Learning Determinantal Point Processes by Corrective Negative Sampling

Zelda Mariet
zelda@csail.mit.edu
Massachusetts Institute of Technology

Mike Gartrell
m.gatrell@criteo.com
Criteo AI Lab

Suvrit Sra
suvrit@mit.edu
Massachusetts Institute of Technology

Abstract

Determinantal Point Processes (DPPs) have attracted significant interest from the machine-learning community due to their ability to elegantly and tractably model the delicate balance between quality and diversity of sets. DPPs are commonly learned from data using maximum likelihood estimation (MLE). While fitting observed sets well, MLE for DPPs may also assign high likelihoods to unobserved sets that are far from the true generative distribution of the data. To address this issue, which reduces the quality of the learned model, we introduce a novel optimization problem, *Contrastive Estimation (CE)*, which encodes information about "negative" samples into the basic learning model. CE is grounded in the successful use of negative information in machine-vision and language modeling. Depending on the chosen negative distribution (which may be static or evolve during optimization), CE assumes two different forms, which we analyze theoretically and experimentally. We evaluate our new model on real-world datasets; on a challenging dataset, CE learning delivers a considerable improvement in predictive performance over a DPP learned without using contrastive information.

1 Introduction

Careful selection of items from a large collection underlies many machine learning applications. Notable examples include recommender systems, information retrieval and automatic summarization methods, among others. Typically, the selected set of items must fulfill a variety of application specific requirements—e.g., when recommending items to a user, the *quality* of each selected item is important. This quality must be, however, balanced by the *diversity* of the selected items to avoid redundancy within recommendations.

But balancing quality with diversity is challenging: as the collection size grows, the number of its subsets grows exponentially. A model that offers an elegant, tractable way to achieve this balance is a Determinantal Point Process (DPP). Concretely, a DPP models a distribution over subsets of a ground set $\mathcal{Y}$ that is parametrized by a semi-definite matrix $\boldsymbol{L} \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$, such that for any $A \subseteq \mathcal{Y}$,

$$\Pr(A) \propto \det(\boldsymbol{L}_A), \tag{1}$$

where $\boldsymbol{L}_A = [\boldsymbol{L}_{ij}]_{i,j\in A}$ is the submatrix of $\boldsymbol{L}$ indexed by A. Informally, $\det(\boldsymbol{L}_A)$ represents the volume associated with subset A, the diagonal entry L_{ii} represents the importance of item i, while entry $L_{ij} = L_{ji}$ encodes similarity between items i and j. Since the normalization constant of (1) is simply $\sum_{A\subseteq\mathcal{Y}} \det(\boldsymbol{L}_A) = \det(\boldsymbol{L}+\boldsymbol{I})$, we have $\Pr(A) = \det(\boldsymbol{L}_A)/\det(\boldsymbol{L}+\boldsymbol{I})$, which suggests why DPPs may be tractable despite their exponentially large sample space.

The key object defining a DPP is its kernel matrix $\boldsymbol{L}$. This matrix may be fixed *a priori* using domain knowledge [Borodin, 2009], or as is more common in machine learning applications, learned from observations using maximum likelihood estimation (MLE) [Gillenwater et al., 2014, Mariet and Sra, 2015]. However, while fitting observed subsets well, MLE for DPPs may also assign high likelihoods to unobserved subsets far from the underlying generative distribution [Chao et al., 2015], since MLE causes the DPP model to maximize the determinantal volume of observed subsets without explicitly minimizing the volume of unobserved subsets. Therefore, the volume of unobserved subsets may be larger than expected, and MLE-based DPP models may thus have modes corresponding to subsets that are close in likelihood, yet differ in how close they are to the true data distribution. Such confusable modes reduce the quality of the learned model, hurting predictions, as shown in Figure 1.

Such concerns when learning generative models over huge sample spaces are not limited to the area of subset-selection: applications in image and text generation have been the driving force in developing techniques for generating high-quality samples. Among their innovations, a particularly successful technique uses generated samples as "negative

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

samples" to train a discriminator, which in turn encourages generation of more realistic samples; this is the key idea behind the Generative Adversarial Nets (GANs) introduced in [Goodfellow et al., 2014].

These observations motivate us to investigate the use of DPP-generated samples with added perturbations as *negatives*, which we then incorporate into the learning task to improve the modeling power of DPPs. Intuitively, negative samples are those subsets that are far from the true data distribution, but to which the DPP erroneously assigns high probability. As there is no closed form way to generate such idealized negatives, we approximate them via an external "negative distribution".

More precisely, we introduce a novel DPP learning problem that incorporates samples from a negative distribution into traditional MLE. Our approach reduces the confusable mode issue associated with MLE for DPPs by augmenting MLE with a term that explicitly minimizes the volume of unobserved subsets that are far from the true data distribution (negative samples). While the focus of our work is on generating the negative distribution *jointly* with $\boldsymbol{L}$, we also investigate outside sources of negative information. Ultimately, our formulation leads to an optimization problem harder than the original DPP learning problem; we show that even approximate solutions greatly improve the performance of the DPP model when evaluated on concrete tasks, such as identifying the best item to add to a subset of chosen objects (*basket-completion*) and discriminating between held-out test data and randomly generated subsets.

Contributions. To our knowledge, this work is the first theoretical or empirical investigation of augmenting the DPP learning problem with negative information.

- Our first main contribution is the Contrastive Estimation (CE) model, which incorporates negative information through *inferred negatives* into the learning task.
- We introduce static and dynamic models for CE and discuss the theoretical and practical trade-offs of such choices. Static models leverage information that does not evolve over time, whereas dynamic models draw samples from a negative distribution that depends on the current model's parameters; dynamic CE posits an optimization problem worthy of independent study.
- We show how to learn CE models efficiently, and furthermore show that the complexity of conditioning a DPP on a chosen sample can be brought from $\mathcal{O}(|\mathcal{Y}|^2)$ to essentially $\mathcal{O}(|\mathcal{Y}|)$. This helps dynamic CE and removes a major bottleneck in computing next-item predictions for a set.

