Style in the Long Tail: Discovering Unique Interests with Latent Variable Models in Large Scale Social E-commerce

Diane Hu
Etsy
Brooklyn, NY
dhu@etsy.com

Rob Hall
Etsy
Brooklyn, NY
rhall@etsy.com

Josh Attenberg
Etsy
Brooklyn, NY
jattenberg@etsy.com

ABSTRACT

Purchasing decisions in many product categories are heavily influenced by the shopper's aesthetic preferences. It's insufficient to simply match a shopper with popular items from the category in question; a successful shopping experience also identifies products that match those aesthetics. The challenge of capturing shoppers' styles becomes more difficult as the size and diversity of the marketplace increases. At Etsy, an online marketplace for handmade and vintage goods with over 30 million diverse listings, the problem of capturing taste is particularly important – users come to the site specifically to find items that match their eclectic styles.

In this paper, we describe our methods and experiments for deploying two new style-based recommender systems on the Etsy site. We use Latent Dirichlet Allocation (LDA) to discover trending categories and styles on Etsy, which are then used to describe a user's "interest" profile. We also explore hashing methods to perform fast nearest neighbor search on a map-reduce framework, in order to efficiently obtain recommendations. These techniques have been implemented successfully at very large scale, substantially improving many key business metrics.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Algorithms; Experimentation; Performance

Keywords

Recommender Systems; Collaborative Filtering; Topic Modeling

1. INTRODUCTION

Describing your style can be tough. There are innumerable details that constitute one's taste that elude simple description. This difficulty in describing style becomes more severe as the set of candidate content becomes larger and increasingly diverse, a "curse of dimensionality" imparted by the inability of simple words to describe what we like about all things.[1] Despite the difficulty of transcribing taste, we generally know when we like something, and often express this in our actions. These actions, if captured and measured, can be used to model taste and power new experiences.

Etsy[2] is an online marketplace for handmade and vintage items, with over 30 million active users and 30 million active listings. This is a marketplace known for its diverse and eclectic content (e.g. Figure 1); people come in order to find those unusual items that match the peculiarities of their style. Indeed, Etsy, in its entirety could be considered part of the e-commerce long tail: in addition to wide ranging functions and styles, the handmade and vintage nature of the site means that most items for sale are unique.

For any large-scale e-commerce site, helping users find relevant content can be a challenge. Sites like Ebay or Amazon surface personalized content to their users [19] by utilizing a wide range of recommendation system technologies. Etsy faces additional challenges when building such systems- the majority of users need to find items not only by category (e.g. a purse, or a desk), but also by style (e.g., modern, cottage, industrial, or geometric). Surfacing categorically relevant items for Etsy buyers is not enough: a query like "wooden desk" will match thousands of relevant listings that buyers must browse through before they find ones that match their style (industrial, rustic, mid-century modern, cottage, antique, etc.). Thus, in Etsy's setting, where there exists an extreme heterogeneity of content, and a corresponding diversity of user behaviors, capturing taste and style is particularly difficult. However, it is precisely this difficulty in describing taste and style that make accurate models for capturing this taste critical. When typical user queries or item descriptions fail to capture intent, users rely on personalization (through implicit or explicit taste modeling) to make the huge marketplace a little smaller.

While it seems that building recommender systems and personalized content at Etsy might be an almost insurmountable challenge, Etsy benefits from an extremely engaged user base, and a suite of social features that lead to the formation of a strong user community. Users connect with each other and share content – listings or shops that they like – in a way that is familiar to any user of the social web. In ad-

KDD'14, August 24–27, 2014, New York, NY, USA.

http://dx.doi.org/10.1145/2623330.2623338 .

[1] Even if you could adequately describe your taste for clothing, would that description capture your taste for furniture or home decor?

[2] `http://www.etsy.com`

dition to creating a more engaging experience, these social features have been a key data point in the development of accurate taste models for a variety of functions.

The goal of this paper is to provide a glimpse into the process of developing and operationalizing recommender systems at web scale, in the difficult setting described above. These recommender systems personalize millions of user experiences every day, and since their inception, have substantially improved some of Etsy's core business metrics.

2. RELATED WORK

Our work touches on a couple of different areas; we give background for each in turn.

2.1 Recommendation Systems

Certainly, recommender systems are nothing new, with the first papers on collaborative filtering appearing in the 1990s [18]. In the subsequent years, motivated by obvious commercial applications, developing experiences that enable shoppers to find what they're looking for more efficiently [19], recommender systems and personalization technology have advanced tremendously. In addition to commerce applications, recommender systems appear in a variety of other settings, for instance, recommending news articles to web surfers [5]. Netflix is a well-known consumer of personalization technology; the Netflix prize has led to great innovation in the recommender system community [13].

Of particular relevance to the work presented here is Gupta et al.'s description of Twitter's "Who to Follow" System [9]. Here, the authors describe the implementation details of a large-scale recommender system, focused on recommending social connections. As with our work here, the authors prescribe a single-server modeling approach in order to reduce system complexity and improve development speed.

The range of techniques available when building recommender systems is vast, too broad to cover here. For a good overview of common techniques, we urge the curious reader to read the survey of Adomavicius and Tuzhilin [1]. Also of note is the work of Koren, Volinsky and others describing the approaches that won the Netflix prize [13, 11].

2.2 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) [4] is an unsupervised, probabilistic, generative model that aims to find a low dimensional description that can summarize the contents of large document collections. LDA works by positing that a summarization of a text document can be achieved by looking at the set of words used. Since words carry very strong semantic information, documents that contain similar content will most likely use a similar set of words. As such, mining an entire corpus of text documents can surface sets of words that frequently co-occur within documents, and be intuitively interpreted as topics. Each discovered topic is characterized by its own particular distribution over words, and each document is characterized as a random mixture of topics indicating the proportion of time the document spends on each topic. This random mixture of topics is essentially a summary of the document: It not only expresses the semantic content of a document in a concise manner, but also gives us a principled approach for describing documents quantitatively.

Beyond document modeling, LDA has also been adapted to numerous settings, including image segmentation [15], part-of-speech tagging [8], and automatic harmonic analysis in music [10], just to name a few. Not surprisingly, LDA has also been used in recommendation tasks, though in substantially different ways than us, to the best of our knowledge. Marlin [16] uses an LDA-like Dirichlet Multinomial model to predict user-item ratings for movie datasets. More recently, Wang et. al [21] proposes a hybrid content and collaborative-filtering based system that recommends scholarly papers and utilizes LDA for the content-based component.

