such an approach we describe a recommender system that uses these features.

There are, of course, many recommender systems that work well without relying on a model of the features that account for a users preferences for a particular item type. Collaborative filtering approaches have been used successfully in a variety of domains [3, 13]. It is well known that this approach is subject to certain limitations, mostly related to situations that result in sparse data. First, collaborative filtering does not do well for newly introduced items because, by definition, they have not been available long enough to accumulate statistics about other items that are suitable complements. More generally, when the set of items available for recommendation changes frequently, as is the case for clothing in apparel stores, it is difficult to accumulate enough data on co-occurring pairs of items to reliably offer recommendations. Finally, if we want to make recommendations across item categories, once again, in most cases we are unlikely to have enough data to support usable recommendations. For example, if a department store knows the apparel a customer has bought and must now recommend furniture, it is very unlikely that they have enough data to look at combinations of people who have bought a particular shirt and couch. This is because the clothing offered changes quickly, the number of couches is large, and because people don't buy couches very frequently.

Content-based approaches have been used to address some of these limitations. These systems operate by comparing text descriptions or other representations associated with an item, for example book or movie descriptions [10, 9]. When usable product descriptions are available, content-based approaches can overcome some of the problems associated with sparse data. Still, they cannot enable cross category recommendations. This is because items across categories tend to be described in very different ways - even when they are suitable complements to each other. A couch in our department store, for example, is not likely to be described in a way that will easily match a blouse.

It may seem like a rather obscure problem: recommending couches to someone who bought a particular blouse. The point, in our view, is not any one cross category recommendation. Instead, the value of such an approach is that it begins to provide a model of customers and retailers that enables us to draw conclusions beyond a narrow range of products and reason at the level of customer tendencies and lifestyle. Such capabilities will arguably allow companies that deal with a variety of disparate customer needs, such as department stores or web portals to interact with their customers more intelligently. For example, if a customer that we've learned prefers flashy, trendy apparel, browses furniture, we can begin to recommend contemporary, daring furniture.

In this paper, we describe our initial work towards a recommender system that is capable of inferring these kinds of attributes enabling us to enhance product databases. The system learns these attributes from the marketing language associated with a product by applying text learning techniques to the product descriptions found on retailer web sites. These descriptions are written by marketers to position the product in the consumer's mind in a manner that implicitly suggests these softer attributes. By analyzing these descriptions, we are able to automatically extract these features and enhance our knowledge of products. This knowledge can then be used to create profiles of individuals that can be used for recommendations. Although the work described here is limited to the apparel domain and a particular set of features, we believe that this approach is relevant to a wider class of products and that abstracting from the product layer to the softer attributes (such as personal tastes and lifestyle) can add a potentially valuable dimension to existing recommender systems.

2 Overview of our approach

At a high level, our system deals with products in the apparels domain and extracts a predefined set of semantic features for each item. These features can generally be extracted from any text materials designed to appeal to a consumer's taste but in this paper, we only use the descriptions associated with each item on a retailer's website. The features extracted are then used to populate a knowledge base, which we call the product semantics knowledge base.

Knowledge Base Construction Phase

1. Collect information about apparel items/products
2. Define the set of features to be extracted
3. Label the data with values of the features defined in step 2
4. Train a classifier/extractor to use the labeled training data to now extract features from unseen data
5. Extract features from new products by using the trained classifier
6. Populate a knowledge base with the products and corresponding features

Recommendation Phase

1. Watch the user browse items/products
2. Extract the product name and description from the pages user visits

3. Infer features from new products by using the trained classifier
4. Build a user profile in terms of these semantic features
5. Recommend new items from the knowledge base that match this user profile

3 Data Collection

To collect data, we constructed a web crawler to visit web sites of several large apparel retail stores and extract names, urls, descriptions, prices and categories of all products available. This was done very cheaply by exploiting regularities in the html structure of the websites and manually writing wrappers. We realize that this restricts the collection of data from websites where we can construct wrappers and although automatically extracting names and descriptions of products from arbitrary websites would be an interesting application area for information extraction or segmentation algorithms using Markov Models [15], we decided to take the manual approach. The extracted items and features were placed in a database and a random subset was chosen to be labeled.

4 Defining the set of features to extract

After discussions with domain experts, we defined a set of features that would be useful to extract for each product. We believe that the choice of features should be made with particular applications in mind and that domain knowledge should be used. We currently infer values for 8 kinds of attributes for each item but are in the process of identifying more features that are potentially interesting. The features we currently extract are Age Group, Functionality, Price point, Formality, Degree of Conservativism, Degree of Sportiness, Degree of Trendiness and Degree of Brand Appeal. More details including the possible values for each feature are given in Table 1.

