Deep Generative Models

Assistant Prof. Sajjad Amini

Electrical Engineering

Ehsan Merikhi   400101967

HW1

November 2, 2025

Deep Generative Models

HW1

Ehsan Merikhi 400101967

Contents

Properties of Gaussian Distributions

In this section we are going to review some basics and take a look at the properties of the Gaussian distribution!

Linear Transformation on Gaussians

Let $\overrightarrow{X} \in \mathbb{R}^d$ be Gaussian: $\overrightarrow{X} \sim \mathcal{N}(\overrightarrow{\mu}, \Sigma)$. Let A be a fixed matrix $(m \times d)$ and $\overrightarrow{b} \in \mathbb{R}^m$.
Show that $\overrightarrow{Y} = A\overrightarrow{X} + \overrightarrow{b}$ is Gaussian and give its mean and covariance.

Soloution

We will use the moment-generating function method. The moment-generating function of $\vec{X}$ is:

$$m_{\vec{X}}(\vec{t}) = E\left[e^{\vec{t}^\top \vec{X}}\right] = \exp\left(\vec{t}^\top \vec{\mu} + \frac{1}{2}\vec{t}^\top \Sigma \vec{t}\right), \quad \vec{t} \in \mathbb{R}^d$$

Now consider the moment-generating function of $\vec{Y}$:

$$m_{\vec{Y}}(\vec{s}) = E\left[e^{\vec{s}^\top \vec{Y}}\right] = E\left[e^{\vec{s}^\top (A\vec{X}+\vec{b})}\right], \quad \vec{s} \in \mathbb{R}^m$$

Simplifying:

$$m_{\vec{Y}}(\vec{s}) = E\left[e^{\vec{s}^\top A\vec{X}} e^{\vec{s}^\top \vec{b}}\right] = e^{\vec{s}^\top \vec{b}} E\left[e^{(A^\top \vec{s})^\top \vec{X}}\right]$$

Recognizing that $E\left[e^{(A^\top \vec{s})^\top \vec{X}}\right] = m_{\vec{X}}(A^\top \vec{s})$, we substitute:

$$m_{\vec{Y}}(\vec{s}) = e^{\vec{s}^\top \vec{b}} \exp\left((A^\top \vec{s})^\top \vec{\mu} + \frac{1}{2}(A^\top \vec{s})^\top \Sigma (A^\top \vec{s})\right)$$

Simplifying further:

$$m_{\vec{Y}}(\vec{s}) = \exp\left(\vec{s}^\top \vec{b} + \vec{s}^\top A\vec{\mu} + \frac{1}{2}\vec{s}^\top A\Sigma A^\top \vec{s}\right)$$

Combining the linear terms:

$$m_{\vec{Y}}(\vec{s}) = \exp\left(\vec{s}^\top (A\vec{\mu} + \vec{b}) + \frac{1}{2}\vec{s}^\top (A\Sigma A^\top)\vec{s}\right)$$

This is the moment-generating function of a Gaussian distribution with:

- Mean: $\vec{\mu}_Y = A\vec{\mu} + \vec{b}$
- Covariance: $\Sigma_Y = A\Sigma A^\top$

Therefore, $\vec{Y} = A\vec{X} + \vec{b} \sim \mathcal{N}(A\vec{\mu} + \vec{b}, A\Sigma A^\top)$.

Sum of Independent Gaussians

Let $\overrightarrow{X} \sim \mathcal{N}(\overrightarrow{\mu}_X, \Sigma_X)$ and $\overrightarrow{Y} \sim \mathcal{N}(\overrightarrow{\mu}_Y, \Sigma_Y)$ be independent vectors in $\mathbb{R}^d$.
Show that $\overrightarrow{Z} = (a\overrightarrow{X} + b\overrightarrow{Y})$ is Gaussian and compute its mean $\overrightarrow{\mu}_Z$ and covariance Σ_Z.

Soloution

We will use the moment-generating function method. Since $\vec{X}$ and $\vec{Y}$ are independent, their joint moment-generating function factors.
The moment-generating function of $\vec{X}$ is:

$$m_{\vec{X}}(\vec{t}) = E\left[e^{\vec{t}^\top \vec{X}}\right] = \exp\left(\vec{t}^\top \vec{\mu}_X + \frac{1}{2}\vec{t}^\top \Sigma_X \vec{t}\right), \quad \vec{t} \in \mathbb{R}^d$$

The moment-generating function of $\vec{Y}$ is:

$$m_{\vec{Y}}(\vec{t}) = E\left[e^{\vec{t}^\top \vec{Y}}\right] = \exp\left(\vec{t}^\top \vec{\mu}_Y + \frac{1}{2}\vec{t}^\top \Sigma_Y \vec{t}\right), \quad \vec{t} \in \mathbb{R}^d$$

Now consider the moment-generating function of $\vec{Z} = a\vec{X} + b\vec{Y}$:

$$m_{\vec{Z}}(\vec{s}) = E\left[e^{\vec{s}^\top \vec{Z}}\right] = E\left[e^{\vec{s}^\top (a\vec{X} + b\vec{Y})}\right], \quad \vec{s} \in \mathbb{R}^d$$

Simplifying:

$$m_{\vec{Z}}(\vec{s}) = E\left[e^{a\vec{s}^\top \vec{X}} e^{b\vec{s}^\top \vec{Y}}\right]$$

Since $\vec{X}$ and $\vec{Y}$ are independent:

$$m_{\vec{Z}}(\vec{s}) = E\left[e^{a\vec{s}^\top \vec{X}}\right] E\left[e^{b\vec{s}^\top \vec{Y}}\right] = m_{\vec{X}}(a\vec{s}) \cdot m_{\vec{Y}}(b\vec{s})$$

