Learning Latent Permutations with Gumbel-Sinkhorn Networks

Gonzalo Mena, David Belanger, Scott Linderman, Jasper Snoek

2021.10.14

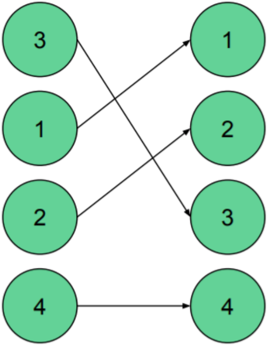
Outlines

1. Introduction
2. Sinkhorn Operator
3. Sinkhorn Networks
4. Experiments
5. Summary

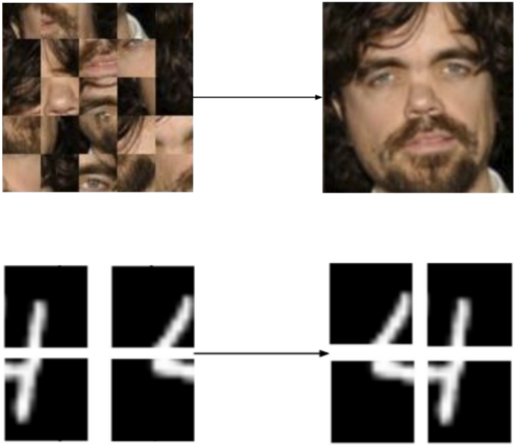
1. Introduction
2. Sinkhorn Operator
3. Sinkhorn Networks
4. Experiments
5. Summary

Why Latent Permutations and Matchings?

- Matchings and permutations are important for aligning, canonicalizing and sorting data.
- We want to solve matchings and permutations where they are not provided and have to be latent.



Permutation



Representing Permutations

Given a permutation mapping, $\pi : \{1, \ldots, m\} \to \{1, \ldots, m\}$, we can represent it as

1. $\begin{pmatrix} 1 & 2 & \cdots & m \\ \pi(1) & \pi(2) & \cdots & \pi(m) \end{pmatrix}$, or
2. multiplication of the identity matrix with permuted rows:

$$P_\pi \mathbf{g} = \begin{bmatrix} \mathbf{e}_{\pi(1)} \\ \mathbf{e}_{\pi(2)} \\ \vdots \\ \mathbf{e}_{\pi(n)} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ \vdots \\ n \end{bmatrix} = \begin{bmatrix} \pi(1) \\ \pi(2) \\ \vdots \\ \pi(n) \end{bmatrix}$$

where

$$P_\pi = \begin{bmatrix} \mathbf{e}_{\pi(1)} \\ \mathbf{e}_{\pi(2)} \\ \mathbf{e}_{\pi(3)} \\ \mathbf{e}_{\pi(4)} \\ \mathbf{e}_{\pi(5)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

.

Birkhoff Polytope

- The Birkhoff polytope B_n is the convex polytope in $\mathbf{R}^N$ $\left(\text{where } N = n^2\right)$ whose points are the doubly stochastic matrices, i.e., the $n \times n$ matrices whose entries are non-negative real numbers and whose rows and columns each add up to 1.
- B_n has $n!$ vertices, one for each permutation on n items.
- The extreme points of the Birkhoff polytope are the permutation matrices.
- Any doubly stochastic matrix may be represented as a convex combination of permutation matrices.

Contributions

- The non-differentiable parameterization of a permutation can be approximated in terms of a differentiable relaxation (Sinkhorn operator).
- Sinkhorn networks.
- Gumbel-Sinkhorn distribution.

1. Introduction
2. Sinkhorn Operator
3. Sinkhorn Networks
4. Experiments
5. Summary

Starting from Softmax

One sensible way to approximate a discrete category by continuous values is by a temperature-dependent softmax function

- $\mathrm{softmax}_{\tau}(x)_i = \exp(x_i/\tau) \,/\, \sum_{j=1} \exp\left(x_j/\tau\right)$.
- For positive values of $\tau, \mathrm{softmax}_{\tau}(x)_i$ is a point in the probability simplex.
- In the limit $\tau \to 0, \mathrm{softmax}_{\tau}(x)_i$ converges to a vertex of the simplex, a one-hot vector corresponding to the largest x_i.

