Learning Probabilistic Submodular Diversity Models Via Noise Contrastive Estimation

Sebastian Tschiatschek
ETH Zurich
tschiats@inf.ethz.ch

Josip Djolonga
ETH Zurich
josipd@inf.ethz.ch

Andreas Krause
ETH Zurich
krausea@ethz.ch

Abstract

Modeling diversity of sets of items is important in many applications such as product recommendation and data summarization. Probabilistic submodular models, a family of models including the determinantal point process, form a natural class of distributions, encouraging effects such as diversity, repulsion and coverage. Current models, however, are limited to small and medium number of items due to the high time complexity for learning and inference. In this paper, we propose FLID, a novel log-submodular diversity model that scales to large numbers of items and can be efficiently learned using noise contrastive estimation. We show that our model achieves state of the art performance in terms of model fit, but can be also learned orders of magnitude faster. We demonstrate the wide applicability of our model using several experiments.

1 INTRODUCTION

Suppose that a search query for images of *Venice* returns only images of *St. Mark's Square* — this is probably not matching the expectations of a user seeking images that are both of high quality and represent all important sights of Venice. For example, in addition to *St. Mark's Square*, one would also expect images of the *canals* and the famous *Carnival of Venice*. There are many more applications for which diversity is an important property, e.g., product recommendation in which given a partial shopping basket, recommended products should be complementary to the products in the basket [1], and extractive text summarization in which selected sentences from a text should not only summarize the text, but also be diverse and non-redundant [2].

Determinantal point processes (DPPs) are a prominent tool [3, 4, 5] for modeling diversity of item sets, offering the benefit of quantifying uncertainty. They have been successfully used in several applications, such as search result diversification [6] and video summarization [7]. However, DPPs scale poorly to large item sets because the inference problem is computationally challenging (the time complexity is roughly $\mathcal{O}(N^3)$, where N is the number of items). This renders them inapplicable for the nowadays large scale and big data applications, e.g., image search over millions of items.

As it turns out, DPPs are a special instance of Probabilistic Submodular Models (PSMs) [8, 9], i.e., distributions over subsets S of some finite ground set V, that assign probabilities of the form $P(S) \propto \exp(F(S))$ for some submodular function $F\colon 2^V \to \mathbb{R}$. Such log-submodular distributions can specify expressive probabilistic models and generalize several existing probabilistic models, e.g., repulsive Ising models and DPPs. Because of the submodularity property, these distributions are natural candidates for modeling the notions of representativeness and diversity. Moreover, this family of distributions is closed under conditioning, and there is an efficient algorithm that yields a $\frac{1}{2}$-approximation to the MAP configuration [10]. Most existing literature on log-submodular distributions deals with performing inference for fixed (known) submodular functions [8, 11, 12]. A natural and important direction, which we pursue in this paper, is to investigate the problem of *learning* these distributions.

In this paper, we propose the FLID *(Facility LocatIon Diversity) model*, a novel probabilistic submodular model based on embedding each item in a latent vector space. Furthermore, even though the model cannot generally be normalized efficiently, we show how it can be efficiently fitted via noise contrastive estimation (NCE) [13]. In extensive experiments on

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51.

several applications, we show that our model provides fits on par to current state of the art models, but at the same time offers vastly superior scalability to large sets of items.

2 DIVERSITY MODEL: FLID

We will now introduce our novel probabilistic submodular diversity model. A probability distribution $P(S)$ over subsets $S \subseteq V$, where w.l.o.g. $V = \{1, \ldots, N\}$, is *log-submodular* if it can be written as $P(S) \propto \exp(F(S))$, where $F\colon 2^V \to \mathbb{R}$ is a submodular set function. Submodular functions F are characterized by a natural diminishing returns property, i.e.,

$$F(A \cup \{i\}) - F(A) \geq F(B \cup \{i\}) - F(B),$$

for all $A \subseteq B \subseteq V \setminus \{i\}$. Intuitively, this property means that the gain of adding an item i in the context of the smaller set A is larger than the gain of adding that element in the context of the larger set B. This property renders submodular functions natural candidates for modeling *coverage* and *diversity*.

The first component of our model is an item relevance term. Specifically, to each item i in the ground set V we will associate some number $u_i \in \mathbb{R}$ that quantifies the *quality* of that item. For example, in the image search example from the introduction we would expect the better looking images to have higher relevance terms. The first model that one may try, which we call *modular*, is to use a distribution of the form

$$P(S) \propto \exp\left(\sum_{i \in S} u_i\right),$$

which is completely factorized. However, this can only capture the frequencies of the individual items, ignoring any dependences. We hence add an extra diversity term to model the extent to which items are substitutes of each other. To this end, we will assign to each item i an L-dimensional vector $\mathbf{w}_i \in \mathbb{R}^L_{\geq 0}$. The intuition is that each of these L dimensions will capture some concept (e.g. *image shows St. Mark's Square*, *image shows people in carnival costumes*, etc.), and we interpret $w_{i,d}$ as a quantification of how *relevant* that item i is for that specific concept d. To quantify the diversity of some items $S \subseteq V$ with respect to dimension d, we propose the term $\max_{i \in S} w_{i,d} - \sum_{i \in S} w_{i,d}$. This term is a nonpositive penalty, evaluating to 0 iff S contains at most one item i with positive value $w_{i,d} > 0$. Summing over all latent dimensions d, the complete model that we propose is:

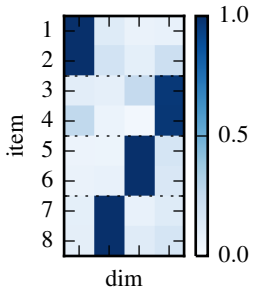


Figure 1: Weights $\mathbf{W}$ learned from the synthetic experiments. Each row corresponds to the factored representation of an item. Different groups are separated by a horizontal dotted line.

