CS480/680: Introduction to Machine Learning

Lec 14: Flows

Yaoliang Yu

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS
DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

March 14, 2024

Recap: MLE

- Given training data $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \sim q(\mathbf{x})$, the data density
- Parameterize $p_{\boldsymbol{\theta}}(\mathbf{x})$, the model density through push-forward:

Theorem: Representation through push-forward

Let r be any continuous distribution on $\mathbb{R}^h$. For any distribution p on $\mathbb{R}^d$, there exist push-forward maps $\mathbf{T} : \mathbb{R}^h \to \mathbb{R}^d$ such that $\mathsf{Z} \sim r \implies \mathbf{T}(\mathsf{Z}) \sim p$.

- Estimate $\boldsymbol{\theta}$ by minimizing some "distance":

$$\min_{\boldsymbol{\theta}} \ \mathsf{KL}(q \| p_{\boldsymbol{\theta}}) \quad \equiv \quad \int -\log p_{\boldsymbol{\theta}}(\mathbf{x}) \cdot q(\mathbf{x}) \, \mathrm{d}\mathbf{x} \quad \approx \quad -\frac{1}{n} \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(\mathbf{x}_i)$$

- Need a training sample from q and an explicit form of $p_{\boldsymbol{\theta}}$

Change-of-variable Formula

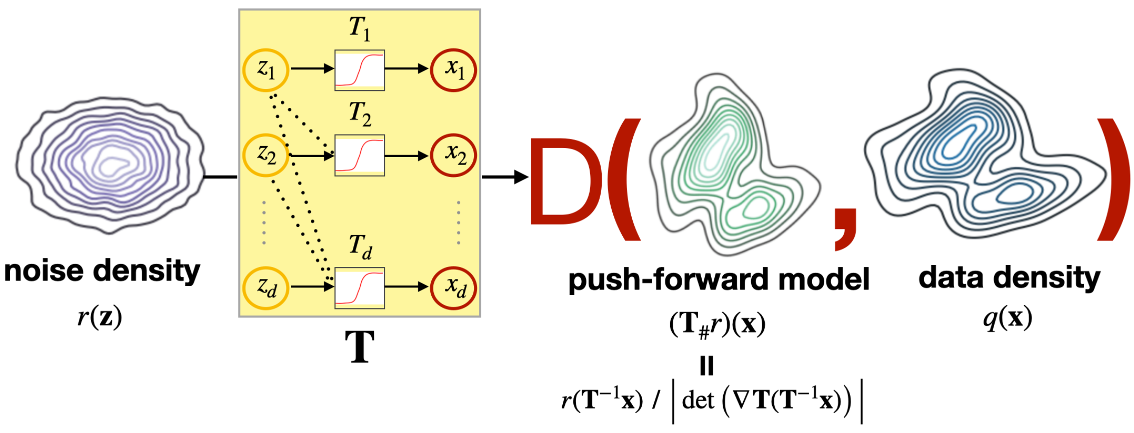
Theorem: Push-forward as change-of-variable

Let r be any continuous distribution on $\mathbb{R}^d$. If the push-forward map $\mathbf{T} : \mathbb{R}^d \to \mathbb{R}^d$ is invertible, then the density of $\mathsf{X} := \mathbf{T}(\mathsf{Z})$ is

$$p(\mathbf{x}) = r(\mathbf{T}^{-1}\mathbf{x}) \cdot |\det(\nabla\mathbf{T}^{-1}\mathbf{x})| = r(\mathbf{T}^{-1}\mathbf{x}) \;/\; |\det(\nabla\mathbf{T}(\mathbf{T}^{-1}\mathbf{x}))|$$