Extension to Permutations: Sinkhorn Operator

We define the Sinkhorn operator $S(X)$ over an N dimensional square matrix X as:

$$\begin{aligned} S^0(X) &= \exp(X) \\ S^l(X) &= \mathcal{T}_c\left(\mathcal{T}_r\left(S^{l-1}(X)\right)\right) \\ S(X) &= \lim_{l\to\infty} S^l(X) \end{aligned}$$

- $\mathcal{T}_r(X) = X \oslash \left(X\mathbf{1}_N\mathbf{1}_N^\top\right)$, and $\mathcal{T}_c(X) = X \oslash \left(\mathbf{1}_N\mathbf{1}_N^\top X\right)$ are the row and column-wise normalization operators of a matrix.
- $\oslash$ denotes the element-wise division.

Remark: Sinkhorn (1964) proved that $S(X)$ must belong to the Birkhoff polytope, the set of doubly stochastic matrices, that we denote B_N.

Matching Operator

The choice of a permutation P through a square matrix X can be parameterized as the solution to the linear assignment problem:

$$M(X) = \arg\max_{P \in \mathcal{P}_N} \langle P, X \rangle_F.$$

where $\mathcal{P}_N$ denotes the set of permutation matrices and $\langle A, B \rangle_F = \mathrm{trace}\,(A^\top B)$ is the (Frobenius) inner product of matrices.
We call $M(\cdot)$ the matching operator, through which we parameterize the hard choice of a permutation.
Remark: M is non-differentiable and requires considering $n!$ permutations.

Connections between S and M

Theorem 1. For a doubly-stochastic matrix P, define its entropy as $h(P) = -\sum_{i,j} P_{i,j} \log\left(P_{i,j}\right)$. Then, one has,

$$S(X/\tau) = \arg\max_{P \in \mathcal{B}_N} \langle P, X \rangle_F + \tau h(P)$$

Now, assume also the entries of X are drawn independently from a distribution that is absolutely continuous with respect to the Lebesgue measure in $\mathbb{R}$. Then, almost surely, the following convergence holds:

$$M(X) = \lim_{\tau \to 0^+} S(X/\tau)$$

1. Introduction
2. Sinkhorn Operator
3. Sinkhorn Networks
4. Experiments
5. Summary

Sinkhorn Layer

We construct a layer to encode the representation of a permutation.

- Task: we want to learn a mapping from scrambled objects $\tilde{X}$ to non-scrambled ones X.
- We formulate it as a regression problem:

$$X_i = P^{-1}_{\theta,\tilde{X}_i}\tilde{X}_i + \varepsilon_i$$

  where ϵ_i is a noise term, and $P_{\theta,\tilde{X}_i}$ is the permutation matrix mapping X_i to $\tilde{X}_i$, which depends on $\tilde{X}_i$ and parameters θ.
- The error between X and a permutation of $\tilde{X}$: $f(\theta, X, \tilde{X}) = \sum_i^M \left\| X_i - P^{-1}_{\theta,\tilde{X}_i}\tilde{X}_i \right\|^2$.
- We use a network to parameterise a solution: $P_{\theta,\tilde{X}} = M(g(\tilde{X},\theta))$, where g can be a neural network.
- This is non-differentiable because of M.
- To make it differentiable, we can use $S(g(\tilde{X},\theta)/\tau)$ instead.