Using findings obtained from extensive experiments conducted on small datasets, we show on a large dataset that CE learning significantly improves the modeling power of DPPs: CE learning improves DPP performance for next-item basket completion, as well as DPP discriminative power, as evaluated by the model's ability to distinguish held-out test data from randomly generated subsets.

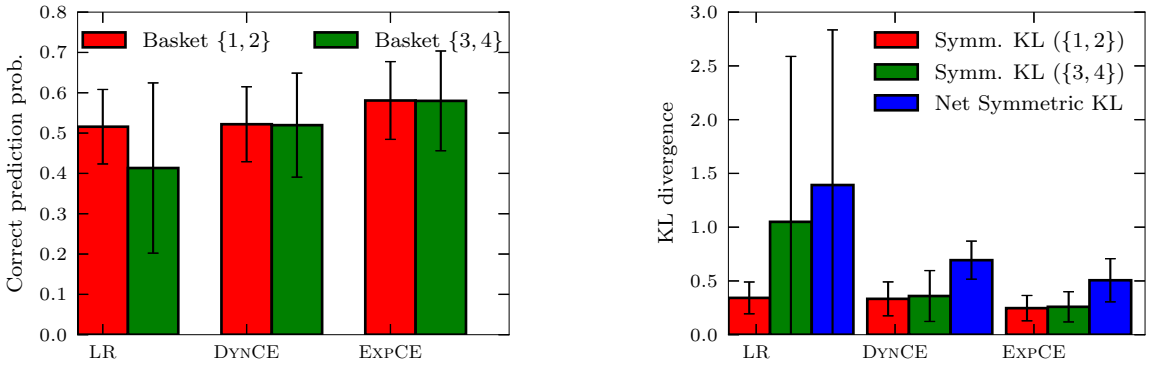
We present a review of related work in Section 2. In Section 3, we introduce Contrastive Estimation and its dynamic and static variants. We discuss how the CE problem can be optimized efficiently in Section 4, as well as how DPP conditioning for basket-completion predictions can be performed with improved complexity. In Section 5, we show that CE learning leads to remarkable empirical improvements of DPP performance metrics.

2 Background and related work

First introduced to model fermion behavior by Macchi [1975], DPPs have gained popularity due to their elegant balancing of quality and subset diversity. DPPs are studied both for their theoretical properties [Kulesza and Taskar, 2012, Borodin, 2009, Affandi et al., 2014, Kulesza, 2013, Gillenwater, 2014, Decreusefond et al., 2015, Lavancier et al., 2015], which include fast sampling [Rebeschini and Karbasi, 2015, Li et al., 2016, Anari et al., 2016], and for their machine learning applications: object retrieval [Affandi et al., 2014], summarization [Lin and Bilmes, 2012, Chao et al., 2015], sensor placement [Krause et al., 2008], recommender systems [Gartrell et al., 2016], neural network compression [Mariet and Sra, 2016a], and minibatch selection [Zhang et al., 2017].

Gillenwater et al. [2014] study DPP kernel learning via EM, while Mariet and Sra [2015] present a fixed-point method. DPP kernel learning has leveraged Kronecker [Mariet and Sra, 2016b] and low-rank [Dupuy and Bach, 2016, Gartrell et al., 2017] structures. Learning guarantees using DPP graph properties are studied in [Urschel et al., 2017]. Aside from Tschiatschek et al. [2016], Djolonga et al. [2016], who learn a Facility LocatIon Diversity (FLID) distribution (as well as more complex FLIC and FLDC models) by contrasting it with a "negative" product distribution, little attention has been given to using negative samples to learn richer subset-selection models.

Nonetheless, leveraging negative information is a widely used in other applications. In object detection, negative mining corrects for the skewed simple-to-difficult negative distribution by training the model on its false positives [Sung, 1996, Canévet and Fleuret, 2014, Shrivastava et al., 2016]. In language modeling, Noise Contrastive Estimation (NCE) [Gutmann and Hyvärinen, 2012], which tasks the model with distinguishing positive samples from generated negatives, was first applied in [Mnih and Teh, 2012] and has been instrumental in Word2Vec [Mikolov et al., 2013]. Since then, variants using adaptive noise [Chen et al., 2017] have been introduced. NCE is also the method used



(a) Probabilities of making the right prediction

(b) KL divergences

Figure 1: Results for experiments on a synthetic toy dataset. This toy dataset was generated by replicating the baskets $\{1, 2\}$ and $\{3, 4\}$ 1000 times each. We randomly select 80% of this dataset for training, and 20% for test. We train each model to convergence, and then compute the next-item predictive probabilities for each unique pair, along with the symmetric KL divergence (over areas of shared support) between the predictive and empirical next-item distributions. Net symmetric KL divergence is computed by adding the symmetric KL divergences for each of the two unique baskets. Experiments were run 10 times, with $|\mathcal{A}^+|/|\mathcal{A}^-|$ set to the optimal value for each model; α is set to its optimal LR value. See Section 3 for details on the DYN and EXP negative sampling models. The LR (low-rank DPP) model assigns relatively high probabilities to modes that represent incorrect predictions: as Maximum Likelihood Estimation learning teaches the model to maximize the volume of observed subsets without explicitly minimizing the volume of unobserved subsets, volumes of unobserved subsets may be larger than expected. The DYN and EXP methods we introduce in Section 3 reduce this confusable mode issue, resulting in predictive distributions that are closer to the true distribution and much smaller variances.

by Tschiatschek et al. [2016] for subset-selection.

An alternate approach to negative samples within submodular language models was introduced as Contrastive Estimation in [Smith and Eisner, 2005a,b]. Negative sampling is also used in GANs [Goodfellow et al., 2014], where a generator network competes with a discriminative network which distinguishes between positives and generated negatives. An adversarial approach to Contrastive Estimation has been recently introduced in [Bose et al., 2018], where ideas from GANs for discrete data are used to implement an adversarial negative sampler that augments a conventional negative sampler.

