Low-Variance Black-Box Gradient Estimates for the Plackett-Luce Distribution

Artyom Gadetsky,[1*†] **Kirill Struminsky,**[1*] **Christopher Robinson,**[2]
Novi Quadrianto,[1,2] **Dmitry Vetrov**[1‡]
[1]National Research University Higher School of Economics
[2]Predictive Analytics Lab (PAL), University of Sussex

Abstract

Learning models with discrete latent variables using stochastic gradient descent remains a challenge due to the high variance of gradient estimates. Modern variance reduction techniques mostly consider categorical distributions and have limited applicability when the number of possible outcomes becomes large. In this work, we consider models with latent permutations and propose control variates for the Plackett-Luce distribution. In particular, the control variates allow us to optimize black-box functions over permutations using stochastic gradient descent. To illustrate the approach, we consider a variety of causal structure learning tasks for continuous and discrete data. We show that our method outperforms competitive relaxation-based optimization methods and is also applicable to non-differentiable score functions.

Introduction

The vast majority of modern machine learning advancements share one central method - gradient-based optimization. Stochastic gradients give a scalable solution for learning, applicable when the loss function is too slow to compute due to the size of data or even intractable. The latter is often the case when the loss function includes an expectation over random latent variables. The objectives of this kind naturally arise in multiple settings, including probabilistic latent variable models (Neal and Hinton 1998) and reinforcement learning (Williams 1992). Often the distribution of random variables also depends on the optimizable parameters of the loss function, which in turn makes gradient estimation harder and less reliable due to the high variance of stochastic gradients.

Despite the recent breakthroughs in gradient estimation for continuous latent variables (Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014; Mohamed et al. 2019), gradient estimation for discrete latent variables remains a challenge. Currently, general-purpose estimators (Williams 1992; Mnih and Gregor 2014) remain unreliable and the state-of-the-art methods (Tucker et al. 2017; Grathwohl et al. 2018; Yin and Zhou 2018) exclusively consider the categorical distribution. Although the reduction to the categorical case allows benefiting from gradient estimators for continuous relaxations, such solutions are hard to translate to discrete distributions with large support.

In this work, we consider a gradient estimator for the Plackett-Luce distribution, a distribution over permutations. Permutations naturally occur in various setting, such as ranking problems (Guiver and Snelson 2009), optimal routing (Bello et al. 2016) and causal inference (Friedman and Koller 2003). However, the support of the distribution is superexponential in the number of items k, which makes representing a distribution as a categorical distribution intractable even for dozens of items. At the same time, the Plackett-Luce distribution has $O(k)$ parameters and allows sampling in $O(k \log k)$.

We translate the recent variance reduction techniques (Tucker et al. 2017; Grathwohl et al. 2018) to the case of Plackett-Luce distributions. Similarly to REBAR, we use the difference of the REINFORCE estimator and the reparametrized estimator for the relaxed model. In particular, we derive the conditional marginalization step (Tucker et al. 2017) for the Plackett-Luce case. In our experiments, we recast causal inference tasks as a variational optimization over permutations and solve it using a gradient optimization method. We show that our method outperforms competitive relaxation-based approaches for optimization over permutations (Grover et al. 2019; Mena et al. 2018) for differentiable score functions and is applicable in a wider range of scenarios.

Our main contributions are the following:

- We derive a low-variance gradient estimator for the Plackett-Luce distribution.
- We apply the gradient estimator to solve variational optimization tasks for black-box functions and concentrate primarily on causal inference tasks for continuous and discrete data.
- For differentiable functions, we show that relaxation-based gradient optimization does not work out-of-the-box

*Both authors contributed equally to this work.
†Corresponding author. E-mail: artygadetsky@yandex.ru
‡Samsung-HSE Laboratory

for causal inference tasks and propose additional constraints to achieve competitive results.

A Brief Tour of Gradient Estimation

We consider a general optimization task $\min_\theta \mathbb{E}_{p(b|\theta)}[f(b)]$, where b is a discrete random variable parametrized by θ. The expectation can be intractable, for instance when b is a vector of categorical variables and the support of b is exponential in the vector length. The standard solution is to construct a stochastic estimate for the gradient $\hat{g}(f) := \frac{\partial}{\partial\theta}\mathbb{E}_{p(b|\theta)}[f(b)]$ without explicitly computing the expectation. In this section, we briefly review the gradient estimation algorithms.

REINFORCE

The REINFORCE estimator (Williams 1992) gives us a widely-applicable unbiased estimate for the gradient

$$\hat{g}_{\textit{REINFORCE}}(f) = f(b)\frac{\partial}{\partial\theta}\log p(b \mid \theta), \;\; b \sim p(b \mid \theta). \quad (1)$$

Although an unbiased gradient estimate is sufficient to guarantee convergence of stochastic gradient descent, in practice, the algorithm may not converge due to the high variance of the estimate (Tucker et al. 2017). The variance of the REINFORCE estimator can be reduced using control variates. A Control variate is a function $c(b)$ with a zero mean $\mathbb{E}_{p(b|\theta)}[c(b)] = 0$ that can be used to define another unbiased estimator

$$\hat{g}_{\textit{CV}}(f) = \hat{g}_{\textit{REINFORCE}}(f) - c(b). \quad (2)$$

The variance of the new estimator $\hat{g}_{\textit{CV}}(f)$ is lower than the variance of $\hat{g}_{\textit{REINFORCE}}(f)$ if $c(b)$ is positively correlated with the random variable $f(b)$. As an illustration, the gradient of probability $\frac{\partial}{\partial\theta}\log p(b \mid \theta)$ has zero mean, therefore it can be used as a control variate (Mnih and Gregor 2014).

Reparametrization Gradients for Continuous Relaxations

The reparametrization trick (Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014) is an alternative unbiased low-variance gradient estimator, applicable when f is differentiable and the latent variable $b_{\textit{cont}}$ is continuous. The estimator represents the latent variable as a differentiable determinisitc transformation $b_{\textit{cont}} = T(v, \theta)$ of a fixed distribution sample v and parameters θ and estimates the gradient as

$$\hat{g}_{\textit{reparam}}(f) = \frac{\partial}{\partial\theta}f(b_{\textit{cont}}) = \frac{\partial f}{\partial T}\frac{\partial T}{\partial\theta}, \quad (3)$$

$$v_i \sim \text{uniform}[0,1], \; i = 1, \ldots, k. \quad (4)$$

Although the reparametrization trick is not applicable when the latent variable b is discrete, (Jang, Gu, and Poole 2016; Maddison, Mnih, and Teh 2016) proposed the Gumbel-softmax estimator, a modification of the reparametrization trick for the relaxed categorical distribution.