Substituting the known moment-generating functions:

$$m_{\vec{Z}}(\vec{s}) = \exp\left(a\vec{s}^\top \vec{\mu}_X + \frac{1}{2}a^2 \vec{s}^\top \Sigma_X \vec{s}\right) \cdot \exp\left(b\vec{s}^\top \vec{\mu}_Y + \frac{1}{2}b^2 \vec{s}^\top \Sigma_Y \vec{s}\right)$$

Combining the exponents:

$$m_{\vec{Z}}(\vec{s}) = \exp\left(a\vec{s}^\top \vec{\mu}_X + b\vec{s}^\top \vec{\mu}_Y + \frac{1}{2}a^2 \vec{s}^\top \Sigma_X \vec{s} + \frac{1}{2}b^2 \vec{s}^\top \Sigma_Y \vec{s}\right)$$

Rearranging terms:

$$m_{\vec{Z}}(\vec{s}) = \exp\left(\vec{s}^\top (a\vec{\mu}_X + b\vec{\mu}_Y) + \frac{1}{2}\vec{s}^\top (a^2 \Sigma_X + b^2 \Sigma_Y)\vec{s}\right)$$

This is the moment-generating function of a Gaussian distribution with:

- Mean: $\vec{\mu}_Z = a\vec{\mu}_X + b\vec{\mu}_Y$
- Covariance: $\Sigma_Z = a^2 \Sigma_X + b^2 \Sigma_Y$

Therefore, $\vec{Z} = a\vec{X} + b\vec{Y} \sim \mathcal{N}(a\vec{\mu}_X + b\vec{\mu}_Y, a^2 \Sigma_X + b^2 \Sigma_Y)$.

Conditional Gaussian (Marginalization)

Let $\overrightarrow{X} \in \mathbb{R}^{d_x}$ and $\overrightarrow{Y} \in \mathbb{R}^{d_y}$. Suppose $\overrightarrow{X} \sim \mathcal{N}(\overrightarrow{\mu}_X, \Sigma_{XX})$ and conditional on $\overrightarrow{X}$, $(\overrightarrow{Y} \mid \overrightarrow{X}) \sim \mathcal{N}(A\overrightarrow{X} + \overrightarrow{b}, \Sigma_{YY|X})$ (here A is $d_y \times d_x$ and note that $\Sigma_{YY|X}$ is the covariance matrix for $(\overrightarrow{Y} \mid \overrightarrow{X})$).

Show the marginal distribution of $\overrightarrow{Y}$ is Gaussian and compute its mean and covariance.

Soloution

We compute the marginal distribution:

$$p(\overrightarrow{Y}) = \int p(\overrightarrow{Y} \mid \overrightarrow{X}) p(\overrightarrow{X})\, d\overrightarrow{X}$$

where:

$$p(\overrightarrow{X}) = \mathcal{N}(\overrightarrow{X} \mid \overrightarrow{\mu}_X, \Sigma_{XX}), \quad p(\overrightarrow{Y} \mid \overrightarrow{X}) = \mathcal{N}(\overrightarrow{Y} \mid A\overrightarrow{X} + \overrightarrow{b}, \Sigma_{YY|X})$$

Product of Gaussians

The product is:

$$\begin{aligned} p(\overrightarrow{Y} \mid \overrightarrow{X}) p(\overrightarrow{X}) =& \frac{1}{(2\pi)^{d_y/2} |\Sigma_{YY|X}|^{1/2}} \exp\left[-\frac{1}{2} (\overrightarrow{Y} - A\overrightarrow{X} - \overrightarrow{b})^\top \Sigma_{YY|X}^{-1} (\overrightarrow{Y} - A\overrightarrow{X} - \overrightarrow{b}) \right] \\ & \times \frac{1}{(2\pi)^{d_x/2} |\Sigma_{XX}|^{1/2}} \exp\left[-\frac{1}{2} (\overrightarrow{X} - \overrightarrow{\mu}_X)^\top \Sigma_{XX}^{-1} (\overrightarrow{X} - \overrightarrow{\mu}_X) \right] \end{aligned}$$

Combine exponents:

$$Q = (\overrightarrow{Y} - A\overrightarrow{X} - \overrightarrow{b})^\top \Sigma_{YY|X}^{-1} (\overrightarrow{Y} - A\overrightarrow{X} - \overrightarrow{b}) + (\overrightarrow{X} - \overrightarrow{\mu}_X)^\top \Sigma_{XX}^{-1} (\overrightarrow{X} - \overrightarrow{\mu}_X)$$

Complete the Square in $\overrightarrow{X}$

Expand Q:

$$\begin{aligned} Q =& \overrightarrow{X}^\top A^\top \Sigma_{YY|X}^{-1} A\overrightarrow{X} - 2\overrightarrow{X}^\top A^\top \Sigma_{YY|X}^{-1} (\overrightarrow{Y} - \overrightarrow{b}) + (\overrightarrow{Y} - \overrightarrow{b})^\top \Sigma_{YY|X}^{-1} (\overrightarrow{Y} - \overrightarrow{b}) \\ & + \overrightarrow{X}^\top \Sigma_{XX}^{-1} \overrightarrow{X} - 2\overrightarrow{X}^\top \Sigma_{XX}^{-1} \overrightarrow{\mu}_X + \overrightarrow{\mu}_X^\top \Sigma_{XX}^{-1} \overrightarrow{\mu}_X \end{aligned}$$

Group terms:

$$\begin{aligned} Q =& \overrightarrow{X}^\top \left(A^\top \Sigma_{YY|X}^{-1} A + \Sigma_{XX}^{-1} \right) \overrightarrow{X} \\ & - 2\overrightarrow{X}^\top \left[A^\top \Sigma_{YY|X}^{-1} (\overrightarrow{Y} - \overrightarrow{b}) + \Sigma_{XX}^{-1} \overrightarrow{\mu}_X \right] \\ & + (\overrightarrow{Y} - \overrightarrow{b})^\top \Sigma_{YY|X}^{-1} (\overrightarrow{Y} - \overrightarrow{b}) + \overrightarrow{\mu}_X^\top \Sigma_{XX}^{-1} \overrightarrow{\mu}_X \end{aligned}$$