3 Learning DPPs with negative samples

Motivated by the similarities between DPP learning and crucial structured prediction problems in other ML fields, we introduce an optimization problem that leverages negative information. We refer to this problem as Contrastive Estimation (CE) due to its ties to a notion discussed in [Smith and Eisner, 2005a].

3.1 Contrastive Estimation

In conventional DPP learning, we seek to maximize determinantal volumes of sets drawn from the true distribution μ (that we wish to model), by solving the following MLE problem, where samples in the training set $\mathcal{A}^+$ are assumed to be drawn i.i.d.:

$$\text{Find } \boldsymbol{L} \in \operatorname*{argmax}_{\boldsymbol{L}\succeq 0} \phi_{\text{MLE}}(\boldsymbol{L}) \triangleq \frac{1}{|\mathcal{A}^+|} \sum_{A\in\mathcal{A}^+} \log\det(\boldsymbol{L}_A) - \log\det(\boldsymbol{L}+\boldsymbol{I}). \tag{2}$$

We augment problem (2) to incorporate additional information from a *negative* distribution ν, which we wish to have the DPP distribution move away from. The ensuing optimization problem is the main focus of our paper.

Definition 1 (Contrastive Estimation)**.** Given a training set of positive samples $\mathcal{A}^+$ on which ϕ_{MLE} is defined and a negative distribution ν over $2^{\mathcal{Y}}$, we call *Contrastive Estimation* the problem

$$\text{Find } \boldsymbol{L} \in \operatorname*{argmax}_{\boldsymbol{L}\succeq 0} \phi_{\text{CE}}(\boldsymbol{L}) \triangleq \phi_{\text{MLE}}(\boldsymbol{L}) - \mathbb{E}_{A\sim\nu}[\log \mathcal{P}_{\boldsymbol{L}}(A)], \tag{3}$$

where we write $\mathcal{P}_{\boldsymbol{L}}(A) \equiv \det(\boldsymbol{L}_A)/\det(\boldsymbol{L}+\boldsymbol{I})$.

The expectation can be approximated by drawing a set of samples $\mathcal{A}^-$ from ν: ϕ_{CE} then becomes[1]

$$\phi_{\text{CE}}(\boldsymbol{L}) = \frac{1}{|\mathcal{A}^+|} \sum_{A\in\mathcal{A}^+} \log \mathcal{P}_{\boldsymbol{L}}(A) - \frac{1}{|\mathcal{A}^-|} \sum_{A\in\mathcal{A}^-} \log \mathcal{P}_{\boldsymbol{L}}(A) \tag{4}$$

[1] With a slight abuse of notation, we continue writing ϕ_{CE} despite the sample approximation to $\mathbb{E}_{A\sim\nu}[\cdot]$.

If $|\mathcal{A}^-| = 0$, the CE objective (3) reduces to ϕ_{MLE}. Conversely, ϕ_{MLE} can be viewed as a sample-based approximation of the value $\mathbb{E}_{A\sim\mu}[\log \boldsymbol{P}_{\boldsymbol{L}}(A)]$, where μ is the true distribution generating the samples in $\mathcal{A}^+$. Interestingly, another reformulation of (3) suggests an even broader class of DPP kernel learning: indeed, let y_A be $\frac{1}{|\mathcal{A}^+|}$ (resp. $-\frac{1}{|\mathcal{A}^-|}$) for $A \in \mathcal{A}^+$ (resp. $\mathcal{A}^-$), and define

$$\mathcal{A} = \{(y_A, A) : A \in \mathcal{A}^+\} \cup \{(y_A, A) : A \in \mathcal{A}^-\},$$

where the y_A should be viewed as belonging in $\{-1, 1\}$ with an additional normalization coefficient. Then, we can rewrite equation (4) in the following form

$$\phi_{\text{CE}}(\boldsymbol{L}) = \sum_{(y_A, A)\in\mathcal{A}} y_A \Big[\log\det \boldsymbol{L}_A - \det(\boldsymbol{L} + \boldsymbol{I}) \Big]. \quad (5)$$

Formulation (5) suggests the use of a broader scope of continuous labels y_A; we do not cover this variation in the present work, but note that (5) permits the use of *weighted* samples for learning.

Remark 1. Compared to the traditional Noise Contrastive Estimation (NCE) approach, which requires full knowledge of the negative distribution, CE does not suffer any such limitation: we only require an estimate of $\mathbb{E}_{\nu}[\log \mathcal{P}_{\boldsymbol{L}}(A)]$.

Remark 2. Eq. (3) can be made to go to $+\infty$ with pathological negative samples (i.e. $\mathcal{P}_{\boldsymbol{L}}(A^-) = 0$); hence, choosing the negative distribution is a crucial concern for CE. In practice, we do not observe this pathological behavior (cf. Section 5).

Remark 3. CE is a non-convex optimization problem, and thus admits the same guarantees as DPP MLE learning when learned using Stochastic Gradient Ascent with decreasing step sizes; however, the convergence rate will depend on the choice of ν.

Indeed, to fully specify the CE problem one must first choose the negative distribution ν, or equivalently, choose a procedure to generate negative samples to obtain (4). We consider below two classes of distributions ν with considerably different ramifications: dynamic and static negatives; their analysis is the focus of the next two sections.

3.2 Dynamic negatives

In most applications leveraging negative information (e.g., negative mining, GANs), negative samples evolve over time based on the state of the learned model. We call any ν that depends on the state of the model a *dynamic negative distribution*: at iteration k of the learning procedure with kernel estimate $\boldsymbol{L}_k$, we use a ν parametrized by $\boldsymbol{L}_k$.

More specifically, we focus on the setting where negative samples themselves are generated by the current DPP, with the goal of reducing overfitting. Given a positive sample A^+, we generate a negative A^- by replacing $i \in A^+$ with j that yields a high probability $\mathcal{P}_{\boldsymbol{L}_k}(A^+\backslash\{i\} \cup \{j\})$ (Alg. 1). We generate the samples probabilistically rather than via mode maximization so that a sample A^+ can lead to different A^- negatives when we generate more negatives than positives.