To sample from a relaxed categorical distribution $p(b \mid \theta)$ with probabilities $\frac{\exp\theta_i}{\sum_j \exp\theta_j}$, Gumbel-Softmax first samples a vector of independent Gumbel random variables $z_i \sim \mathcal{G}(\theta_i, 1), i = 1, \ldots, k$

$$z_i = T(\theta_i, v_i) = \theta_i - \log(-\log(v_i)) \quad (5)$$

$$v_i \sim \text{uniform}[0,1], \; i = 1, \ldots, k \quad (6)$$

with location parameter θ. According to the **Gumbel-max trick** (Maddison, Tarlow, and Minka 2014), the index of the maximal element $H(z) = \arg\max(z)$ is a categorical random variable with distribution $p(b \mid \theta)$. Then, to make the sampler differentiable, the Gumbel-softmax trick replaces $\arg\max(z)$ with a relaxation $\text{soft}\max(z) = \frac{1}{\sum \exp z_i}(\exp z_1, \ldots, \exp z_k)$. The gradient estimate is the reparametrization gradient for the relaxed categorical distribution:

$$\hat{g}_{\textit{Gumbel}}(f) = \frac{\partial}{\partial\theta}f(b) = \frac{\partial f}{\partial b}\frac{\partial b}{\partial z}\frac{\partial z}{\partial\theta}, \quad (7)$$

$$b = \text{soft}\max(z), \quad (8)$$

$$z_i \sim \mathcal{G}(\theta_i, 1), i = 1, \ldots, k. \quad (9)$$

The resulting reparametrization gradient $\hat{g}_{\textit{Gumbel}}(f)$ has much lower variance than $\hat{g}_{\textit{REINFORCE}}(f)$, but is generally biased due to the relaxation.

Relaxation-based Control Variates

Recently, Tucker et al. (2017) and Grathwohl et al. (2018) proposed control variates for REINFORCE estimator based on the relaxed conditional distribution. Both works use the REINFORCE gradient estimator for the relaxed categorical distribution as a control variate for the non-relaxed estimator. To eliminate the bias of the REINFORCE estimator, they subtract the low-variance reparametrization gradient estimator.

The key insight of Tucker et al. (2017) is the conditional marginalization step used to correlate the non-relaxed REINFORCE estimator and the control variate. Importantly, the conditional marginalization relies on reparametrization trick for the conditional distribution $p(z \mid b, \theta)$, obtained from the joint distribution $p(b, z \mid \theta) = p(b|z)p(z \mid \theta)$ of the Gumbel random vector z and the output of the Gumbel-max trick $b = H(z) = \arg\max(z)$. Tucker et al. (2017) derive a reparametrizable sampling scheme for $p(z \mid b, \theta)$

$$\tilde{z}_i = \begin{cases} -\log(-\log v_i) & i = b \\ -\log\left(-\frac{\log v_i}{\exp\theta_i} + \exp(-\tilde{z}_b)\right) & i \neq b \end{cases}, \quad (10)$$

where vector v is a uniform i.i.d. vector $v \sim \text{uniform}[0,1]^k$. This gives a two-step generative process for the distribution $p(z \mid b, \theta)$. On the first step we sample the maximum variable v_b from the Gumbel distribtuion and on the second step we sample the other variables $v_i, i \neq b$ from the Gumbel distribution trunctated at $\tilde{z}_b$ with location parameter θ_i.

The unbiased RELAX estimator from Grathwohl et al. (2018) is

$$\hat{g}_{\textit{RELAX}}(f) = [f(b) - c_\phi(\tilde{z})]\frac{\partial}{\partial\theta}\log p(b \mid \theta) + \frac{\partial}{\partial\theta}c_\phi(z) - \frac{\partial}{\partial\theta}c_\phi(\tilde{z}) \quad (11)$$

$$b = H(z), \; z \sim p(z \mid \theta), \; \tilde{z} \sim p(z \mid b, \theta) \quad (12)$$

where $c_\phi(z)$ is a parametric function optimized to reduce the variance of the estimator.

Similarly, for a differentiable function f the REBAR estimator by Tucker et al. (2017) uses the function f with the relaxed argument $\text{soft}\max(z)$ and tunes the scalar parameter η

$$\begin{aligned}\hat{g}_{REBAR}(f) =&[f(b) - \eta f(\text{soft}\max(\tilde{z}))]\frac{\partial}{\partial\theta}\log p(b \mid \theta) \\ &+ \eta\frac{\partial}{\partial\theta}f(\text{soft}\max(z)) \\ &- \eta\frac{\partial}{\partial\theta}f(\text{soft}\max(\tilde{z}))\end{aligned} \tag{13}$$

$$b = H(z),\ z \sim p(z \mid \theta),\ \tilde{z} \sim p(z \mid b, \theta) \tag{14}$$

Constructing Control Variates for the Plackett-Luce Distribution

In this paper, we extend the stochastic gradient estimators $\hat{g}_{REBAR}(f)$ and $\hat{g}_{RELAX}(f)$ from the categorical distribution to the Plackett-Luce distribution. With a slight abuse of notation, below we use letter b to denote an integer vector $b = (b_1, \dots, b_k) \in S_k$ that represent a permutation, θ to denote the parameters of the Plackett-Luce distribution and $p(b \mid \theta)$ to denote the Plackett-Luce distribution.

The goal of this section is to define the two components required to apply the aforementioned gradient estimators: the mapping $b = H(z)$ and the two reparametrizable conditional distributions $p(z \mid \theta)$ and $p(z|b, \theta)$. After this we apply the estimators as defined in eq. 11 and eq. 13, but to emphasize the difference we refer to them as PL-RELAX and PL-REBAR.

Definition 1. *The Plackett-Luce distribution (Luce 2005; Plackett 1975) with scores $\theta = (\theta_1, \dots, \theta_k)$ is a distribution over permutations S_k with the probability of outcome $b \in S_k$*

$$p(b|\theta) = \prod_{j=1}^{k}\frac{\exp\theta_{b_j}}{\sum_{u=j}^{k}\exp\theta_{b_u}}. \tag{15}$$

Intuitively, a sample from the Plackett-Luce distribution $b = (b_1, \dots, b_k)$ is generated as a sequence of samples from categorical distributions. The first component b_1 comes from the categorical distribution with logits θ, then the second components b_2 comes from the categorical distribution with the logits θ without the component θ_{b_1} and so on.

The Plackett-Luce can be used for variational optimization (Staines and Barber 2012). Indeed, at the lower temperatures $\theta \to \frac{\theta}{T}, T \ll 1$ the distribution converges to a divergent distribution. The mode of the Plackett-Luce distribution is the descending order permutation of the scores $b^0 : \theta_{b_1^0} \geq \dots \geq \theta_{b_k^0}$, because b^0 permutation maximizes each factor in the product in eq. 15.

Now we will give an alternative definition of the Plackett-Luce distribution.

Lemma 1. *(appears in (Grover et al. 2019; Yellott Jr 1977)) Let z be a vector of k independent Gumbel random variables with location parameters specified by score vector θ*

$$z_i = \theta_i - \log(-\log(v_i)),\ v_i \sim \text{uniform}[0, 1]. \tag{16}$$

Then for a permutation $b \in S_k$ the probability of event $\{z_{b_1} \geq \dots \geq z_{b_k}\}$ is

$$p(z_{b_1} \geq \dots \geq z_{b_k}) = \prod_{j=1}^{k}\frac{\exp\theta_{b_j}}{\sum_{u=j}^{k}\exp\theta_{b_u}}. \tag{17}$$

Similarly to the **Gumbel-max trick**, Lemma 1 shows that an order of a Gumbel-distributed vector is distributed according to the Plackett-Luce distribution. Following the lemma, for Plackett-Luce distributions we define $p(z \mid \theta)$ to be a Gumbel-distributed vector and $H(z)$ to be a sorting operation

$$z_i \sim \mathcal{G}(\theta_i, 1),\ i = 1, \dots, k \tag{18}$$

$$H(z) = \arg\text{sort}(z) \tag{19}$$

Our principal discovery is that, similarly to the categorical case, the conditional distribution $p(z|b, \theta)$ factorizes into a sequence of truncated Gumbel distributions. As a consequence, the distribution is reparametrizable and can be used to construct a control variate for a gradient estimator.