Sinkhorn Layer

- The value of τ must be chosen with caution:
  - If τ is too small, gradients vanish almost everywhere, as $S(g(\tilde{X},\theta)/\tau)$ approaches the non-differentiable $M(g(\tilde{X},\theta))$.
  - If τ is too large, $S(X/\tau)$ may be far from the vertices of the Birkhoff polytope, and reconstructions $P^{-1}_{\theta,\tilde{X}}\tilde{X}$ may be nonsensical.
- Importantly, we will always add noise to the output layer $g(\tilde{X},\theta)$ as a regularization device: by doing so we ensure uniqueness of $M(g(\tilde{X},\theta))$, which is required for convergence in Theorem 1.

Permutation Equivariance

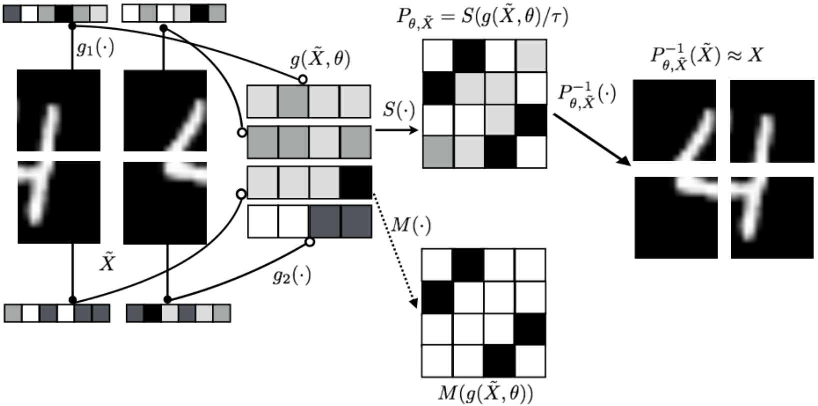
- Intuition: reconstructions of objects should not depend on how pieces were scrambled, but only on the pieces themselves.

$$P_{\theta,P'\tilde{X}}\left(P'\tilde{X}\right) = P'\left(P_{\theta,\tilde{X}}\tilde{X}\right)$$

  where P' is an arbitrary permutation.

- Solution: using the same network to process each piece of $\tilde{X}$, throwing an N dimensional output.

Scheme of Sinkhorn Network

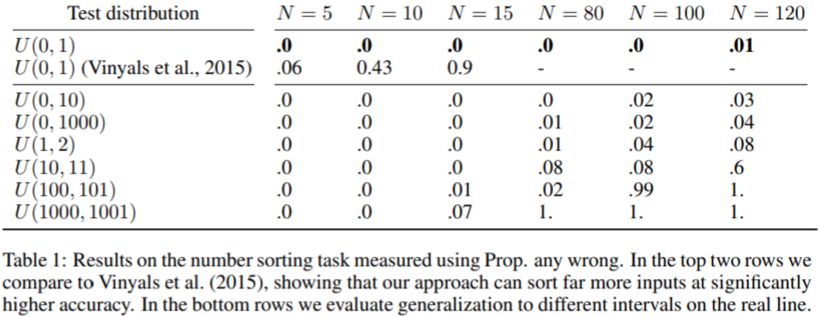


$g_1(\cdot)$
$g(\tilde{X},\theta)$
$P_{\theta,\tilde{X}} = S(g(\tilde{X},\theta)/\tau)$
$S(\cdot)$
$P^{-1}_{\theta,\tilde{X}}(\cdot)$
$P^{-1}_{\theta,\tilde{X}}(\tilde{X}) \approx X$
$M(\cdot)$
$\tilde{X}$
$g_2(\cdot)$
$M(g(\tilde{X},\theta))$

1. Introduction
2. Sinkhorn Operator
3. Sinkhorn Networks
4. **Experiments**
5. Summary

Sorting Numbers

g_1, g_2: both fully connected layers.