2.3 Art and Style Classification

Since our work revolves around identifying aesthetic styles, we also look at literature for style-based recommendation and search tasks. We find that the majority of this kind of work relies solely on image processing features and visual cues, without any leverage from social influences or networks. Di et. al [7] seeks to match similar styles of clothing with a combination of crowd-sourced tags and bag-of-words image features. Zujovic et. al [22] uses Gabor-like image features as well as color features to try to classify digital pictures of paintings by artistic genre. Arora et. al [2] provides a survey of different methods for fine-art painting style, comparing the performance of discriminative versus generative models on various semantic-level and low-level image features. We note that our work for identifying visual style is substantially different from all of these approaches given that we use no image cues, and solely rely on observing social-network based user behavior.

3. IDENTIFYING USER INTERESTS

Etsy differs from many other e-commerce sites not only by the nature of the items which are for sale, but by the emphasis on social interaction amongst our users. Because this social component played such a pivotal role in the success of our taste models, it makes sense to start with a detailed background covering how users interact with the Etsy site, and how we gather implicit feedback from their activities.

3.1 Social E-Commerce, a Description of Etsy

On Etsy, there are three important entities:

- **User:** Anyone registered on Etsy, including sellers
- **Seller:** Etsy user who own a shop
- **Shop:** A collection of items sold by the same seller. Each shop has its own online storefront.
- **Listing:** Products/items listed in a shop, each with its unique listing id.

To give an idea of scale, we currently have approximately 1 million active sellers/shops, 30 million active listings, and 30 million active members.

Importantly, unlike many other e-commerce sites, users come to Etsy not only to purchase things – users often come to Etsy just to browse the site, with no specific intention of purchasing anything. During a recent user study, someone described browsing Etsy as "flipping through a magazine." Users will also connect with other users of similar interests, and share their discoveries though an activity feed, similar to those seen in other popular social websites. This social component is manifested in several interactions, including:

- **Favorite listing:** a way for users to bookmark listings that they like, and share their affinity with their followers. There are many possible intentions for favoriting a listing – the user may have the intention of purchasing it later, simply do it for curation purposes, or they may favorite simply to share with others.
- **Favorite shop:** a way for users to bookmark shops that they like, similar to favoriting a listing
- **Follow other user:** a way for users to keep track of what other users are favoriting (and thus being exposed to more content on the site). This will be discussed in more detail in section 4.1.

3.2 Inferring User Interests

As discussed in section 1, Etsy users not only want to find functionally/categorical relevant products – they also want to find ones that specifically match their style. The notion of a style can be very subjective, and very difficult to be describe with words. Thus, we rely on user activity patterns to define these styles for us.

While a variety of approaches were attempted, the most successful approaches for modeling user interests were based on Latent Dirichlet Allocation (LDA), an unsupervised, probabilistic, generative model for discovering latent semantic topics in large collections of text [4]. However, instead of using LDA to model listing content text (as is done traditionally), we use LDA to find patterns in user behavior and cluster them accordingly, akin to how matrix factorization is used for collaborative filtering.

Our use of LDA is based on the premise that users with similar interests will act upon similar listings. We chose to use the social action of "favoriting" listings as a a reliable signal for user style. This is done in lieu of more traditional user intent signals, for instance "purchasing" as is commonly done in collaborative filter development. The reasons for this choice are several fold: 1) user purchases only show a small subset of items that users are actually interested in, 2) user purchases are biased toward lower-priced items, and 3) the unique nature of Etsy's marketplace means that only one user has the opportunity to purchase an item. It is possible to have many users with very similar taste to have no overlap in their purchase vectors. Note that many of the techniques discussed below do not have any specific requirement that favoriting be the source of intent. Experimenting with different information sources to provide a broader picture of user intent is a subject of ongoing research.

Our treatment of LDA is as follows: Each user is represented as a "document," and each of the user's favorite listings are treated a "word". As such, each discovered topic can be interpreted as an "interest profile" – a distribution over all products, with highly weighted products belonging to a similar category or style (or sometimes both). Our model is formalized as follows: Assume there are K topics, or interests that we would like to discover, and V total listings. Then, β is a $K \times V$ matrix, where β_K is a distribution over the fixed vocabulary of listings. A user's list of favorited listings is posited to have been produced by the following generative process:

For each user u_j,

1. Draw u_j's interest profile $\theta_j \sim Dirichlet(\alpha)$.
2. For each favorited listing that u_j has,
   (a) Draw an interest group $z_{jn} \sim Multi(\theta_j)$
   (b) Draw a listing $x_{jn} \sim Mult(\beta_{z_{jn}})$

Note that the underlying model is no different from the original LDA model, making it easy to use existing libraries and implementations of LDA into our workflow. We fit the model using a multithreaded implementation of collapsed Gibbs sampling (see e.g., [17]) from Mallet[3]. Once the LDA model is fitted to our user-listings data, we obtain the topic-listing matrix β, which describes each of the K interests as a distribution over listings. Listings that have the highest weights within each topic are most indicative of the style of that group. We also obtain a K-dimensional user profile vector, θ, which indicates the proportion of time each user spends favoriting items from each interest group. This random mixture of topics gives a concise profile of each user's taste, and gives us a principled approach for describing user interests quantitatively. Section 4 describes the more practical aspects of our system and workflow in more detail.

One of the advantages of a topic-modeling approach to collaborative filtering, as we have presented here, is that the latent factors are easily visualized. In the following figures, we present some example topics in which both categories and styles are captured. For example, Figure 1 shows topics that center around a certain kind of interest, while spanning many different categories such as apparel, jewelry, home decor, etc. Here, (A) shows a fox theme, (B) shows a cephalopod theme, and (C) shows a Legend of Zelda theme. Figure 2 shows three topics that contain listings from the same furniture category, but span different styles: (A) shows a rustic, wooden style, (B) shows a french, cottage chic style, and (C) shows a mid-century modern style. Similarly, Figure 3 shows art from six very different styles: (A) botanical/hand-lettered, (B) haunting landscape photography, (C) whimsical animal illustrations, (D) abstract paintings, (E) fairytale doll prints, and (F) abstract whimsical paintings. Visualizing these topics have been imperative to understanding the different clusters of interests that users have on Etsy.