Proposition 1. *Let $p(b, z \mid \theta)$ be the joint distribution with $z_i \sim \mathcal{G}(\theta_i, 1)$, $b = \arg\text{sort}(z)$ and normalized parameters $\sum_{j=1}^{k}\exp\theta_j = 1$. Then for uniform i.i.d samples $v_i \sim \text{uniform}[0, 1]$ and $\Theta_i = \sum_{j=i}^{k}\exp\theta_{b_j}$ for $i = 1, \dots, k$ the vector $\tilde{z} = (\tilde{z}_1, \dots, \tilde{z}_k)$*

$$\tilde{z}_{b_i} = \begin{cases} -\log(-\log v_i) & i = 1 \\ -\log(-\frac{\log v_i}{\Theta_i} + \exp(-\tilde{z}_{b_{i-1}})) & i \geq 2, \end{cases} \tag{20}$$

is a sample from the conditional distriubtion $p(z \mid b, \theta)$.

The proof of the proposition is given in the appendix.

The sampling procedure from Proposition 1 has two principal differences from the sampling scheme for the categorical case (see eq. 10). First, the truncation parameter $\tilde{z}_{b_{i-1}}$ now depends on the previous component $i - 1$, while for the categorical case the truncation parameter is defined by the maximum component. Second, the location parameter is now a cumulative sum and depends on the previous scores.

Related Work

Jang, Gu, and Poole; Maddison, Mnih, and Teh (2016; 2016) use the Gumbel distribution and Gumbel-max trick to define continuous relaxations of discrete distributions, by providing a gradient estimator which replaces the sampling of a categorical distribution with a differentiable sample from a Gumbel-Softmax distribution.

The Gumbel-Softmax distribution does not scale to permutations, as distribution over k-dimensional permutations is equivalent to that over $k!$ categories. Recently, a line of work proposed various for optimization over permutations. Linderman et al. (2018) relaxes the discrete set of permutations to Birkhoff polytope, the set of doubly-stochastic matrices, and extend stick-breaking approach (Sethuraman

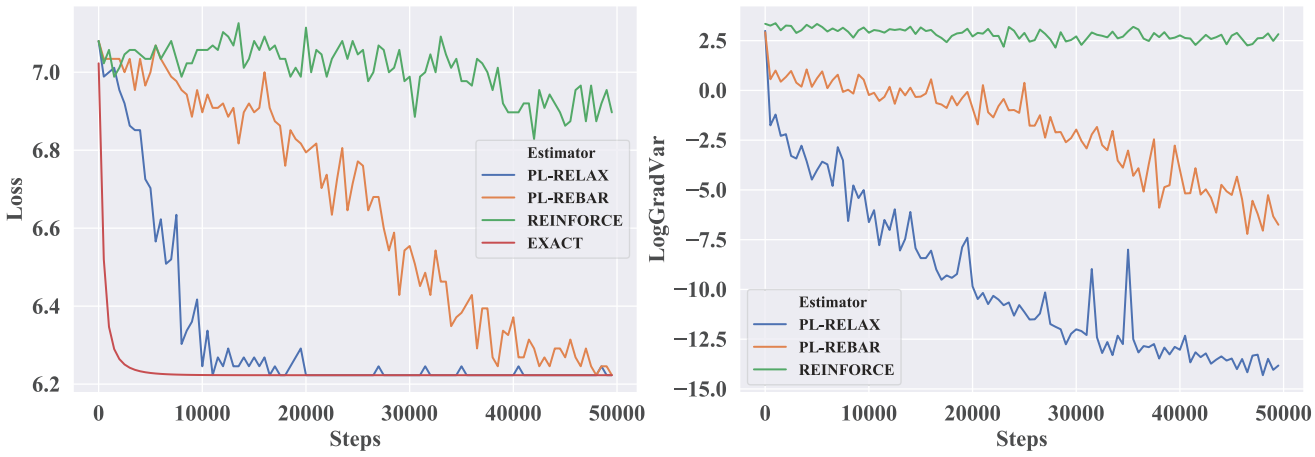

Figure 1: Training curves and log-variance of gradient estimators for different estimators on a toy problem: $\mathbb{E}_{p(b|\theta)}\|P_b - P_{0.05}\|_F^2$

1994) to satisfy polytope constraints. Mena et al. (2018) obtain doubly-stochastic matrices by applying the Sinkhorn operator. They use the Gumbel-Softmax distribution to define a distribution over latent matchings, the implicit Gumbel-Sinkhorn distribution. Grover et al. (2019) define new relaxation to the set of unimodal row-stochastic matrices, the set of matrices that have a unique maximal element in every row.

Grathwohl et al. (2018) extend Tucker et al. (2017) and derive control variate for black-box function optimization combining the REINFORCE estimator and reparametrization trick. Yin and Zhou (2018) propose gradient estimator that estimates the gradients of discrete distribution parameters in an augmented space.

For the special case of TSP, (Bello et al. 2016; Kool, van Hoof, and Welling 2018) introduce an amortized family of distributions over permutations using a deep autoregressive model and design control variates that exploit the structure of the loss function.

Experiments

We demonstrate the effectiveness of the proposed method with a simple toy task similar to Tucker et al. (2017) and then continue to the more challenging task of optimization over topological orderings for solving causal structure learning problems. Our PyTorch (Paszke et al. 2017) implementation of the gradient estimators is available at https://github.com/agadetsky/pytorch-pl-variance-reduction .

Toy Experiment

As a proof of concept we perform an experiment in minimizing $\mathbb{E}_{p(b|\theta)}\|P_b - P_t\|_F^2 = \mathbb{E}_{p(b|\theta)} f(P_b)$ as a function of θ where $p(b|\theta) = \text{Plackett-Luce}(b|\theta)$. P_b is permutation matrix with elements $p_{i,b_i} = 1$ and P_t is a matrix with $\frac{1}{k} + t$ on the main diagonal and $\frac{1}{k} - \frac{t}{k-1}$ in the remaining positions. This problem can be seen as linear sum assignment problem with specifically constructed doubly stochastic matrix P_t. It is easy to note that taking $k = 2$ and $t = 0.05$ leads to toy problem similar to that of Tucker et al. (2017). We focus on $t = 0.05$ and $k = 8$ to enable computation of exact gradients. For the PL-REBAR estimator we take $c_\phi(z) = \eta f(\sigma(z, \tau))$ where $\sigma(z, \tau)$ is the continuous relaxation of permutations described by Grover et al. (2019). For the PL-RELAX estimator we take $c_\phi(z) = f(\sigma(z, \tau)) + \rho_\phi(z)$ where $\rho_\phi(z)$ is a simple neural network with two linear layers and ReLU activation between them. Figure 1 shows the relative performance and gradient log-variance of REINFORCE, PL-REBAR and PL-RELAX. Although the REINFORCE estimator is unbiased, we can see that the variance of the estimator is too large even for the simple toy task, therefore the method is completely inapplicable for optimization over permutations. On the other hand, the proposed method significantly reduces variance of the gradient and thus converges to optimal. Also, similarly to the toy experiment from Grathwohl et al. (2018) paper, we observe better performance of the PL-RELAX estimator due to free-form control variate parameterized by a neural network.

