Diverse Personalization with Determinantal Point Process Eigenmixtures

James Foulds
University of California, Irvine*
jfoulds@ics.uci.edu

Dilan Görür
Yahoo Labs
dilan@yahoo-inc.com

1 Introduction

Personalization has become an important part of recommendation systems for online products including news, search, media and advertising. Real world recommender systems need to also take into account the diversity and serendipity of the set of recommended items so as to not overwhelm the user with too similar items and to discover user interests that were previously unknown to the system (Szpektor et al., 2013). Recently developed methods that include diversity as a part of the model have achieved promising results (Yue & Joachims, 2008; Kulesza & Taskar; Raman et al., 2012).

Here, we consider a model-based approach for diverse and personalized recommendations using determinantal point processes (DPPs) (Hough et al., 2006; Kulesza & Taskar, 2012). DPPs are probability models for sets of points which tend to repel each other, i.e. are diverse. They can conveniently model both diversity and quality. However, in most personalization applications we would also like to be able to tune this diversity/quality trade-off, as well as other aspects such as the user's long-term versus short-term interests and personalized versus popular/trending items, and DPPs do not provide a simple mechanism to accomplish this.

We propose to address this by introducing models for blending the behaviors of multiple DPPs. We consider several models, in increasing order of sophistication. For the final method, we first offer an alternative perspective on the DPP. Using this new perspective, we generalize the DPP in a natural way by exploiting the eigenstructure of the DPP kernels. The resulting model allows for fine-grained control of the extent to which the behavior of the model mimics the most important properties of each component DPP. We show how to learn the mixture parameters of the models, and demonstrate the utility of the proposed methods on a news recommendation task.

2 Determinantal Point Processes

A *point process* $\mathcal{P}$ on a discrete set $\mathcal{D} = \{x_1, \ldots, x_M\}$ is a probability distribution on the power set $2^{\mathcal{D}}$ of $\mathcal{D}$. Such a process is called a *determinantal* point process (Macchi, 1975; Hough et al., 2006) if there is a positive semidefinite matrix $\mathbf{K}$ with eigenvalues less than or equal to one such that for every subset of a set $S \sim \mathcal{P}$ the inclusion probability of A is given by $\mathcal{P}(A \subseteq S) = \det(\mathbf{K_A})$. Here, $\mathbf{K}_A \triangleq [K_{ab}]_{x_a, x_b \in S}$ is the matrix $\mathbf{K}$ restricted to the rows and columns indexed by elements of A. The matrix $\mathbf{K}$ is referred to as the *marginal kernel* of the DPP as it defines the marginal probability of each subset being included in the drawn set S. We will focus on an alternative $\mathbf{L}$-*ensemble* construction of DPPs (Macchi, 1975; Borodin & Rains, 2005) for the ease of representation it provides in modeling. An $\mathbf{L}$-ensemble defines a point process via the atomic probabilities of each set

$$\mathcal{P}_L(S) = \frac{\det(\mathbf{L}_S)}{\det(\mathbf{L} + \mathbf{I})}\,, \tag{1}$$

*This work was performed while James Foulds was under internship at Yahoo Labs.

Algorithm 1 Sampling from a DPP $\mathcal{P}_L$

1: **Given:** the L-ensemble kernel $\mathbf{L}$
2: Compute eigenvector/value pairs $\{(\mathbf{v}_n, \lambda_n)\}$ of $\mathbf{L}$
3: Sample a subset V of eigenvectors, selecting $\mathbf{v}_n$ with probability $\frac{\lambda_n}{\lambda_n+1}$
4: Construct an elementary DPP $\mathcal{P}^V$ with marginal kernel $\mathbf{K}^V = \sum_{\mathbf{v}\in V} \mathbf{v}\mathbf{v}^\intercal$
5: Sample items from $\mathcal{P}^V$