- Roughly, $p(\mathbf{x})\,\mathrm{d}\mathbf{x} = r(\mathbf{z})\,\mathrm{d}\mathbf{z}$: preservation of mass
- $\mathbf{T} \circ \mathbf{T}^{-1} = \mathrm{Id} \implies \nabla\mathbf{T}(\mathbf{T}^{-1}) \cdot \nabla\mathbf{T}^{-1} = \mathrm{Id}$ and $\det(A^{-1}) = 1/\det(A)$
  - $\mathbf{T} = (T_1, \ldots, T_d) : \mathbb{R}^d \to \mathbb{R}^d$ and $\nabla\mathbf{T} = (\nabla T_1, \ldots, \nabla T_d) : \mathbb{R}^d \to \mathbb{R}^d \otimes \mathbb{R}^d$
- Input dim = output dim
- From now on, use the notation $p = \mathbf{T}_{\#}r$ to denote the push-forward

MLE Revisited



$$\min_{\mathbf{T}:\mathbb{R}^d\to\mathbb{R}^d} \mathsf{KL}(q \,\|\, T_{\#}r) \approx \max_{\mathbf{T}} \frac{1}{n}\sum_{i=1}^{n}\left[\log r\left(\mathbf{T}^{-1}\mathbf{x}_i\right) - \log\left|\det\nabla\mathbf{T}\left(\mathbf{T}^{-1}\mathbf{x}_i\right)\right|\right]$$

Pick Your Poison

- Inverse in training, easy in sampling

$$\max_{\mathbf{T}} \frac{1}{n}\sum_{i=1}^{n}\Big[\log r\big(\mathbf{T}^{-1}\mathbf{x}_i\big) - \log\big|\det \nabla\mathbf{T}\big(\mathbf{T}^{-1}\mathbf{x}_i\big)\big|\Big]$$

- Easy in training, inverse in sampling

$$\max_{\mathbf{S}=\mathbf{T}^{-1}} \frac{1}{n}\sum_{i=1}^{n}\Big[\log r\big(\mathbf{S}\mathbf{x}_i\big) + \log\big|\det \nabla\mathbf{S}(\mathbf{x}_i)\big|\Big]$$

- Bottleneck in inverse $\mathbf{T}^{-1}$ and determinant $\det\nabla$
- Can apply GAN to avoid both inverse and determinant —> minimax game

Increasing Triangular Map

$$
\begin{aligned}
x_1 &= T_1(z_1) \\
x_2 &= T_2(z_1, z_2) \\
&\vdots \\
x_d &= T_d(z_1,\ z_2,\ z_3,\ \ldots,\ z_d)
\end{aligned}
$$

$$
\nabla \mathbf{T}(\mathbf{z}) = \begin{bmatrix} \frac{\partial T_1}{\partial z_1} & 0 & 0 & \cdots & 0 \\ \frac{\partial T_2}{\partial z_1} & \frac{\partial T_2}{\partial z_2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial T_d}{\partial z_1} & \frac{\partial T_d}{\partial z_1} & \frac{\partial T_d}{\partial z_3} & \cdots & \frac{\partial T_d}{\partial z_d} \end{bmatrix}
$$

- Triangular: j-th output x_j depends only on the first j inputs $z_1, \ldots, z_j$
  - $\nabla \mathbf{T}$ is a (lower) triangular matrix
- Increasing: $T_j(z_1, \ldots, z_{j-1}, z_j)$ is increasing w.r.t. z_j for any $z_1, \ldots, z_{j-1}$
  - diagonal of ∇T is positive, i.e. $\frac{\partial T_j}{\partial z_j} > 0$

Practical Implications of Increasing Triangular Maps

- It is easy to compute the inverse $\mathbf{T}^{-1} : \mathbb{R}^d \to \mathbb{R}^d$
  - Fix x_1, compute z_1 such that $T_1(z_1) = x_1$
  - T_1 is increasing w.r.t. z_1, so bisection suffices
  - Fix x_j, compute z_j such that $T_j(z_1, \ldots, z_{j-1}, z_j) = x_j$
  - T_j is increasing w.r.t. z_j while $z_1, \ldots, z_{j-1}$ are already computed, so bisection suffices
- It is easy to compute the determinant $\det(\nabla \mathbf{T})$
  - $\nabla \mathbf{T}$ is triangular, so det = product of the diagonal

Theoretical Implications of Increasing Triangular Maps

Theorem: Uniqueness for increasing triangular maps

For any two densities r and p on $\mathbb{R}^d$, there exists a unique (up to permutation) increasing triangular map $\mathbf{T}$ so that $p = \mathbf{T}_{\#} r$.