Algorithm 1 Generate dynamic negative

Input: Positive sample A^+, current kernel $\boldsymbol{L}_k$
Sample $i \in A^+$ prop. to its empirical probability in $\mathcal{A}^+$
$A^- := A^+\backslash\{i\}$
Sample j w.p. proportional to $\mathcal{P}_{\boldsymbol{L}_k}(A^- \cup \{j\})$
$A^- \leftarrow A^- \cup \{j\}$
return A^-

As ν evolves along with $\boldsymbol{L}_k$, the second term of ϕ_{CE} acts as a moving target that must be continuously estimated during the learning procedure. For this reason, we choose to optimize ϕ_{CE} by a two-step procedure described in Alg. 2, similarly to an alternating maximization approach such as EM.

Algorithm 2 Optimizing dynamic CE

Input: Positive samples $\mathcal{A}^+$, initial kernel $\boldsymbol{L}_0$, maxIter.
$k \leftarrow 1$
while $k++ <$ maxIter **and** not converged **do**
  $\mathcal{A}^- \leftarrow$ GENERATEDYNAMICNEGATIVES$(\boldsymbol{L}_k, \mathcal{A}^+)$
  $\boldsymbol{L}_{k+1} \leftarrow$ OPTIMIZECE$(\boldsymbol{L}_k, \mathcal{A}^+, \mathcal{A}^-)$
end while
return $\boldsymbol{L}_k$

Note that this approach bears strong similarities with GANs, in which both the generator and discriminator evolve during training (dynamic negatives also appear in a discussion by Goodfellow [2014] as a theoretical tool to analyze the difference between NCE and GANs).

Once the generated negative A^- has been used in an iteration of the optimization of ϕ_{CE}, it is less likely to be sampled again.[2] Crucially, such dynamic negatives also avoid the problem alluded to in Remark 2, since by construction they have a non-zero probability under $\mathcal{P}_{\boldsymbol{L}_k}$ at iteration k.

3.3 Static negatives

Conversely, we can simplify the optimization problem by considering a *static* negative distribution: ν does not depend on the current kernel estimate. A considerable theoretical advantage of static negatives lies in the simpler optimization problem: given a static negative distribution ν, the optimization objective ϕ_{CE} does not evolve during training, and

[2] If A^- happens to be a false negative (i.e. appears in $\mathcal{A}^+$), A^- will be comparatively sampled more frequently as a positive, and so will contribute on average as a positive sample. Additional precautions such as the ones mentioned in [Bose et al., 2018] can also be leveraged if necessary.

is amenable to a simple invocation of stochastic gradient descent [Bottou, 1998].

Theorem 1. *Let ν be a static distribution over $2^{\mathcal{Y}}$ and let $k > 0$ be such that $k \geq \max\{|S|\colon S \in \mathcal{A}^+ \cup \operatorname{supp}(\nu)\}$. Let K be a bounded subspace of all $|\mathcal{Y}|\times|\mathcal{Y}|$ positive semi-definite matrices of rank k. Projected stochastic gradient ascent applied to the CE objective with negative distribution ν and space K with step sizes η_i such that $\sum \eta_i = \infty$, $\sum \eta_i^2 < \infty$ will converge to a critical point.*

Note, however, that such distributions may suffer from the fundamental theoretical issue in Rem. 2, and hence careful attention must be paid to ensure that the learning algorithm does not converge to a spurious optimum that assigns a probability $\boldsymbol{P_L}(A) = 0$ to $A \in \mathcal{A}^-$. In practice, we observed that the local nature of stochastic gradient ascent iterations was sufficient to avoid such behavior.

Let us now discuss two classical choices for fixed ν.

Product negatives. A common choice of negative distribution in other machine learning areas is the *product distribution*, which is the standard "noise" distribution used in NCE. It is defined by

$$\nu(A) = \prod_{i \in A} \hat{p}(i) \prod_{i \notin A} (1 - \hat{p}(i)) \tag{6}$$

where $\hat{p}(i)$ is the empirical probability of $\{i\}$ in $\mathcal{A}^+$. Although [Mikolov et al., 2013] reports better results by raising the $\hat{p}$ to the power $\frac{3}{4}$, we did not observe any improvements when using exponentiated power distributions; for this reason, by *product negatives*, we always indicate the baseline distribution (6).

The product distribution is in practice a mismatch for DPPs, as it lacks the negative association property of DPPs which enables them to model the repulsive interactions between similar items[3].

Explicit negatives. Alternatively, we may have prior knowledge of a class of subsets that our model should *not* generate. For example, we might know that items i and j are negatively correlated and hence unlikely to co-occur. We may also learn via user feedback that some generated subsets are inaccurate. We refer to negatives obtained using such outside information as *explicit negatives*.

A fundamental advantage of explicit negatives is that they allow us to incorporate prior knowledge and user feedback as part of the learning algorithm. The ability to incorporate such information, to our knowledge, is in itself a novel contribution to DPP learning.

[3] DPPs belong to the family of *Strongly Rayleigh* measures, which have been shown to verify a broad range of negatively associated properties; we refer the interested reader to the fascinating work [Pemantle, 2000, Borcea et al., 2009a,b, Borcea and Brändén, 2009a,b, 2010].

Although such knowledge may be costly and/or only available at rare intervals, a form of continuous learning that would regularly update the state of our prior knowledge (and hence ν) would bring the explicit negative distribution into the realm of dynamic distributions, as described by Alg. 2.

4 Efficient learning and prediction

We now describe how the Contrastive Estimation problem for DPPs can be optimized efficiently. In order to efficiently generate dynamic negatives, which rely on DPP conditioning, we additionally generalize the dual transformation leveraged in [Osogami et al., 2018] to speed up basket-completion tasks with DPPs. This speed-up impacts the broader use of DPPs, outside of CE learning.