Let:

$$M = A^\top \Sigma_{YY|X}^{-1} A + \Sigma_{XX}^{-1}, \quad \overrightarrow{m} = A^\top \Sigma_{YY|X}^{-1} (\overrightarrow{Y} - \overrightarrow{b}) + \Sigma_{XX}^{-1} \overrightarrow{\mu}_X$$

Then:

$$Q = (\overrightarrow{X} - M^{-1}\overrightarrow{m})^\top M (\overrightarrow{X} - M^{-1}\overrightarrow{m}) + R(\overrightarrow{Y})$$

where:

$$\begin{aligned} R(\overrightarrow{Y}) =& (\overrightarrow{Y} - \overrightarrow{b})^\top \Sigma_{YY|X}^{-1} (\overrightarrow{Y} - \overrightarrow{b}) + \overrightarrow{\mu}_X^\top \Sigma_{XX}^{-1} \overrightarrow{\mu}_X \\ & - \overrightarrow{m}^\top M^{-1} \overrightarrow{m} \end{aligned}$$

Integrate Over $\overrightarrow{X}$

The integral becomes:

$$\begin{aligned}p(\overrightarrow{Y}) =& \frac{1}{(2\pi)^{(d_x+d_y)/2}|\Sigma_{XX}|^{1/2}|\Sigma_{YY|X}|^{1/2}} \\ & \times \int \exp\left[-\frac{1}{2}(\overrightarrow{X} - M^{-1}\overrightarrow{m})^\top M(\overrightarrow{X} - M^{-1}\overrightarrow{m})\right] d\overrightarrow{X} \\ & \times \exp\left[-\frac{1}{2}R(\overrightarrow{Y})\right]\end{aligned}$$

The Gaussian integral gives:

$$\int \exp\left[-\frac{1}{2}(\overrightarrow{X} - M^{-1}\overrightarrow{m})^\top M(\overrightarrow{X} - M^{-1}\overrightarrow{m})\right] d\overrightarrow{X} = (2\pi)^{d_x/2}|M|^{-1/2}$$

So:

$$p(\overrightarrow{Y}) = \frac{1}{(2\pi)^{d_y/2}|\Sigma_{XX}|^{1/2}|\Sigma_{YY|X}|^{1/2}|M|^{1/2}} \exp\left[-\frac{1}{2}R(\overrightarrow{Y})\right]$$

Simplify $R(\overrightarrow{Y})$

After algebraic manipulation (using Woodbury identity), we find:

$$R(\overrightarrow{Y}) = (\overrightarrow{Y} - \overrightarrow{\mu}_Y)^\top \Sigma_{YY}^{-1}(\overrightarrow{Y} - \overrightarrow{\mu}_Y) + \text{constant}$$

where:

$$\overrightarrow{\mu}_Y = A\overrightarrow{\mu}_X + \overrightarrow{b}, \quad \Sigma_{YY} = \Sigma_{YY|X} + A\Sigma_{XX}A^\top$$

The constant is absorbed into normalization.

Final Result

Thus:

$$p(\overrightarrow{Y}) = \frac{1}{(2\pi)^{d_y/2}|\Sigma_{YY}|^{1/2}} \exp\left[-\frac{1}{2}(\overrightarrow{Y} - \overrightarrow{\mu}_Y)^\top \Sigma_{YY}^{-1}(\overrightarrow{Y} - \overrightarrow{\mu}_Y)\right]$$

i.e.,

$$\overrightarrow{Y} \sim \mathcal{N}(\overrightarrow{\mu}_Y, \Sigma_{YY})$$

with:

$$\overrightarrow{\mu}_Y = A\overrightarrow{\mu}_X + \overrightarrow{b}, \quad \Sigma_{YY} = \Sigma_{YY|X} + A\Sigma_{XX}A^\top$$

General Joint Gaussian Conditional Distribution

Let $\begin{pmatrix}\vec{X}\\ \vec{Y}\end{pmatrix}$ be jointly Gaussian with mean $\begin{pmatrix}\vec{\mu}_X\\ \vec{\mu}_Y\end{pmatrix}$ and covariance blocks $\Sigma = \begin{pmatrix}\Sigma_{XX} & \Sigma_{XY}\\ \Sigma_{YX} & \Sigma_{YY}\end{pmatrix}$.
Assume Σ_{XX} invertible.
Derive the conditional distribution $(\vec{Y} \mid \vec{X} = \vec{x})$ (mean and covariance).

Soloution

Let:

$$\vec{m} = \vec{\mu}_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(\vec{x} - \vec{\mu}_X).$$

This represents the conditional mean of $\vec{Y}$ given $\vec{X} = \vec{x}$. The term $\Sigma_{YX}\Sigma_{XX}^{-1}$ acts as a regression matrix that linearly transforms the deviation of $\vec{X}$ from its mean to predict the corresponding deviation in $\vec{Y}$.
Then, after algebraic manipulation, we obtain:

$$Q = (\vec{y} - \vec{m})^\top \Sigma_{YY|X}^{-1}(\vec{y} - \vec{m}) + (\vec{x} - \vec{\mu}_X)^\top \Sigma_{XX}^{-1}(\vec{x} - \vec{\mu}_X).$$

This shows that the joint quadratic form Q separates into two independent parts: one involving only $\vec{y}$ (centered around the conditional mean $\vec{m}$) and one involving only $\vec{x}$.