- In theory, any property of a probabilistic density is captured in the deterministic map $\mathbf{T}$, and vice versa!

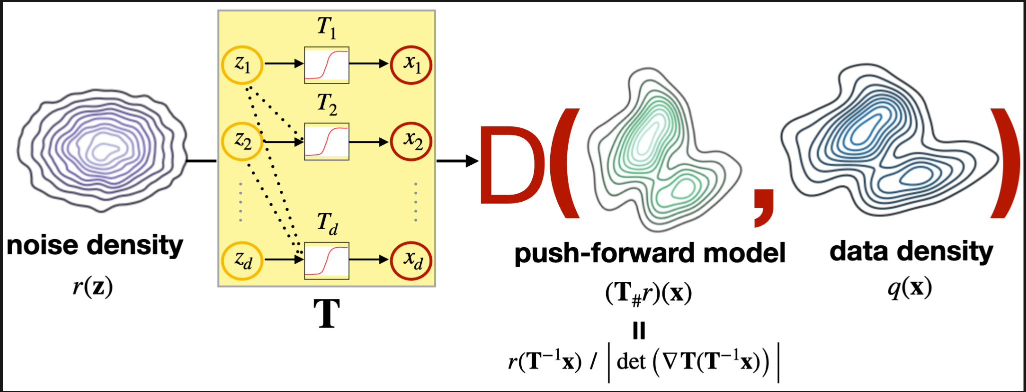
Example: Linear transformation of Gaussian is Gaussian

Let $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathrm{Id})$ and $L = \mathrm{Cholesky}(S)$. Then, $\mathbf{x} = L\mathbf{n} + \boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\mu}, S)$.

- We see now this is the only way, when restricted to increasing triangular maps!

V. I. Bogachev et al. "Triangular transformations of measures". *Sbornik: Mathematics*, vol. 196, no. 3 (2005), pp. 309–335.

MLE Revisited

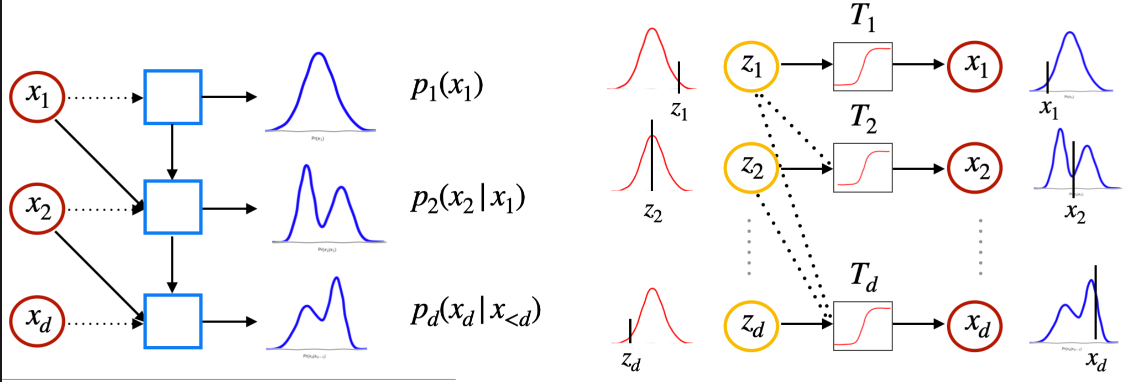


$$\min_{\mathbf{T}:\mathbb{R}^d \to \mathbb{R}^d} \mathsf{KL}(q \,\|\, T_{\#}r) \approx \max_{\mathbf{T}} \frac{1}{n}\sum_{i=1}^{n}\left[\log r\left(\mathbf{T}^{-1}\mathbf{x}_i\right) - \sum_{j=1}^{d}\log \nabla_j T_j\left(\mathbf{T}^{-1}\mathbf{x}_i\right)\right]$$