4.1 Optimizing ϕ_{CE}

We propose to optimize the CE problem by exploiting a low-rank factorization of the kernel, writing $\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^\top$, where $\boldsymbol{V} \in \mathbb{R}^{M \times K}$ and $K \leq M$ is the rank of the kernel, which is fixed *a priori*.

This factorization ensures that the estimated kernel remains positive semi-definite, and enables us to leverage the low-rank computations derived in [Gartrell et al., 2017] and refined in [Osogami et al., 2018]. Given the similar forms of the MLE and CE objectives, we use the traditional stochastic gradient ascent algorithm introduced by [Gartrell et al., 2017] to optimize (3). In the case of dynamic negatives, we re-generate $\mathcal{A}^-$ after each gradient step; less frequent updates are also possible if the negative generation algorithm is very costly.

We furthermore augment ϕ_{CE} with a regularization term $R(\boldsymbol{V})$, defined as

$$R(\boldsymbol{V}) = \alpha \sum_{i=1}^{M} \frac{1}{\mu_i} \|\boldsymbol{v}_i\|_2^2,$$

where μ_i counts the occurrences of i in the training set, $\boldsymbol{v}_i$ is the corresponding row vector of $\boldsymbol{V}$ and $\alpha > 0$ is a tunable hyperparameter. Note that this is the same regularization as introduced in [Gartrell et al., 2017]. This regularization tempers the strength of $\|\boldsymbol{v}_i\|_2$, a term interpretable as to the popularity of item i [Kulesza and Taskar, 2012, Gillenwater, 2014], based on its *empirical* popularity μ_i. Experimentally, we observe that adding $R(\boldsymbol{V})$ has a strong impact on the predictive quality of our model.

The reader may wonder if other approaches to DPP learning are also applicable to the CE problem.

Remark 4. Gradient ascent algorithms require that the estimate $\boldsymbol{L}$ be projected onto the space of positive semi-definite matrices; however, doing so can lead to almost-diagonal kernels [Gillenwater et al., 2014] that cannot model negative interactions. Riemannian gradient ascent methods were

considered, but deemed too computationally demanding by [Mariet and Sra, 2015]. Furthermore, the update rule for the fixed-point approach in [Mariet and Sra, 2015] does not admit a closed form solution for CE, rendering it impractical (App. A).

The low-rank formulation allows us to apply CE (as well as NCE, as discussed in Section 5) to learn large datasets such as the Belgian retail supermarket dataset (described in Section 5) without prohibitive learning runtimes. We show below that by leveraging the idea described in [Osogami et al., 2018], the low-rank formulation can also lead to additional speed ups during prediction.

4.2 Efficient conditioning for predictions

Dynamic negatives rely upon conditioning a DPP on a chosen sample A (see Alg. 1: $\mathcal{P}_{\boldsymbol{L}_k}(A^- \cup \{j\})$ can be efficiently computed for all j by a preprocessing step that conditions $\boldsymbol{L}_k$ on set A^-). For this reason, we now describe how low-rank DPP conditioning can be significantly sped up.

In [Gartrell et al., 2017], conditioning has a cost of $\mathcal{O}(K|\bar{A}|^2+|A|^3)$, where $\bar{A} = \mathcal{Y} - A$. Since $|\mathcal{Y}| \gg |A|$ for many datasets, this represents a significant bottleneck for conditioning and computing next-item predictions for a set. We show here that this complexity can be brought down significantly.

Proposition 1. *Given $A \subseteq \{1, \ldots, M\}$ and a* DPP *of rank K parametrized by $\boldsymbol{V}$, where $\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^\top$, we can derive the conditional marginal probabilities in the* DPP *parametrization $\boldsymbol{L}^A$ in only $\mathcal{O}(K^3 + |A|^3+K^2|A|^2+|\bar{A}|K^2)$ time.*

Proof. Let $\boldsymbol{V}$ be the low-rank parametrization of the DPP kernel ($\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^\top$) and $A \subseteq \mathcal{Y}$. As in Gillenwater [2014], we first compute the dual kernel $\boldsymbol{C} = \boldsymbol{B}^\top\boldsymbol{B}$, where $\boldsymbol{B} = \boldsymbol{V}^\top$. We then compute

$$\boldsymbol{C}^A = (\boldsymbol{B}^A)^\top\boldsymbol{B}^A = \boldsymbol{Z}^A\boldsymbol{C}\boldsymbol{Z}^A,$$

with $\boldsymbol{Z}^A = \boldsymbol{I} - \boldsymbol{B}_A(\boldsymbol{B}_A^\top\boldsymbol{B}_A)^{-1}\boldsymbol{B}_A^\top$, and where $\boldsymbol{C}^A$ is the DPP kernel conditioned on the event that all items in A are observed, and $\boldsymbol{B}_A$ is the restriction of $\boldsymbol{B}$ to the rows and columns indexed by A.

Computing $\boldsymbol{C}^A$ costs $\mathcal{O}(K^3 + |A|^3+K^2|A|^2)$. Next, following Kulesza and Taskar [2012], we eigendecompose $\boldsymbol{C}^A$ to compute the conditional (marginal) probability P_i of every possible item i in $\bar{A}$:

$$P_i = \sum_{n=1}^{K} \frac{\lambda_n}{\lambda_n+1}\left(\frac{1}{\sqrt{\lambda_n}}\boldsymbol{b}_i^A\hat{\boldsymbol{v}}_n\right)^2$$

where $\boldsymbol{b}_i^A$ is column vector for item i in $\boldsymbol{B}^A$ and $(\lambda_n, \hat{\boldsymbol{v}}_n)$ are an eigenvalue/vector of $\boldsymbol{C}^A$.

The computational complexity for computing the eigendecomposition is $\mathcal{O}(K^3)$, and computing P_i for all items in $\bar{A}$ costs $\mathcal{O}(|\bar{A}|K^2)$. Therefore, we have an overall computational complexity of $\mathcal{O}(K^3 + |A|^3+K^2|A|^2+|\bar{A}|K^2)$ for computing next-item conditionals/predictions for the low-rank DPP using the dual kernel, which is significantly superior to the typical cost of $\mathcal{O}(K|\bar{A}|^2+|A|^3)$. □

As in most cases $K \ll |\bar{A}|$, this represents a substantial improvement, allowing us condition in time essentially linear in the size of the item catalog.