4. GENERATING RECOMMENATIONS

The interest clusters discovered in the previous section are used not only to summarize trending styles across the Etsy marketplace, but also to describe users' interests. In the LDA model, for each user u_j, we learn an interest profile vector, θ_j, that is a distribution over the discovered topics, indicating the amount of interest u_j has in each of the K groups. In the sections below, we show how the interest profiles are used in our user and shop recommendation systems.

4.1 User Recommendations

As mentioned above, Etsy combines a social experience with more traditional e-commerce. Etsy users (buyers and sellers alike) can opt to share their activity by connecting to other users, and sharing what interests them. In turn, users often discover content that is relevant to them by seeing the activities of others. To manifest this social networking behavior, Etsy has a page called the *activity feed*. The activity feed is linked from each user's signed-in homepage, and is similar to Facebook's mini-feed: a continuous stream of rectangular story cards that describe some sort of activity

[3] `http://mallet.cs.umass.edu`

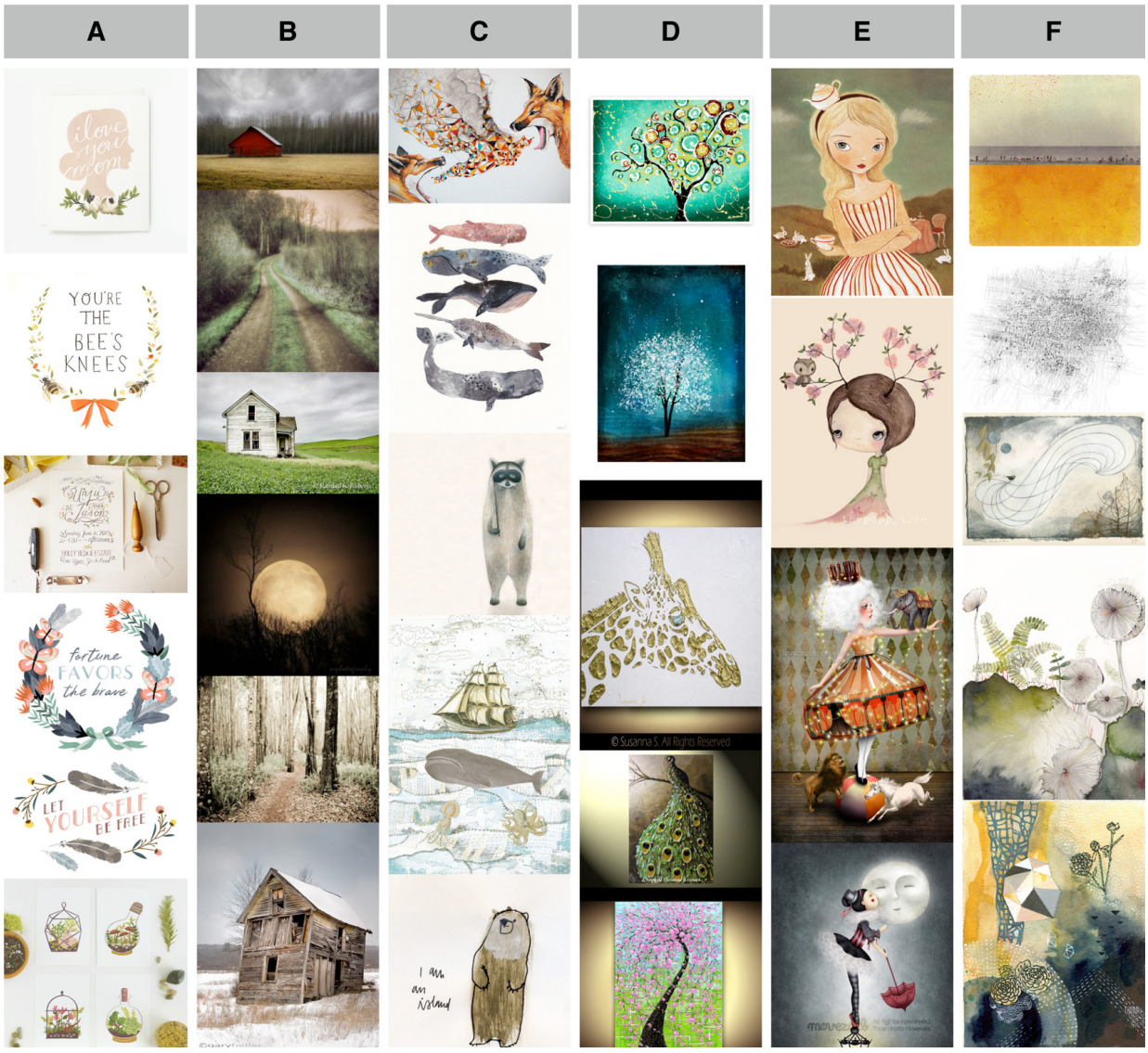

Figure 3: Six different styles of art discovered on Etsy. Each column contains high-ranking items from a topic. Note that all items come from unique sellers.

or behavior from another user that the feed owner is "following". These stories flow from the top to the bottom of the screen in the order in which the activity took place. Some examples include: "User X started following member Y" or "User X has favorited item Z", where X is a user that the feed owner follows.

More specifically, the "following" mechanism can be described as follows: Users can "follow" each other on Etsy in the same way that users can follow other users on Twitter. When user A follows user B, user B's activity (for example: products or shops that user B favorites, or even other users that user B follows) will be shown on user A's activity feed in the form of story cards (Figure 5). The idea is that a user will want to follow another user who has similar interests, so that it is more likely that user B's activity will interest user A. Before the deployment of our recommendation system, Etsy users found other users to follow by either 1) knowing the user in person, or 2) stumbling upon them while browsing the site. Thus, the purpose of the user recommendation system was to make the process of finding users with similar interests less arbitrary and more intentional.

4.1.1 *Algorithm & Implementation*

Once we obtain each user's interest profile (as described in section 3.2), we conduct a nearest neighbor search across all eligible users on Etsy (i.e. those active users who do not have private settings turned on) to find the top 100 users with the most similar θ's, which we recommend. These are users, presumably, with the most similar styles and interests.