$$P(S \mid \mathbf{u}, \mathbf{W}) = \underbrace{\frac{1}{Z} \exp\left(\sum_{i \in S} u_i + \underbrace{\sum_{d=1}^{L} \left(\max_{i \in S} w_{i,d} - \sum_{i \in S} w_{i,d}\right)}_{\mathrm{Div}(S)}\right)}_{\widetilde{P}(S \mid \mathbf{u}, \mathbf{W})}, \tag{1}$$

Hereby, $\mathbf{u}$ is the vector of item qualities $\mathbf{u} = [u_i]_{i \in V}$, and $\mathbf{W} = [w_{i,d}]_{i \in V, 1 \leq d \leq L}$ represents the latent diversity embedding for all items $i \in V$. We refer to this model as the FLID (Facility LocatIon Diversity) model[1]. As a concrete example, consider the latent properties encoded by the matrix shown in Figure 1. In this example, we have that $\mathrm{Div}(\{1, 2\}) \approx -1$ (items 1 and 2 have roughly the same latent properties), while $\mathrm{Div}(\{1, 5, 7\}) = 0$ (these items have roughly pairwise *orthogonal* latent properties). Also, note that $\mathrm{Div}(S)$ is zero whenever $|S| = 1$. This is a desired property because there is no meaningful notion of diversity for sets containing only a single item.

Using ideas similar to the normalization of submodular point processes [14], for any fixed L, we can compute the partition function, and hence marginals, conditionals etc., of the FLID model in time polynomial in the ground set size by Algorithm 1 (the algorithm and the proof are presented in the Appendix):

Proposition 1 (Partition Function of FLID)**.** *The time complexity for computing the partition function of the* FLID *model using Algorithm 1 is* $\mathcal{O}(|V|^{L+1})$.

[1]Submodular functions $F(S) = \sum_i \max_{j \in S} m_{i,j}$ for some matrix $\mathbf{M} \geq 0$ are called *facility location functions*. The idea is that $m_{i,j}$ models the value that a selected facility at location j may provide to customer i. Each customer subscribes to the facility that provides largest value. $F(S)$ is thus the total value provided to all customers.

For the case of $L = 2$, the time complexity for normalizing the FLID model is the same as for DPPs. For larger values of $|V|$ and L, the time complexity may be prohibitive. In these cases, the partition function can be estimated, e.g., using Gibbs sampling [11].

Note that exact sampling from the FLID model is in general difficult. However, under certain conditions on $\mathbf{W}$ sampling from FLID can be performed efficiently using Gibbs sampling [11]. In contrast to our model, there are customized and efficient algorithms for sampling from DPPs. The main bottleneck of these algorithms is however the eigendecomposition of the kernel matrix which can be computationally expensive for large ground sets V [4].

An important quantity, that is useful, e.g., for recommender systems, is the probability of adding a *single* item to some set of already chosen items $S \subseteq V$. Formally, for any such S we will define the quantities $Q_S(i) = \frac{P(\{i\}|S)}{\sum_{j \in V \setminus S} P(\{j\}|S)}$ for any $i \in V \setminus S$. Note that these quantities can be computed efficiently for FLID as we have to sum up over only $|V \setminus S|$ items and the normalization constants cancel.

3 EFFICIENT TRAINING

Unfortunately, maximum likelihood estimation of the FLID model is intractable for large L, due to the presence of the partition function in the log-likelihood. Instead, we will use an alternative estimation method known as *noise contrastive estimation* (NCE) [13]. The idea behind NCE is to pose the estimation problem as a discriminative learning task in which the goal is to discriminate between samples from the data distribution and from a known noise distribution. This noise distribution can be chosen almost arbitrarily under mild conditions, cf. §4.3 for a brief discussion.

We now describe the parameter estimation in more detail. Assume that we are given a labeled data set $\mathcal{A} = \{(S, Y_S)\}$, where $S \subseteq V$ and $Y_S \in \{0, 1\}$, formed from a set of data samples $\mathcal{D}$ and noise samples $\mathcal{N}$ by labeling them as 1 and 0, respectively. That is, $\mathcal{A} = \{(S, 1) \colon S \in \mathcal{D}\} \cup \{(S, 0) \colon S \in \mathcal{N}\}$. The data samples are from the unknown and to be estimated data distribution P_d, while the noise samples are drawn from some known and properly normalized noise distribution P_n. Then, the conditional likelihood that $S \in \mathcal{D} \cup \mathcal{N}$ came from $\mathcal{D}$ is

$$P(Y_S = 1 \mid S) = \frac{P(S \mid Y_S = 1)}{P(S \mid Y_S = 1) + \eta P(S \mid Y_S = 0)} = \frac{P_d(S)}{P_d(S) + \eta P_n(S)},$$

where $\eta = P(N)/P(D) = |\mathcal{N}|/|\mathcal{D}|$. As P_d is unknown, we cannot compute $P(Y_s = 1 \mid S)$. In NCE, P_d is thus substituted by the (un-normalized) model distribution to be estimated including a parameter $\hat{Z}$ to scale the model, i.e., in our case P_d is substituted by $\frac{1}{\hat{Z}} \widetilde{P}(S \mid \mathbf{u}, \mathbf{W})$. The parameters of the model together with $\hat{Z}$, i.e., $\theta = [\mathbf{u}_{\text{NCE}}, \mathbf{W}_{\text{NCE}}, \hat{Z}]$, are then estimated by maximizing the conditional log-likelihood of the labels Y_S given S for $S \in \mathcal{D} \cup \mathcal{N}$, which is equivalent to maximizing the following objective:

$$g(\theta) = \sum_{S \in \mathcal{D}} \log P(Y_S = 1 \,|\, S, \theta) + \sum_{S \in \mathcal{N}} \log P(Y_S = 0 \,|\, S, \theta) \tag{2}$$

For maximizing $g(\theta)$, the parameters of the FLID model must be adjusted to effectively discriminate between data and noise samples. Furthermore, the parameter $\hat{Z}$ is adjusted — this corresponds to a rescaling of $\widetilde{P}(S \mid \mathbf{u}, \mathbf{W})$. For convenience, $\hat{Z}$ can also be included in the model itself yielding an approximately normalized model after NCE [13].