Causal Structure Learning Through Order Search

Directed acyclic graph (DAG) models are popular tools for describing causal relationships and for guiding attempts to learn them from data. Learning the structure of a DAG remains challenging because of the combinatorial acyclicity constraint. A common way to model causal relations is a structural equation model (SEM). Let X be k-dimensional random variable, then relations are described as follows:

$$X_i = f_i(X_{pa(i)}, \varepsilon_i), \tag{21}$$

where $pa(i)$ is the set of parent vertices of variable X_i and ε_i is independent noise. Edge set $\{\cup_{i=1}^k \cup_{j \in pa(i)} j \to i\}$ describes DAG G on k vertices associated with joint distribution $\mathbb{P}_G(X) = \prod_{i=1}^k \mathbb{P}(X_i|pa(X_i))$. The basic structure learning problem can therefore be formulated as follows: let $\boldsymbol{X}$ be data matrix consisting of n i.i.d. samples of random

Table 1: Results for ER and SF graphs of 10 nodes

	ER1				ER4			
	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID
PL-RELAX	-0.2±1.7	5.2±2.5	5.8±3.2	13.0±9.6	12.2±26.3	29.4±1.9	35.0±4.4	67.0±1.8
SINKHORN$_{ECP}$	1.8±5.3	5.6±2.7	6.4±2.9	14.2±10.2	4.8±10.4	31.2±2.6	33.6±2.7	69.6±2.3
URS$_{ECP}$	13.5±26.9	7.4±3.7	7.4±3.6	16.0±8.9	12.4±6.2	29.8±3.6	32.8±4.9	67.4±2.1
SINKHORN	85.9±101.2	12.0±3.7	12.0±3.7	29.4±17.3	4019.6±3138.0	36.6±2.4	37.8±1.7	79.8±6.9
URS	71.4±128.9	10.8±2.9	11.0±3.2	26.0±10.5	1894.9±1704.8	34.6±2.2	36.8±2.8	74.4±2.7
GREEDY-SP	N/A	2.2±2.9	2.4±3.9	8.8±15.4	N/A	29.8±1.1	35.4±5.0	71.6±3.8
RANDOM	122.3±184.3	18.8±2.5	18.8±2.6	27.2±14.3	10078.2±10770.5	25.4±3.2	33.0±4.5	65.8±5.9
	SF1				SF4			
	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID
PL-RELAX	-0.7±0.3	2.2±1.5	2.4±1.5	2.6±2.2	-1.3±1.4	8.2±3.1	8.8±3.3	15.4±5.9
SINKHORN$_{ECP}$	0.6±2.9	2.8±3.2	3.0±3.2	7.6±12.3	-0.3±4.0	6.6±1.5	7.0±1.8	11.8±4.0
URS$_{ECP}$	1.6±1.8	5.0±1.7	5.4±2.2	7.0±2.2	2.1±2.3	12.8±2.5	13.4±2.2	24.8±5.6
SINKHORN	22.6±22.4	9.0±0.0	9.2±0.4	17.4±3.8	232.4±251.8	17.2±2.8	17.6±3.4	34.4±9.9
URS	10.1±5.2	9.6±1.2	9.6±1.2	14.6±2.1	69.6±81.6	14.6±1.4	14.6±1.2	29.2±5.8
GREEDY-SP	N/A	0.8±0.4	0.0±0.0	2.8±1.6	N/A	5.0±9.5	4.2±9.4	11.0±12.7
RANDOM	35.4±22.4	17.2±2.6	18.0±2.6	23.6±6.4	240.3±251.0	34.4±2.6	35.6±2.2	31.4±11.2

Table 2: Results for ER and SF graphs of 20 nodes

	ER1				ER4			
	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID
PL-RELAX	15.7±27.3	14.4±5.3	16.0±6.2	61.0±48.7	468.8±208.4	71.0±5.9	72.6±3.9	289.6±9.1
SINKHORN$_{ECP}$	10.4±8.7	15.8±4.7	17.0±6.0	84.8±56.3	2519.0±3715.2	78.0±6.1	78.8±5.5	302.2±15.8
URS$_{ECP}$	27.5±34.2	20.6±6.3	21.4±7.2	96.8±74.6	1011.4±745.5	75.8±2.9	76.6±2.9	300.2±20.3
SINKHORN	1651.2±3050.4	24.0±6.1	25.0±6.7	131.2±76.5	126284.6±194386.3	88.8±6.0	91.0±5.7	330.0±14.1
URS	1189.4±1815.5	26.4±8.4	26.6±8.6	134.2±75.0	7179677.6±7874489.3	93.0±3.8	94.4±4.5	328.0±11.5
GREEDY-SP	N/A	18.6±13.5	18.0±16.6	74.0±53.5	N/A	103.4±10.9	105.6±10.5	288.6±14.7
RANDOM	895.1±1270.3	37.8±5.2	38.8±4.9	146.8±79.9	109891.2±74968.7	113.0±4.9	114.4±4.1	330.6±9.2
	SF1				SF4			
	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID
PL-RELAX	-1.5±0.2	4.0±0.6	4.6±0.5	4.2±0.7	-5.8±1.2	20.0±4.3	20.0±4.1	48.4±16.2
SINKHORN$_{ECP}$	1.9±4.3	6.6±2.2	6.6±2.4	10.4±5.0	-0.4±2.4	25.6±5.6	25.8±5.9	58.6±19.7
URS$_{ECP}$	3.0±2.0	10.6±2.0	10.6±1.6	14.4±4.0	8.5±11.8	30.2±5.8	30.6±5.2	72.2±25.0
SINKHORN	38.3±26.2	19.0±0.0	19.0±0.0	35.0±2.4	158.2±99.9	44.6±5.8	44.8±6.1	103.6±20.8
URS	38.3±26.2	19.0±0.0	19.0±0.0	35.0±2.4	140.7±140.6	42.0±5.4	42.8±5.1	89.8±20.4
GREEDY-SP	N/A	2.0±1.4	0.0±0.0	7.0±5.1	N/A	50.6±31.5	49.8±32.3	69.0±43.2
RANDOM	94.0±36.4	36.2±2.6	36.6±2.3	48.6±14.7	635.5±182.6	98.2±6.1	99.2±5.5	168.8±29.6