The problem of the nearest neighbor search, of course, is that examining every pair of users to determine the distance between them (the "brute force" approach) is unfeasible due to the large number of users. Therefore, we experimented with two different hashing methods, both of which center around the idea of hashing the interest profiles θ into buckets, and then computing distances only between users that

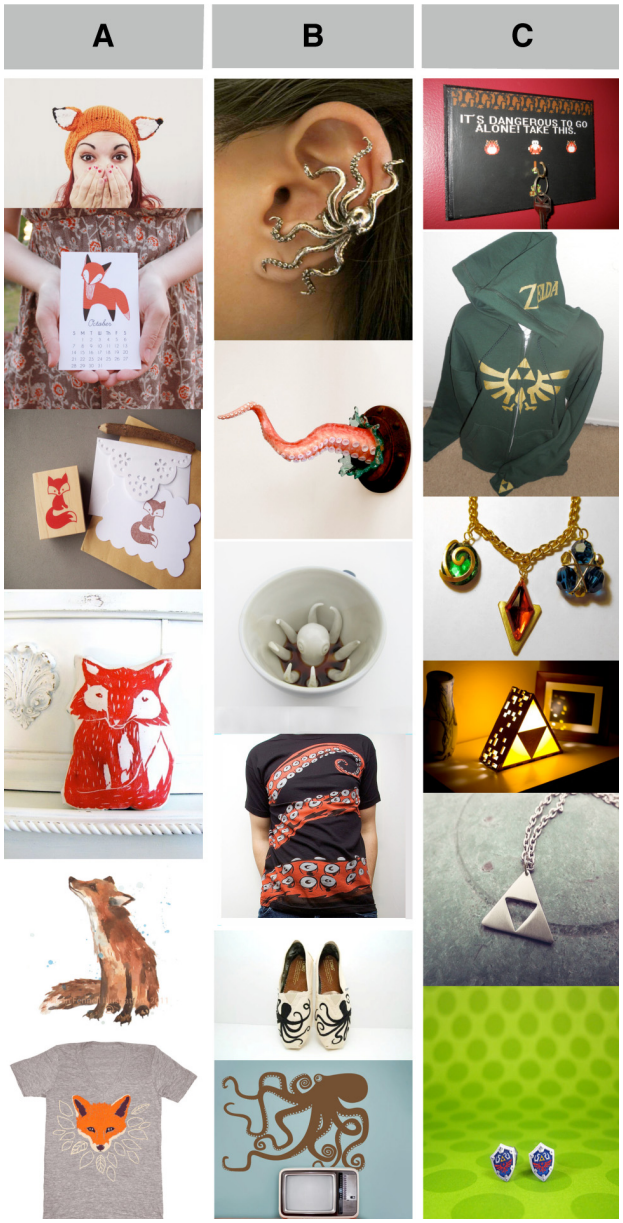

Figure 1: Interests/styles that span different categories (clothing, stationery, jewelry, etc.).

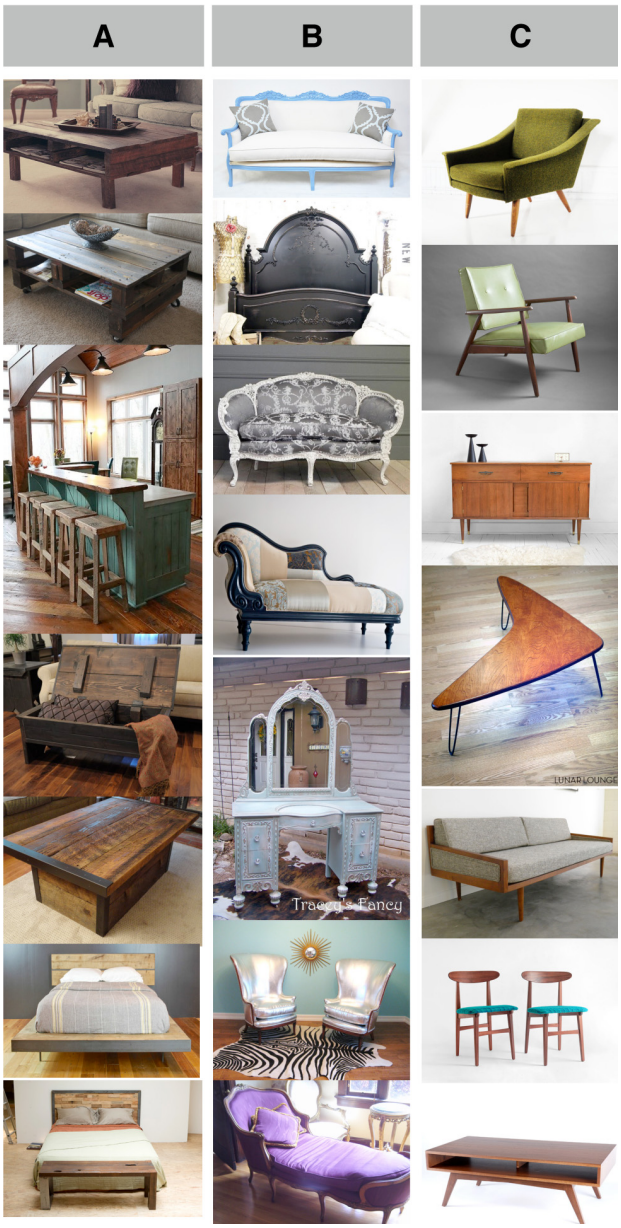

Figure 2: Three different styles within the same furniture category.

fall into the same hash bucket. This gives an approximate nearest neighbors method where the overall time complexity is dictated by the size of the largest hash bucket, which we can manage directly. A similar nearest neighbor search on the simplex was considered in [14], but focus there was placed on the setting in which all user vectors fit into one machine in memory. In contrast, we consider using mapreduce to compute the nearest neighbors in parallel so that we may scale to millions of users and high dimensional topic models, without memory or running time becoming an issue. The two hashing methods are as follows:

Locality Sensitive Hashing. Locality Sensitive Hashing (LSH) works by splitting the input space into several cones and numbering each one (see e.g., [6]). Then the elements contained in each cone are mapped to its corresponding number. We generate m random planes which pass through the origin (in d-dimensions, the normal vector to each plane is generated from a d-dimensional isotropic Gaussian), denote these normal vectors v^i then map each point $\theta \in \mathbb{R}^d$ to

$$H_{\text{LSH}}(\theta) = \sum_{i=1}^{m} 2^{i-1} \mathbf{1}\left\{\theta^T v^i \geq 0\right\}.$$

Where $\mathbf{1}\{.\}$ is a function that takes a value of 1 whenever the operand is true, and 0 otherwise. Thus each point is hashed to an m-bit integer. In our experiment we use $m = 16$. Finally, note that while this algorithm maps some nearby points to the same integer, there may be points which are close by, but separated by one of the planes. In order to mit-

Method	Number of Comparisons	20-NN Precision
Brute force	260000000	1.0
LSH-10	17711	0.37
LSH-21	38498	0.56
LSH-45	79141	0.80
TopK-5	45195	0.58
TopK-7	96630	0.68
TopK-10	197762	0.75

Table 1: Computational expense versus precision of the retrieved 20 nearest neighbors.

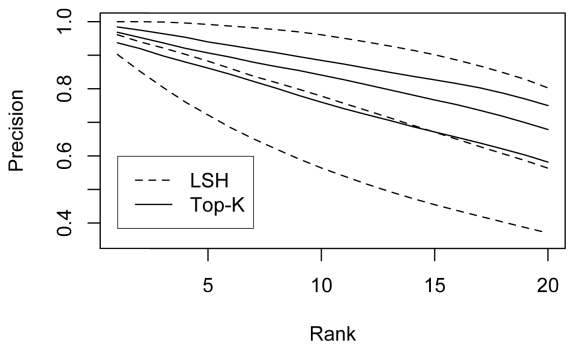

Figure 4: Precision at rank graph for experimental methods. In both cases, the higher curves correspond to higher parameter settings (i.e., the top curve is LSH-45).

igate this problem, we perform the hashing multiple times, compute nearest neighbors in each hash bucket, and then combine the results of the different hashes.

"Top-K" Hashing. We propose a second hashing method which takes advantage of the sparsity that we anticipate in the topic mixture vectors from LDA. It is plausible that the nearest neighbors to a certain user will share some subset of top-k interests. Therefore, we map each topic vector to the set of all pairs of topic indices from the top-k topics. The reason for taking pairs rather than individual topic indices is to make more specific hash buckets which will have smaller capacity. Note that in this method, each vector gets mapped into several hash buckets. We compute nearest neighbors in each bucket, then combine these across buckets, and take the nearest neighbors from among those candidates.

Comparison of Hashing Methods. We compare the performance of the above nearest-neighbor search methods on the grounds of their approximation quality and computational expense. For this experiment we used a test set of approximately 800K users. Their topic vectors were inferred from a previously trained LDA model, with 1000 topics. In order to establish the performance of the above hashing methods, we compare to the exact nearest-neighbors. Since generating these for the entire set of users is computationally expensive, we restrict the experiment to a subset of 300 Etsy users.

Both hashing methods are parameterized in a way that allows control over the number of hash bins that each user is assigned to, and we test three settings for each method. For the LSH method we use 16 bit hash keys, and 10, 21, and 45 hashes per user respectively. For the top-k hashing we set k to 5, 7, and 10 and hash the vectors according to pairs of topics in the top k (leading to 10, 21 and 45 hashes per vector). We report the number of pairwise comparisons between user vectors that are computed in Table 1, and then the precision at rank n, for the 20 nearest neighbors in Figure 4. The results demonstrate that for our LDA model, both hashing methods perform adequately, although the LHS method seems to perform slightly better than the top-k hashing method, both in terms of computational cost and the quality of the approximation.

4.1.2 Extensions

In addition to the learning done in LDA, we also experimented with two extensions that modified the interest profiles, θ, slightly to better reflect or expand upon each user's interests.

Finding Correlated Interests. Many of the discovered interest groups are quite complementary; there is no doubt that users who like a certain style would be more likely to be interested in another, related style. In order to surface these correlations, we compute the covariance matrix, Σ, from the aggregate of the user interest profile vectors, resulting in a $K \times K$ matrix that shows which interest groups are most highly correlated. We then transform each user's θ vector by projecting it onto Σ, in order to strategically "expand" the user's interests. This can be seen as a simpler way of achieving what the Correlated Topic Model [3] does, but with an easier implementation that is trivial to do in a map-reduce framework.

Special Interest Signals. Though evaluating user recommendations can often be subjective, user feedback revealed certain recommendations that should never happen. These cases include: 1) recommending users with very feminine favorites (i.e. headbands, dresses, lingerie) to a male user, 2) recommending users with mature content to those users who have never expressed interested in mature content, and 3) recommending users with very special interests (such as wedding or baby related favorited items) to users who have no use for such item. For ease of implementation, we built several linear classifiers to predict the likelihood of each user's interest level in the above three areas, and used these as additional dimensions tacked onto the original user interest profile vector. Using the "Top-K" hashing method, these additional features were used in the hashing, as to only bucket together users with a similar level of interest in these specific areas.

4.1.3 Product Design

The final recommended users are displayed as a new story type in the activity feed, as shown in figure 5 (highlighted in orange). These user recommendation stories are inserted once every 12 hours into each user's feed, and moves down the feed over time with all of the other story cards. We chose to display the user recommendations in the activity feed specifically to encourage users who currently follow few users, in hopes that more user follows will result in more interesting feed content. Our goal here is to see an increase in the overall number of users following other users (i.e. a more connected social graph), more engagement on the Etsy site, and of course, a higher conversion rate – all of which would indicate that we are properly surfacing relevant personalized content.