Although NCE is not as common as maximum likelihood estimation, it enjoys several nice theoretical guarantees. Assuming that the true data distribution is in the model class, that P_n is nonzero whenever P_d is nonzero, and two other mild technical conditions (having their counterparts in maximum likelihood estimation), the NCE estimator is guaranteed to converge to the true parameters in probability [13] in the limit of infinite data.

For our model, the objective $g(\theta)$ is non-convex, hence we can only aim to identify good local optima. This is akin to similar non-convexity challenges arising when fitting DPPs (despite the fact that these can be efficiently normalized). In the following, we show how gradient-based techniques can be efficiently implemented for optimizing $g(\theta)$.

Gradient-based optimization In our experiments, cf., §5, we optimize the NCE objective (2) using adaptive gradient descent (AdaGrad) [15]. Thus, in every iteration of AdaGrad, the gradient $\nabla \log P(Y_S = Y \mid S)$ of a labeled sample $(S, Y) \in \mathcal{A}$ must be computed (at points at which the function is not differentiable, we break ties arbitrarily). This gradient is given as

$$\nabla \log P(Y_S = Y \mid S) = \left(Y - \frac{1}{1 + \eta \frac{P_n(S)}{\frac{1}{\hat{Z}} \widetilde{P}(S|\mathbf{u}, \mathbf{W})}} \right) \nabla \log \frac{1}{\hat{Z}} \widetilde{P}(S \mid \mathbf{u}, \mathbf{W}), \tag{3}$$

with

$$\left(\nabla_u \log \frac{\widetilde{P}(S \mid \mathbf{u}, \mathbf{W})}{\hat{Z}}\right)_i = \begin{cases} 1 & i \in S \\ 0 & \text{otherwise} \end{cases}$$

$$\left(\nabla_{\mathbf{W}} \log \frac{\widetilde{P}(S \mid \mathbf{u}, \mathbf{W})}{\hat{Z}}\right)_{i,d} = \begin{cases} -1 & i \neq \arg\max_{j \in S} w_{j,d} \\ 0 & \text{otherwise} \end{cases}$$

$$\nabla_{\hat{Z}} \log \frac{\widetilde{P}(S \mid \mathbf{u}, \mathbf{W})}{\hat{Z}} = -\frac{1}{\hat{Z}},$$

where $(\nabla_u \log \widetilde{P}(S \mid \mathbf{u}, \mathbf{W}))_i$ denotes the i^{th} entry of the gradient with respect to u and $(\nabla_{\mathbf{W}} \log \widetilde{P}(S \mid \mathbf{u}, \mathbf{W}))_{i,d}$ the $(i,d)^{\text{th}}$ entry of the gradient with respect to $\mathbf{W}$, respectively. Evaluating (3) requires $\mathcal{O}(L|S|)$ time. A full pass through all the samples (including the noise samples) takes $\mathcal{O}(|\mathcal{D} \cup \mathcal{N}|\kappa L)$ time, where $\kappa = \max_{S \in \mathcal{D} \cup \mathcal{N}} |S|$. This is a noteworthy smaller cost than that of performing one iteration of projected gradient descent for learning DPPs, as explained in more detail in §4.3.

To ensure that all weights remain non-negative, they have to be projected onto $\mathbb{R}_{\geq 0}^{|V| \times L}$ after each gradient step. Let $\mathbf{W}'$ be the weights after a gradient step. Then, their projection is $\mathbf{W} = [\max\{0, w'_{i,d}\}]_{i \in V, 1 \leq d \leq L}$. This requires an additional $\mathcal{O}(L|S|)$ time for sample S.

4 EXPERIMENTAL SETUP

We now describe the setup for our experiments in §5, including the used datasets, details on the application of NCE, baselines and the considered metrics.

4.1 Datasets

Amazon Baby Registries This dataset consists of baby registries collected from Amazon [16] and is a standard benchmark dataset for evaluating DPPs. These registries are split into sub-registries according to categories, e.g., *safety* and *feeding*. For each category, the data consists of a set of products V and a set of registries over these products. The number of items in the sub-registries ranges from 32 to 100, and eachsubregistry contains about 5,000 to 13,300 instances. More details on the dataset can be found in [16].

For demonstrating the scalability of our model, we furthermore considered a larger version of the Amazon baby registries data. For creating this data, we did not split the originally collected data into categories, but considered the whole data. We filtered out all items that did not appear in at least 10 registries, resulting in a total of 7,058 items and 32,468 registries.

In the experiments, we used 10 fold cross-validation for estimating statistics.

Product Recommendation We adopted the baby registry data from above for a product recommendation task. In this task, we aim to predict an item that best complements a given registry. For this task we created new test datasets

$$\mathcal{T}' = \{(S \setminus \{i\}, i) \colon S \in \mathcal{T}, |S| \geq 2, i \in S\}$$

from the original test datasets $\mathcal{T}$. In other words, $\mathcal{T}'$ consists of partial registries obtained from $S \in \mathcal{T}$ by removing a single element $i \in S$, keeping also the element that was removed. We will use the shorthand $\check{S}_i = S \setminus \{i\}$.

Image Collection Data For the image collection summarization task in §5.5 we considered the dataset from [17] consisting of a total of 14 image collections with 100 images each. The image collections were, for the most part, taken during holiday trips. For each image collection, several hundred human generated summaries of size 10 of that collection were obtained using Amazon Mechanical Turk.