The last four features (conservative, sportiness, trendiness, and brand appeal) have five possible values 1 to 5 where 1 corresponds to low and 5 is the highest (e.g. for trendiness, 1 would be not trendy at all and 5 would be extremely trendy).

5 Labeling Training Data

The data (product name, descriptions ,categories, price) collected by crawling websites of apparel retailers was placed into a database and a small subset (600 products) was given to a group of fashion-aware people to label with respect to each of the features described in the previous section. They were presented with the description of the predefined set of features and the possible values that each feature could take (listed in Table 1).

Table 1. Details of features extracted from each product description

Feature Name	Possible Values	Description
Age Group	Juniors, Teens, GenX, Mature, All Ages	For what ages is this item most appropriate?
Functionality	Loungewear, Sportswear, Eveningwear, Business Casual, Business Formal	How will the item be used?
Formality	Informal , Somewhat Formal, Very Formal	
Conservative	1(gray suits) to 5 (Loud, flashy clothes)	Does this suggest the person is conservative or flashy?
Sportiness	1 to 5	
Trendiness	1 (Timeless Classic) to 5 (Current favorite)	Is this item popular now but likely to go out of style? or is it more timeless?
Brand Appeal	1(Brand makes the product unappealing) to 5 (high)	Is the brand known and appealing to a sizable group

6 Training from the Labeled Data

In this section, we describe the process of training models to learn the attributes associated with each of the products extracted from the retailers websites. We treat the learning problem as a traditional text classification problem and create one text classifier for each "semantic feature". For example, in the case of the Age Group feature, we classify the product into one of five classes (Juniors, Teens, GenX, Mature, All Ages). The initial learning algorithm we use for classifying the product names and descriptions is Naive Bayes which is a simple but effective text classification algorithm for learning from labeled data[8, 7]. Naive Bayes builds probabilistic models for each attribute and when given new product descriptions, calculates the likelihood of that product being associated with each of the attribute values.

Table 2 gives a list of words which had high weights for some of the features that we used the naive bayes classifier to extract. These words were selected by scoring all the words according to their log-odds-ratio scores and picking the top 10 words. Looking at the words gives us an intuitive idea of what type of words are informative of each attribute and verifies our initial hypothesis that the marketing language does correspond to these softer attributes.

6.1 Incorporating Unlabeled Data using EM

In our initial data collection phase ,we collected names and descriptions of thousands of women's apparel items from websites. Since the labeling process was expensive, we only labeled about 700 of those, leaving the rest as unlabeled. Recently, there has been much recent interest in supervised learning algorithms that combine information from labeled and unlabeled data. Such approaches include

using Expectation-Maximization to estimate maximum a posteriori parameters of a generative model [12], using a generative model built from unlabeled data to perform discriminative classification [5],using redundant feature splits for bootstrapping classifiers with co-training [1, 11] and using transductive inference for support vector machines to optimize performance on a specific test set [6]. These results have shown that using unlabeled data can significantly decrease classification error, especially when labeled training data are sparse.

For the case of textual data in general, and product descriptions in particular, obtaining the data is very cheap. A simple crawler can be build and large amounts of unlabeled data can be collected for very little cost. Since we had a large number of product descriptions that were collected but unlabeled, we decided to use the Expectation-Maximization algorithm to combine labeled and unlabeled data for our task. EM is an iterative statistical technique for maximum likelihood estimation in problems with incomplete data [2]. In our scenario, the class labels of the unlabeled data are treated as the missing values. It has been shown by [12] that this technique can significantly increase text classification accuracy when given limited amounts of labeled data and large amounts of unlabeled data.

Table 2. For each attribute, the table shows the top words ranked according to the weighted log-odds ratio scores.

Brand Appeal =5 (high)	Conservative =5 (high)	Conservative =1 (low)	Formality=Informal	Somewhat Formal
lauren	lauren	rose	jean	jacket
ralph	ralph	special	tommy	fully
dkny	breasted	leopard	jeans	button
kenneth	seasonless	chemise	denim	skirt
cole	trouser	straps	sweater	lines
imported	jones	flirty	pocket	york
	sport	spray	neck	seam
	classic	silk	tee	crepe
	blazer	platform	hilfiger	leather