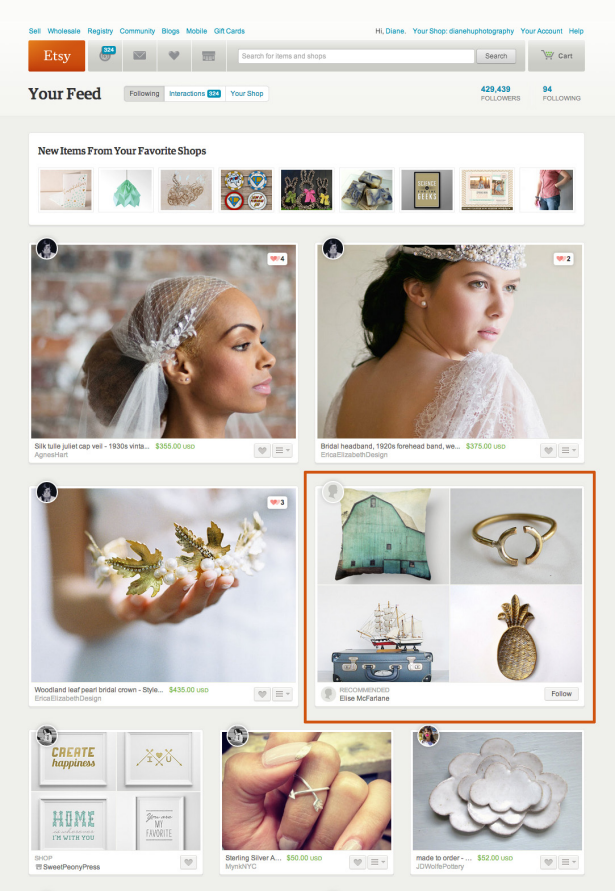

Figure 5: A section of a user's activity feed. Each story card displays an activity from a user that the feed owner is following. The story card highlighted in orange is a user recommendation story card, which highlights the recommended user's most recent favorites. A "follow" button in the bottom right-hand corner prompts the user to begin following the recommended user.

4.2 Shop Recommendations

Besides encouraging user follows, we would also like to more readily expose users to shops that are relevant to their style. Currently, users find shops by using the search engine to find specific items, through which they can find the shop that produced the item. Our goal with a shop recommendation system is to make relevant shop discovery more efficient and abundant.

4.2.1 Algorithm & Implementation

Using a very similar topic modeling approach, we developed a shop recommender system in order to encourage users to visit and favorite more shops. Here, inferring the interest profile vector, θ, is slightly different than in section 3.2. Instead of representing documents as a list of users' favorite items, we replace each favorite item with its corresponding shop id instead, and also concatenate this list with a list of shop ids of the user's favorite shops. Instead of representing listing ids as words, we use shop ids instead. The inferred topic-word matrix, β, thus becomes a distribution

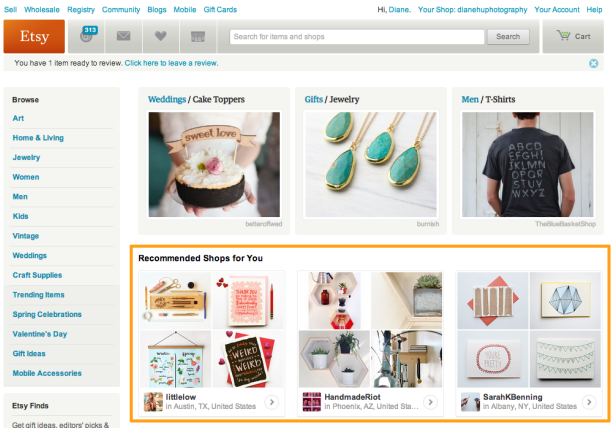

Figure 6: Shop recommendations on the homepage, highlighted in orange

over shops instead items, and the resulting interest groups are described by clusters of shops instead of clusters of items.

To obtain a single shop recommendations for user u_j, we do the following:

1. Draw an interest group $z_{jn} \sim Multi(\theta_j)$
2. Recommend shop $y_{jn} \sim Mult(\beta_{z_{jn}})$

In the spirit of collaborative filtering by matrix factorization, multiplying each user's interest profile vector by the topic-shop matrix $(\theta_j \beta)$ would have been a more traditional approach. However, this matrix multiplication is quite expensive, and it is unnecessary to have a ranked list of all possible shops, as we are only concerned with highly-ranked shops for each user. Thus, we chose this sampling approach which is more efficient and theoretically, should have comparable accuracy.

4.2.2 Product Design

The shop recommendations are currently displayed as a small module on the front page of Etsy, for signed in users (Figure 6). The three shops are swapped out every two hours so that the user will always have fresh, personalized content on their homepage.

5. SYSTEM OVERVIEW

In this section, we discuss the workflow of our deployed recommender systems. Figure 7 gives an overview of the process. First, for both recommender systems, we estimate the model on all users with at least a minimum number of favorite products. After thresholding, there are approximately 4 million such users, and the resulting data is small enough that the model could be estimated on a single machine. We used a server with 100Gb of RAM and sixteen CPU cores to run the multithreaded implementation of collapsed Gibbs sampling (see e.g., [17]) from Mallet.[4]

We have experimented with topic models that consist of anywhere between 10 to 1000 topics, and found that while this number changes the content of the granularity of the interest groups greatly, it does not affect the accuracy of the

[4]`http://mallet.cs.umass.edu`

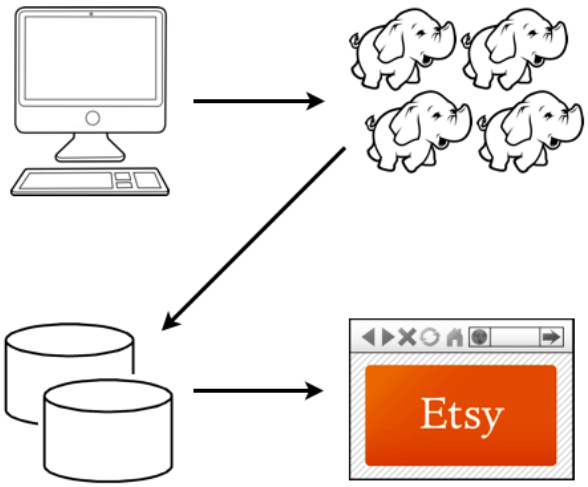

Figure 7: High-level overview of our recommendation systems: 1) Train LDA model on single machine. 2) Infer topic distributions for remaining users on Hadoop cluster; perform nearest neighbor search 3) Store recommendation datasets in Redis databases. 4) Front-end Etsy code retrieves recommendations.

user interest profiles (as evidenced from user recommendation experiments). In the end, we wound up choosing 200 and 1000 topics, depending on the specifics of the application. In both cases, the estimation of the parameters of a model takes on the order of half a day. Since the resulting model consists of a subset of all the products on Etsy, any products which are not included are treated as the nearest included product by TF-IDF distance of the text attributes of the product (the title and tags). These topic models are re-trained once a week.