4.2 Details on the Application of NCE

Noise Distribution In all experiments using NCE, we must make use of a noise distribution for contrasting the model distribution against, cf. §3. The performance of NCE depends on the quality of this noise distribution. Intuitively, the noise distribution should be close to the data distribution to efficiently estimate its properties [13]. Furthermore, evaluating the probability of a sample under the noise distribution P_n must be efficient so that $g(\theta)$ can be optimized efficiently. Additionally, it must be efficient to sample from P_n to obtain the noise samples for NCE. Therefore, one natural choice for the noise distribution is a product (log-modular) distribution. For such distributions, the maximum likelihood parameters and the partition function can be computed in closed form.

Parameter Initialization As our objective $g(\theta)$ is non-convex, initialization is an important issue. We initialize the utilities u_i corresponding to the empirical marginal distribution of $\{i\}$. The weights $\mathbf{W}$ were initialized randomly to small non-zero values drawn from a uniform distribution on $[0, 0.001]$.

Weight Projection During Gradient Descent As discussed in §2, the weights must be projected during the application of AdaGrad to ensure that they stay non-negative. We found that instead of projecting the parameters $\mathbf{W}'$ after each gradient step via $\mathbf{W} = [\max\{0, w'_{i,d}\}]_{i \in V, 1 \leq d \leq L}$, better performance can be achieved by adding small random noise to weights that would have had become clipped, i.e., the weights

are *projected* by

$$w_{i,d} = \begin{cases} w_{i,d} = w'_{i,d} & w'_{i,d} \geq 0 \\ w_{i,d} \sim \mathcal{U}([0, 0.001]) & w'_{i,d} < 0, \end{cases}$$

where $w_{i,d} \sim \mathcal{U}([0, 0.001])$ means that $w_{i,d}$ is drawn from a uniform distribution on $[0, 0.001]$. The randomness introduced in this way may help to escape local optima.

4.3 Baselines

For benchmarking our model, we use modular distributions estimated using maximum likelihood estimation and DPPs as baselines. Determinantal point processes (DPP) [18] are probably the most well-known log-submodular diversity models. They are parameterized by a positive semi-definite matrix $A \succcurlyeq 0$, and the mass assigned to any set $S \subseteq V$ is given by $P(S) = \det A_S / \det(A + I)$, where A_S denotes the submatrix obtained from A by taking the rows and columns S. As evident from the formula, the model is already normalized and there is no need for approximate inference if one can afford the cost of computing the determinant. The task of learning the matrix A from data is very challenging, however, as it results in non-convex problems. In [16] the authors develop an EM algorithm for learning these models, and a faster scheme has been proposed recently in [19]. We refer to this faster scheme as *fixed-point iterations* (FP). One may be tempted to estimate DPPs using NCE to reduce computational complexity. However, a naive approach fails: Computing the gradient of the likelihood of the DPP requires computation of matrix inverses which has complexity $\mathcal{O}(|V|^3)$ [16]. This renders this approach infeasible, even for medium-scale values of $|V|$.

4.4 Evaluation Metrics

To quantify the model fit, we report the relative improvement in log-likelihood over a modular distribution (fully factorized distribution over the items, i.e., no diversity term) fitted using maximum likelihood estimation, i.e., we report

$$\text{LLRI} = 100 \cdot \frac{\mathcal{L}_{\text{method}} - \mathcal{L}_{\text{modular}}}{|\mathcal{L}_{\text{modular}}|}, \tag{4}$$

where $\mathcal{L}_{\text{modular}}$ is the log-likelihood of the test data for the modular distribution and $\mathcal{L}_{\text{method}}$ for the evaluated methods (DPPs with EM, DPPs with FP, FLID with NCE), respectively. We call this measure *log-likelihood relative improvement (LLRI)*.

For assessing the performance of the considered models on the product recommendation task we use the following two metrics:

- *Accuracy.* Given a partial registry $(\check{S}_i, i) \in \mathcal{T}'$, the models are asked to *complete* $\check{S}_i$ by proposing an element $j \in V \setminus \check{S}_i$. For the FLID and modular model, we proposed the most likely element under $Q_{\check{S}_i}$. For DPPs, we proposed the element with largest marginal probability given $\check{S}_i$. The *accuracy* of this proposal is computed as $|\{i\} \cap \{j\}|$, i.e., equal to 1 if j was the removed item and 0 otherwise. These accuracies are then averaged over all partial registries in $\mathcal{T}'$.

- *Mean reciprocal rank (MRR).* Given test data $(\check{S}_i, i)$, we compute using each model the probability of adding single items $Q_{\check{S}_i}(j) \propto P(\{j\} \mid \check{S}_i)$ over the candidates $j \in V \setminus \check{S}_i$, which we have already discussed at the end of §2. The rank $\text{rank}_j^{\check{S}_i}$ of item $j \in V \setminus \check{S}_i$ in the context of $\check{S}_i$ is m if $Q_{\check{S}_i}(j)$ is the m^{th} largest probability (we break ties arbitrarily). The MRR score is then computed as

$$\text{MRR} = \frac{100}{|\mathcal{T}'|} \sum_{\check{S}_i \in \mathcal{T}'} \frac{1}{\text{rank}_i^{\check{S}_i}}.$$

5 EXPERIMENTS

This section consists of three parts: (i) we show that the FLID models estimated using NCE match our intuition on synthetic data, (ii) we evaluate our model on a standard benchmark dataset for learning DPPs, and (iii) we demonstrate that FLID models can be easily scaled to large ground sets and datasets consisting of thousands of samples, beyond reach of DPPs.