where $\mathbf{L}$ is a positive semidefinite matrix, and $\det(\mathbf{L}_\emptyset) \triangleq 1$. $\mathbf{L}$-ensembles are determinantal point processes with marginal kernel $\mathbf{K} = \mathbf{L}(\mathbf{L}+\mathbf{I})^{-1}$. Since $\mathbf{L}$ is positive semidefinite, it can be factorized as a Gram matrix, $\mathbf{L} = \mathbf{X}^\intercal\mathbf{X}$. In other words, if we represent the items $x_a \in \mathcal{D}$ by a column vector $\mathbf{X}_a$, L_{ab} is the dot product of $\mathbf{X}_a$ and $\mathbf{X}_b$. Writing each feature representation as a product of a scalar $q(x_a)$ and a unit vector $\phi(x_a)$, we can write the entries of the kernel as

$$L_{ab} = q(x_a)\phi(x_a)^\intercal\phi(x_b)q(x_b),$$

where $q(x_a)$ measures the *quality* of item a and $\phi(x_a)^\intercal\phi(x_b)$ measures the similarity between a and b. The probability of a set S, $\mathcal{P}_L(S) \propto \det(\mathbf{L}_S) = \left(\prod_{x_a\in S} q(x_a)^2\right)\det(\phi(S)^\intercal\phi(S))$ is equal to the squared volume of the parallelotope spanned by the column vectors $\mathbf{X}_a$, $x_a \in S$. The volume of this parallelotope, and thus the probability of the set S according to the DPP, increases as the vectors become closer to being orthogonal, i.e. more dissimilar. The volume also increases as we increase the length of the vectors $q(x_a)$, encoding a preference for items with high quality or relevance.

2.1 Sampling from DPPs

A DPP whose marginal kernel has eigenvalues in $\{0, 1\}$ is called an *elementary* DPP. Given a set of orthonormal vectors V, we denote by $\mathcal{P}^V$ an *elementary* DPP with marginal kernel $\mathbf{K}^V = \sum_{\mathbf{v}\in V} \mathbf{v}_n\mathbf{v}_n^\intercal$. Determinantal point processes can be expressed as mixtures of elementary DPPs (Hough et al., 2006). For a DPP $\mathcal{P}_L$ with L-ensemble kernel matrix $\mathbf{L}$ eigendecomposed as $\mathbf{L} = \sum_{n=1}^N \lambda_n\mathbf{v}_n\mathbf{v}_n^\intercal$ we have

$$\mathcal{P}_L(S) = \frac{1}{\det(\mathbf{L}+\mathbf{I})} \sum_{J\subseteq\{1,\ldots,N\}} \mathcal{P}^{V_J}(S) \prod_{n\in J} \lambda_n \,, \tag{2}$$

where $V_J = \{\mathbf{v}_j : j \in J\}$ is the set of eigenvectors of $\mathbf{L}$ specified by the index set J.

There is an efficient algorithm for sampling from elementary DPPs, which can be exploited to draw from arbitrary DPPs by first drawing an elementary DPP from the mixture (Algorithm 1) (Hough et al., 2006; Tao, 2009).

2.2 A novel interpretation of elementary DPPs

Rewriting the eigendecomposition of an elementary DPP's marginal kernel as $\mathbf{K}^V = \mathbf{B}^\intercal\mathbf{B}$, where the rows of $\mathbf{B}$ are the eigenvectors in V, $\mathcal{P}^\mathcal{V}$ selects exactly $|V|$ items proportionally to the squared volume spanned by the corresponding columns of $\mathbf{B}$ (c.f. Kulesza & Taskar (2012)). We can therefore reinterpret $\mathcal{P}^\mathcal{V}$ as a k-DPP (a variant of a DPP which samples exactly k items (Kulesza & Taskar, 2011)) with the implicit feature matrix $\mathbf{B}$ giving rise to an L-ensemble kernel $\mathbf{L}^V = \mathbf{B}^\intercal\mathbf{B} = \mathbf{K}^V$. When sampling from an arbitrary L-ensemble DPP, we can interpret the eigenvectors chosen in step 3 of Algorithm 1 as features in a new latent space in which the items are re-represented. The sampling procedure finally selects $|V|$ items proportionally to their squared volume in the new latent space.