5 Experiments

We run next-item prediction and AUC-based classification experiments[4] on two recommendation datasets for DPP evaluation: the UK retail dataset [Chen, 2012], which after clipping all subsets to a maximum size[5] of 100, contains 4070 items and 20059 subsets, and the Belgian Retail Supermarket dataset[6], which contains 88,163 subsets, of a total of 16,470 unique items [Brijs et al., 1999, Brijs, 2003]. We compare the following Contrastive Estimation approaches:

- EXP: explicit negatives learned with CE. As to our knowledge there are no datasets with explicit negative information, we generate approximations of explicit negatives by removing one item from a positive sample and replacing it with the least likely item (Algorithm 3).
- DYN: dynamic negatives learned with CE.

As our work revolves around improving DPP performance, we focus on the two following baselines, which are targeted to learning DPP parametrizations from data:

- NCE: Noise Contrastive Estimation using product negatives.
- LR: the standard low-rank DPP stochastic gradient ascent algorithm from [Gartrell et al., 2017].

NCE learns a model by contrasting $\mathcal{A}^+$ with negatives drawn from a "noisy" distribution p_n, training the model to distinguish between sets drawn from μ and sets drawn from p_n. NCE has gained popularity due to its ability to model distributions μ with untractable normalization coefficients, and has been shown to be a powerful technique to improve submodular recommendation models [Tschiatschek et al., 2016]. NCE learns by maximizing the following conditional

[4] All code is implemented in Julia and will be made publicly available upon publication.

[5] This allows us to use a low-rank matrix factorization for the DPP that scales well in terms of train and prediction time.

[6] http://fimi.ua.ac.be/data/retail.pdf

(a) UK dataset				(b) Belgian dataset		
		Improvement over LR			Improvement over LR	
Metric	LR	EXP	DYN	LR	EXP	DYN
MPR	80.07	**3.75 ± 0.16**	**3.74 ± 0.16**	79.42	**9.58 ± 0.15**	**9.64 ± 0.13**
AUC	0.57297	**0.41465 ± 0.01334**	**0.41467 ± 0.01339**	0.6162	**0.3705 ± 8.569e-5**	**0.3702 ± 4.447e-5**

Table 1: Results over the UK and Belgian datasets. Both explicit and dynamic CE obtain statistically significant improvements in MPR and AUC metrics, confirming that CE learning enhances recommender value of the model and its ability to distinguish data drawn from the target distribution from fake samples. The impact on precisions@k metrics is not reported as we did not observe statistically significant deviations from LowRank performance.

log-likelihood:

$$\phi_{\text{NCE}}(\boldsymbol{L}) = \sum_{A \in \mathcal{A}^+} \log P(A \in \mathcal{A}^+ \mid A) + \sum_{A \in \mathcal{A}^-} \log P(A \in \mathcal{A}^- \mid A). \tag{7}$$

The key difference between NCE and CE lies in how negative information is used: whereas CE learns to assign a low probability to negative subsets, NCE's task is more indirect, learning to distinguish positive from negative examples. As a consequence of NCE's objective function (Eq. 7), NCE requires knowledge of the distribution of negative samples, making it difficult to apply when explicit negative samples are available, but not the form of their distribution.

In our experiments, we learn the NCE objective with stochastic gradient ascent for our low-rank model, since $\nabla \log \Pr(A \in \mathcal{A}^* | A, \boldsymbol{V}\boldsymbol{V}^\top)$ is given by

$$\left(\epsilon^* - \left(1 + \frac{|\mathcal{A}^-|}{|\mathcal{A}^+|} \frac{p_n(A)}{\mathcal{P}_{\boldsymbol{V}\boldsymbol{V}^\top}(A)}\right)^{-1}\right) \nabla_{\boldsymbol{V}} \log \mathcal{P}_{\boldsymbol{V}\boldsymbol{V}^\top}(A). \tag{8}$$

where $\epsilon^* = 1$ if $\mathcal{A}^* = \mathcal{A}^+$ and 0 otherwise.

Algorithm 3 Approximate explicit negative generation

input: Positive sample A^+
Sample $i \neq j \in A^+$ w.p. $p_i \propto \widehat{P}(\{i\})$
Sample $k \notin A^+$ w.p. $p_k \propto 1 - \widehat{P}(\{i, k\})$.
return $(A^+ \backslash \{j\}) \cup \{k\}$

This allows us to approximate true explicit negatives, as we use the empirical data to derive "implausible" sets. Note, however, that when using such negatives we have no guarantee that objective function will be well behaved, as opposed to the theoretically grounded dynamic negatives.

5.1 Experimental setup

The performance of all methods are compared using standard recommender system metrics: Mean Percentile Rank (MPR). MPR is a recall-based metric which evaluates the model's predictive power by measuring how well it predicts the next item in a basket, and is a standard choice for recommender systems [Hu et al., 2008, Li et al., 2010].

Specifically, given a set A, let $p_{i,A} = \Pr(A \cup \{i\} \mid A)$. The percentile rank of an item i given a set A is defined as

$$\text{PR}_{i,A} = \frac{\sum_{i' \notin A} \mathbb{1}(p_{i,A} \geq p_{i',A})}{|\mathcal{Y} \backslash A|} \times 100\%$$

The MPR is then computed as

$$\text{MPR} = \frac{1}{|\mathcal{T}|} \sum_{A \in \mathcal{T}} \text{PR}_{i, A \backslash \{i\}}$$

where $\mathcal{T}$ is the set of test instances and i is a randomly selected element in each set A. An MPR of 50 is equivalent to random selection; a MPR of 100 indicates that the model perfectly predicts the held out item.

