

Adaptive, Personalized Diversity for Visual Discovery

Choon Hui Teo
Amazon
choonhui@amazon.com

Sriram Srinivasan
UC Santa Cruz
ssriniv9@ucsc.edu

Houssam Nassif
Amazon
houssamn@amazon.com

Mitchell Goodman
Amazon
migood@amazon.com

S. V. N. Vishwanathan
Amazon & UC Santa Cruz
vishy@amazon.com

Daniel Hill
Amazon
daniehil@amazon.com

Vijai Mohan
Amazon
vijaim@amazon.com

ABSTRACT

Search queries are appropriate when users have explicit intent, but they perform poorly when the intent is difficult to express or if the user is simply looking to be inspired. Visual browsing systems allow e-commerce platforms to address these scenarios while offering the user an engaging shopping experience. Here we explore extensions in the direction of adaptive personalization and item diversification within Stream, a new form of visual browsing and discovery by Amazon. Our system presents the user with a diverse set of interesting items while adapting to user interactions. Our solution consists of three components (1) a Bayesian regression model for scoring the relevance of items while leveraging uncertainty, (2) a submodular diversification framework that re-ranks the top scoring items based on category, and (3) personalized category preferences learned from the user's behavior. When tested on live traffic, our algorithms show a strong lift in click-through-rate and session duration.

Keywords

Machine Learning; Submodular Functions; Diversity; Personalization; Explore-Exploit; Multi-Armed Bandits

1. INTRODUCTION

The brick-and-mortar shopping experience is characterized by visual browsing where the user is able to quickly scan a large number of potential purchases. The user has a high potential to discover new items while maintaining the ability to focus attention on items of particular interest. This problem of discoverability is more challenging in e-commerce where it can be difficult to expose the entirety of an online retailer's catalog. The in-store browsing experience is not well-replicated by search engines that restrict item discovery to items relevant to an explicit search query.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. RecSys '16 September 15-19, 2016, Boston, MA, USA

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4035-9/16/09.

DOI: <http://dx.doi.org/10.1145/2959100.2959171>



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs International 4.0 License.

Therefore, an online visual browsing experience may greatly aid users in item discovery.

One effort in this direction is Amazon Stream (figure 1), a new website for fashion discovery developed by Amazon (www.amazon.com/stream). Stream enables users to easily discover popular, new, and relevant fashionable items without the need for search queries or to sieve through less relevant items. Toward this end, we have built a personalizable system that ranks and diversifies items scored by an explore-exploit algorithm. These items are presented to the user as an infinite scroll where each item can be interacted with by clicking. This paper outlines elements of our diverse and personalized visual shopping experience approach.

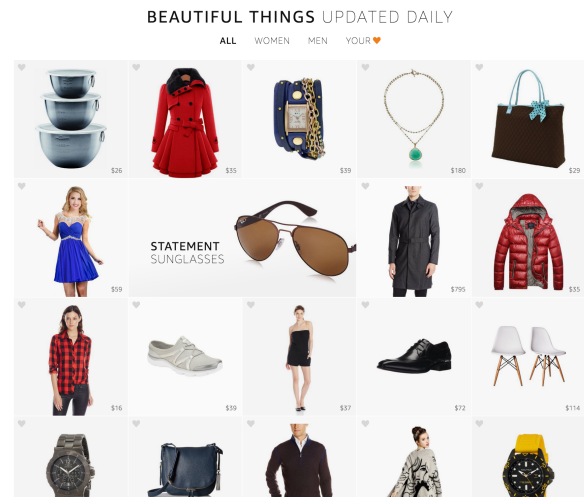


Figure 1: Screenshot of www.amazon.com/stream.

2. SCORING ITEM RELEVANCE

As the first step in generating the stream, we score each item in the Stream catalog for relevance to our users. We use click probability $P(\text{click} \mid \text{item is viewed})$ to quantify relevance. Click refers to any of the following activities: save item as favorite, visit the item's detail page, or open the modal window for the item.