Table 3: Results for ER and SF graphs of 50 nodes

	ER1				ER4			
	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID
PL-RELAX	-1.8±1.3	19.2±6.9	20.6±7.8	103.2±55.5	1863.1±1703.2	220.6±42.8	221.4±43.5	1779.6±193.1
SINKHORN$_{ECP}$	5.5±7.0	30.0±6.3	30.8±5.8	151.8±35.1	43463.9±70904.3	221.0±14.7	223.2±15.2	1846.4±158.3
URS$_{ECP}$	10.3±4.7	41.0±2.4	40.0±2.7	177.6±17.1	22997.9±38346.1	239.4±31.6	240.2±31.5	1789.8±154.4
SINKHORN	90.3±35.8	49.6±4.3	49.6±4.3	275.0±42.5	231304.8±290019.0	248.6±18.5	250.4±19.1	1966.8±135.5
URS	90.3±35.8	49.6±4.3	49.6±4.3	275.0±42.5	546793216.7±984510739.7	320.2±26.8	320.8±27.1	2119.0±130.5
GREEDY-SP	N/A	38.2±21.6	38.2±24.6	151.6±84.3	N/A	525.6±35.5	526.8±34.7	1951.4±50.3
RANDOM	271.0±71.6	99.4±9.3	99.8±9.5	301.2±60.4	477442.0±661243.9	360.8±23.5	361.0±23.2	2175.0±52.6
	SF1				SF4			
	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID
PL-RELAX	-3.9±0.5	11.4±3.3	11.8±2.9	14.4±2.7	-1.1±7.6	70.0±9.9	70.6±11.2	219.0±20.3
SINKHORN$_{ECP}$	25.1±18.2	28.6±6.5	28.4±6.1	58.4±12.1	124.3±126.0	94.4±22.7	95.6±23.0	257.2±25.8
URS$_{ECP}$	32.1±44.3	33.4±10.2	33.6±10.7	55.6±32.7	164.4±53.1	110.6±12.8	111.4±13.7	319.6±18.1
SINKHORN	138.2±68.2	49.0±0.0	49.0±0.0	110.6±5.5	10238.2±15850.1	139.0±8.3	139.6±8.1	387.0±37.2
URS	138.2±68.2	49.0±0.0	49.0±0.0	110.6±5.5	7966.9±4838.0	142.8±11.8	144.2±12.1	527.4±86.8
GREEDY-SP	N/A	38.8±39.3	35.4±39.6	54.8±20.6	N/A	381.2±76.2	384.2±77.0	963.0±475.7
RANDOM	380.1±207.8	97.8±7.3	97.8±7.3	155.4±31.2	10109.8±2027.0	312.0±14.9	312.4±15.0	807.0±101.7

variable X. Also let $\mathbb{D}$ be space of DAGs. Then, given observations $\boldsymbol{X}$ the task is to find DAG $G \in \mathbb{D}$ or so-called Bayesian Network for joint distribution $\mathbb{P}(X)$:

$$\min_{G \in \mathbb{D}} Q(G, \boldsymbol{X}) \tag{22}$$

where Q is function that scores DAG G given data.

To incorporate permutations in the objective (22) we consider parametrization of DAG adjacency matrix using nilpotent matrices which are upper triangular in basis induced by topological ordering, namely $W_G = PAP^T$ where A is strictly upper triangular adjacency matrix which describes parent sets of variables and permutation matrix P which describes topological ordering. Then optimization over DAGs can thus be seen as an optimization over topological orderings

$$\min_{P \in \mathcal{P}_k} \widehat{Q}(P, \boldsymbol{X}), \tag{23}$$

where $\widehat{Q}$ scores topological ordering P and $\mathcal{P}_k$ is the set of permutation matrices of size k. Optimization over A is usually hidden in the computation of $\widehat{Q}$. It is worth noting that this approach is similar to order MCMC (Friedman and Koller 2003), however our work considers gradient-based optimization over permutations matrices rather than discrete order changes.

Continuous data We consider linear additive noise SEMs:

$$X = W^T X + \varepsilon \tag{24}$$

where $W = PAP^T$ and non-zero elements of A describe linear coefficients and parent sets for each variable X_i. As score function $\widehat{Q}$ we take regularized mean squared loss combined with sparsity-inducing $L1$ regularization term

$$\widehat{Q}(P, \boldsymbol{X}) = \min_{A \in \mathbb{A}} \frac{1}{2n} \|\boldsymbol{X} - PAP^T \boldsymbol{X}\|_F^2 + \lambda \|\operatorname{vec}(A)\|_1, \tag{25}$$

where $\mathbb{A}$ is the set of strictly upper triangular matrices. Computing $\widehat{Q}$ itself involves optimization problem, which can be efficiently solved using accelerated proximal gradient for convex composite function optimization (Nesterov 2013). To apply the proposed method, we reformulate (23) as variational optimization with respect to parameters of a Plackett-Luce distribution:

$$\min_{\theta} \mathop{\mathbb{E}}_{p(b|\theta)} \widehat{Q}(P_b, \boldsymbol{X}) \tag{26}$$

where $p(b|\theta) = \text{Plackett-Luce}(b|\theta)$, and P_b is a permutation matrix with $p_{i,b_i} = 1$. For variational optimization, we only apply PL-RELAX and treat $\widehat{Q}(P, \boldsymbol{X})$ as a black-box function to avoid unrolling the optimizer to compute gradients.

As a concurrent approach, we consider work by Mena et al. (2018) which proposes relaxing optimization over a set of permutations to a set of doubly-stochastic matrices using the Sinkhorn operator. Another recent work by Grover et al. (2019) proposes relaxation to the set of unimodal row-stochastic matrices (URS) which intersects the set of doubly-stochastic matrices and contains the set of all permutation matrices. Since these methods can't be used to optimize black-box functions we reformulate (26) as:

$$\min_{\phi} \min_{A \in \mathbb{A}} \frac{1}{2n} \|\boldsymbol{X} - P(\phi) A P(\phi)^T \boldsymbol{X}\|_F^2 + \lambda \|\operatorname{vec}(A)\|_1 \tag{27}$$

where ϕ are the parameters of the corresponding relaxation. We optimize (27) coordinate-wise using gradient descent with respect to ϕ and accelerated proximal gradient optimization with respect to A. We refer to the optimization of this objective as SINKHORN or URS according to the used relaxation.

We also try an alternative approach for the above relaxations. Since $P(\phi)$ is not a permutation matrix during training we extend (27) with an orthogonality constraint and replace $\|vec(A)\|$ with $H_\mu(vec(PAP^T))$ where H_μ is the Huber relaxation of $L1$ norm and μ is a hyperparameter controlling tightness of relaxation:

$$\begin{aligned} \min_{\phi} \min_{A \in \mathbb{A}} \quad & \frac{1}{2n} \|\boldsymbol{X} - P(\phi) A P(\phi)^T \boldsymbol{X}\|_F^2 + \\ & + \lambda H_\mu(\operatorname{vec}(P(\phi) A P^T(\phi))) \\ \text{s.t.} \quad & \|P(\phi) P^T(\phi) - I_k\|_F^2 = 0 \end{aligned} \tag{28}$$

We use an Augmented Lagrangian (Nemirovski 1999) to solve this equality constrained optimization problem (ECP) (28) and refer to the solutions as SINKHORN$_{ECP}$ or URS$_{ECP}$ correspondingly.