Y. Marzouk et al. "Sampling via Measure Transport: An Introduction". In: *Handbook of Uncertainty Quantification*. Ed. by R. Ghanem et al. Springer, 2016, pp. 1–41, P. Jaini et al. "Sum-of-squares Polynomial Flow". In: *International Conference on Machine Learning (ICML)*. 2019.

Autoregressive Models

$$p(\mathbf{x}) = \prod_{j=1}^{d} p_j(x_j | x_1, \ldots, x_{j-1})$$

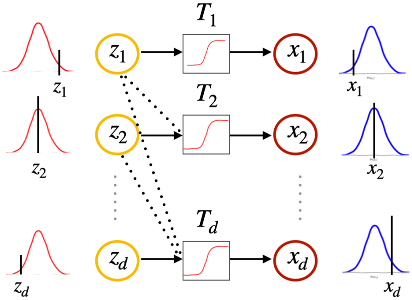


H. Larochelle and I. Murray. "The neural autoregressive distribution estimator". In: *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*. 2011, pp. 29–37.

AR with Gaussian Conditionals

$$p(x) = p_1(x_1) \cdot p_2(x_2 \mid x_1) \cdot \cdots \cdot p_d(x_d \mid x_{<d})$$

$$\mathcal{N}(\mu_1, \sigma_1^2) \quad \mathcal{N}\big(\mu_2(x_1), \sigma_2^2(x_1)\big) \quad \mathcal{N}\big(\mu_d(x_1, \ldots, x_{d-1}), \sigma_d^2(x_1, \ldots, x_{d-1})\big)$$



$$x_1 = \sigma_1 \cdot z_1 + \mu_1 =: T_1(z_1)$$

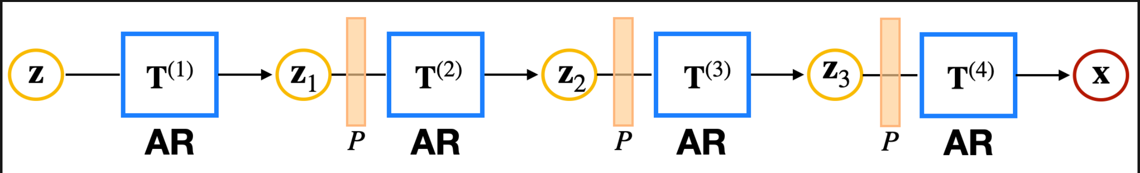
$$x_2 = \sigma_2(z_1) \cdot z_2 + \mu_2(z_1) := T_2(z_2 \, ; z_1)$$

$$\vdots$$

$$x_d = \sigma_d(z_{<d}) \cdot z_d + \mu_d(z_{<d}) := T_d(z_d \, ; z_{<d})$$

D. P. Kingma and P. Dhariwal. "Glow: Generative flow with invertible 1x1 convolutions". In: *Advances in Neural Information Processing Systems*. 2018.