We learn this probability distribution by using a Bayesian

Linear Probit regression [7] that maps item attributes onto click probabilities. Item attributes are represented by categorical variables. They include product features such as brand, color, department, price, size, as well as a unique identifier for each item. The regression model learns a set of weights where each weight represents the contribution of a single attribute value to the click probability. When evaluating an item, we say that a weight x_i is active if its corresponding attribute is present in the item. Thus, the expected click probability of an item is given by

$$P(\text{click} \mid \text{item is viewed}) = \Phi^{-1}\left(\sum_{x_i \text{ is active}} x_i\right) \quad (1)$$

where $\Phi^{-1}(\cdot)$ is the probit function. We assume model weights are generated independently from a prior Normal distribution. As we observe user actions, we update the active weights posterior distributions accordingly [7].

Our catalog consists of hundreds of thousands of items with thousands of items added or purged daily. The expected click probability may unduly favor popular items over underexposed or recently added items. This is because underexposed items will likely be associated with undertrained model weights that potentially underestimate the true click probability. This poses an exploration-exploitation dilemma: balancing exploiting highly relevant items with exploring undersampled sections of the catalog. Multi-armed bandit algorithms such as Thompson sampling [13] and Upper Confidence Bound [3] solve this problem. Here we use a Thompson sampling approach to score items as it performs well under delayed feedback [4].

We apply Thompson sampling by sampling each model weight from its posterior distribution conditioned on the training data. We then use the sampled weights to compute equation (1) for each item. Under this procedure, an item will receive a score close to its expected click probability unless its attributes are associated with highly uncertain model weights. Therefore, the items with the highest scores will be a mix of popular products with high expected values (an exploitation strategy) and underexposed products that were randomly scored high above their expected value (an exploration strategy).

Finally, we note that the customer controls the number of products viewed via scrolling. For simplicity we have omitted a model of customer browsing that predicts the number of views per session. However, such a model would allow us to estimate the expected clicks per session directly rather than the expected clicks per item viewed.

3. SUBMODULAR DIVERSIFICATION

We now have a set of items that are scored based on both their relevancy and exposure. Similar items tend to be assigned similar scores, so ranking these items by their scores creates a list where highly similar items are clustered together. However, our user experience studies indicate that a diverse item list helps users discover relevant items faster, as it exposes more variety in a fixed number of slots [11]. Similar to cases in web search [1], blog posts [5], news articles [2] and music recommendation [11], we apply diversification to alleviate information redundancy.

We formulate the selection and ranking of a diverse subset of items as a submodular optimization problem [6]. Let $\mathcal{A} := \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ denote the attribute vectors for a set

of n items. All attribute vectors are one-hot encoded in d -dimensions such that $\mathbf{a}_i \in \{0, 1\}^d$. Let \mathbf{w} denote a d -dimensional vector which encodes user preferences for each of the d item attributes. In addition, let $s(\mathbf{a})$ denote a function which maps an item to a real-valued utility score such as the click probability. A submodular objective function ρ for selecting k items from \mathcal{A} is given by

$$\rho(\mathcal{A}_k, \mathbf{w}) = \left\langle \mathbf{w}, \log \left(1 + \sum_{\mathbf{a}_i \in \mathcal{A}_k} \mathbf{a}_i \right) \right\rangle + \sum_{\mathbf{a}_i \in \mathcal{A}_k} s(\mathbf{a}_i), \quad (2)$$

where \mathcal{A}_k is a subset of \mathcal{A} of size k . We use the convention of applying the logarithm to a vector's individual components. The optimal subset is given by

$$\mathcal{A}_k^* := \underset{\mathcal{A}_k \subseteq \mathcal{A}, |\mathcal{A}_k|=k}{\operatorname{argmax}} \rho(\mathcal{A}_k, \mathbf{w}). \quad (3)$$

In equation (2), the weight w_j emphasizes the importance of the j^{th} attribute to a user. By summing over \mathbf{a}_i we obtain the number of times that attribute occurs in the collection \mathcal{A}_k . However, the utility of each attribute does not grow linearly with its count. Instead, the logarithmic term ensures that the incremental utility of the j^{th} attribute diminishes as the number of items with j attributes increases in \mathcal{A}_k . In other words, if a user loves shoes, showing only shoes does not lead to a good user experience. The scale of $s(\mathbf{a})$ can be tuned to adjust the relative importance of the item's utility.