We simulated graphs from two well-known random graph models with different degree distributions: Erdos-Renyi random graphs and Scale-free networks with *k* and *4 k* expected number of edges, denoted by ER1, ER4, SF1, SF4 respectively. Given a random acyclic graph we assigned edge weights independently from $U([-2; -0.5] \cup [0.5; 2])$ to obtain weight matrix W. To generate data matrix $\boldsymbol{X}$ we follow generating process of linear SEM (24) with standard Gaussian noise.

As a sanity check, we also introduce a simple baseline. We generate Erdos-Renyi random graphs with the corresponding expected number of edges and refer to it as RANDOM baseline. For comparison, we also include the Greedy Sparse Permutation (Greedy-SP) algorithm (Solus et al. 2017). This algorithm casts DAG structure learning as a linear programming problem with graph sparsity as the linear objective function, and a sub-polytope of the permutohedron as the feasible region. Whilst this algorithm also searches permutations as a proxy to DAGs to reduce the size of the search space, it is in essence a constraint-based method - rather than optimising a DAG score function, it searches for the sparsest DAG which satisfies the conditional independence relations found. Conversely, our gradient-based method does not rely on these conditional independence tests, which typically require the simplifying assumptions of CI tests, and is able to use linear as well as non-linear objective functions (e.g. in the discrete data experiment, the quotient normalized maximum likelihood score is non-linear and non-differentiable).

For each method we report the score difference $\widehat{Q}$ (25) between learned and ground truth DAGs on additionally generated validation samples $\boldsymbol{X}_{val}$, as well as three DAG met-

rics from causal inference literature. The quoted score difference shows the effectiveness of our method for optimizing the chosen score function, while the DAG metrics show how well it performs on the problem itself. *Structural hamming distance* (SHD) is the number of edge additions, removals, and reversals required to get from the learned structure to the ground truth. Multiple DAGs can represent the same set of conditional independence relations, forming a Markov equivalence class; this can be represented by a completed partially directed acyclic graph (CPDAG). We also report SHD-CPDAG - the SHD between the CPDAG the learned structure belongs to and that of the true structure. *Structural interventional distance* (SID) (Peters and Bühlmann 2013) quantifies the distance between two DAGs in terms of their respective causal inference statements. This gives an indication of accuracy of computed interventions using the learned graph.

We consider graphs of 10, 20 and 50 nodes. For PL-RELAX we take the mode of the distribution after training. For SINKHORN relaxation we apply the Hungarian algorithm to find the closest permutation matrix. For URS we use the argmax permutation property to obtain the permutation matrix. Regularization coefficient λ is set to 0.5 for all methods.

Tables 1-3 show the performance of all methods for varying number of nodes k averaged across 5 random seeds (the error ranges represent standard deviation). We can see that the proposed method outperforms baselines in the majority of settings. Also, it is worth mentioning that SINKHORN and URS perform poorly in terms of score function values due to the fact that the optimization is carried out over the set of relaxed matrices. This leads to deterioration in score value $\widehat{Q}$ when relaxation is transformed to permutation. As we can see there is no such problem with ECP versions of relaxations, though they perform worse than PL-RELAX and require additional constrained optimization techniques to be applied. Also, one more observation should be explained: PL-RELAX almost always ends up with better solutions in terms of score function than the ground truth DAG, therefore solves the optimization problem well. However, it is not ideal in terms of metrics. Peters and Bühlman (2014) proved that given enough data, it is possible to identify the ground truth DAG if data was generated from linear SEM with Gaussian homogeneous noise. Authors used $L0$-regularized mean squared error score function, but it is non-convex and hard to optimize, therefore $L1$-regularization is used in practice. Because of relaxation of the $L0$ norm and finite amount of data all guarantees vanish, and we observe inconsistency between the metrics of interest and values of the surrogate score function $\widehat{Q}$.

Discrete data Due to the discrete and nonlinear nature of categorical data, it cannot be modeled with the SEM defined previously. Discrete variable networks can however be modeled as generated by sampling each node's conditional probability table, depending only on the configuration of its parent nodes. In the standard general form this is $X_i = f_i(X_{pa(i)})$, where f_i is assumed to be multinomial, thus

$$f_i(X_{pa(i)}) \sim Multinomial(\Theta_{X_i}|Pa(X_i))$$

where $\Theta_{X_i}|Pa(X_i)$ are the conditional probabilities $\theta_{i,j,k} = P(X_i = k|Pa(X_i) = j)$.

Rather than learning the optimal A for a given P by minimising a training loss, we can therefore instead try to maximise the marginal likelihood based on the above model

$$Q(P, \mathbf{X}) = \max_{A \in \mathbb{A}} P(\mathbf{X}|A, P) \tag{29}$$

which can be found using the factorisation

$$P(\mathbf{X}|A, P) = \prod_{i=1}^{d} \prod_{j=1}^{q_i} P(\mathbf{X}_{i,pa(i)=j}; \alpha). \tag{30}$$

As a result of the decomposition of the score by node in equation (30), the *maximum a posteriori* (MAP) parent set can be selected from the set of parents permitted by the topological ordering for each node, independently of the rest. Due to the ordering, the graph resulting from combining each of these MAP parent connections is guaranteed to be acyclic, thus the exact MAP DAG for a given ordering can be found. Due to the combinatorial size of even this reduced search space, the set of permitted parents for a given node is reduced further, to only those that cannot be easily proven to be conditionally independent - as determined by a standard constraint-based method (in this case the PC-stable algorithm (Colombo and Maathuis 2014)). As this finds the exact solution for a reduced search space, the result is an approximation of the best score possible for the ordering. Whilst this provides an approximate score for any given order, it is a non-differentiable black-box function; therefore whilst our method can be applied to this permutation optimization, options are severely limited - the SINKHORN and URS methods used for continuous SEM graph benchmarks for example cannot be used. For a simple evaluation, Table 4 shows the result of our method on data sampled from the standard ALARM network compared against random orders, and permutations optimized by order MCMC (Friedman and Koller 2003), all using the same MAP DAG method described above, maximizing the quotient normalized maximum likelihood score (Silander et al. 2018). Higher Val $\widehat{Q} - \widehat{Q}^*$ is better, other metrics lower is better. Whilst Table 4 shows our algorithm to be less effective than MCMC for this task, the comparison is not particularly favorable - MCMC is performed directly on permutations, rather than attempting to learn the Plackett-Luce distribution over permutations - thus the MCMC simply attempts to find a good local minimum in the score space. To give a lower bound to performance, we also compare to the MAP DAGs of 1000 random permutations, computed in the same way as for MCMC and our algorithm, showing sampling the learned Plackett-Luce distribution gives permutations far better than random.

Conclusion

In this work we proposed a gradient-based optimization method, with unique capabilities for application to Plackett-Luce distributions over permutations. A proof of concept

Table 4: Results for ALARM graph (37 nodes)

	Val $\widehat{Q} - \widehat{Q}^*$	SHD	SHD-CPDAG	SID
PL-RELAX	-15645.2±3255.8	14.6±1.7	19.0±2.3	214.2±31.8
SINKHORN		N/A		
URS		N/A		
ORDER MCMC	-13404.7±2224.6	8.6±1.1	10.6±0.5	104.4±20.8
RANDOM	-75022.7±9647.7	25.8±3.7	30.0±3.9	478.8±70.8

experiment shows our method outperforms existing methods for differentiable objective functions, whilst also generalizing to non-differentiable black-box functions, and being applicable to permutation learning despite the factorial complexity. This allowed us to extend Plackett-Luce distribution based causal graphical model structure learning beyond the simple SEM based methods, to the more general case of DAGs of arbitrary variable types.