3 Methods for Blending DPPs

We consider different approaches to extending DPPs that will lead to more flexible models with a capacity to fine tune their behavior by combining the properties of several DPPs. Suppose we have DPPs with L-ensemble kernels $\{\mathbf{L}_{(i)}\}$ and mixture weights $\{\alpha_i\}, \sum\alpha_i = 1$ specifying the extent

Algorithm 2 Simple methods for combining DPPs

(a) Sampling from a mixture of DPPs

1: **Given:** a mixture of DPPs $\mathcal{P}^{\boldsymbol{\alpha}}_{L_{1:M}}$
2: Select a DPP with probability $\boldsymbol{\alpha}$
3: Sample from the selected DPP using Algorithm 1

(b) Sampling from a DPP with a mixture of kernels

1: **Given:** a collection of kernels with corresponding mixing weights
2: compute the convex combination of the kernels
3: Sample from the DPP with the resulting kernel using Algorithm 1

to which the model should respect each component DPP. A straightforward way to combine DPPs, previously proposed by Kulesza & Taskar (2011), is to use a mixture model over them (Algorithm 2a). A single draw from this model will in general result in a set containing high-probability items from only one of the DPPs.

Alternatively, we can create a new kernel by taking a convex combination of the component L-ensemble kernels, $\mathbf{L}^{\boldsymbol{\alpha}} = \sum_{i=1}^{M} \alpha_i \mathbf{L}_{(i)}$, and drawing from the resulting DPP (Algorithm 2b). Convex combinations of positive semidefinite matrices are positive semidefinite, so the resulting matrix is a valid kernel. This approach is reminiscent of kernel learning methods, e.g. for support vector machines (Lanckriet et al., 2004). Note that as each of the component L-ensemble kernels is a Gram matrix, each entry of these matrices corresponds to a dot product between the feature vectors for a pair of items. This approach interpolates their dot products, with the extent of the interpolation depending on the weight vector α. In terms of the feature space representation, this can be thought of as defining a new extended feature space by appending the different feature spaces of each kernel, scaled with the square root of the corresponding mixture weight.

3.1 Determinantal Point Process Eigenmixtures

While the mixture of kernels interpolates between the properties of multiple DPPs, some applications may require more control over the distribution of items by specifying the proportion of samples coming from each DPP. To enable this, instead of concatenating the (rescaled) features of the items we can concatenate the *latent* features implied by the eigendecomposition of the kernel matrices. The model, which we call the *DPP eigenmixture*, follows the mixture model representation of DPPs (Equation 2), except that the latent features are chosen from the eigenvectors of all M component DPPs. The number of eigenvectors s_m per component m can be specified in advance by the modeler, or chosen from a multinomial distribution with parameter vector α. The full model is as follows:

$$\mathbf{s} \sim \text{Mult}(\alpha, k) \tag{3}$$

$$\mathcal{P}_L(\mathcal{Y}; \mathbf{L}_{(1)}, \ldots, \mathbf{L}_{(M)}, s_1, \ldots, s_m) \propto \sum_{J = j^{(1)} \cup j^{(2)} \cup \ldots \cup j^{(M)}, j^{(m)} \in \binom{V^{(m)}}{s_m}} \mathcal{P}^k_L(\mathcal{Y}; \mathbf{L}^{V_J}) \prod_{n^{(m)} \in J} \lambda_{n^{(m)}} ,$$