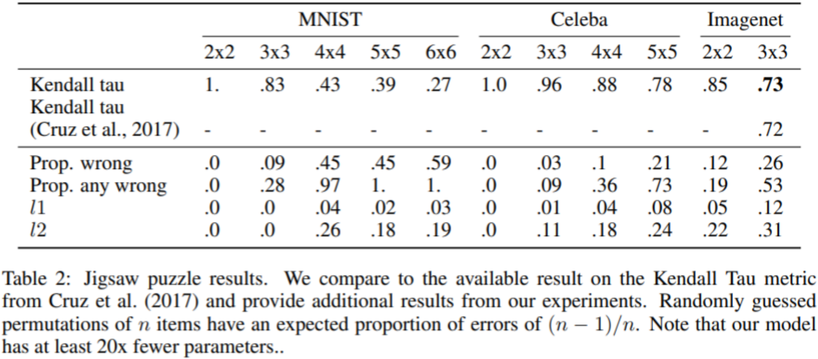


Test distribution	$N = 5$	$N = 10$	$N = 15$	$N = 80$	$N = 100$	$N = 120$
$U(0, 1)$	**.0**	**.0**	**.0**	**.0**	**.0**	**.01**
$U(0, 1)$ (Vinyals et al., 2015)	.06	0.43	0.9	-	-	-
$U(0, 10)$	.0	.0	.0	.0	.02	.03
$U(0, 1000)$	.0	.0	.0	.01	.02	.04
$U(1, 2)$	.0	.0	.0	.01	.04	.08
$U(10, 11)$	.0	.0	.0	.08	.08	.6
$U(100, 101)$	.0	.0	.01	.02	.99	1.
$U(1000, 1001)$	.0	.0	.07	1.	1.	1.

Table 1: Results on the number sorting task measured using Prop. any wrong. In the top two rows we compare to Vinyals et al. (2015), showing that our approach can sort far more inputs at significantly higher accuracy. In the bottom rows we evaluate generalization to different intervals on the real line.

Jigsaw Puzzles

g_1: a simple CNN (convolution + max pooling)



	MNIST					Celeba				Imagenet	
	2x2	3x3	4x4	5x5	6x6	2x2	3x3	4x4	5x5	2x2	3x3
Kendall tau	1.	.83	.43	.39	.27	1.0	.96	.88	.78	.85	**.73**
Kendall tau (Cruz et al., 2017)	-	-	-	-	-	-	-	-	-	-	.72
Prop. wrong	.0	.09	.45	.45	.59	.0	.03	.1	.21	.12	.26
Prop. any wrong	.0	.28	.97	1.	1.	.0	.09	.36	.73	.19	.53
*l*1	.0	.0	.04	.02	.03	.0	.01	.04	.08	.05	.12
*l*2	.0	.0	.26	.18	.19	.0	.11	.18	.24	.22	.31

Table 2: Jigsaw puzzle results. We compare to the available result on the Kendall Tau metric from Cruz et al. (2017) and provide additional results from our experiments. Randomly guessed permutations of n items have an expected proportion of errors of $(n-1)/n$. Note that our model has at least 20x fewer parameters..

Jigsaw Puzzles

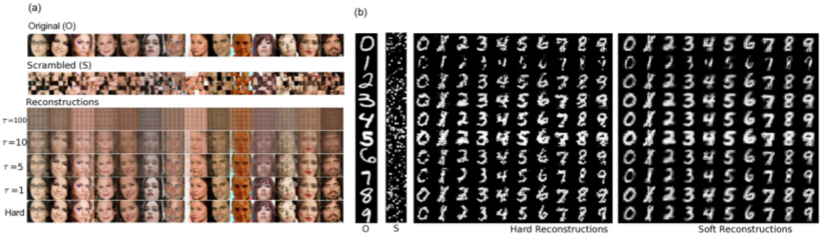


Figure 2: (a) Sinkhorn networks can be trained to solve Jigsaw Puzzles. Given a trained model, ‘soft’ reconstructions are shown at different τ using $S(X/\tau)$. We also show hard reconstructions, made by computing $M(X)$ with the Hungarian algorithm (Munkres, 1957). (b) Sinkhorn networks can also be used to learn to transform any MNIST digit into another. We show hard and soft reconstructions, with $\tau = 1$.

