Density Estimation and Autoregressive Models

Kyungseon Lee

2026-04-08

Seoul National University

Density Estimation

- Observed data $X_1, \ldots, X_n \in \mathbb{R}^D$
- Assume they come from the same unknown distribution P_0.
- If P_0 has a density function p_0, then the probability of a set A is given by integrating p_0 over A.

$$P_0(A) = \int_A p_0(x)\, d\mu(x), \qquad A \subset \mathbb{R}^D.$$

- The goal of density estimation is to train a model p_θ that approximates the true density p_0 well.

Why is it difficult?

- **Curse of dimensionality**
- **Model flexibility**
- **Trade-off between expressiveness and tractability**
- **Untractability**: In many models, the normalization constant or the likelihood cannot be computed exactly.

Density Estimation

1. **Classical / Nonparametric**
   - Kernel density estimator, finite mixture model
2. **Unnormalized density learning**
   - Score matching, noise-contrastive estimation
3. **Autoregressive models**
   - NADE, MADE, PixelCNN, Transformer-based estimators
4. **Normalizing flows**
   - Coupling flow, autoregressive flow, continuous flow
5. **Score-based / diffusion models**
   - DDPM, score SDE

Autoregressive Density Estimation

- The joint probability density of a vector $x = (x_1, \ldots, x_D)$ can be decomposed, for any ordering o,

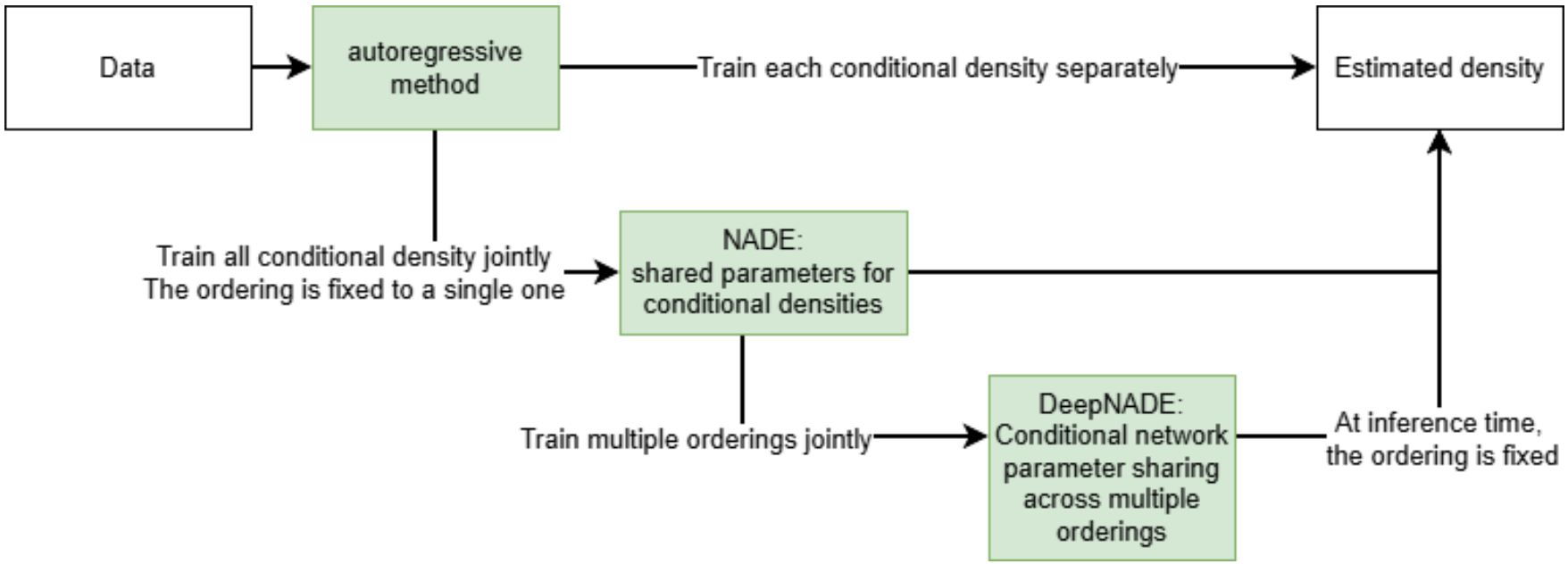
$$p(x) = \prod_{j=1}^{D} p\left(x_{o_j} \mid x_{o_1}, \ldots, x_{o_{j-1}}\right)$$

- The main advantage: **exact likelihood**

Paper Map

- **NADE – Neural Autoregressive Distribution Estimation**
  - The basic structure of a tractable autoregressive neural network model with a fixed ordering.
- **A Deep and Tractable Density Estimator**
  - Reducing the ordering dependence of NADE by using order-agnostic training.

Paper Map


Data
autoregressive method
Train each conditional density separately
Estimated density
Train all conditional density jointly
The ordering is fixed to a single one
NADE:
shared parameters for conditional densities
Train multiple orderings jointly
DeepNADE:
Conditional network parameter sharing across multiple orderings
At inference time,
the ordering is fixed

NADE

Background

- RBM (Restricted Boltzmann Machine) is powerful, but it is hard to compute the likelihood.
- On the other hand, simple directed models make likelihood computation easy, but they are not good at capturing complex high-dimensional dependence.