5.1 Synthetic Experiments

In our synthetic experiments, we demonstrate that our proposed FLID model can effectively discover groups of items where all items within a particular group are substitutes of each other. For this purpose, we assume a ground set $V = \cup_{i=1}^{G} \mathcal{G}_i$ composed of G groups of mutually disjoint sets of items $\mathcal{G}_i$. We now create random subsets of V such that smaller subsets are more likely and subsets never contain multiple items from any particular group. In more detail, we created subsets $S_i \in V$ as follows:

1. Sample $|S_i|$ according to a truncated geometric distribution, i.e., $|S_i| \propto \exp(-\lambda |S_i|)$, with $|S_i| \in \{1, \dots, G\}$, where $\lambda > 0$ is a parameter.
2. Select $|S_i|$ groups uniformly at random, and draw one element uniformly at random from each of these groups. The drawn items constitute S_i.

Following this sampling scheme, every set S_i contains at most one item from each group $\mathcal{G}_1, \dots, \mathcal{G}_G$.

We instantiate the above setup for $G = 4$, $\lambda = 1$ and $\mathcal{G}_i = \{2i-1, 2i\}$ for all $i \in \{1, \ldots, G\}$ and drew 500 samples. Using $L = G$ latent dimensions, we estimate a model of the form (1) by NCE using 1,000 noise samples. The learned weights $\mathbf{W}$ are shown in Figure 1. One can observe that

(i) items from the same group have similar factored representations (there is a *penalty* for including multiple items from a single group), and that

(ii) the factored representations from items of different groups are approximately orthogonal to each other (there is no *penalty* for including items from different groups).

5.2 Amazon Baby Registries – Benchmark Comparison

We next considered the Amazon baby registry data. The task was, given the registries of a category, to fit a probabilistic model and use that model for inference. More concretely, we simply computed the likelihood of the test data. We compared the model fit and running time of our FLID model with the baseline models. We fitted FLID models using NCE and $L = 3$ latent dimensions (this small number of latent dimensions already allows for a clustering of the items; experiments with larger values of L are considered in §5.3). The step size of AdaGrad was set to 1. For optimization, AdaGrad was shown $20 \cdot |\mathcal{D} \cup \mathcal{N}|$ samples selected uniformly at random from the training data and the noise samples (we created $10 \cdot |\mathcal{D}|$ noise samples).

We report the LLRI score based on 10-fold cross-validation in Figure 2a. We observe that our model has the largest improvement over the modular distribution for categories with smaller $|V|$, e.g., *safety* and *furniture*. For some of the categories with larger $|V|$, e.g., *feeding*, we significantly outperform DPPs. For the other categories, our model performs on par with DPPs estimated using EM and FP, respectively.

For running time comparisons with DPPs we used the implementation provided by the authors[2], while our model is implemented in Python/C++.[3] Results showing the cumulative running time for training on all 10 folds are presented in Figure 2b. We observe that for all categories, our model is significantly faster than DPPs — even faster than DPPs estimated using FP. The gap in runtime between our model and DPPs would become even more significant for datasets with even larger ground sets. This is a consequence of the linear cost per iteration of the FLID model, compared to the cubic per iteration cost of DPPs. In more detail: The per iteration cost for learning DPPs using FP is $\mathcal{O}(|\mathcal{D}|\kappa^3 + |V|^3)$, where $\kappa = \max_{S \in \mathcal{D}} |S|$. Thus, the cubic complexity in κ and $|V|$ (which may be much larger for typical data), is the limiting factor for scaling DPPs to large ground sets.[4] In contrast, our model has a per iteration cost of $\mathcal{O}(|\mathcal{D} \cup \mathcal{N}|\kappa' D)$ time, where $\kappa' = \max_{S \in \mathcal{D} \cup \mathcal{N}} |S|$. Thus, the per iteration cost scales only linear with $\kappa' \leq |V|$.

5.3 Amazon Baby Registries – Product Recommendation

For learning our FLID model, we used $L = 10$ (for smaller L the performance degrades) latent dimensions to allow for rich structure in $\mathbf{W}$ and $|\mathcal{N}| = 10|\mathcal{D}|$ noise samples. The step size of AdaGrad was set to 1. The accuracy and mean reciprocal rank results results are summarized in Figures 2c and Figures 2d, respectively. We can observe that FLID compares favorably with both the modular distributions and DPPs. FLID improves significantly over the modular distributions in several cases. Compared to DPPs, FLID performs mostly on par, improving over DPPs significantly on a few datasets, e.g., *carseats* and *strollers*.