1 Introduction

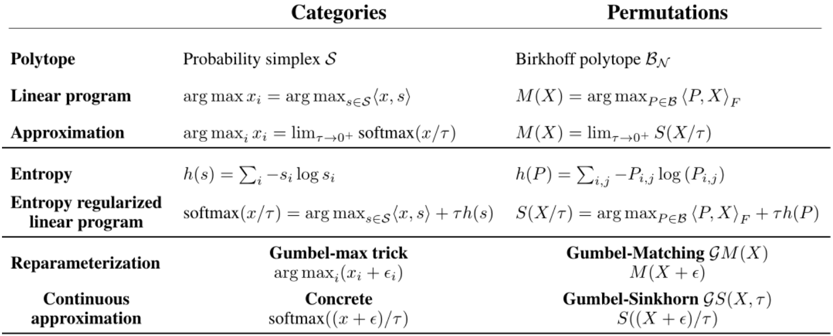
2 Sinkhorn Operator

3 Sinkhorn Networks

4 Experiments

5 Summary

Analogies between Permutation and Categories



	Categories	**Permutations**
Polytope	Probability simplex $\mathcal{S}$	Birkhoff polytope $\mathcal{B}_{\mathcal{N}}$
Linear program	$\arg\max x_i = \arg\max_{s \in \mathcal{S}} \langle x, s \rangle$	$M(X) = \arg\max_{P \in \mathcal{B}} \langle P, X \rangle_F$
Approximation	$\arg\max_i x_i = \lim_{\tau \to 0^+} \text{softmax}(x/\tau)$	$M(X) = \lim_{\tau \to 0^+} S(X/\tau)$
Entropy	$h(s) = \sum_i -s_i \log s_i$	$h(P) = \sum_{i,j} -P_{i,j} \log \left(P_{i,j}\right)$
Entropy regularized linear program	$\text{softmax}(x/\tau) = \arg\max_{s \in \mathcal{S}} \langle x, s \rangle + \tau h(s)$	$S(X/\tau) = \arg\max_{P \in \mathcal{B}} \langle P, X \rangle_F + \tau h(P)$
Reparameterization	**Gumbel-max trick** $\arg\max_i (x_i + \epsilon_i)$	**Gumbel-Matching** $\mathcal{G}M(X)$ $M(X + \epsilon)$
Continuous approximation	**Concrete** $\text{softmax}((x + \epsilon)/\tau)$	**Gumbel-Sinkhorn** $\mathcal{G}S(X, \tau)$ $S((X + \epsilon)/\tau)$

Reparameterizing the Birkhoff Polytope for Variational Permutation Inference

Scott Linderman, Gonzalo Mena, Hal Cooper, Liam Paninski, John Cunningham

2021.10.14

Outlines

1. Introduction

2. Background

3. Variational Permutation Inference via Reparameterization

4. Experiments

1. Introduction
2. Background
3. Variational Permutation Inference via Reparameterization
4. Experiments

Permutation Inference and Contributions

- Permutation inference is central to many modern machine learning problems, *e.g.*, multiple-object tracking, ranking problems to search and recommender systems.
- We proposes 2 transformations that enable variational inference over doubly stochastic matrices.
- We exploit these transformations and reparameterization gradients to introduce variational inference over permutation matrices

1. Introduction
2. Background
3. Variational Permutation Inference via Reparameterization
4. Experiments

Permutation and Birkhoff Polytope

- A permutation is a bijective mapping of a set onto itself: both the rows and columns of X must sum to one.
- When this set is finite, the mapping is conveniently represented as a binary matrix $X \in \{0,1\}^{N\times N}$ where entry $x_{m,n} = 1$ implies that element m is mapped to element n.
- Birkhoff-von Neumann theorem: the convex hull of the set of permutation matrices is the set of doubly-stochastic matrices.
- The set is called the Birkhoff polytope. Let $\mathcal{B}_N$ denote the Birkhoff polytope of $N \times N$ doubly-stochastic matrices.