Conditional Density

The conditional density is:

$$p(\vec{y} \mid \vec{x}) = \frac{p(\vec{x}, \vec{y})}{p(\vec{x})}.$$

The marginal $p(\vec{x})$ is $\mathcal{N}(\vec{\mu}_X, \Sigma_{XX})$, so:

$$p(\vec{x}) \propto \exp\left[-\frac{1}{2}(\vec{x} - \vec{\mu}_X)^\top \Sigma_{XX}^{-1}(\vec{x} - \vec{\mu}_X)\right].$$

Therefore:

$$p(\vec{y} \mid \vec{x}) \propto \exp\left[-\frac{1}{2}(\vec{y} - \vec{m})^\top \Sigma_{YY|X}^{-1}(\vec{y} - \vec{m})\right].$$

This is the density of a Gaussian distribution with mean $\vec{m}$ and covariance $\Sigma_{YY|X}$. The $\vec{x}$-dependent terms cancel out in the ratio, leaving only the $\vec{y}$-dependent quadratic form.

Final Result

The conditional distribution is:

$$(\vec{Y} \mid \vec{X} = \vec{x}) \sim \mathcal{N}(\vec{\mu}_{Y|X}, \Sigma_{YY|X}),$$

where:

$$\vec{\mu}_{Y|X} = \vec{\mu}_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(\vec{x} - \vec{\mu}_X),$$
$$\Sigma_{YY|X} = \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}.$$

The conditional covariance $\Sigma_{YY|X}$ represents the remaining uncertainty in $\vec{Y}$ after accounting for the information provided by $\vec{X}$. It is always smaller (in the positive definite sense) than the marginal covariance Σ_{YY}.

KL Divergence between Two Multivariate Gaussians (Closed Form)

KL divergence is one of the mathematical tools for measuring the difference between two probability distributions. Write down the KL divergence $D_{KL}(\mathcal{N}(\vec{\mu}_0, \Sigma_0)\|\mathcal{N}(\vec{\mu_1}, \Sigma_1))$ for full-rank covariances $\Sigma_0, \Sigma_1 \in \mathbb{R}^{d\times d}$. Write it down for the case $d = 1$ as well.

Soloution

Definition of KL Divergence

The Kullback-Leibler (KL) divergence between two probability distributions P and Q is defined as:

$$D_{KL}(P\|Q) = \int p(\vec{x}) \log\left(\frac{p(\vec{x})}{q(\vec{x})}\right) d\vec{x}$$

For two multivariate Gaussian distributions:

- $P = \mathcal{N}(\vec{\mu}_0, \Sigma_0)$
- $Q = \mathcal{N}(\vec{\mu}_1, \Sigma_1)$

we can compute this integral in closed form.

Multivariate Gaussian Case (d-dimensional)

The probability density functions are:

$$p(\vec{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma_0|^{1/2}} \exp\left[-\frac{1}{2}(\vec{x} - \vec{\mu}_0)^\top \Sigma_0^{-1}(\vec{x} - \vec{\mu}_0)\right]$$
$$q(\vec{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma_1|^{1/2}} \exp\left[-\frac{1}{2}(\vec{x} - \vec{\mu}_1)^\top \Sigma_1^{-1}(\vec{x} - \vec{\mu}_1)\right]$$

The log-ratio is:

$$\log\left(\frac{p(\vec{x})}{q(\vec{x})}\right) = \frac{1}{2}\left[\log\left(\frac{|\Sigma_1|}{|\Sigma_0|}\right) - (\vec{x} - \vec{\mu}_0)^\top \Sigma_0^{-1}(\vec{x} - \vec{\mu}_0) + (\vec{x} - \vec{\mu}_1)^\top \Sigma_1^{-1}(\vec{x} - \vec{\mu}_1)\right]$$

Taking the expectation under P gives:

$$D_{KL}(P\|Q) = \frac{1}{2}\left[\log\left(\frac{|\Sigma_1|}{|\Sigma_0|}\right) - \mathbb{E}_P[(\vec{x} - \vec{\mu}_0)^\top \Sigma_0^{-1}(\vec{x} - \vec{\mu}_0)] + \mathbb{E}_P[(\vec{x} - \vec{\mu}_1)^\top \Sigma_1^{-1}(\vec{x} - \vec{\mu}_1)]\right]$$

Using properties of Gaussian expectations:

- $\mathbb{E}_P[(\vec{x} - \vec{\mu}_0)^\top \Sigma_0^{-1}(\vec{x} - \vec{\mu}_0)] = \mathrm{tr}(\Sigma_0^{-1}\Sigma_0) = d$
- $\mathbb{E}_P[(\vec{x} - \vec{\mu}_1)^\top \Sigma_1^{-1}(\vec{x} - \vec{\mu}_1)] = \mathrm{tr}(\Sigma_1^{-1}\Sigma_0) + (\vec{\mu}_1 - \vec{\mu}_0)^\top \Sigma_1^{-1}(\vec{\mu}_1 - \vec{\mu}_0)$

Thus, the final formula is:

$$\boxed{D_{KL}(\mathcal{N}(\vec{\mu}_0, \Sigma_0)\|\mathcal{N}(\vec{\mu}_1, \Sigma_1)) = \frac{1}{2}\left[\log\left(\frac{|\Sigma_1|}{|\Sigma_0|}\right) - d + \mathrm{tr}(\Sigma_1^{-1}\Sigma_0) + (\vec{\mu}_1 - \vec{\mu}_0)^\top \Sigma_1^{-1}(\vec{\mu}_1 - \vec{\mu}_0)\right]}$$

Univariate Case ($d = 1$)

For one-dimensional Gaussians:

- $P = \mathcal{N}(\mu_0, \sigma_0^2)$
- $Q = \mathcal{N}(\mu_1, \sigma_1^2)$

The formula simplifies to:

$$\boxed{D_{KL}(\mathcal{N}(\mu_0, \sigma_0^2) \| \mathcal{N}(\mu_1, \sigma_1^2)) = \log\left(\frac{\sigma_1}{\sigma_0}\right) + \frac{\sigma_0^2 + (\mu_0 - \mu_1)^2}{2\sigma_1^2} - \frac{1}{2}}$$

This can be derived from the multivariate formula by noting:

- $\operatorname{tr}(\Sigma_1^{-1}\Sigma_0) = \frac{\sigma_0^2}{\sigma_1^2}$
- $(\vec{\mu}_1 - \vec{\mu}_0)^\top \Sigma_1^{-1} (\vec{\mu}_1 - \vec{\mu}_0) = \frac{(\mu_1 - \mu_0)^2}{\sigma_1^2}$

Autoregressive (AR) Models of Order p

An Autoregressive model of order p, denoted as AR(p), is used to describe a time-dependent process where the current value depends linearly on its previous p values plus a random error term. The model is expressed as:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

where: y_t is the observation at time t, $\phi_1, \phi_2, \ldots, \phi_p$ are the parameters (coefficients) of the model and ε_t is the error term at time t, assumed i.i.d. with a normal distribution $N(0, \sigma^2)$. You are required to estimate the parameters $\phi_1, \phi_2, \ldots, \phi_p$ and σ^2 that maximize the likelihood function given the observed data $y_1, y_2, \ldots, y_n$.

Likelihood Function

Formulate the likelihood function for the AR(p) model based on the assumption that the error terms are normally distributed and independent. Then derive the log-likelihood function in terms of $\phi_1, \phi_2, \ldots, \phi_p$ and σ^2.

Soloution

Assume when $t - i < 0$, the corresponding y_{t-i} is replaced with 0.
The AR(p) model is:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma^2) \text{ i.i.d.}$$

For $t = 1, \ldots, n$, the conditional density of y_t given $y_{t-1}, \ldots, y_{t-p}$ is:

$$f(y_t \mid y_{t-1}, \ldots, y_{t-p}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2} \left(y_t - \sum_{i=1}^{p} \phi_i y_{t-i} \right)^2 \right].$$

We use the conditional MLE approach. The conditional likelihood function is:

$$L(\phi_1, \ldots, \phi_p, \sigma^2) = \prod_{t=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2} \left(y_t - \sum_{i=1}^{p} \phi_i y_{t-i} \right)^2 \right].$$

The log-likelihood function is:

$$\ell(\boldsymbol{\phi}, \sigma^2) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{t=p+1}^{n} \left(y_t - \sum_{i=1}^{p} \phi_i y_{t-i} \right)^2.$$

Let

$$S(\phi) = \sum_{t=p+1}^{n} \left(y_t - \sum_{i=1}^{p} \phi_i y_{t-i} \right)^2.$$

Then:

$$\ell(\phi, \sigma^2) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{S(\phi)}{2\sigma^2}.$$

Maximum Likelihood Estimation (MLE)

Find the set of equations for parameter values that maximize the log-likelihood function and provide explicit formulas for the estimators of $\phi_1, \phi_2, \ldots, \phi_p$ and σ^2. Include detailed derivations of the likelihood and log-likelihood functions and explain each step clearly.

Soloution

To find the maximum likelihood estimators, we maximize $\ell(\phi, \sigma^2)$ with respect to $\phi_1, \ldots, \phi_p$ and σ^2.

Estimation of $\phi_1, \ldots, \phi_p$

For fixed σ^2, maximizing $\ell(\boldsymbol{\phi}, \sigma^2)$ is equivalent to minimizing $S(\boldsymbol{\phi})$:

$$\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}} \sum_{t=p+1}^{n} \left(y_t - \sum_{i=1}^{p} \phi_i y_{t-i} \right)^2$$

This is a linear regression problem. Define:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad X = \begin{bmatrix} y_1 & 0 & \cdots & 0 \\ y_2 & y_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ y_{n-1} & y_{n-2} & \cdots & y_{n-p} \end{bmatrix}, \quad \boldsymbol{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix}$$

The normal equations are:

$$X\hat{\boldsymbol{\phi}} = Y \implies X^T X \hat{\boldsymbol{\phi}} = X^T Y$$

Solving for $\hat{\boldsymbol{\phi}}$ gives:

$$\hat{\boldsymbol{\phi}} = (X^T X)^{-1} X^T Y$$

Estimation of σ^2

Differentiate the log-likelihood with respect to σ^2 and set the derivative to zero:

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{S(\boldsymbol{\phi})}{2(\sigma^2)^2} = 0 \implies \hat{\sigma}^2 = \frac{S(\hat{\boldsymbol{\phi}})}{n}$$

Therefore:

$$\hat{\sigma}^2 = \frac{S(\hat{\boldsymbol{\phi}})}{n} = \frac{1}{n} \sum_{t=1}^{n} \left(y_t - \sum_{i=1}^{p} \hat{\boldsymbol{\phi}}_i y_{t-i} \right)^2$$

Linking AR(p) and Markov property

Prove that an AR(1) process

$$y_t = \phi y_{t-1} + \varepsilon_t,$$

where $\{\varepsilon_t\}$ is a white noise sequence, is a (time-homogeneous) Markov process of order 1.

(a) Derive the one-step transition density $p(y_t \mid y_{t-1})$.

Soloution

Given y_{t-1}, the distribution of y_t is determined by ε_t. We know ε_t is independent of the past and normally distributed, hence y_t is a normal distribution with a mean shift of ϕy_{t-1}.
The one-step transition density is:

$$y_t \mid y_{t-1} \sim N(\phi y_{t-1}, \sigma^2). \implies p(y_t \mid y_{t-1}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_t - \phi y_{t-1})^2}{2\sigma^2} \right).$$

(b) Assuming $|\phi| < 1$, derive the stationary distribution $\pi(y)$ of y_t and show its mean and variance.