In future, our method could be combined with other standard scoring functions from Bayesian network literature - providing they decompose as described in equation (30) - for DAG structure learning of continuous data from different model types. Other potential applications include approximate inference for probabilistic models with latent permutations, routing problems and combinatorial problems for permutations.

Acknowledgements

This work was partly supported by Sberbank AI Lab and UK EPSRC project EP/P03442X/1. Kirill Struminsky proved proposition 1 and was supported by the Russian Science Foundation grant no. 19-71-30020. The authors thank NRU HSE for providing computational resources, NVIDIA for GPU donations, and Amazon for AWS Cloud Credits.

Appendix

We prove Proposition 1 in this section. We first discuss the properties of Gumbel distribution. Then we discuss the generative processes for the densities used for $p(z \mid b, \theta)$ in Eq. 20. Then we show that $p(b \mid z)p(z \mid \theta) = p(b \mid \theta)p(z \mid b, \theta)$ for the unconditional Gumbel density $p(z \mid \theta)$ and the Plackett-Luce distribution $p(b \mid \theta)$.

Density for the Gumbel distribution and the truncated Gumbel distribution

The density function of the Gumbel distribution with location parameter μ is

$$\phi_\mu(z) = \exp(-z + \mu)\exp(-\exp(-z + \mu)) \tag{31}$$

and the cumulative density function is

$$\Phi_\mu = \exp(-\exp(-z + \mu)). \tag{32}$$

Our derivation of the conditional distribution $p(b \mid z, \theta)$ relies on the additive property of the cumulative density function of the Gumbel distribution

$$\Phi_{\log(\exp\mu + \exp\nu)}(z) = \exp(-\exp(z)(\exp\mu + \exp\nu)) = \Phi_\mu(z)\Phi_\nu(z), \tag{33}$$

which we enfold in the following auxiliary claim.

Lemma 2. *For permutation $b \in S_k$, score vector $\theta \in \mathbb{R}^k$ and $i = 1, \dots, k$ and the argument vector $z \in \mathbb{R}^k$ we have*

$$\phi_{\theta_{b_i}}(z_{b_i})\Phi_{\log(\sum_{j=i+1}^k \exp\theta_{b_j})}(z_{b_i}) \tag{34}$$

$$= \frac{\exp\theta_{b_i}}{\sum_{j=i}^k \exp\theta_{b_j}}\phi_{\log(\sum_{j=i}^k \exp\theta_{b_j})}(z_{b_i}). \tag{35}$$

Proof. For brevity, we denote $\exp\theta_i$ as p_i. We then rewrite the density $\phi_{\log p_{b_i}}(z_{b_i})$ through the exponent $\exp(-z_{b_i} + \log p_{b_i})$ and c.d.f. $\Phi_{\log p_{b_i}}(z_{b_i})$ and apply the additive property in Eq. 38:

$$\phi_{\log p_{b_i}}(z_{b_i})\Phi_{\log(\sum_{j=i+1}^k p_{b_j})}(z_{b_i}) \tag{36}$$

$$= p_{b_i}\exp(-z_{b_i})\Phi_{\log p_{b_i}}(z_{b_i})\Phi_{\log(\sum_{j=i+1}^k p_{b_j})}(z_{b_i}) \tag{37}$$

$$= p_{b_i}\exp(-z_{b_i})\Phi_{\log(\sum_{j=i}^k p_{b_j})}(z_{b_i}) \tag{38}$$

$$= p_{b_i}\frac{\sum_{j=i}^k p_{b_j}}{\sum_{j=i}^k p_{b_j}}\exp(-z_{b_i})\Phi_{\log(\sum_{j=i}^k p_{b_j})}(z_{b_i}) \tag{39}$$

$$= \frac{p_{b_i}}{\sum_{j=1}^k p_{b_j}}\phi_{\log(\sum_{j=i}^k p_{b_j})}(z_{b_i}). \tag{40}$$

The last step collapses the exponent and the c.d.f. into the density function $\phi_{\log(\sum_{j=i}^k p_{b_j})}(z_{b_i})$. □

Finally, to define the density of conditional distribution $p(b \mid z, \theta)$ we define the density of the truncated Gumbel distribution $\phi_\mu^{z_0}(z) \propto \phi_\mu(z)I[z \le z_0]$:

$$\phi_\mu^{z_0}(z) = \frac{\phi_\mu(z)}{\Phi_\mu(z_0)}(z)I[z \le z_0], \tag{41}$$

where the superscript z_0 denotes the truncation parameter.

Reparametrization for the Gumbel distribution and the truncated Gumbel distribution

The reparametrization trick requires representing a draw from a distribution as a deterministic transformation of a fixed distribution sample and a distribution parameter. For a sample z from the Gumbel distribution $\mathcal{G}(\mu, 1)$ with location parameter μ the representation is

$$z = \mu - \log(-\log v),\ v \sim \text{uniform}[0, 1]. \tag{42}$$

For the Gumbel distribution truncated at z_0 (Maddison, Tarlow, and Minka 2014) proposed an analogous representation

$$z = \mu - \log(-\log v + \exp(-z_0 + \mu))$$

$$= -\log\left(-\frac{\log v}{\exp\mu} + \exp(-z_0)\right) \tag{43}$$

$$v \sim \text{uniform}[0, 1]. \tag{44}$$

In particular, the sampling schemes in Eq. 10 and Eq. 20 generate samples from the truncated Gumbel distribution.

The derivation of the conditional distribution

We now derive the conditional distribution and the sampling scheme defined in Proposition 1.

The joint distribution of the permutation b and the Gumbel samples z is

$$p(b, z \mid \theta) = p(b \mid z)p(z \mid \theta) \tag{45}$$

$$= \phi_{\theta_{b_1}}(z_{b_1}) \prod_{i=2}^{k} \left(\phi_{\theta_{b_i}}(z_{b_i}) I[z_{b_{i-1}} \geq z_{b_i}] \right) \tag{46}$$

We first multiply and divide the joint density by the c.d.f. $\Phi_{\log(\sum_{i=2}^{k} \exp \theta_{b_i})}(z_{b_1})$ and apply Lemma 2

$$\frac{\Phi_{\log(\sum_{i=2}^{k} \exp \theta_{b_i})}(z_{b_1})}{\Phi_{\log(\sum_{i=2}^{k} \exp \theta_{b_i})}(z_{b_1})} \phi_{\theta_{b_1}}(z_{b_1}) \prod_{i=2}^{k} \dots \tag{47}$$

$$= \frac{\exp \theta_{b_1}}{\sum_{i=1}^{k} \exp \theta_{b_i}} \frac{\phi_{\log(\sum_{i=1}^{k} \exp \theta_{b_i})}(z_{b_1})}{\Phi_{\log(\sum_{i=2}^{k} \exp \theta_{b_i})}(z_{b_1})} \prod_{i=2}^{k} \dots . \tag{48}$$