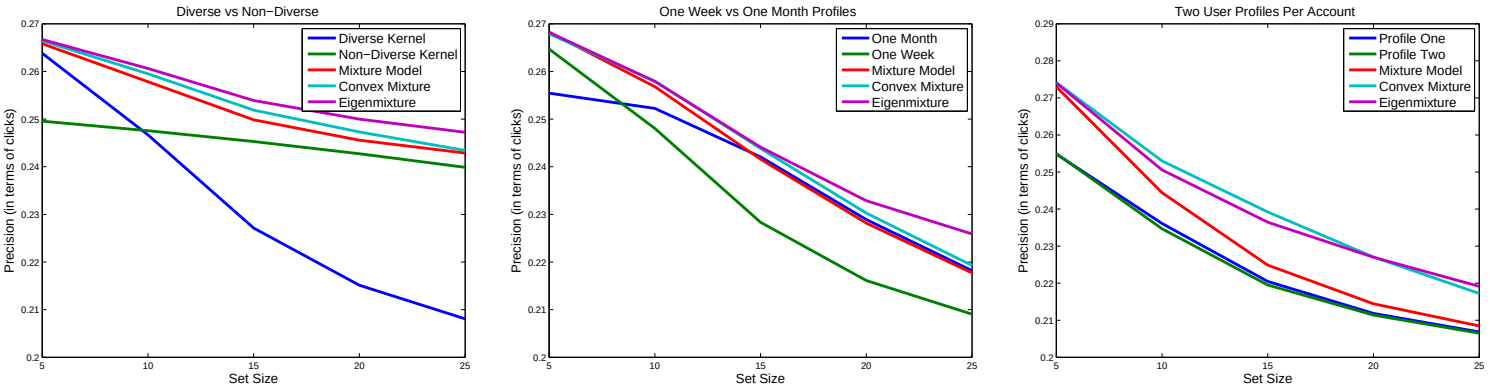
where $\binom{S}{k}$ is the set of all k-combinations (i.e. sets of size k) of the set S and $\mathcal{P}^k_L(\mathcal{Y}; \mathbf{L})$ is a k-DPP, i.e. a DPP restricted to drawing exactly k items. The selected eigenvectors are not in general eigenvectors of the new kernel, so we will refer to them instead as "*eigenfeatures*." The number of eigenfeatures s_m from each component DPP m determines the extent to which the behavior of the model follows the behavior of that DPP. Specifically, in expectation the associated s_m-dimensional subspace of eigenfeatures represents each item such that the s_m-dimensional squared volume spanned by any set of s_m items is proportional to the volume spanned by those items' feature vectors, as specified by the original Gram matrix $\mathbf{L}^{(m)}$. We can sample from the model using Algorithm 3.

While a DPP is a mixture over elementary DPPs, DPP eigenmixtures are in general mixtures over non-elementary DPPs. This means that we must resort to the full DPP sampler in the final sampling step before we can exploit the properties of elementary DPPs to sample more efficiently. It also means that the new kernel $\mathbf{L}^V$ or its dual must be eigendecomposed in order to sample from it. Fortunately, the number of features in the new space is equal to the number of items to draw k, which is very small in most applications. This means that the dual DPP sampler of Kulesza & Taskar (2010) may be used instead, which in this case only requires a decomposition of a $k \times k$ matrix.

Algorithm 3 Sampling from DPP eigenmixtures

Given: a collection of kernels with corresponding mixture weights
Sample s_m eigenvectors $V^{(m)}$ from each kernel m, $\mathbf{s} \sim \text{Mult}(\alpha, k)$, $\mathcal{P}(V^{(m)}) \propto \prod_{v \in V^{(m)}} \lambda_v$
Concatenate all selected eigenvectors V to form a new latent representation $\bar{\mathbf{B}} = [\mathbf{v}_1^\intercal; \ldots; \mathbf{v}_{|V|}^\intercal]$
Compute the resulting kernel matrix $\mathbf{L}^V = \bar{\mathbf{B}}^\intercal \bar{\mathbf{B}}$
Sample from the DPP with the new kernel
$\mathbf{Y} \sim \mathcal{P}_L^k(\mathcal{Y}; \mathbf{L}^V)$ using Algorithm 1

Figure 1: Personalized news recommendation, blending diversity vs preference (left), short-term and long-term interests (middle) and multiple users' interests (right).