Age Group = Juniors	Functionality =Loungewear	Functionality =Partywear	Sportiness =5 (high)	Trendiness =1 (low)
jrs	chemise	rock	sneaker	lauren
dkny	silk	dress	camp	seasonless
jeans	kimono	sateen	base	breasted
tee	calvin	length:	rubber	trouser
collegiate	klein	skirt	sole	pocket
logo	august	shirtdress	white	carefree
tommy	lounge	open	miraclesuit	ralph
polo	hilfiger	platform	athletic	blazer
short	robe	plaid	nylon	button
sneaker	gown	flower	mesh	

7 Recommendation Phase

Being able to analyze the text associated with products and map it to the set of predefined semantic features in real-time gives us the ability to create instant profiles of customers shopping in an online store. As the shopper browses products in a store, the system running in the background can extract the name and description of the items and using the trained classifiers, can infer semantic features of that product. This process can be used create instant profiles based on viewed items without knowing the identity of the shopper or the need to retrieve previous transaction data. This can be used to suggest subsequent products to new and infrequent customers for whom past transactional data may not be available. Of course, if historical data is available, our system can use that to build a better profile and recommend potentially more targeted products. We believe that this ability to engage and target new customers tackles one of the challenges currently faced by commercial recommender systems [14] and can help retain new customers.

We have built a prototype of a recommender system for women's apparel items by using our knowledge base of product semantics. The knowledge base is populated with thousands of items and their associated semantic attributes inferred by the learning algorithm described in earlier sections. Our system monitors the browsing behavior of user browsing a retailer's website and in real-time, extracts names and descriptions of products that they browse. The description text is then passed through our learned models and the semantic attributes of the products are inferred. For each product browsed, our system calculates $P(A_{i,j}\,|Product)\forall i,j$, where $A_{i,j}$ is the jth value of the ith attribute. The attributes are the semantic features described in Table 1 and the possible values for each attribute are also listed in the table. The user profile is constructed by combining these probabilities for each product browsed: User Profile =

$$\Pr(U_{i,j}\,|PastNItems) = \frac{1}{N}\sum_{k=1}^{N}\Pr(A_{i,j}\,|Item_k) \tag{1}$$

The user profile is also stored in terms of probabilities for each attribute value which allows us flexibility to include mixture models in future work in addition to being more robust to changes over time.

As the user browses products, the system compares the evolving profile against the products in the knowledge base, which has products classified into the same taxonomy of semantic features, and recommends the closest matching ones. Currently, we give equal weight to all products browsed when constructing the profile. In future work, we plan to experiment with different weighting schemes such as weighting recent items more than older ones.

We believe that our approach improves on collaborative filtering as it works for new products which users haven't browsed yet and can also present the user with explanations as to why they were recommended certain products (in terms of the semantic attributes). We believe that our system also performs better than standard content-based systems. Although content-based systems also use the

words in the descriptions of the items, they traditionally use those words to learn one scoring function. For example, a classical content-based recommendation engine takes the text from the descriptions of all the items that user has browsed or bought and learns a model (usually a binary target function: "recommend" or "not recommend"). In contrast, our system abstracts the feature space from words (thousands of features) to the eight semantic attributes. This still enables us to recommend a wide variety of products unlike most content-based systems. Table 3 lists a set of products that were recommended by our system in response to a user's interest in certain products. These recommendations are only in response to a single product browsed by the user and are similar in "taste" to the product browsed.

Table 3. Examples of products recommended by our system.

Product Browsed	Inferred Attributes	Products Recommended
I.N.C Pinstripe Shirt Traditional pinstripes combined with ruching, create a fresh, contemporary button-front shirt. Ruching at center front, sleeves and slit cuffs	AgeGroup=GenX Function=Bus. Casual Formality=High Conservativeness=High Trendiness=High Sportiness=Very Low Brand Appeal=High	Liz Claiborne Textured Suit Jacket Jones New York Sateen Pant CITY DKNY "Selie" Loafer Lizsport Check Skirt
Levi's Super-Low Boot-Cut Jean: Low-riding style prevails in the super sexy, low-rise boot-cut jean. From the best, in the business, Levi's delivers denim.	AgeGroup=Teens Functionality=Partywear Formality=Low Conservativeness=Low Sportiness=Very Low BrandAppeal=VeryHigh	Tommy Jeans Split-Flare Jean DKNY Mock-Fur Halter Sweater DKNY Camo Top CK Jeans Short Denim Skirt Kenneth Cole Racer-Back Sweater

A unique ability of our approach is to connect products that occur in seemingly different domains. As discussed earlier, both content-based and collaborative filtering approaches are not suitable for recommending products across different categories. The assumption underlying our claim is that consumer preferences are related across product categories. The way these preference attributes manifest themselves across products can vary and different product categories will have slightly different attributes. Nevertheless, once we are able to recognize relevant attributes in product categories of interest, we can have a manageable number of mappings across product-product preferences, enabling us to make such recommendations. For example, assuming again, that people who like flashy clothing also like contemporary furniture, consider how different the product descriptions are (see Table 4), yet once we have extracted these attributes, the mapping is straightforward. By constructing a domain model in terms of features that try to define the consumer's preferences for a general class of products, our system can then map the different domains and recommend a wide variety of products.