We also evaluate the discriminative power of each model using the AUC metric. For this task, we generate a set of negative subsets uniformly at random. For each positive subset A^+ in the test set, we generate a negative subset A^- of the same length by drawing $|A^+|$ samples uniformly at random, while ensuring that the same item is not drawn more than once for a subset. We then compute the AUC for the model on these positive and negative subsets, where the score for each subset is the log-likelihood that the model assigns to the subset. This task measures the ability of the model to discriminate between positive subsets (ground-truth subsets) and randomly generated subsets.

In all experiments, 80% of subsets are used for training; the remaining 20% served as test; convergence is reached when the relative change in the validation log-likelihood is below a pre-determined threshold ϵ, set identically for all methods. All results are averaged over 5 learning trials.

5.2 Amazon registries

We conducted an experimental analysis on the largest 7 sub-datasets included in the Amazon Registry dataset, which has become a standard dataset for DPP modeling [Gillenwater et al., 2014, Mariet and Sra, 2015, Gartrell et al., 2017]. Given the small size of these datasets (the largest has 100 items), these experiments serve only to provide insight into the general behavior of the baselines and CE methods as well as the influence of the hyperparameters on convergence.

Table 2 reports the average time to convergence for each method. As generating the dynamic negatives has a high

complexity due to DPP conditioning, DYN is 2.7x slower than EXP. LR is the fastest method, as it does not need to process any negatives. NCE is by far the most time-consuming.

Table 2: Runtime to convergence (s) on the feeding Amazon registry ($\alpha = 1$, $|\mathcal{A}^-|/|\mathcal{A}^+| = 0.5$, $K = 30$).

METHOD	LR	EXP	DYN	NCE
RUNTIME	0.83 ± 0.54	2.69 ± 0.02	7.13 ± 0.28	27.59 ± 2.20

We found that explicit and dynamic CE are not very sensitive to the α and $|\mathcal{A}^-|/|\mathcal{A}^+|$ hyperparameters. For this reason, in all further results, we set $\alpha = 1$ and $|\mathcal{A}^-|/|\mathcal{A}^+| = .5$ in all further experiments. In previous work on low-rank DPP learning [Gartrell et al., 2017], $\alpha = 1$ was found to be a reasonably optimal value, ensuring a fair comparison between all methods.

Further experiments reporting the MPR, AUC and various precisions for the Amazon registries are described in App. B.

5.3 UK and Belgian Retail Datasets

Following [Gartrell et al., 2016], for both the UK and the Belgian dataset, we set the rank K of the kernel to be the size of the largest subset in the dataset (K=100 for the UK dataset, K=76 for the Belgian dataset): this optimizes memory costs while still modeling all ground-truth subsets. Based on our results on the smaller Amazon dataset, we fix $|\mathcal{A}^-|/|\mathcal{A}^+| = 0.5$ and $\alpha = 1$.

Finally, corroborating our timing results on the Amazon registry, we saw that one iteration of NCE required nearly 11 hours on the Belgian dataset (compared to 5 minutes for one iteration of CE). For this reason, we remove NCE as a baseline from all remaining experiments, as it is not feasible in the general case.

Tables 4 (a) and (b) summarize our results; the negative methods show significant MPR improvement over LR, with both DYN and EXP performing almost 10 points higher on the Belgian dataset, and 3 points higher on the UK dataset. This is a striking improvement, compounded by small standard deviations confirming that these results are robust to matrix initialization.

We also see a dramatic improvement over LR in AUC, with an improvement of approximately 0.41 for the UK dataset and 0.37 for the Belgian dataset, across both DYN and EXP methods. Both DYN and EXP perform quite well, with an AUC score of approximately 0.9864 or higher for both models. These results suggest that for larger datasets, CE can be effective at improving the discriminative power of the DPP.

6 Conclusion and future work

We introduce the Contrastive Estimation (CE) optimization problem, which optimizes the difference of the traditional DPP log-likelihood and the expectation of the DPP model's log-likelihood under a *negative* distribution ν. This increases the DPP's fit to the data while simultaneously incorporating inferred or explicit domain knowledge into the learning procedure.

CE lends itself to intuitively similar but theoretically different variants, depending on the choice of ν: a static ν leads to significantly faster learning but allows spurious optima; conversely, allowing ν to evolve along with model parameters limits overfitting at the cost of a more complex optimization problem. Optimizing dynamic CE is in of itself a theoretical problem worthy of independent study.

Additionally, we show that low-rank DPP conditioning complexity can be improved by a factor of M by leveraging the dual representation of the low-rank kernel. This not only improves prediction speed on a trained model, but allows for more efficient dynamic negative generation.

Experimentally, we show that CE with dynamic and explicit negatives provide comparable, significant improvements in the predictive performance of DPPs, as well as on the learned DPP's ability to discriminate between real and randomly generated subsets.

Our analysis also raises both theoretical and practical questions: in particular, a key component of future work lies in better understanding how explicit domain knowledge can be incorporated into the generating logic for both dynamic and static negatives. Furthermore, the CE formulation in Eq. (5) suggests the possibility of using continuous labels for weighted samples within CE.

Acknowledgements

This work was partially supported by a Criteo Faculty Research Award, and NSF-IIS-1409802.

References

R. Affandi, E. Fox, R. Adams, and B. Taskar. Learning the parameters of Determinantal Point Process kernels. In *ICML*, 2014.

Nima Anari, Shayan Oveis Gharan, and Alireza Rezaei. Monte carlo markov chain algorithms for sampling strongly rayleigh distributions and determinantal point processes. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 103–115, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

Julius Borcea and Petter Brändén. The Lee-Yang and Pólya-Schur programs I Linear operators preserving stability. *Inventiones mathematicae*, 177(3), 2009a.

Julius Borcea and Petter Brändén. The Lee-Yang and Pólya-Schur programs II Theory of stable polynomials and applications. *Communications on Pure and Applied Mathematics*, 62(12), 2009b.

Julius Borcea and Petter Brändén. Multivariate Pólya–Schur classification problems in the Weyl algebra. *Proceedings of the London Mathematical Society*, 101(1), 2010.