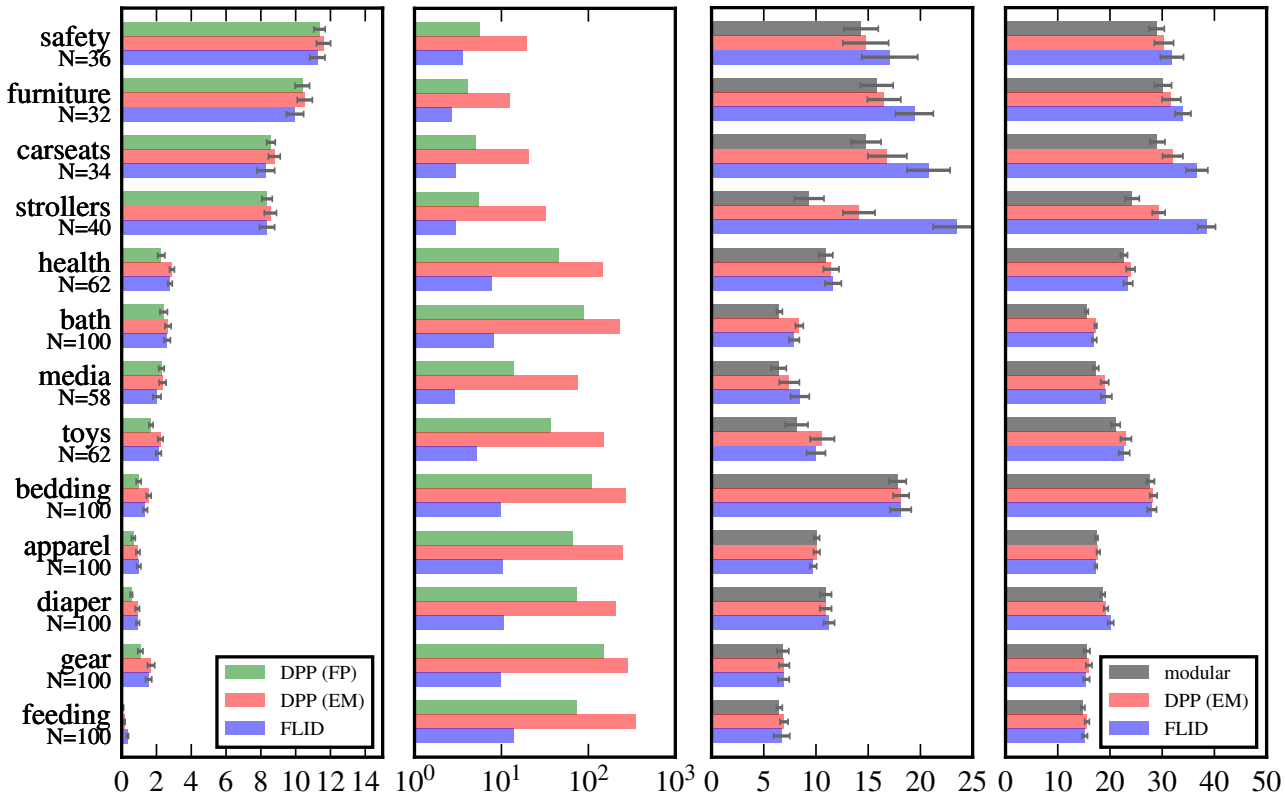
5.4 Large Scale Experiments

We performed the product recommendation experiments from the last section on the 7,058 item version of the Amazon baby registry data. For estimating the FLID model using NCE, we used $L = 40$ latent dimensions and $50 \cdot |\mathcal{D}|$ noise samples. In contrast to the experiments before we used stochastic gradient descent for optimization because this gave better results. We used a step size of $\frac{10^{-2}}{t^{0.1}}$, where t is the iteration, and performed a total of $50 \cdot |\mathcal{D} \cup \mathcal{N}|$ iterations. Training on a single fold took roughly 7 minutes. Training of DPPs did not finish within 2 hours. We achieved a small improvement over the modular distributions on this task, i.e., a relative improvement in accuracy of $6.56\% \pm 4.49$, and a relative improvement in MRR of $1.35\% \pm 1.19$, respectively. This is considerable given only a very limited number of registries in relation to the relatively large number of items.

[2]We found that computing the termination criterion used in [19] is substantial more time consuming than performing the actual fixed-point iterations. Therefore, we modified the authors' code to check the termination criterion only in every 10^{th} iteration resulting in a speedup for the whole training procedure.

[3]The source code will be publicly released after publication. Experiments were run on an OS X system with 2.8 GHz Intel Core i5 and 16 GB memory.

[4]The per iteration cost for using the EM algorithm from [16] is similar, i.e., quadratic in κ and cubic in $|V|$.



(a) Log-likelihood relative improvement (b) Cumulative runtime in seconds (c) Accuracy (d) MRR

Figure 2: Experimental results for the Amazon baby registry data. The horizontal grey bars in the figures indicate the variance of the shown metrics across the cross-validation folds (their widths equal two standard deviations). The FLID model improves the LLRI, accuracy and MRR metrics over the modular baseline on all datasets. In several cases, FLID also improves significantly over DPPs, e.g., in terms of LLRI on *feeding*, in terms of accuracy on *strollers* and *carseats*, and in terms of MRR on *strollers* and *diaper*. FLID can be trained much faster than DPPs.

5.5 Image Collection Summarization

Finally, we considered an image collection summarization application. Here, the task is, given a set of images, to select a subset of those images that represent the whole collection best and is as little redundant as possible. We used 90% of the summaries for fitting a FLID model using NCE with $D = 10$. As noise, we used 200,000 samples. To illustrate the usefulness of the resulting model, we considered summary completion, i.e., given a partial summary $\check{S} \subseteq S$ created from a human summary S, we sampled completions of that summary in an item-wise manner. In more detail: starting from a partial summary $\check{S}$, we computed the probability $Q_{\check{S}}(i)$ of adding the single element i to $\check{S}$, cf. §2. We then sampled an item according to $Q_{\check{S}}(i)$ and added it to $\check{S}$. We iterated this procedure until $\check{S}$ had the size of S. Illustrative results are shown in Figure 3b. One can observe that the proposed images are mainly complementary to the given images and that there is little redundancy within the completed summary. Furthermore, one can observe similarities of the images removed from the human summary and the images proposed by our model.

6 RELATED WORK

Probabilistic modeling with submodularity The study of Probabilistic Submodular Models (PSMs), i.e., Gibbs distributions associated with general submodular set functions, has been recently initiated by [8]. PSMs define distributions of the form $P(S) \propto \exp(+F(S))$ or $P(S) \propto \exp(-F(S))$ for some submodular function F, called *log-submodular* and *log-supermodular* respectively. While exact inference is intractable, both variational [8, 9] and sampling-based [12, 11] inference methods have been developed. Some special cases, such as the determinantal point process [18] or attractive and repulsive Ising models [20, 21], have been explored extensively. Another

(a) Image collection

(b) Completed summaries

Figure 3: Results for completing image summaries. (a) Whole image collection; (b) Summary completion results. The first row shows a partial human generated summary provided to our model. The row (H) shows the images removed from the human summary. The rows (1) to (4) show completions of the partial summary computed from our model. The proposed completions are complementary to the given images and every completion consists of diverse images. Every proposed completion has at least a one image overlap with the images removed from the human summary.

way of utilizing submodular functions to define distributions are *submodular point processes* (SPPs), introduced in [14]. Some of these models can be efficiently normalized, but in contrast to PSMs, they are not closed under multiplication, which makes it difficult to use them as priors in Bayesian models.