Equation (3) is a special case of the NP-hard maximum set cover problem. Nevertheless, we can use an iterative greedy procedure to obtain a near-optimal solution [12]:

$$\mathcal{A}_0 := \emptyset \text{ and } \mathcal{A}_{i+1} := \mathcal{A}_i \cup \left\{ \underset{\mathbf{a} \in \mathcal{A} \setminus \mathcal{A}_i}{\operatorname{argmax}} \rho(\mathcal{A}_i \cup \{\mathbf{a}\}, \mathbf{w}) \right\}, \quad (4)$$

with a runtime complexity of $O(dkn)$. We further boost efficiency by using the CELF lazy evaluation algorithm [5, 9].

We also optimize the diversification process by transforming the attribute vector space into a single categorical space. This is achieved by mapping each possible attribute vector to a unique category. We restrict the size of this categorical space by coarsening our representation of attributes, such as price, and eliminating attributes that do not need to be diversified, such as average rating in customer reviews. The final set for diversification includes on the order of 100 mutually exclusive categories.

4. PERSONALIZATION

Category weights \mathbf{w} in equation (2) control the trade-off between item utility and category popularity for selecting and ranking items. Unlike methods such as [14] and [2], we do not optimize these weights directly. This allows us to exploit user behavioral data that was not subject to diversification. In the following sections, we describe the procedure for learning global and personalized category weights.

4.1 Learning Adaptive Global Weights

Though items in the stream are displayed in a grid layout (figure 1), we assume that users scan them linearly from left to right and top to bottom. Hence, we can leverage existing work on click modeling with linear positional bias correction.

To learn the weights \mathbf{w} in equation (2), we compared logistic regression, clicks over expected clicks [15], and click-thru-rate (CTR) with additive smoothing. Each model was

tested on several weeks of historical click logs. Even though the CTR approach does not include corrections for position bias, we found that the three methods gave similar results. We therefore opted to use CTR with additive smoothing as it is simple to implement yet provides additional controls to handle update cycle fluctuations.

Let c_j and v_j represent the number of clicks and views, respectively, for items from the j^{th} category. Under the CTR with additive smoothing model, the weight for the j^{th} category w_j is estimated as:

$$w_j = \frac{c_j + \alpha}{v_j + \alpha + \beta}, \quad (5)$$

where α and β are priors that enforce minimum category weights. α and β can be tuned based on traffic volume to avoid dramatic fluctuations in update intervals. In fact, one can interpret w_j as distributed according to a Binomial distribution whose parameter p_j is drawn from a Beta distribution with parameters α and β estimated from data.

4.2 Learning User Specific Weights

The adaptively tuned category weights described in the previous section were derived from the aggregate clicks of all users. They capture the general popularity of the categories but not the interests of particular users. To personalize the presentation of items and categories, we need to first build a user model. Unlike some personalized applications where user feedback is immediate and irreversible, our model must handle delayed feedback and react quickly and efficiently to *evolving* preferences where a positive indicator of interest may no longer be predictive at a later date.

4.2.1 User Modeling

User modeling methods such as matrix factorization [8] and hierarchical Bayesian models [10] are not directly applicable in our case because these methods work best with a massive collection of data from all users, and are thus less sensitive to changing patterns in a small fraction of training records. Instead, we model the u^{th} user's level of interest in the d categories as a random vector \mathbf{w}_u drawn from a Dirichlet distribution parameterized by α_0 . The distribution of a user's clicks on each category is denoted as \mathbf{c}_u and is modeled as a random variable drawn from a Multinomial distribution parameterized by the interest vector and the user's total number of clicks m_u :

$$\mathbf{w}_u \sim \text{Dirichlet}(\alpha_0), \quad (6)$$

$$\mathbf{c}_u \sim \text{Multinomial}(\mathbf{w}_u, m_u). \quad (7)$$

The posterior distribution of the user interest vector \mathbf{w}_u is given proportionally by

$$P(\mathbf{w}_u | \mathbf{c}_u, \alpha_0) \propto P(\mathbf{c}_u | \mathbf{w}_u)P(\mathbf{w}_u | \alpha_0). \quad (8)$$