Soloution

The stationary distribution $\pi(y)$ is the distribution that does not evolve with time.
Let $\pi(y)$ be a normal distribution with mean μ and variance ν^2. Then from the model:

$$y_t = \phi y_{t-1} + \varepsilon_t,$$

we have $\mathbb{E}[y_t] = \phi\mathbb{E}[y_{t-1}]$, so $\mu = \phi\mu$, hence $\mu = 0$.
For the variance:

$$\mathrm{Var}(y_t) = \phi^2 \mathrm{Var}(y_{t-1}) + \sigma^2,$$

so $\nu^2 = \phi^2\nu^2 + \sigma^2$, giving $\nu^2 = \frac{\sigma^2}{1-\phi^2}$.
Therefore, the stationary distribution is:

$$\pi(y) = \frac{1}{\sqrt{2\pi\frac{\sigma^2}{1-\phi^2}}} \exp\left(-\frac{y^2}{2\frac{\sigma^2}{1-\phi^2}}\right),$$

with mean 0 and variance $\frac{\sigma^2}{1-\phi^2}$.

Nonlinear and Multi-Step Autoregressive Models

In this problem, you will study nonlinear extensions of autoregressive (AR) models and explore how to use them for prediction and sequence generation. You will also examine issues of stability and long-term behavior in autoregressive generative models.

Nonlinear Autoregressive Model

Consider a sequence of real-valued random variables $x = (x_1, x_2, ..., x_T)$. Suppose we model the conditional distribution of each x_t given its past as:

$$p_\theta(x_t|x_{<t}) = \mathcal{N}(f_\theta(x_{t-1}, ..., x_{t-p}), \sigma^2)$$

where: f_θ is a nonlinear function parametrized by a neural network with parameters θ, σ^2 is a fixed variance and p is the model order. This model is referred to as a nonlinear autoregressive (NAR) model.

- Derive the NLL (Negative Log-Likelihood) of the observed data sequence $x_1, x_2, ..., x_T$ under this model.

Soloution

1. Negative Log-Likelihood (NLL) Derivation The joint probability of the sequence $x_1, x_2, \ldots, x_T$ under the nonlinear autoregressive (NAR) model is:

$$p_\theta(x_1, \ldots, x_T) = p(x_1, \ldots, x_p) \prod_{t=p+1}^{T} \mathcal{N}\left(x_t \mid f_\theta(x_{t-1}, \ldots, x_{t-p}), \sigma^2\right),$$

where $p(x_1, \ldots, x_p)$ is the initial distribution. The negative log-likelihood (NLL) is:

$$\text{NLL} = -\log p_\theta(x_1, \ldots, x_T) = -\sum_{t=p+1}^{T} \log \mathcal{N}\left(x_t \mid f_\theta(x_{t-1}, \ldots, x_{t-p}), \sigma^2\right) + \text{constant}.$$

Substituting the Gaussian density:

$$\text{NLL} = \frac{1}{2\sigma^2} \sum_{t=p+1}^{T} \left(x_t - f_\theta(x_{t-1}, \ldots, x_{t-p})\right)^2 + \frac{T-p}{2} \log(2\pi\sigma^2).$$

The constant term is often dropped during optimization, reducing the NLL to a mean squared error (MSE) objective.

- Then explain how the gradients of the NLL with respect to θ can be computed using backpropagation through time.

Soloution

2. Gradient Computation via Backpropagation Through Time (BPTT) The gradient of the NLL with respect to θ is:

$$\nabla_\theta \text{NLL} = -\frac{1}{\sigma^2} \sum_{t=p+1}^{T} \left(x_t - f_\theta(x_{t-1}, \ldots, x_{t-p})\right) \nabla_\theta f_\theta(x_{t-1}, \ldots, x_{t-p}).$$

Since f_θ is a neural network, $\nabla_\theta f_\theta$ is computed using backpropagation. For sequences, this is extended to **backpropagation through time (BPTT)**:

- The computation graph is "unrolled" over time steps, with shared parameters θ.
- Gradients are accumulated across all time steps $t = p+1, \ldots, T$.
- The chain rule is applied recursively through the network's layers and across time.

- Discuss how the autoregressive structure affects gradient flow, especially for large p or long sequences T.

Soloution

3. Autoregressive Structure and Gradient Flow The autoregressive structure impacts gradient flow as follows:

- **Vanishing/Exploding Gradients**: For large p or T, repeated application of the chain rule can cause gradients to vanish (e.g., with saturating activations like sigmoid) or explode (e.g., with unstable dynamics or large weights).

Architecture for large vs. small context

Assume that the model order p can vary from very small (e.g. $p = 1$) to very large ($p \gg 1$, full context).

- When $p \gg 1$, what types of neural architectures would you recommend for the nonlinear function f_θ? Discuss design choices and practical limitations.

Soloution

Large Context ($p \gg 1$) When the model order p is very large, the nonlinear function f_θ must effectively capture long-range dependencies while remaining computationally feasible. Recommended architectures include:

- **Recurrent Neural Networks (RNNs) with Gating Mechanisms**:
  - **LSTMs (Long Short-Term Memory)**: Use input, forget, and output gates to control information flow, mitigating vanishing gradient problems.
  - **GRUs (Gated Recurrent Units)**: Simplified gating mechanism with reset and update gates, offering similar benefits with fewer parameters.
  - Both architectures can theoretically handle infinite context, though practical limitations exist.
- **Attention Mechanisms and Transformers**:
  - **Self-attention**: Computes weighted combinations of all previous time steps, allowing direct access to relevant historical information.
  - **Causal masking**: Ensures autoregressive property by masking future time steps.
  - Excellent for capturing long-range dependencies but computationally intensive with $O(T^2)$ complexity.

- When p is small, what architectures are more suitable and why? Compare them in terms of accuracy, computational efficiency, and training stability.

Soloution

Small Context (p small) When p is small, simpler architectures are often sufficient and more efficient:

- - **Multilayer Perceptrons (MLPs)**:
    - ∗ Direct mapping from p inputs to output prediction.
    - ∗ Simple, fast to train and evaluate, with stable gradients.
    - ∗ Limited capacity but effective when true dependencies are short-range.
  - **Convolutional Neural Networks (CNNs)**:
    - ∗ Use 1D convolutions with kernel size $\leq p$.
    - ∗ Parameter sharing across time provides efficient feature extraction.
    - ∗ Good inductive bias for local temporal patterns.
  - **Comparison**:
    - ∗ **Accuracy**: For small p, MLPs and CNNs often match or exceed complex architectures by avoiding overfitting.
    - ∗ **Computational Efficiency**: MLPs/CNNs have $O(1)$ inference time per step vs. $O(T)$ for RNNs; much faster training than Transformers.
    - ∗ **Training Stability**: Simpler architectures have fewer gradient issues, converge faster, and require less hyperparameter tuning.