Masked AR Flows



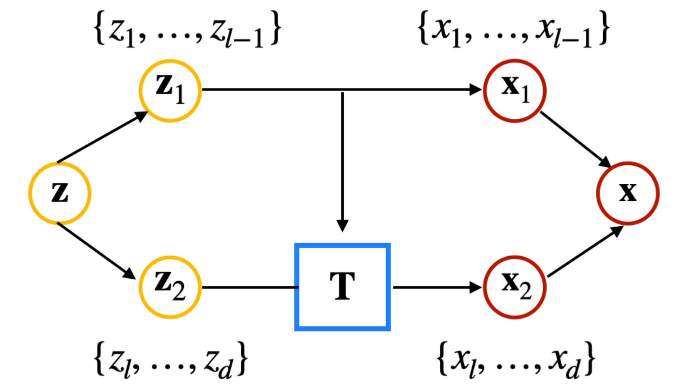
$$(\mathbf{T}_{\#}r)(\mathbf{x}) = r(\mathbf{z}) \ / \ \det\left(\nabla \mathbf{T}^{(1)}\mathbf{z}\right) \ / \ \det\left(\nabla \mathbf{T}^{(2)}\mathbf{z}_1\right) \ / \ \det\left(\nabla \mathbf{T}^{(3)}\mathbf{z}_2\right) \ / \ \det\left(\nabla \mathbf{T}^{(4)}\mathbf{z}_3\right)$$

$$x_j = z_j \cdot \exp\left(\alpha_j(z_1, \ldots, z_{j-1})\right) + \mu_j(z_1, \ldots, z_{j-1}) =: T_j(z_1, \ldots, z_{j-1}, z_j)$$

- Stack multiple layers to get deep
- What happens if the number of layers goes to ∞ ?

G. Papamakarios et al. "Masked autoregressive flow for density estimation". In: *Advances in Neural Information Processing Systems*. 2017, pp. 2338–2347.

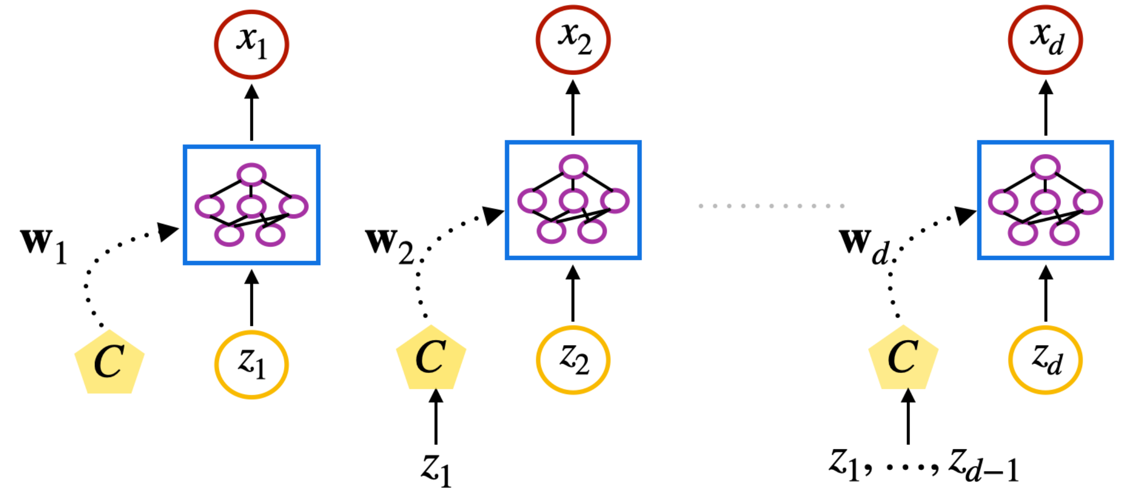
real-NVP



$$T_j(z_j\ ;\ z_1, \ldots, z_{l-1}) = \exp\left(\alpha_j(z_1, \ldots, z_{l-1}) \cdot \mathbf{1}_{j \notin [l-1]}\right) \cdot z_j + \mu_j(z_1, \ldots, z_{l-1}) \cdot \mathbf{1}_{j \notin [l-1]}$$