As Dirichlet is conjugate prior for the Multinomial distribution, this posterior is also Dirichlet with mean $\hat{\mathbf{w}}_u$:

$$\hat{\mathbf{w}}_u = (\mathbf{c}_u + \alpha_0) \| \mathbf{c}_u + \alpha_0 \|_1^{-1}. \quad (9)$$

We estimate the personalized weight vector \mathbf{w}_u from the user's past clicks $\mathbf{c}_u = (c_{u1}, \dots, c_{ud})$ where $c_{uj} \in \mathbb{N}$ is the number of clicks for category j . Dirichlet prior parameter vector α_0 can be chosen to match pre-specified business rules or to highlight certain category preferences.

In addition to being simple to implement, this Multinomial-Dirichlet user model allows straightforward incorporation of

different types of user signals by adding them as counts to \mathbf{c}_u . Similarly, evolving preferences are handled by deducting older signals from \mathbf{c}_u . Finally, the personalized diversification can be achieved by plugging the estimated user interest vector $\hat{\mathbf{w}}_u$ into equation (2).

4.2.2 User Click Signal Diffusion

User interests are correlated between different categories. For example, a male interested in high end fashion is likely interested in both suits and expensive watches. This concept is especially useful when users have concentrated clicks on only a few categories, which would otherwise tend to inhibit diversified recommendations.

To ensure a user is exposed to related categories, we diffuse our Dirichlet updates between categories. In addition to a user's click incrementing the item's specific category count, the update is diffused across *related* categories. Let \mathbf{M} be a d -by- d matrix where the entry M_{ij} is the ratio of the number of users who clicked items from both categories i and j to the number of users who clicked items from category j . To alleviate noise, we use only the top categories each user has actively interacted with. The *smoothed* interest vector $\tilde{\mathbf{w}}_u$ of the user interest vector $\hat{\mathbf{w}}_u$ of equation (9) becomes:

$$\tilde{\mathbf{w}}_u = \mathbf{M} \hat{\mathbf{w}}_u \| \mathbf{M} \hat{\mathbf{w}}_u \|_1^{-1}. \quad (10)$$

There are several common approaches for quantifying the pairwise association between two categories, such as point-wise mutual information or Pearson's correlation coefficients. We use the asymmetric conditional probability that "users who like category i also like category j " because we have observed such user behavior patterns in our data. Moreover, the leading eigenvector of the co-interest probability matrix \mathbf{M} can be seen as a global interest vector, towards which equation (10) smooths the user interest vector $\hat{\mathbf{w}}_u$.

5. EXPERIMENTS

We conducted several online experiments to test the effectiveness of the methods presented above. We evaluated the treatment impact on engagement by calculating session duration, number of items viewed, and total click-through-rate. Each experiment lasted at least one week and involved at least 100,000 users per group. Some statistics are reported for population subsets of at least 10,000 users per group. Statistical significance was determined via Welch's t-test. Table 1 shows experimental results. Columns should not be compared directly to each other, as each experiment's control is different. Columns confer incremental improvements on top of the previous treatment.

5.1 Submodular Diversification

We first compared our submodular diversifier of Section 3 to a simpler approach where categorical weights and item scores are considered separately. The model of section 2 assigns a score to each item. Within each category, we rank the items in descending order. We calculate a multinomial distribution over item categories using global click propensities. For each slot in the stream, we first sample a category from the multinomial, and then fill it with that category's highest scoring item.