Multi-Step Prediction

In practice, we often want to predict not just the next value x_{t+1}, but several steps into the future. Let $\hat{x}_{t+k}$ denote the model's k-step-ahead prediction at time t, obtained recursively by feeding back previous predictions into the model.

- Write down the recursive equations for generating k-step predictions $\hat{x}_{t+1}, \hat{x}_{t+2}, ..., \hat{x}_{t+k}$ given the trained model $f_\theta(\cdot)$ and the most recent p true observations $x_{t-p+1:t}$.

Soloution

Recursive equations for k-step predictions:
Given a trained AR(p) model $f_\theta(x_{t-1}, \ldots, x_{t-p})$ and the most recent p true observations $x_{t-p+1:t}$, the k-step predictions are generated recursively as follows:
For $j = 1$:

$$\hat{x}_{t+1} = f_\theta(x_{t-1}, \ldots, x_{t-p})$$

For $j = 2$ to k:

$$\hat{x}_{t+j} = f_\theta(\hat{x}_{t+j-1}, \ldots, \hat{x}_{t+j-p})$$

where for $i \geq 1$, $\hat{x}_{t+i} = x_{t+i}$ if $t + i \leq t$ (using actual observations when available), otherwise using previous predictions.

- Explain mathematically why prediction errors in autoregressive models tend to accumulate over multiple steps ahead & Propose a way to mitigate this issue when training or evaluating the model.

Soloution

Prediction error accumulation:
Let $\varepsilon_{t+1} = x_{t+1} - \hat{x}_{t+1}$ be the one-step prediction error. For k-step predictions, errors accumulate because the model recursively uses its own predictions as inputs:

$$\hat{x}_{t+k} = f_\theta(x_{t+k-1}, x_{t+k-2}, \ldots, x_{t+k-p}) + \varepsilon_{t+k}$$
$$\hat{x}_{t+k} = f_\theta(\hat{x}_{t+k-1}, \hat{x}_{t+k-2}, \ldots, \hat{x}_{t+k-p})$$

The prediction error $e_{t+j} = x_{t+j} - \hat{x}_{t+j}$ propagates through the recursive application of f_θ. Since f_θ is generally a nonlinear function, the error dynamics are complex, but errors from previous steps directly affect future predictions through the input sequence.
Mitigation strategy:
To mitigate error accumulation:

- **Direct multi-step training**: Instead of training only for one-step prediction, include multi-step prediction loss during training:

$$\mathcal{L}(\theta) = \sum_{k=1}^{K} w_k \mathbb{E}[(x_{t+k} - \hat{x}_{t+k})^2]$$

  where w_k are weights prioritizing different prediction horizons.

- **Scheduled sampling**: During training, randomly replace some inputs with model predictions rather than true observations to expose the model to its own prediction errors.

- **Ensemble methods**: Generate multiple prediction trajectories with different noise realizations and average them to reduce variance.

Conditional Nonlinear Autoregressive Modeling

Suppose now you want to model a sequence $x = (x_1, x_2, ..., x_T)$ conditional on another sequence $c = (c_1, c_2, ..., c_T)$. Define the conditional AR model:

$$p_\theta(x_t | x_{<t}, c_{1:t}) = \mathcal{N}(f_\theta(x_{t-1}, ..., x_{t-p}; c_{1:t}), \sigma^2)$$

- Derive the expression for the conditional log-likelihood of the full sequence $x_{1:T}$ given $c_{1:T}$, and write the training objective that maximizes this conditional likelihood.

Soloution

The conditional log-likelihood of the full sequence $x_{1:T}$ given $c_{1:T}$ can be derived by factorizing the joint conditional distribution using the autoregressive structure:

$$p_\theta(x_{1:T} \mid c_{1:T}) = \prod_{t=1}^{T} p_\theta(x_t \mid x_{<t}, c_{1:t})$$

Assuming the model starts at $t = 1$, and using the given conditional AR formulation:

$$p_\theta(x_t \mid x_{<t}, c_{1:t}) = \mathcal{N}(f_\theta(x_{t-1}, \ldots, x_{t-p}; c_{1:t}), \sigma^2)$$

The conditional log-likelihood is:

$$\log p_\theta(x_{1:T} \mid c_{1:T}) = \sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t}, c_{1:t})$$

$$= -\frac{T}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{T}\left(x_t - f_\theta(x_{t-1}, \ldots, x_{t-p}; c_{1:t})\right)^2$$

The training objective that maximizes this conditional likelihood is:

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t}, c_{1:t}) = -\frac{1}{2\sigma^2}\sum_{t=1}^{T}\left(x_t - f_\theta(x_{t-1}, \ldots, x_{t-p}; c_{1:t})\right)^2 + \text{constant}$$

- Explain how this model could be used for conditional generation (e.g., generating speech given text, or motion given control inputs).

Soloution

This conditional AR model can be used for conditional generation in the following way:

1. **Initialization**: Start with initial values $x_{1-p}, \ldots, x_0$ (which could be zeros, random, or specified based on the task).
2. **Recursive generation**: For each time step $t = 1, 2, \ldots, T$:
   - Compute the conditional mean: $\mu_t = f_\theta(x_{t-1}, \ldots, x_{t-p}; c_{1:t})$
   - Sample $x_t \sim \mathcal{N}(\mu_t, \sigma^2)$ (or take μ_t directly for deterministic generation)

- Compare conditional and unconditional AR models in terms of: The factorization of their joint distributions, The dependence structure, The types of data they can model.