4 Experimental Results / Conclusions

A particularly compelling application for the models is the recommendation of news articles on personalized news aggregation websites. We investigated the performance of the methods on the news recommendation task for the front page of the popular media website Yahoo. In the experiments, we considered a set of approximately 600 users who each clicked on between 3–20 articles per day between the beginning of April and the first week of May, 2013. The sets articles for which each user clicked on, and the articles for which they skipped over without clicking, were collected during this time period and used as candidate items for the models to choose from. The feature representations of the articles and the user profiles were extracted using proprietary algorithms belonging to Yahoo, based on data from April.

For each user, the mixtures were trained to optimize the metric used for evaluation, namely expected precision on the number of clicked articles out of the generated sets of items, using a grid search on parameter space. At each location in the grid search, and at test time, the expected precision was estimated by simulating 1000 sets of items. The models were evaluated on the data from the first week of May, using a 5-fold cross-validation scheme to select items for training the mixtures.

We explored the use of the models for varying several dimensions of the recommendation, namely diversity versus estimated user preference, short-term versus long-term interests (with user profiles trained on one week and one month of data) and multiple users per account (simulated by dividing the active features into two separate user profiles). Precision results are shown in Figure 1. Interestingly, for small set sizes the diverse kernel was better than the non-diverse kernel, while the reverse was true for larger sets. Similarly, short-term profiles were better than long-term profiles for small set sizes only. The blended methods were able to handle this gracefully, dominating the single DPPs throughout the curve. The convex mixture and eigenmixture performed better than the mixture model, with the eigenmixture typically being the best with larger set sizes.

In ongoing work, we are exploring the use of the gradient-free optimization algorithm SPSA (Spall, 1987, 1998) to learn the blended models with many components . We envision that the techniques proposed here will be useful for a wider variety of applications that can be cast as subset selection such as extractive summarization.

References

Borodin, Alexei and Rains, Eric M. Eynard–Mehta theorem, Schur process, and their Pfaffian analogs. *Journal of Statistical Physics*, 121(3-4):291–317, 2005.

Hough, J Ben, Krishnapur, Manjunath, Peres, Yuval, and Virág, Bálint. Determinantal processes and independence. *Probability Surveys*, 3:206–229, 2006.

Kulesza, Alex and Taskar, Ben. Learning determinantal point processes. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, pp. 419–427.

Kulesza, Alex and Taskar, Ben. Structured determinantal point processes. In *Advances in Neural Information Processing Systems*, pp. 1171–1179, 2010.

Kulesza, Alex and Taskar, Ben. k-dpps: Fixed-size determinantal point processes. In *Proceedings of the International Conference on Machine Learning*, pp. 1193–1200, 2011.

Kulesza, Alex and Taskar, Ben. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2-3):123–286, 2012.

Lanckriet, Gert RG, Cristianini, Nello, Bartlett, Peter, Ghaoui, Laurent El, and Jordan, Michael I. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.

Macchi, Odile. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, pp. 83–122, 1975.

Raman, K., Shivaswamy, P., and Joachims, T. Online learning to diversify from implicit feedback. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 705–713, 2012.

Spall, James C. A stochastic approximation technique for generating maximum likelihood parameter estimates. In *American Control Conference, 1987*, pp. 1161–1167. IEEE, 1987.

Spall, James C. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, 19(4):482–492, 1998.

Szpektor, Idan, Maarek, Yoelle, and Pelleg, Dan. When relevance is not enough: promoting diversity and freshness in personalized question recommendation. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 1249–1260, 2013.

Tao, Terence. Determinantal processes. http://terrytao.wordpress.com/2009/08/23/determinantal-processes/, August 2009.

Yue, Yisong and Joachims, T. Predicting diverse subsets using structural SVMs. In *International Conference on Machine Learning (ICML)*, pp. 271–278, 2008.