Variational Inference

Given a model with data y, likelihood $p(y \mid x)$, and prior $p(x)$, variational Bayesian inference algorithms aim to approximate the posterior distribution $p(x \mid y)$ with a more tractable distribution $q(x; \theta)$.
We find this approximate distribution by searching for the parameters θ that maximize the evidence lower bound (ELBO),

$$\mathcal{L}(\theta) \triangleq \mathbb{E}_q[\log p(x, y) - \log q(x; \theta)]$$

- Use SGD to optimize ELBO.
- When computing $\nabla_\theta \mathcal{L}(\theta)$, ELBO contains an expectation with respect to a distribution that depends on these parameters.

Reparameterization Trick

When x is a continuous random variable, we can sometimes leverage the reparameterization trick. Specifically, in some cases we can simulate from q via the following equivalence,

$$x \sim q(x;\theta) \quad \Longleftrightarrow \quad z \sim r(z), \quad x = g(z;\theta)$$

where r is a distribution on the "noise" z and where $g(z;\theta)$ is a deterministic and differentiable function:

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{r(z)} \left[\nabla_\theta \log p \left(g(z;\theta) \mid y \right) - \nabla_\theta \log q(g(z;\theta);\theta) \right]. \tag{1}$$

- This gradient can be estimated with Monte Carlo.
- The gradient can only be computed if x is continuous.

Continuous Relaxations: Gumbel-Softmax

We can relax an atomic density to continuous densities on the simplex by viewing x as a vertex of the simplex:

$$\begin{aligned} z_n &\overset{\text{iid}}{\sim} \text{Gumbel}(0,1) \\ \psi_n &= \log \theta_n + z_n \\ x &= \text{softmax}(\psi/\tau) \\ &= \left(\frac{e^{\psi_1/\tau}}{\sum_{n=1}^N e^{\psi_n/\tau}}, \cdots, \frac{e^{\psi_N/\tau}}{\sum_{n=1}^N e^{\psi_n/\tau}} \right) \end{aligned} \tag{2}$$

The output x is now a point on the simplex, and the parameter $\theta = (\theta_1, \ldots, \theta_N) \in \mathbb{R}_+^N$ can be optimized via stochastic gradient ascent with the reparameterization trick.
As the temperature τ goes to zero, the softmax converges to the argmax function.

1. Introduction
2. Background
3. Variational Permutation Inference via Reparameterization
4. Experiments

Overview

We develop two transformations to map $O\left(N^2\right)$-dimensional random variates to points in or near the Birkhoff polytope:

1. stick-breaking
2. rounding

Stick-breaking transformations to the Birkhoff polytope

- Let B be a matrix in $[0,1]^{(N-1)\times(N-1)}$; we will transform it into a doubly-stochastic matrix $X \in [0,1]^{N\times N}$.
- Intuition: Each row and column has an associated unit-length "stick" and we allot to its entries.

Stick-Breaking: Categorical

Transform a real-valued vector $\psi \in \mathbb{R}^{N-1}$ to a point in the simplex.

1. Reparameterize ψ by $\theta = (\mu_n, \nu_n)_{n=1}^{N-1}$:

$$\begin{aligned} z_n &\overset{\text{iid}}{\sim} \mathcal{N}(0,1) \\ \psi_n &= \mu_n + \nu_n z_n, \quad 1 \leq n \leq N-1 \end{aligned} \tag{3}$$

2. Map ψ to the unit hypercube in a temperature-controlled manner: $\beta_n = \sigma(\psi_n/\tau)$ where $\sigma(u) = (1 + e^{-u})^{-1}$ is the logistic function.
3. Transform the unit hypercube to a point in the simplex:

$$\begin{aligned} x_1 &= \beta_1 \\ x_n &= \beta_n \left(1 - \sum_{m=1}^{n-1} x_m\right), \quad 2 \leq n \leq N-1 \\ x_N &= 1 - \sum_{m=1}^{N-1} x_m \end{aligned}$$

Here, β_n is the fraction of the remaining ”stick” of probability mass assigned to x_n.