Learning submodular functions The problem of learning submodular functions has been considered in the literature, albeit only in the non-probabilistic setting. Goemans et al. in [22] and Balcan et al. in [23] analyze the setting of learning in a value oracle setting. Yue and Guestrin have considered submodular maximization in a bandit setting [24]. Several authors have proposed learning mixtures (nonnegative linear combinations) of submodular functions in a max-margin framework [17, 25].

Diversity models Models with diversity components have been used for many different problems. The problem of document summarization, i.e., the selection of sentences from a document that compactly represent the information in it, is one example that has received much attention, e.g., [26, 25]. The related problem of summarizing image collections has been also attacked [27, 17]. El-Arini et al. [28] use a coverage function to select relevant and diverse posts from the blogosphere. For probabilistically modelling diversity, the most well-known model is probably the determinantal point process (DPP) [18], cf. §4 for some more details.

7 CONCLUSIONS

In this paper, we proposed FLID, a novel log-submodular diversity model that easily scales to thousands of items. We showed how it can be efficiently learned using noise contrastive estimation. Our experiments demonstrate that FLID achieves state of the art performance in terms of model fit, while offering superior scalability (linear vs. cubic as is the case for DPPs). We believe that our results present an important step towards modeling and estimating complex, high-order probabilistic dependencies from data.

Acknowledgments. This research was supported in part by SNSF grant 200021 137528, ERC StG 307036, Microsoft Research Faculty Fellowship and a Google European Doctoral Fellowship.

A APPENDIX

A.1 Exact Computation of the Partition Function

In this section we provide an algorithm for computing the partition function of the model in (1), i.e.,

$$Z = \sum_{S \subseteq V} \exp(F(S)) \tag{5}$$

$$= \sum_{S \subseteq V} \exp\left(\sum_{i \in S} u_i + \sum_{d=1}^{L} \left(\max_{i \in S} w_{i,d} - \sum_{i \in S} w_{i,d} \right) \right). \tag{6}$$

The above equation can be rewritten by collapsing the modular terms $\sum_{i \in S} u_i$ and $-\sum_{i \in S} w_{i,d}$, i.e.,

$$Z = \sum_{S \subseteq V} \exp\left(\sum_{i \in S} u'_i + \sum_{d=1}^{L} \max_{i \in S} w_{i,d} \right), \tag{7}$$

where $u'_i = u_i - \sum_{d=1}^{L} w_{i,d}$.

The idea behind the algorithm for exactly computing the partition function is based on the following three observations[5]:

1. Although there are $2^{|V|}$ subsets for a ground set of size $|V|$, the expression $\sum_{d=1}^{L} \max_{i \in S} w_{i,d}$ can only take $\mathcal{O}(|V|^L)$ different values. For cases where $L \ll |V|$, $|V|^L$ can be substantially smaller than $2^{|V|}$.

2. Assuming we don't know S but only that $a_d = \max_{i \in S} w_{i,d}$, we can infer several properties of S. Firstly, $k_d \in S$ for $a_d = w_{k_d,d}$. Secondly, S must not contain any item j with $w_{j,d} > a_d$, i.e., $S \cap X^d_{a_d} = \emptyset$ where $X^d_{a_d} = \{j \colon w_{j,d} > a_d\}$. Thirdly, S can contain any item j with $w_{j,d} < a_d$, i.e., $(V \setminus \{k_1\}) \setminus (X^d_{a_d})$.

3. Assuming again that we don't know S but only the values $a_1, \ldots, a_L$ we can infer several properties of S. Firstly, $k_1, \ldots, k_L \in S$. Secondly, $S \cap (\cup_{d=1}^{L} X^d_{a_d}) = \emptyset$ (if this violates $k_1, \ldots, k_L \in S$ then the given values $a_1, \ldots, a_L$ can never be taken). Thirdly, S can contain the items $(V \setminus \{k_1, \ldots, k_L\}) \setminus (\cup_{d=1}^{L} X^d_{a_d})$.

Exploiting the above observations, we can rewrite the partition function as

$$Z = \sum_{k_1, \ldots, k_L} \Big[k_1, \ldots, k_L \text{ feasible} \Big] \exp\left(\sum_{d=1}^{L} (w_{k_d,d} + u'_{k_d}) \right) \cdot \sum_{S \subseteq (V \setminus \{k_1, \ldots, k_L\}) \setminus (\cup_{d=1}^{L} X^d_{a_d})} \exp\left(\sum_{i \in S} u'_i \right).$$

Note that the last sum above is easy to compute as $\exp(u'(S))$ factorizes over the items in S. Pseudocode for computing the above expression is given in Algorithm 1.

The time complexity of $\mathcal{O}(|V|^{L+1})$ now follows immediately from the algorithm by noting that for every d the set $\{w_{1,d}, \ldots, w_{|V|,d}\}$ can be sorted outside of the loop in time $|V| \log |V|$. Then, X and V' can be computed in time linear in $|V|$ and also the update of Z in line 12 is linear in $|V|$.