Experimental results (Table 1, column 2) show that, within the same session duration, the amount of items viewed by the users subjected to submodular diversification decreased

	Submodular diversifier	Adaptive weights	Personalized weights
Duration	0.05%	5.39%	1.10%
Views	-1.32%	1.08%	-4.95%
CTR	9.82%	8.29%	12.58%

Table 1: Experimental results evaluated using per-session engagement metrics: duration, aggregated view count, and aggregated click-through-rate. Column 2 shows incremental changes between a *submodular vs multinomial* diversifier. Column 3 shows incremental changes of *adaptive global category weights vs static manual weights*. Column 4 shows incremental changes for *personalized category weights vs global weights*. Highlighted results are statistically significant at the 0.05 level.

by 1.32% while CTR increased by 9.82%. The submodular diversifier produces a more diverse stream as categories are penalized more aggressively for being shown repeatedly. These results are consistent with our expectation of users being more interested in diversified streams.

5.2 Adaptive Category Weights

In a second experiment, we tested the impact of learning adaptive category weights from global click propensities as described in Section 4.1. We compared a submodular diversifier that uses the adaptive weights, versus one that uses a set of weights chosen manually by Amazon fashion experts. Experimental results (Table 1, column 3) show that adaptive weights greatly out-performed manual weights. CTR increased by 8.29% and session duration increased by 5.39%. This indicates that users are highly sensitive to the categorical mix of their experience.

5.3 Personalization

In our final experiment, we exposed users to category weights that were adapted to their own click behavior, as described in Section 4.2. These weights were calculated for users whose history included at least 10 clicks, so this experiment only applied to the most active users. In the control group, users were exposed to streams built using non-personalized weights. Experimental results (Table 1, column 4) show that personalized weights yielded a 12.58% increase in CTR with almost no impact on duration. Interestingly, the amount of items that were viewed in the stream decreased by 4.95%. This indicates that users with personalized diversity weights had a more efficient experience where they interacted with more items while scrolling through less.

6. CONCLUSION

We present Amazon Stream, a visual browsing system that emphasizes adaptive and personalized diversity in user experience. Such methods can be applied to any online system where the user seeks to discover content in the absence of an explicit search query. Our experimental results show the tangible incremental impact on key engagement metrics of submodular diversification, adaptive global weights, and personalization.

Acknowledgments.

The authors thank Charles Elkan, Matthias Seeger, and the anonymous reviewers for their helpful comments.

References

- [1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Jeong. Diversifying search results. In *Proceedings of the Second ACM*

International Conference on Web Search and Data Mining, pages 5–14. ACM, 2009.

- [2] A. Ahmed, C. H. Teo, S. V. N. Vishwanathan, and A. Smola. Fair and balanced: Learning to present news stories. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 333–342. ACM, 2012.
- [3] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research*, 3:397–422, 2003.
- [4] O. Chapelle and L. Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.
- [5] K. El-Arini, G. Veda, D. Shahaf, and C. Guestrin. Turning down the noise in the blogosphere. In *Proceedings of SIGKDD international conference on Knowledge discovery and data mining*, pages 289–298, 2009.
- [6] S. Fujishige. *Submodular functions and optimization*, volume 58. Elsevier, 2005.
- [7] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft’s bing search engine. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 13–20, 2010.
- [8] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [9] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.
- [10] Y. Low, D. Agarwal, and A. J. Smola. Multiple domain user personalization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 123–131. ACM, 2011.
- [11] H. Nassif, K. O. Cansizlar, M. Goodman, and S. V. N. Vishwanathan. Diversifying music recommendations. In *Proceedings of Machine Learning for Music Discovery Workshop at 33rd International Conference on Machine Learning (ICML)*, 2016.
- [12] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions i. *Mathematical Programming*, 14(1):265–294, 1978.
- [13] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pages 285–294, 1933.
- [14] Y. Yue and C. Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 2483–2491, 2011.
- [15] W. V. Zhang and R. Jones. Comparing click logs and editorial labels for training query rewriting. In *WWW’07 Workshop on Query Log Analysis: Social And Technological Challenges*, 2007.