Problem

- The chain rule decomposition itself is exact. But if we learn each conditional distribution $p(x_d \mid x_1, \ldots, x_{d-1})$ with a separate model, the number of parameters and the computation cost become very large.

Main idea

- The key idea is that we do not need a separate model for each conditional distribution.

NADE

- For a binary vector $x = (x_1, \ldots, x_D) \in \{0, 1\}^D$ the joint distribution is written as

$$p(x) = \prod_{d=1}^{D} p(x_d \mid x_{<d}), \qquad x_{<d} = (x_1, \ldots, x_{d-1}).$$

- If the number of hidden units is H

$$h_d = \sigma(W_{\cdot,<d} x_{<d} + c), \qquad p(x_d = 1 \mid x_{<d}) = \sigma(v_d^\top h_d + b_d).$$

  - $W \in \mathbb{R}^{H \times D}$, $c \in \mathbb{R}^H$: shared input parameters used for all d.
  - $W_{\cdot,<d}$: the submatrix of W that uses only the columns before d.
  - $v_d \in \mathbb{R}^H$, $b_d \in \mathbb{R}$: output parameters for the d-th conditional distribution.

RNADE: Real-valued NADE

- For a real-valued vector $x = (x_1, \ldots, x_D) \in \mathbb{R}^D$

$$p(x) = \prod_{d=1}^{D} p(x_d \mid x_{<d}).$$

- The hidden representation is the same as in binary NADE. But each conditional distribution is modeled by a Gaussian mixture:

$$p(x_d \mid x_{<d}) = \sum_{c=1}^{C} \pi_{d,c} \mathcal{N}(x_d; \mu_{d,c}, \sigma_{d,c}^2).$$

  - C: number of mixture components.
  - $\pi_{d,c}$: mixing weight of the c-th Gaussian component.
  - $\mu_{d,c},\ \sigma_{d,c}$: mean and standard deviation of the c-th component.
  - $\mathcal{N}(x; \mu, \sigma^2)$: Gaussian density with mean μ and variance σ^2

A Deep and Tractable Density Estimator

Background

- NADE depends on one fixed ordering. The goal is to reduce this dependence on ordering.

Problem

- There are $D!$ possible orderings. So it is not realistic to train a separate NADE for every ordering.
    - To learn many orderings at once, we can sample an ordering for each data point during training.

A Deep and Tractable Density Estimator

Order-agnostic training

- Randomly choose an ordering $o = (o_1, \ldots, o_D)$, Then we train the conditional distributions $p_\theta\left(x_{o_j} \mid x_{o_1}, \ldots, x_{o_{j-1}}\right),\ j = 1, \cdots, D$ for all j using the same shared parameters θ.
- The training objective is to maximize the average log-likelihood over many orderings. This allows one network to work for many different orderings.

A Deep and Tractable Density Estimator

$$NLL(\theta) = \mathbb{E}_{o \in D!}[-\log p(X \mid \theta, o)] \tag{7}$$

$$\propto \mathbb{E}_{o \in D!} \mathbb{E}_{x^{(n)} \in X} \Big[-\log p(x^{(n)} \mid \theta, o) \Big] \tag{8}$$

$$= \mathbb{E}_{o \in D!} \mathbb{E}_{x^{(n)} \in X} \sum_{d=1}^{D} -\log p(x_{o_d}^{(n)} \mid x_{o<d}^{(n)}, \theta, o) \tag{9}$$

$$= \mathbb{E}_{x^{(n)} \in X} \sum_{d=1}^{D} \mathbb{E}_{o<d} \mathbb{E}_{o_d} \Big[-\log p(x_{o_d}^{(n)} \mid x_{o<d}^{(n)}, \theta, o < d, o_d) \Big] \tag{11}$$

- $x^{(n)} \in X$: the n-th training example in train data X
- D: number of variables, $d \in \{1, \ldots, D\}$: d-th ordering
- $o \in D!$: an ordering of the conditional densities.
- $o_{<d}$: the set of variables before position d, o_d: the variable at position d
- θ: model parameters shared across all orderings

A Deep and Tractable Density Estimator

$$\mathbb{E}_{x^{(n)} \in X} \sum_{d=1}^{D} \mathbb{E}_{o<d} \mathbb{E}_{o_d} \Big[-\log p(x_{o_d}^{(n)} \mid x_{o<d}^{(n)}, \theta, o < d, o_d) \Big] \tag{11}$$

Equation (11) can be approximated by sampling $x^{(n)}$, d, and $o_{<d}$.

$$\widehat{NLL}(\theta) = \frac{D}{D - d + 1} \sum_{o_d} -\log p(x_{o_d}^{(n)} \mid x_{o<d}^{(n)}, \theta, o < d, o_d) \tag{12}$$

A Deep and Tractable Density Estimator

What changed from NADE?

- NADE shares parameters across conditional distributions within one fixed ordering.
- Deep and Tractable Density Estimator shares parameters not only within one ordering, but also across multiple orderings.

Limitation

- At test time, we still need to choose one ordering or use an ensemble over several orderings. Because of this, the inference cost can still be high.

References

- Larochelle, H., & Murray, I. (2011). The Neural Autoregressive Distribution Estimator. AISTATS, PMLR 15:29–37.
- Benigno Uria, Iain Murray, Hugo Larochelle. *A Deep and Tractable Density Estimator*. ICML, 2014.
- Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, Hugo Larochelle. *Neural Autoregressive Distribution Estimation*. JMLR, 2016.