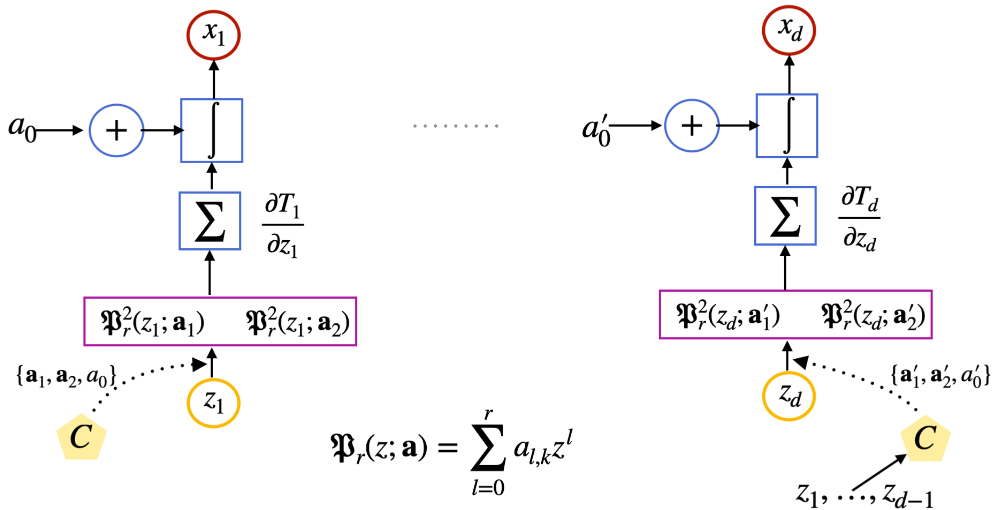
L. Dinh et al. "Density Estimation using Real NVP". In: *Proceedings of the 5th International Conference on Learning Representation*. 2017.

Neural AR Flow

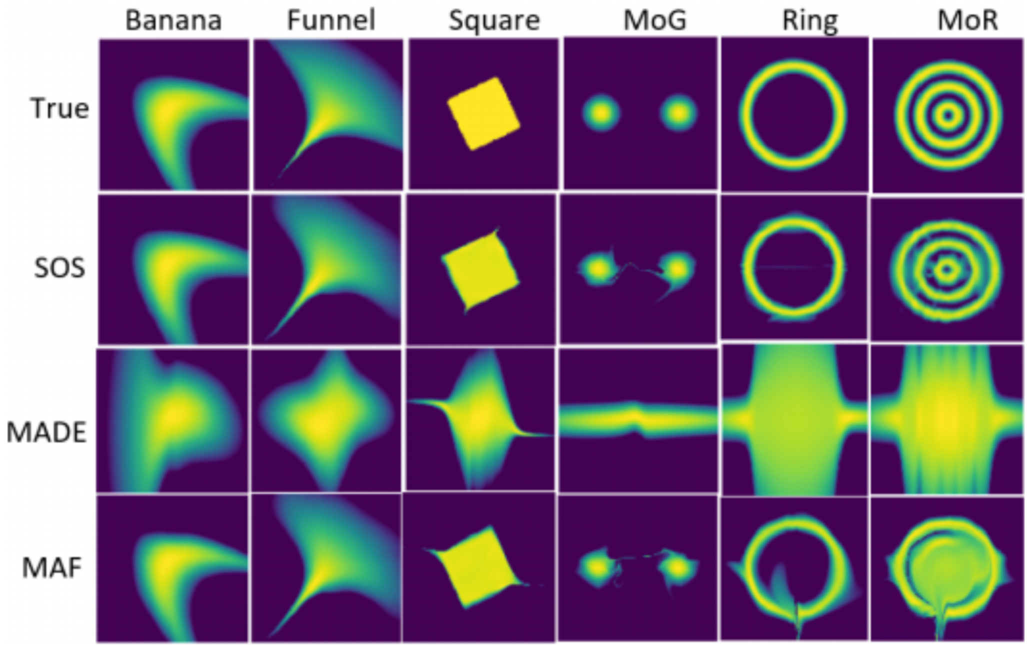


C.-W. Huang et al. "Neural Autoregressive Flows". In: *Proceedings of the 35th International Conference on Machine Learning*. 2018.