Temperature in Stick-Breaking

The temperature τ controls how concentrated $p(x)$ is at the vertices of the simplex.

- $\tau \to 0$: we can recover any categorical distribution;
- $\tau \to \infty$: the density concentrates on a point in the interior of the simplex determined by the parameters.
- For intermediate values, the density is continuous on the simplex.

Stick-Breaking: Permutations

1. Start from a standard Gaussian matrix $Z \in \mathbb{R}^{(N-1)\times(N-1)}$:

$$\psi_{mn} = \mu_{mn} + \nu_{mn} z_{mn},$$
$$\beta_{mn} = \sigma\left(\psi_{mn}/\tau\right)$$

   where $\theta = \left\{\mu_{mn}, \nu_{mn}^2\right\}_{m,n=1}^{N}$ are the mean and variance parameters of the intermediate Gaussian matrix Ψ.

2.

$$x_{1n} = \beta_{1n}\left(1 - \sum_{k=1}^{n-1} x_{1k}\right) \quad \text{for } n = 2, \dots, N-1$$
$$x_{1N} = 1 - \sum_{n=1}^{N-1} x_{1n}$$

As $\tau \to 0$, the values of β_{mn} are pushed to either zero or one.
As a result, the doubly-stochastic output matrix X is pushed toward the extreme points of the Birkhoff polytope, the permutation matrices.

Rounding

Rounding moves points in $\mathbb{R}^{N\times N}$ nearer to the closest permutation matrix.

1. Let e_n denote a one-hot vector with n-th entry equal to one. Define the rounding operator,

$$\text{round}(\psi) = e_{n^*}$$

where

$$n^* = \arg\min_n \|e_n - \psi\|^2$$

Rounding effectively partitions the space into N disjoint "Voronoi" cells:

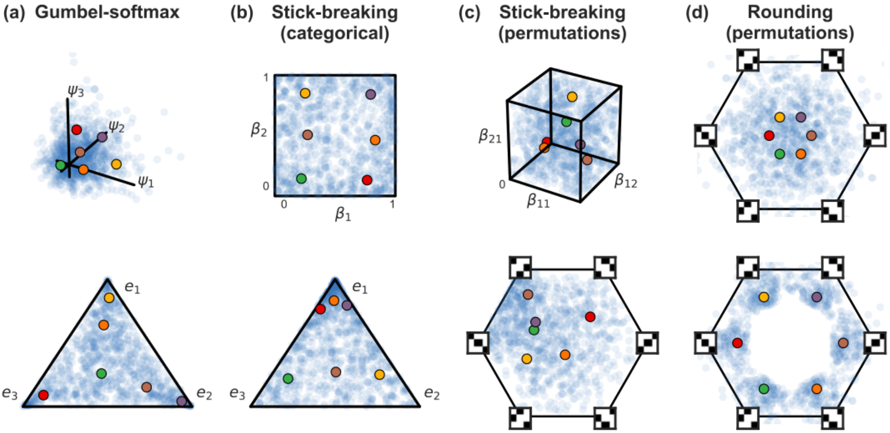
$$V_n = \left\{\psi \in \mathbb{R}^N : (\psi_n \geq \psi_m \forall m) \land (\psi_n > \psi_m \forall m < n)\right\}$$

By definition, $\text{round}(\psi) = e_{n^*}$ for all $\psi \in V_{n^*}$.