Algorithm 1 Exact computation of the partition function

Require: Ground set V, utilities $\mathbf{u} \in \mathbb{R}^{|V|}$, weights $\mathbf{W} \in \mathbb{R}^{|V| \times L}_{\geq 0}$

1: $u'_i \leftarrow u_i - \sum_{d=1}^{L} w_{id} \quad \forall i \in V$
2: $Z \leftarrow 0$
3: **for all** $(k_1, \ldots, k_L) \in V^L$ **do**
4: $\quad a_d \leftarrow w_{k_d,d} \quad \forall d$
5: $\quad I \leftarrow \cup_{d=1}^{L} \{k_i\}$
6: $\quad X \leftarrow \cup_{d=1}^{L} \{j \colon w_{j,d} > a_d\}$
7: $\quad$ **if** $I \cap X \neq \emptyset$ **then**
8: $\quad\quad$ continue
9: $\quad$ **end if**
10: $\quad Z \leftarrow Z + \exp(\sum_{d=1}^{L} (w_{k_d,d} + u'_{k_d}))$
11: $\quad V' \leftarrow (V \setminus \{k_1, \ldots, k_L\}) \setminus (\cup_{d=1}^{L} X^d_{a_d})$
12: $\quad Z \leftarrow Z + \prod_{i \in V'} (1 + \exp(u'_i))$
13: **end for**
14: **return** Z

[5]For simplicty assume that for all $d \in \{1, \ldots, L\}$ all values of $w_{i,d}$ are distinct, i.e., $|\{w_{i,d} \colon i \in V\}| = |V|$.

References

[1] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving Recommendation Lists Through Topic Diversification," in *International Conference on World Wide Web (WWW)*, pp. 22–32, ACM, 2005.

[2] J. Carbonell and J. Goldstein, "The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries," in *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336, ACM, 1998.

[3] A. Kulesza, *Learning with Determinantal Point Processes*. PhD thesis, University of Pennsylvania, 2012.

[4] A. Kulesza and B. Taskar, "Determinantal point processes for machine learning," *Foundations and Trends in Machine Learning*, vol. 5, no. 2–3, 2012.

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

[6] A. Kulesza and B. Taskar, "Diverse Sequential Subset Selection for Supervised Video Summarization," in *International Conference on Machine Learning (ICML)*, pp. 1193–1200, 2011.

[7] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse Sequential Subset Selection for Supervised Video Summarization," in *Neural Information Processing Systems (NIPS)*, pp. 2069–2077, 2014.

[8] J. Djolonga and A. Krause, "From MAP to Marginals: Variational Inference in Bayesian Submodular Models," in *Neural Information Processing Systems (NIPS)*, pp. 244–252, 2014.

[9] J. Djolonga and A. Krause, "Scalable variational inference in log-supermodular models," in *International Conference on Machine Learning (ICML)*, 2015.

[10] N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz, "A tight linear time (1/2)-approximation for unconstrained submodular maximization," in *Foundations of Computer Science (FOCS)*, pp. 649–658, IEEE, 2012.

[11] A. Gotovos, H. Hassani, and A. Krause, "Sampling from Probabilistic Submodular Models," in *Neural Information Processing Systems (NIPS)*, 2015.

[12] P. Rebeschini and A. Karbasi, "Fast Mixing for Discrete Point Processes," in *Conference on Learning Theory (COLT)*, pp. 1480–1500, 2015.

[13] M. U. Gutmann and A. Hyvärinen, "Noise-contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics," *Journal of Machine Learning Research*, vol. 13, pp. 307–361, Feb. 2012.

[14] R. Iyer and J. Bilmes, "Submodular Point Processes with Applications to Machine Learning," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

[15] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[16] J. Gillenwater, A. Kulesza, E. Fox, and B. Taskar, "Expectation-Maximization for Learning Determinantal Point Processes," in *Neural Information Processing Systems (NIPS)*, 2014.

[17] S. Tschiatschek, R. Iyer, H. Wei, and J. Bilmes, "Learning Mixtures of Submodular Functions for Image Collection Summarization," in *Neural Information Processing Systems (NIPS)*, 2014.

[18] A. Kulesza and B. Taskar, "Learning Determinantal Point Processes," in *Conference on Uncertainty in Artificial Intelligence*, 2011.

[19] Z. Mariet and S. Sra, "Fixed-point algorithms for learning determinantal point processes," in *International Conference on Machine Learning (ICML)*, pp. 2389–2397, 2015.

[20] M. Jerrum and A. Sinclair, "Polynomial-time approximation algorithms for the ising model," *SIAM Journal on computing*, vol. 22, no. 5, pp. 1087–1116, 1993.

[21] L. A. Goldberg and M. Jerrum, "The complexity of ferromagnetic ising with local fields," *Combinatorics, Probability and Computing*, vol. 16, no. 01, pp. 43–61, 2007.

[22] M. X. Goemans, N. J. Harvey, S. Iwata, and V. Mirrokni, "Approximating submodular functions everywhere," in *Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 535–544, Society for Industrial and Applied Mathematics, 2009.

[23] M.-F. Balcan and N. J. Harvey, "Learning submodular functions," in *Annual ACM symposium on Theory of Computing*, pp. 793–802, ACM, 2011.

[24] Y. Yue and C. Guestrin, "Linear submodular bandits and their application to diversified retrieval," in *Neural Information Processing Systems (NIPS)*, pp. 2483–2491, 2011.

[25] H. Lin and J. Bilmes, "Learning mixtures of submodular shells with application to document summarization," in *Uncertainty in Artificial Intelligence (UAI)*, AUAI, July 2012.

[26] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 510–520, Association for Computational Linguistics, 2011.

[27] I. Simon, N. Snavely, and S. M. Seitz, "Scene summarization for online image collections," in *International Conference on Computer Vision (ICCV)*, pp. 1–8, IEEE, 2007.

[28] K. El-Arini, G. Veda, D. Shahaf, and C. Guestrin, "Turning down the noise in the blogosphere," in *ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 289–298, ACM, 2009.