Julius Borcea, Petter Brändén, and Thomas Liggett. Negative dependence and the geometry of polynomials. *Journal of the American Mathematical Society*, 22(2), 2009a.

Julius Borcea, Petter Brändén, and Boris Shapiro. Classification of hyperbolicity and stability preservers: the multivariate Weyl algebra case. *arXiv preprint math.CA/0606360*, 2009b.

Alexei Borodin. Determinantal Point Processes. *arXiv:0911.1153*, 2009.

Avishek Bose, Huan Ling, and Yanshuai Cao. Adversarial contrastive estimation. *arXiv preprint arXiv:1805.03642*, 2018.

Léon Bottou. On-line learning in neural networks. Cambridge University Press, 1998.

Tom Brijs. Retail market basket data set. In *Workshop on Frequent Itemset Mining Implementations (FIMI'03)*, 2003.

Tom Brijs, Gilbert Swinnen, Koen Vanhoof, and Geert Wets. Using association rules for product assortment decisions: A case study. In *SIGKDD*. ACM, 1999.

Olivier Canévet and Francois Fleuret. Efficient Sample Mining for Object Detection. In *ACML*, JMLR: Workshop and Conference Proceedings, 2014.

Wei-Lun Chao, Boqing Gong, Kristen Grauman, and Fei Sha. Large-margin Determinantal Point Processes. In *Uncertainty in Artificial Intelligence (UAI)*, 2015.

D Chen. Data mining for the online retail industry: A case study of rfm model-based customer segmentation using data mining. *Journal of Database Marketing and Customer Strategy Management*, 19(3), August 2012.

Long Chen, Fajie Yuan, Joemon M. Jose, and Weinan Zhang. Improving negative sampling for word representation using self-embedded features. *CoRR*, abs/1710.09805, 2017.

Laurent Decreusefond, Ian Flint, Nicolas Privault, and Giovanni Luca Torrisi. Determinantal Point Processes, 2015.

Josip Djolonga, Sebastian Tschiatschek, and Andreas Krause. Variational inference in mixed probabilistic submodular models. In *NIPS*, 2016.

Christophe Dupuy and Francis Bach. Learning Determinantal Point Processes in sublinear time, 2016.

Mike Gartrell, Ulrich Paquet, and Noam Koenigstein. Bayesian low-rank Determinantal Point Processes. In *RecSys*, 2016.

Mike Gartrell, Ulrich Paquet, and Noam Koenigstein. Low-rank factorization of Determinantal Point Processes. In *AAAI*, 2017.

J. Gillenwater. *Approximate Inference for Determinantal Point Processes*. PhD thesis, University of Pennsylvania, 2014.

J. Gillenwater, A. Kulesza, E. Fox, and B. Taskar. Expectation-maximization for learning Determinantal Point Processes. In *NIPS*, 2014.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

Ian J. Goodfellow. On distinguishability criteria for estimating generative models, 2014.

Michael U. Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. Mach. Learn. Res.*, 13, February 2012. ISSN 1532-4435.

Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. *JMLR*, 9, 2008.

A. Kulesza. *Learning with Determinantal Point Processes*. PhD thesis, University of Pennsylvania, 2013.

A. Kulesza and B. Taskar. *Determinantal Point Processes for machine learning*, volume 5. Foundations and Trends in Machine Learning, 2012.

Frédéric Lavancier, Jesper Møller, and Ege Rubak. Determinantal Point Process models and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(4), 2015.

Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Fast dpp sampling for nystrom with application to kernel methods. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2061–2070, New York, New York, USA, 20–22 Jun 2016. PMLR.

Yanen Li, Jia Hu, ChengXiang Zhai, and Ye Chen. Improving one-class collaborative filtering by incorporating rich user information. In *CIKM*, 2010.

H. Lin and J. Bilmes. Learning mixtures of submodular shells with application to document summarization. In *UAI*, 2012.

O. Macchi. The coincidence approach to stochastic point processes. *Adv. Appl. Prob.*, 7(1), 1975.

Zelda Mariet and Suvrit Sra. Fixed-point algorithms for learning Determinantal Point Processes. In *ICML*, 2015.

Zelda Mariet and Suvrit Sra. Diversity networks. *Int. Conf. on Learning Representations (ICLR)*, 2016a.

Zelda Mariet and Suvrit Sra. Kronecker Determinantal Point Processes. In *NIPS*, 2016b.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*. 2013.

Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *ICML*, 2012.

Takayuki Osogami, Rudy Raymond, Akshay Goel, Tomoyuki Shirai, and Takanori Maehara. Dynamic Determinantal Point Processes. In *AAAI*, 2018.

Robin Pemantle. Towards a theory of negative dependence. *Journal of Mathematical Physics*, 41(3), 2000.

Patrick Rebeschini and Amin Karbasi. Fast mixing for discrete point processes. In *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 1480–1500, Paris, France, 03–06 Jul 2015. PMLR.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016.

Noah A. Smith and Jason Eisner. Guiding unsupervised grammar induction using contrastive estimation. In *In Proc. of IJCAI Workshop on Grammatical Inference Applications*, 2005a.

Noah A. Smith and Jason Eisner. Contrastive estimation: Training log-linear models on unlabeled data. In *ACL*, ACL '05, 2005b.

Kah Kay Sung. *Learning and Example Selection for Object and Pattern Detection*. PhD thesis, Massachusetts Institute of Technology, 1996.

Sebastian Tschiatschek, Josip Djolonga, and Andreas Krause. Learning probabilistic submodular diversity models via noise contrastive estimation. In *AISTATS*, 2016.

John Urschel, Victor-Emmanuel Brunel, Ankur Moitra, and Philippe Rigollet. Learning Determinantal Point Processes with moments and cycles. In *ICML*, 2017.

Cheng Zhang, Hedvig Kjellström, and Stephan Mandt. Stochastic learning on imbalanced data: Determinantal Point Processes for mini-batch diversification. *CoRR*, abs/1705.00607, 2017.