BRUNO

A Deep Recurrent Model for Exchangeable Data

Mohd Adnan
MASc Candidate

Objective of BRUNO:

To generate a sequence of images (or random variables) which is “unordered”

Exchangeable Random Variables

A stochastic process is exchangeable if for all n and all permutations π:

$$p(x_1, \ldots, x_n) = p\left(x_{\pi(1)}, \ldots, x_{\pi(n)}\right)$$

Theorems for Exchangeability

- De Fenitti's Theorem: It states that a random infinitely exchangeable sequence can be factorised into mixture densities conditioned on some parameter θ which captures the underlying generative process i.e.

$$P(x_1, \ldots, x_n) = \int p(\theta) \prod_{i=1}^{n} p(x_i|\theta)\, d(\theta)$$

In terms of predictive distributions $p(x_n|x_{1:n-1})$, the de Finetti equation can be written as

$$p(x_n|x_{1:n-1}) = \int p(x_n|\theta)p(\theta|x_{1:n-1})d\theta,$$

Problem

Integral in the equation is intractable

$$P(x_1, \ldots, x_n) = \int p(\theta) \prod_{i=1}^{n} p(x_i|\theta)\, d(\theta)$$

Alternative to directly modeling de Finetti's Equation

Learn a mapping from input space to an exchangeable process

Gaussian Process:

A collection of random variables {f(x) : x ∈ X } is said to be drawn from a Gaussian process with mean function m(·) and covariance function k(·, ·) if for any finite set of elements x1, . . . , xm ∈ X , the associated finite set of random variables f(x1), . . . , f(xm) have distribution,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_m) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_m) \\ \vdots & \ddots & \vdots \\ k(x_m, x_1) & \cdots & k(x_m, x_m) \end{bmatrix} \right).$$

Conditional Distribution of Multivariate Distribution

Y~N(μ,Σ)

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \quad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix}$$

$$(\boldsymbol{y}_1 | \boldsymbol{y}_2 = \boldsymbol{a}) \sim \mathcal{N}(\overline{\boldsymbol{\mu}}, \overline{\Sigma}),$$

$$\overline{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \Sigma_{12} \Sigma_{22}{}^{-1} (\boldsymbol{a} - \boldsymbol{\mu}_2)$$

$$\overline{\Sigma} = \Sigma_{11} - \Sigma_{12} \Sigma_{22}{}^{-1} \Sigma_{21}$$

Exchangeable Gaussian Process

Covariance matrix of Gaussian Process can be defined in such way that the process becomes exchangeable.

$$\Sigma = \begin{bmatrix} v & \rho & \rho & \dots & \rho \\ \rho & v & \rho & \dots & \rho \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \rho & \dots & v \end{bmatrix} \quad 0 \leq \rho < v$$

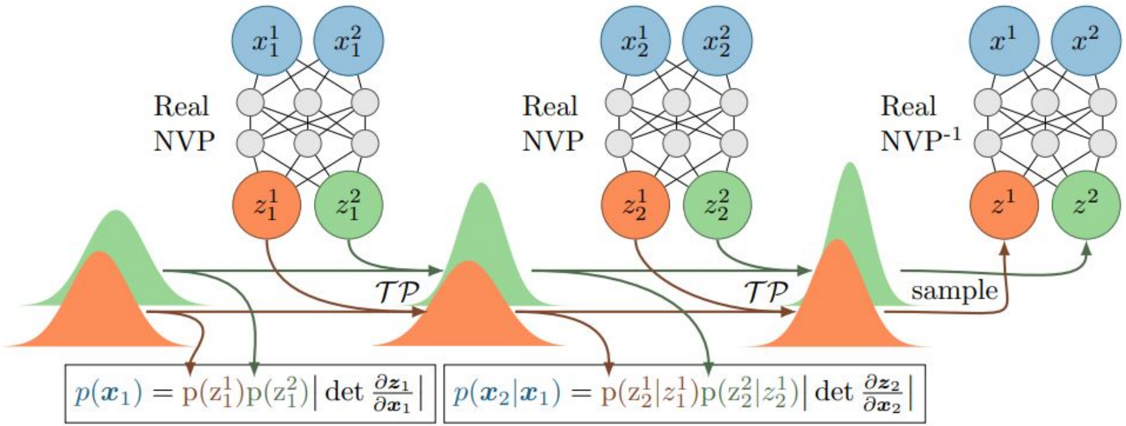
Recurrent Updates

Since the structure of covariance matrix is simple, we can derive recurrent updates.

$$\mu_{n+1} = (1 - d_n)\mu_n + d_n z_n, \quad v_{n+1} = (1 - d_n)v_n + d_n(v - \rho)$$

Normalization Flow:

Use Real NVP to learn a bijective mapping from input space to Exchangeable Gaussian Process.



Training:

Maximize the the joint log-likelihood

$$\mathcal{L} = \sum_{n=0}^{N-1} \log p(\boldsymbol{x}_{n+1} | \boldsymbol{x}_{1:n})$$

Other Application: Few Shot Learning

Model	n=1,k=5	n=5,k=5	n=1,k=20	n=5,k=20
Matching Nets	98.1%	98.9%	93.8%	98.5%
BRUNO	86.3%	95.6%	69.2%	87.7%
BRUNO finetuned	97.1%	99.4%	91.3%	97.8%

Related Papers on Exchangeablity

1. Deep Sets (https://arxiv.org/abs/1703.06114)
2. Set Transformer (https://arxiv.org/abs/1810.00825)
3. Conditional BRUNO (http://bayesiandeeplearning.org/2018/papers/40.pdf)
4. Learning Permutation Invariant Representation using Memory Networks (https://arxiv.org/abs/1911.07984)

Thank You!