Once the hyper parameters of the LDA model are learned, they are copied to our Hadoop cluster, where we infer topic distributions for the remaining users who have fewer than the necessary threshold favorite items. The inference can be done independently for each user, thus making this process completely parallelizable. For user recommendations, the top k nearest neighbors are computed on a map-reduce framework, using the techniques described in section 4.1. For shop recommendations, shops are sampled according to the algorithm in section 4.2. Both of these processes take only a couple of hours, and are computed nightly, in order to incorporate fresh user favoriting data.

The final recommendation datasets (resulting from the map-reduce jobs described above) are then stored in a shared and replicated Redis database cluster for fast retrieval by the systems used to generate Etsy's user facing web content. Earlier versions of these recommender model datasets were stored on shared mysql servers – the same servers that power crucial and complex e-commerce functionality. This proved problematic – loading the massive recommendation datasets (hundreds of millions of rows) at full speed caused a troubling performance hit for some more critical database queries. Throttling the dataset loads meant that they would take unacceptably long. Redis avoids this problem, and has the added benefit of very fast recommendation retrieval.

6. EXPERIMENTS & DISCUSSION

In the following sections, we discuss results from user studies and live experiments that were conducted for both the user and shop recommendation systems. We also compare the LDA-based model to some more traditional recommender algorithms. Before we delve into the experiments though, we give an overview of the kinds of metrics we look at when evaluating a live A/B experiment like this.

6.1 Evaluation Metrics

Most of our evaluation metrics are measured per *visit*. A *visit* is defined as a consecutive sequence of page requests that are associated with a single Google cookie. The page requests must be generated by the same traffic source, and must be viewed at most 30 minutes apart. In very high-level terms, a visit is a discrete occurrence in which a user uses the Etsy site. Visit-based metrics are engrained in Etsy's experiment-driven development culture, offering increased resilience to outliers than many other numbers we could collect. The following are some relevant evaluation metrics used in our experiments:

- **Conversion Rate:** Fraction of visits that end up in at least one purchase
- **Activity Feed Visit Rate:** Fraction of visits that view the Activity Feed, the main social touchpoint on the site
- **Pages Viewed Rate:** Number of pages viewed per visit
- **User Follow Rate:** Fraction of visits in which at least one user is followed
- **Item Favorite Rate:** Fraction of visits in which at least one item is favorited
- **Shop Favorite Rate:** Fraction of visits in which at least one shop is favorited

Our user recommendation system underwent three different phases of user testing. In the first and earliest stage, we created a one-off interface (independent of the Etsy website) that would test the quality of the user recommendations. In this user study, 135 users were presented a randomly interleaved list of 30 recommended users, from 3 different models. The first model uses the topic modeling approach described in section 4.1. The other two models use common recommendation heuristics as follows: 1) *Cosine Similarity*: Represent users as a bag of favorite products, and recommend other users with high cosine similarity, and 2) *Triadic Closure*: Recommend other users who also follow the same users. Users were then asked to rate each of the 30 user recommendations, based on images of the recommended user's most recently favorited items. The possible ratings were: negative ("I would not want to follow this user's activity"), neutral ("I wouldn't mind following this user"), or positive ("I would like to follow this user"). The results shown in Table 2 show that the LDA approach was the overwhelming favorite.

Model Name	# Neg	# Neutral	# Pos	Avg
LDA	196	278	440	2.27
Cosine Similarity	361	357	315	1.96
Triadic Closure	480	248	138	1.61

Table 2: Comparison of LDA with two popular baseline methods. The weighted average attributes 1 point to negative ratings, 2 points to neutral ratings, and 3 points to positive ratings.

Metric	Control (95%)	On (Diff) (5%)
Conversion Rate	–	**+0.32**%
Pages Viewed Rate	–	**+1.18%**
Activity Feed Visit Rate	–	**+7.51%**
User Follow Rate	–	**+13.43%**
Item Favorite Rate	–	**+2.81%**
Shop Favorite Rate	–	**+2.44%**

Table 3: Stage 2 of user recommendation experiments with live A/B user testing. Bolded numbers in the *Diff* column indicate statistical significance.

In the second stage of testing, we introduced the user recommendations in the form of story cards in the activity feed (as described in section 4.1) to live traffic on the Etsy site. Users were randomly bucketed so that 95% of users would receive an unchanged experience on the activity feed (control group), and the remaining 5% would receive the new user recommendation story cards in their activity feed. The goal of this experiment was to observe the effects of this new type of story card, and how it would impact site-wide metrics such as: number of users follows, number of favorited item, overall conversion rate, etc. The experiment ran for approximately two months, and the results were overwhelmingly positive: we saw statistically significant improvements in many site-wide metrics, as shown in Table 3. After the positive results, the on group was increased to a 50% bucketing (with similar results), and later launched to 100%.

In the third and most recent stage of testing, we experimented with four different models for obtaining user recommendations. Three of these models were variations on the original LDA model, and one model implemented a traditional matrix factorization approach. We describe each of these four variants in turn:

- **LDA-1000.** This was the original LDA-based user recommendation model used in the first two experiments described above. The number of topics was set to 1000 and only users with at least 50 favorited items were used to fit the model (resulting in 700K users for model fitting). These parameter values were somewhat arbitrarily chosen based on the perceived quality of the topics shown in section 3.

- **LDA-200.** This is similar to the original LDA-based user recommendation model, but learns only 200 topics instead. Because of the smaller number of topics, we could afford to use more data when fitting the model; here, users with at least 10 favorited items were considered (resulting in a much larger dataset of 4M users).

- **LDA-INTEREST.** This is similar to LDA-200, but with the added special interest features described in section 4.1.2 .