Sum-Of-Squares



P. Jaini et al. "Sum-of-squares Polynomial Flow". In: *International Conference on Machine Learning (ICML)*. 2019.


Banana
Funnel
Square
MoG
Ring
MoR
True
SOS
MADE
MAF

Interpreting $\mathbf{T} =: \mathbf{Q}$

- $\mathbf{T} : \mathbb{R}^d \to \mathbb{R}^d$ pushes the noise Z forward to observation X
- The inverse map $\mathbf{T}^{-1}$ pulls observation X back to noise Z

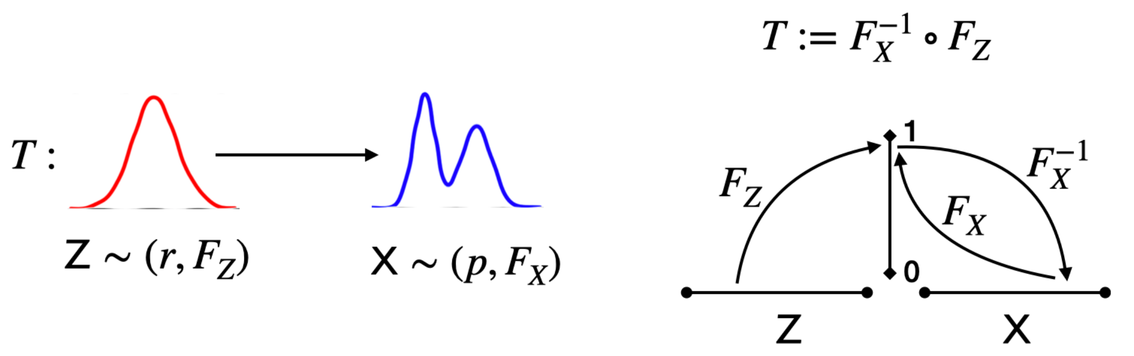
Theorem: Inverse sampling

Let Z $\sim \mathrm{Uniform}(0,1)$, F be the cdf of X, and $Q := F^{-1}$ is the quantile function of X. Then, $Q(\mathsf{Z}) \sim F$.

$$\Pr(Q(\mathsf{Z}) \leq x) = \Pr(\mathsf{Z} \leq Q^{-1}x) = \Pr(\mathsf{Z} \leq F(x)) = F(x)$$

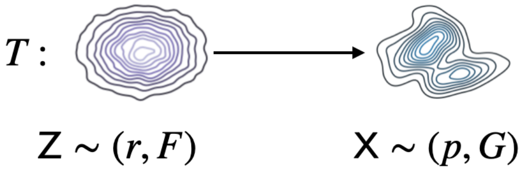
$\mathbf{Q} := \mathbf{T}$ serves as a multivariate generalization of the quantile function!

Univariate Increasing Rearrangement



- $Z \sim F_Z \implies F_Z(Z) \sim \mathrm{Uniform}(0,1)$
- $U \sim \mathrm{Uniform}(0,1) \implies F_X^{-1}(U) \sim F_X$
- $Z \sim F_Z \implies T(Z) \sim F_X$ where $T := F_X^{-1} \circ F_Z$ "rearranges" probability mass