2. We define a map that pulls points toward their rounded values,

$$x = \tau\psi + (1-\tau)\,\text{round}(\psi)$$

Different Reparameterizations of Discrete Polytopes



1. Introduction
2. Background
3. Variational Permutation Inference via Reparameterization
4. Experiments

Synthetic Experiments

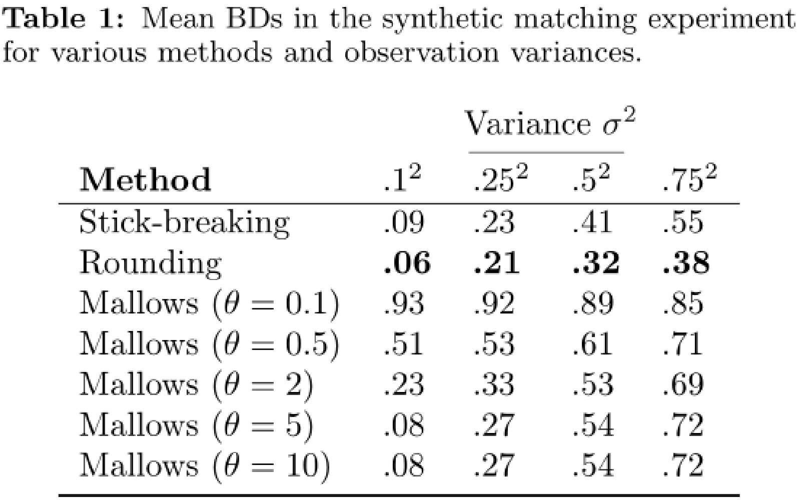


Table 1: Mean BDs in the synthetic matching experiment for various methods and observation variances.

		Variance σ^2		
Method	$.1^2$	$.25^2$	$.5^2$	$.75^2$
Stick-breaking	.09	.23	.41	.55
Rounding	**.06**	**.21**	**.32**	**.38**
Mallows ($\theta = 0.1$)	.93	.92	.89	.85
Mallows ($\theta = 0.5$)	.51	.53	.61	.71
Mallows ($\theta = 2$)	.23	.33	.53	.69
Mallows ($\theta = 5$)	.08	.27	.54	.72
Mallows ($\theta = 10$)	.08	.27	.54	.72

Synthetic Experiments

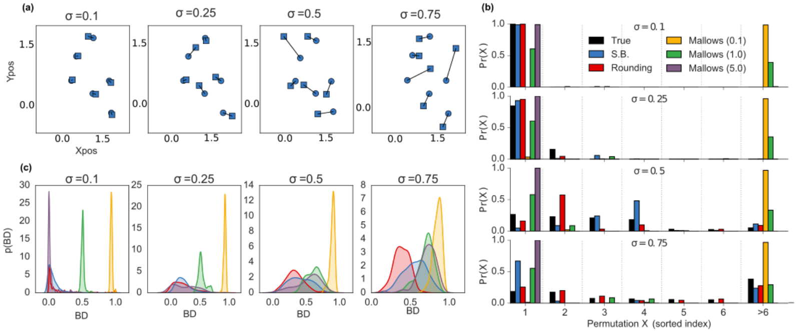


Figure 2: Synthetic matching experiment results. The goal is to infer the lines that match squares to circles. (a) Examples of center locations (circles) and noisy samples (squares), at different noise variances. (b) For illustration, we show the true and inferred probability mass functions for different method (rows) along with the Battacharya distance (BD) between them for a selected case of each σ (columns). Permutations (indices) are sorted from the highest to lowest actual posterior probability. Only the 10 most likely configurations are shown, and the 11st bar represents the mass of all remaining configurations. (c) KDE plots of Battacharya distances for each parameter configuration (based on 200 experiment repetitions) for each method and parameter configuration. For comparison, stick-breaking, rounding, and Mallows ($\theta = 1.0$) have BD's of .36, .35, and .66, respectively, in the $\sigma = 0.5$ row of (b).