- **SVD.** This is the only non-LDA-based user recommendation model. Here, we factorize the user-favorites matrix using the stochastic singular value decomposition (SVD) to obtain a "latent factor" vector for each user [13]. These are scaled to have unit norm, so that the direction captures the profile of their interests (as opposed to the magnitude, which is proportional to number of favorites that a user has)

For each model, we used the locality sensitive hashing (LSH) method to retrieve the nearest neighbors to each users interest profile. In order to find the optimal parameters for the hashing (both in terms of computational efficiency and the quality of the results), we tested the method with varying parameters (number of planes, number of parallel hashings) and compared the recall against the exact 100 nearest neighbors, computed via brute force, for a small subset of the users. This allowed us to arrive at a method which is tractable to compute, and yields approximately 90% precision when compared to recommendations produced by exact nearest neighbor search.

We again launched a live A/B experiment, bucketing all users into one of these four variants, where each variant is shown to 25% of all users. This experiment ran for two weeks, with Table 4 summarizing the results. We can see that the LDA-based approach (LDA_200, in particular) is almost on equal footing as the traditional matrix factorization approach; most differences are statistically insignificant. We also note that the huge improvement we see in LDA-200 over LDA-1000 tells us that more topics doesn't necessarily mean better performance for measuring user similarity, and that fitting the model on a larger dataset possibly makes a huge difference.

We note that while the A/B experiments show little difference, one advantage that the topic modeling approach has over traditional matrix factorization methods is the ability to easily visualize latent factors. We have learned from numerous studies (e.g. [20]) that users appreciate explanations of why certain content is being recommended to them. By being able to visualize each topic as a group of items (Figures 1, 2, and 3), we can show each user exactly how their interest profile were composed.

6.2 Shop Recommendations

Our shop recommendation module sits on the Etsy homepage (described in Section 4.2), and is also being A/B tested on live traffic. The most recent experiment ran for two weeks, and consisted of three variants, each shown to 33.3% of all signed-in Etsy users. The three variants are as follows: 1) No shop recommendations (control), 2) personalized shop recommendations based on LDA, and 3) generic shop recommendations (obtained by finding the most authoritative and popular shops using the well-known Hubs and Authority (HITS) algorithm [12]). Table 5 shows the impact on relevant site-wide metrics: the shop recommendations are prompting users to engage more with the site, as all desired behavior has increased by a significant amount. As predicted, personalized recommendations trump generic recommendations across all evaluation metrics.

Metric	SVD (Control) (25%)	LDA_200 (Diff) (25%)	LDA_1000 (Diff) (25%)	LDA_INTEREST (Diff) (25%)
Conversion Rate	–	-0.14%	**-2.69%**	-1.16%
Pages Viewed Rate	–	**-0.79%**	**-1.25%**	-0.08%
User Follow Rate	–	+0.10%	**-4.46%**	**-7.05%**
Item Favorite Rate	–	-0.18%	**-0.72%**	**-2.38%**
Shop Favorite Rate	–	+0.95%	**-1.13%**	-0.99%

Table 4: Stage 3 of user recommendation experiments with live A/B user testing: experimenting with multiple recommendation models. Bolded numbers in the *Diff* columns indicate statistical significance.

Metric	Control (33%)	Personalized (33%)	Generic (33%)
Conversion Rate	–	**+1.25%**	**+1.08%**
Pages Viewed Rate	–	**+3.17%**	**+2.67%**
Item Favorite Rate	–	**+7.33%**	**+6.25%**
Shop Favorite Rate	–	**+33.18%**	**+27.92%**
Shop Visit Rate	–	**+9.70%**	**+8.52%**

Table 5: Experiments from live A/B user testing of shop recommendations on the homepage. Bolded numbers in the *Diff* columns (personalized and generic) indicate statistical significance.

7. CONCLUSION AND FUTURE WORK

In this paper, we have described the challenges of building two style-based recommendation systems to a large e-commerce site. In particular, we described an untraditional usage of LDA which leverages implicit feedback of user behavior in order to accomplish collaborative filtering. As an advantage over traditional matrix factorization-based models, our LDA-based model is capable of visualizing trending interests and styles across the Etsy marketplace, and intuitively summarizing user's style preferences. We used these models to deploy a fully functional user and shop recommendation system, both of which are currently serving live traffic. We also described methods for large-scale experimentation on the Etsy site.

In the near future, we plan on several new iterations for improving the accuracy and efficiency of our recommendations systems, including: Continuing current experiments for more conclusive results of different recommendation models; incorporating user demographic, gender, and referrer urls as priors to ease the cold-start problem; incorporating text and image features for a more informed system (and also to ease the cold-start problem); and finally, figuring out more ways to utilize inferred interest groups to create a better browsing experience.

8. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 2005.

[2] R. S. Arora and A. M. Elgammal. Towards automated classification of fine-art painting style: A comparative study. In *ICPR*, 2012.

[3] D. M. Blei and J. D. Lafferty. Correlated topic models. In *ICML*, 2006.

[4] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *JMLR*, 2003.

[5] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. In *WWW*, 2007.

[6] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *SCG*, 2004.

[7] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan. Style finder: Fine-grained clothing style detection and retrieval. In *CVPR Workshops*, 2013.

[8] T. Griffiths, M. Steyvers, D. Blei, and J. Tenenbaum. Integrating topics and syntax. In *NIPS*, 2005.

[9] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh. Wtf: The who to follow service at twitter. In *WWW*, 2013.

[10] D. J. Hu and L. K. Saul. A probabilistic topic model for unsupervised learning of musical key-profiles. In *ISMIR*, 2009.

[11] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008.

[12] J. Kleinberg. Hubs, authorities, and communities. In *ACM Computer Survey*, 1999.

[13] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

[14] K. Krstovski, D. Smith, H. M. Wallach, and A. McGregor. Efficient nearest-neighbor search in the probability simplex. In *ICTIR*, 2013.

[15] F.-F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[16] B. Marlin. Modeling user rating profiles for collaborative filtering. In *NIPS*, 2004.

[17] I. Porteous, A. Asuncion, D. Newman, P. Smyth, A. Ihler, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *ACM SIGKDD*, 2008.

[18] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW*, 1994.

[19] J. B. Schafer, J. Konstan, and J. Riedl. Recommender systems in e-commerce. In *EC*, 1999.

[20] N. Tintarev and J. Masthoff. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 2012.

[21] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *ACM SIGKDD*, 2011.

[22] J. Zujovic, L. Gandy, S. Friedman, B. Pardo, and T. Pappas. Classifying paintings by artistic genre: An analysis of features & classifiers. In *MMSP*, 2009.