Table 4. Actual descriptions of two products from different categories

Category	Description
Apparel	Polo Jeans Co. Muscle Logo Tee. Strut your stuff in the Muscle Logo Tee. Flattering on the arms with a close to the body fit
Furniture	A strong, modern appeal with an organic influence and a touch of the tropics. Woven of abaca rope with a rich, rustic mahogany color finish.

Since our goal was not to build the best recommendation system but rather to demonstrate the potential of a system that is aimed at inferring a domain-specific model of the user that describes their personal tastes and lifestyle, we did not explore many approaches to building a user's profile. In future work, we plan to tackle the cases where a user's profile consists of a number of separate profiles. For example, if a user is looking for something for herself and also for someone else, our system should be able to recognize that the items that the user is buying or browsing are inherently different. This could be done through mixture models where we construct a profile using a mixture of different profiles. Another potential solution is to monitor the users profile as they browse more and more products. Since each product can be though of as a point in an n-dimensional Euclidean space (where n is the number of features, in our case, 8), we can calculate the distance of a new product from the current profile of the user. If a new product is "very" different from the current profile of the user (using thresholds based on cross-validation), it can be placed in a separate profile or treated as an outlier. We also plan to conduct user studies to validate the effectiveness of such a recommendation system based on these intermediate-level semantic features for a variety of conditions.

8 Conclusions and Future Work

We described our initial work towards a recommender system capable of inferring semantic attributes of products enabling us to "understand" the customer. The system learns these attributes by applying supervised and semi-supervised learning techniques to the product descriptions found on retailer web sites. One of the main assumptions we make is that descriptions associated with the products accurately convey the semantic attributes. We believe that this assumption is justified because in most cases these descriptions are written by marketers to position the product in the consumer's mind in a manner that implicitly suggests these softer attributes.

The main advantages of our system are that by mapping products to an abstract layer of semantic features, we can not only recommend other items in the same class of products that "match" the user model, but also understand the customer's "tastes" and recommend items across categories. Our approach also allows us to "explain" the recommendations in terms of qualitative features which, we believe, enhances the user experience and helps build the user's confidence in the recommendations[4].

Although the work described here is limited to the apparel domain and a particular set of features, we believe that this approach is relevant to a wider class

of problems and that inferring semantic attributes based on a domain model and using them in recommender systems is potentially valuable. The domain model we construct is not only useful for profiling customers, but also for profiling retailers. Analyzing their product offerings according to these features potentially gives us deeper insight in to the retailer as well as provide a sense of how the retailer positions their offerings. Additionally, the ability to profile retailers enables strategic applications such as competitive comparisons, monitoring brand positioning, tracking trends over time, etc.

We believe that we can be more effective at recommending products when we go beyond the immediately available data, such as the fact that a customer is looking at or bought a product, and start paying attention to what these products mean. Since this is just the beginning of our work, we plan to extend our data sources from retailers websites to incorporate multiple sources of data and expand to different classes of products.

References

1. A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of COLT*, pages 92–100, 1998.
2. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
3. D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *CACM*, 35(12):61–70, 1992.
4. J. Herlocker, J. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *CSCW*, 2000.
5. T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in NIPS 11*, 1999.
6. T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of ICML '99*, 1999.
7. D. D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In *Proceedings of ECML*, 1998.
8. A. McCallum and K. Nigam. A comparison of event models for naive Bayes text classification. In *Learning for Text Categorization: AAAI Workshop*, 1998.
9. P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering. In *Proceedings of the SIGIR Workshop on Recommender Systems*, 2001.
10. R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of DL*, 2000.
11. K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of CIKM*, 2000.
12. K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.
13. P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom, and J. Reidl. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW*, 1998.
14. J. Schafer, J. Konstan, and J. Riedl. Electronic commerce recommender applications. *Journal of Data Mining and Knowledge Discovery*, 5:115–152, 2000.
15. K. Seymore, A. McCallum, and R. Rosenfeld. Learning hidden Markov model structure for information extraction. In *Machine Learning for Information Extraction: Papers from the AAAI Workshop*, 1999. Tech. rep. WS-99-11, AAAI Press.