Next, we apply Lemma 2 to combine the c.d.f. in the denominator $\Phi_{\log(\sum_{j=i}^{k} \exp \theta_{b_j})}(z_{b_{i-1}})$ and the term $\phi_{\theta_{b_i}}(z_{b_i}) I[z_{b_{i-1}} \geq z_{b_i}]$ inside the product

$$\frac{\phi_{\theta_{b_i}}(z_{b_i}) I[z_{b_{i-1}} \geq z_{b_i}]}{\Phi_{\log(\sum_{j=i}^{k} \exp \theta_{b_j})}(z_{b_{i-1}})} \tag{49}$$

$$= \frac{\phi_{\theta_{b_i}}(z_{b_i}) I[z_{b_{i-1}} \geq z_{b_i}]}{\Phi_{\log(\sum_{j=i}^{k} \exp \theta_{b_j})}(z_{b_{i-1}})} \frac{\Phi_{\log(\sum_{j=i+1}^{k} \exp \theta_{b_j})}(z_{b_i})}{\Phi_{\log(\sum_{j=i+1}^{k} \exp \theta_{b_j})}(z_{b_i})} \tag{50}$$

$$= \frac{\exp \theta_{b_i}}{\sum_{j=i}^{k} \exp \theta_{b_j}} \frac{\phi^{z_{b_{i-1}}}_{\log(\sum_{j=i}^{k} \exp \theta_{b_j})}(z_{b_i})}{\Phi_{\log(\sum_{j=i+1}^{k} \exp \theta_{b_j})}(z_{b_i})} \tag{51}$$

and obtain the truncated distribution $\phi^{z_{b_{i-1}}}_{\log(\sum_{j=i}^{k} \exp \theta_{b_j})}(z_{b_i})$ along with one factor of the Plackett-Luce probability $\frac{\exp \theta_{b_i}}{\sum_{j=i}^{k} \exp \theta_{b_j}}$. Also, after the transformation the summation index in the denominator c.d.f. changes from i to $i+1$. This gives us an induction step that we apply sequentially for $i = 2, \dots, k-1$. For $i = k$ the denominator c.d.f. $\Phi_{\log \exp \theta_k}(z_{b_{k-1}})$ and the product term $\phi_{\log \exp \theta_k}(z_{b_k}) I[z_{k-1} \geq z_k]$ combine into the truncated Gumbel distribution with density $\phi^{z_{b_{k-1}}}_{\log \exp \theta_k}(z_{b_k})$.

As a result, we rearrange $p(b, z \mid \theta)$ into the product of the truncated Gumbel distribution densities $p(z \mid b, \theta)$ and the probability of the Plackett-Luce distribution $p(b \mid \theta)$:

$$\prod_{i=1}^{k} \frac{\exp \theta_{b_i}}{\sum_{j=i}^{k} \exp \theta_{b_j}} \left(\phi_0(z_{b_1}) \prod_{i=2}^{k} \phi^{z_{b_{i-1}}}_{\log \sum_{j=i}^{k} \exp \theta_j}(z_{b_i}) \right). \tag{52}$$

Finally, to obtain the claim of Proposition 1 we apply the reparametrized sampling scheme defined in Eq. 43.

References

Bello, I.; Pham, H.; Le, Q. V.; Norouzi, M.; and Bengio, S. 2016. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*.

Colombo, D., and Maathuis, M. H. 2014. Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research* 15(1):3741–3782.

Friedman, N., and Koller, D. 2003. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine learning* 50(1-2):95–125.

Grathwohl, W.; Choi, D.; Wu, Y.; Roeder, G.; and Duvenaud, D. 2018. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *International Conference on Learning Representations*.

Grover, A.; Wang, E.; Zweig, A.; and Ermon, S. 2019. Stochastic optimization of sorting networks via continuous relaxations. In *International Conference on Learning Representations*.

Guiver, J., and Snelson, E. 2009. Bayesian inference for plackett-luce ranking models. In *proceedings of the 26th annual international conference on machine learning*, 377–384. ACM.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kool, W.; van Hoof, H.; and Welling, M. 2018. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*.

Linderman, S.; Mena, G.; Cooper, H.; Paninski, L.; and Cunningham, J. 2018. Reparameterizing the birkhoff polytope for variational permutation inference. In Storkey, A., and Perez-Cruz, F., eds., *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, 1618–1627. Playa Blanca, Lanzarote, Canary Islands: PMLR.

Luce, R. D. 2005. *Individual Choice Behavior: A Theoretical Analysis*. Courier Corporation.

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Maddison, C. J.; Tarlow, D.; and Minka, T. 2014. A* sampling. In *Advances in Neural Information Processing Systems*, 3086–3094.

Mena, G.; Belanger, D.; Linderman, S.; and Snoek, J. 2018. Learning latent permutations with gumbel-sinkhorn networks. In *International Conference on Learning Representations*.

Mnih, A., and Gregor, K. 2014. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*.

Mohamed, S.; Rosca, M.; Figurnov, M.; and Mnih, A. 2019. Monte carlo gradient estimation in machine learning. *arXiv preprint arXiv:1906.10652*.

Neal, R. M., and Hinton, G. E. 1998. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*. Springer. 355–368.

Nemirovski, A. 1999. Optimization ii: Standard numerical methods for nonlinear continuous optimization. *Lecture notes*.

Nesterov, Y. 2013. Gradient methods for minimizing composite functions. *Mathematical Programming* 140(1):125–161.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

Peters, J., and Bühlman, P. 2014. Identifiability of Gaussian structural equation models with equal error variances. *Biometrika* 101(1):219–228.

Peters, J., and Bühlmann, P. 2013. Structural intervention distance (sid) for evaluating causal graphs. *arXiv preprint arXiv:1306.1043*.

Plackett, R. L. 1975. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 24(2):193–202.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

Sethuraman, J. 1994. A constructive definition of dirichlet priors. *Statistica sinica* 639–650.

Silander, T.; Leppä-aho, J.; Jääsaari, E.; and Roos, T. 2018. Quotient normalized maximum likelihood criterion for learning bayesian network structures. In Storkey, A., and Perez-Cruz, F., eds., *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, 948–957. Playa Blanca, Lanzarote, Canary Islands: PMLR.

Solus, L.; Wang, Y.; Matejovicova, L.; and Uhler, C. 2017. Consistency guarantees for permutation-based causal inference algorithms.

Staines, J., and Barber, D. 2012. Variational optimization. *arXiv preprint arXiv:1212.4507*.

Tucker, G.; Mnih, A.; Maddison, C. J.; Lawson, J.; and Sohl-Dickstein, J. 2017. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc. 2627–2636.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Yellott Jr, J. I. 1977. The relationship between luce's choice axiom, thurstone's theory of comparative judgment, and the double exponential distribution. *Journal of Mathematical Psychology* 15(2):109–144.

Yin, M., and Zhou, M. 2018. Arm: Augment-reinforce-merge gradient for discrete latent variable models. *arXiv preprint arXiv:1807.11143*.