Knothe-Rosenblatt Transform

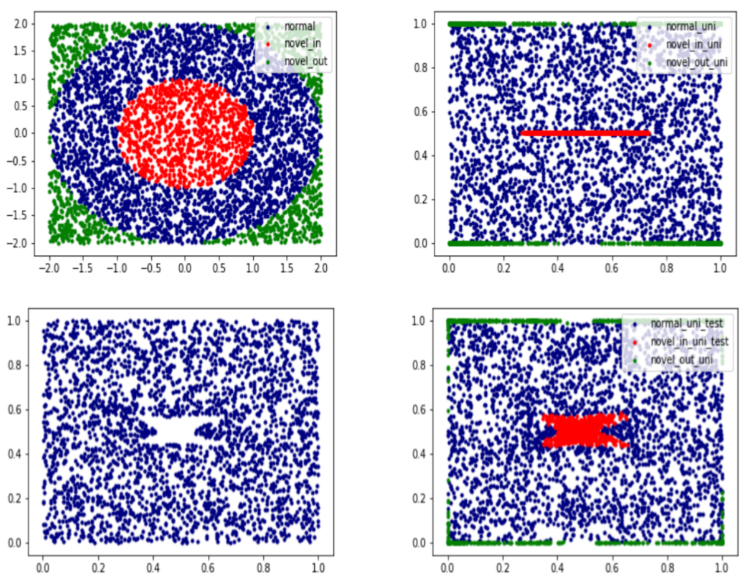


$$T_1(z_1) := G_1^{-1} \circ F_1(z_1)$$

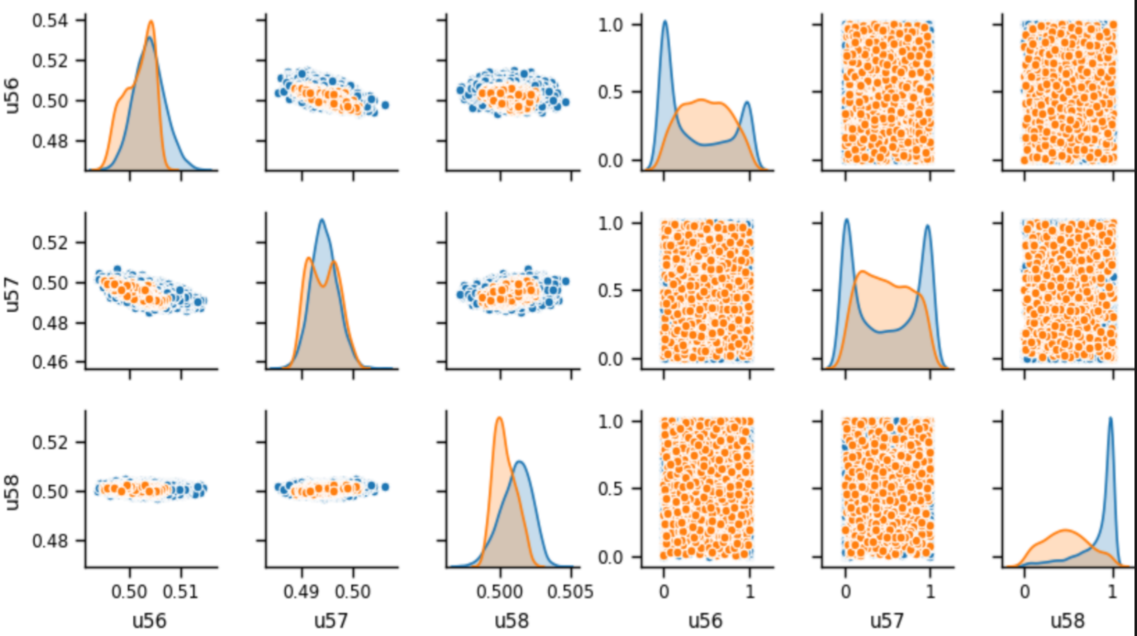
$$T_2(z_2, z_1) := G_{2|1}^{-1} \circ F_{2|1}(z_2)$$

- Iteratively apply univariate increasing rearrangement to the conditionals
- An early precursor of increasing triangular maps

H. Knothe. "Contributions to the theory of convex bodies". *The Michigan Mathematical Journal*, vol. 4, no. 1 (1957), pp. 39–52,
M. Rosenblatt. "Remarks on a Multivariate Transformation". *The Annals of Mathematical Statistics*, vol. 23, no. 3 (1952), pp. 470–472.



J. Wang et al. "Multivariate Triangular Quantile Maps for Novelty Detection". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2019.


u56
u57
u58
0.54
0.52
0.50
0.48
0.46
1.0
0.5
0.0
0.50
0.51
0.49
0.500
0.505
0
1
u56
u57
u58
u56
u57
u58