Soloution

Comparison between conditional and unconditional AR models:

- **Joint distribution factorization**:
  - **Unconditional**: $p(x_{1:T}) = \prod_{t=1}^{T} p(x_t \mid x_{<t})$
  - **Conditional**: $p(x_{1:T} \mid c_{1:T}) = \prod_{t=1}^{T} p(x_t \mid x_{<t}, c_{1:t})$
- **Dependence structure**:
  - **Unconditional**: Depends only on past outputs $x_{<t}$
  - **Conditional**: Depends on both past outputs $x_{<t}$ and current/past conditioning variables $c_{1:t}$
- **Types of data they can model**:
  - **Unconditional**: Suitable for modeling single time series without external inputs (e.g., stock prices, weather patterns)
  - **Conditional**: Can model complex input-output relationships (e.g., text-to-speech, controller-to-motion, prompt-to-text)

KL Divergence between two conditional NAR models

Consider two conditional autoregressive models, one parameterized by f_θ and another by g_ϕ. Write down the expression for the Kullback-Leibler (KL) divergence between the two distributions $p_\theta(x)$ and $q_\phi(x)$. Discuss the computational challenges of directly evaluating this divergence and propose a practical approximation method for high-dimensional sequences.
(Hint: You can describe Monte Carlo sampling here.)

Soloution

The KL divergence between two conditional autoregressive distributions $p_\theta(x_{1:T} \mid c_{1:T})$ and $q_\phi(x_{1:T} \mid c_{1:T})$ is defined as:

$$D_{KL}(p_\theta \| q_\phi) = \mathbb{E}_{x_{1:T} \sim p_\theta} \left[\log \frac{p_\theta(x_{1:T} \mid c_{1:T})}{q_\phi(x_{1:T} \mid c_{1:T})} \right]$$

Using the autoregressive factorization for both models:

$$p_\theta(x_{1:T} \mid c_{1:T}) = \prod_{t=1}^{T} p_\theta(x_t \mid x_{<t}, c_{1:t}) \quad \& \quad q_\phi(x_{1:T} \mid c_{1:T}) = \prod_{t=1}^{T} q_\phi(x_t \mid x_{<t}, c_{1:t})$$

The KL divergence becomes:

$$D_{KL}(p_\theta \| q_\phi) = \mathbb{E}_{x_{1:T} \sim p_\theta} \left[\sum_{t=1}^{T} \log \frac{p_\theta(x_t \mid x_{<t}, c_{1:t})}{q_\phi(x_t \mid x_{<t}, c_{1:t})} \right]$$

Computational challenges:

- **High-dimensional expectation**: The expectation over the entire sequence $x_{1:T}$ requires integration over T dimensions, which becomes intractable for long sequences.
- **Autoregressive sampling**: To compute the expectation exactly, we would need to evaluate all possible sequences, which is computationally infeasible for high-dimensional x_t.
- **Exponential complexity**: The number of possible sequences grows exponentially with sequence length T.

Practical approximation method:
We can use Monte Carlo sampling to approximate the KL divergence:
This Monte Carlo estimator is unbiased and converges to the true KL divergence as $N \to \infty$, making it practical for high-dimensional sequences.

1. Sample N sequences from p_θ: $\{x_{1:T}^{(i)}\}_{i=1}^{N} \sim p_\theta(x_{1:T} \mid c_{1:T})$
2. For each sample, compute the log-ratio:

$$r^{(i)} = \sum_{t=1}^{T} \log \frac{p_\theta(x_t^{(i)} \mid x_{<t}^{(i)}, c_{1:t})}{q_\phi(x_t^{(i)} \mid x_{<t}^{(i)}, c_{1:t})}$$

3. Approximate the KL divergence:

$$D_{KL}(p_\theta \| q_\phi) \approx \frac{1}{N} \sum_{i=1}^{N} r^{(i)}$$

Real NADE Parameters

In this problem, you will study an extension of the Real NADE model. Recall that, given an autoregressive model

$$p(x) = p(x_1)p(x_2 \mid x_{<2}) \cdots p(x_i \mid x_{<i}) \cdots p(x_n \mid x_{<n})$$

and Real NADE models the conditional distribution as

$$p(x_1) = \mathcal{N}(x_1 \mid \mu_1, \exp(s_1))$$

$$\cdots$$

$$p(x_i \mid x_{<i}; W, c, v_i, b_i, u_i, d_i) = \mathcal{N}\left(x_i \mid v_i^\top h_i + b_i, \exp(u_i^\top h_i + d_i)\right),$$

where $h_i, c, v_i, u_i \in \mathbb{R}^d$. Now, we would like to make $p(x_i \mid x_{<i})$ follow a mixture of Gaussians:

$$p(x_i \mid x_{<i}) = \sum_{c=1}^{C} \pi_i^c \mathcal{N}(\mu_i^c, (\sigma_i^c)^2),$$

where $\sum_{c=1}^{C} \pi_i^c = 1$. Now the question is: How do you propose to parameterize $\pi_i^c, \mu_i^c, \sigma_i^c, \forall c \in \{1, \ldots, C\}$ as a function of h_i? Describe the parameters required and the total number of parameters required for a single $p(x_i \mid x_{<i})$.

Soloution

To model each conditional distribution $p(x_i \mid x_{<i})$ as a Gaussian mixture with C components, all mixture parameters are parameterized as functions of the hidden state $h_i \in \mathbb{R}^d$:

- **Mixing coefficients** π_i^c: Use a softmax transformation to ensure they sum to 1:

$$\alpha_i^c = w_\pi^c \cdot h_i + b_\pi^c$$

- **Means** μ_i^c: Use a linear transformation:

$$w_\mu^c \cdot h_i + b_\mu^c$$

- **Standard deviations** σ_i^c: Use the exponential function to ensure positivity:

$$\exp(w_\sigma^c \cdot h_i + b_\sigma^c)$$

Parameter Count:

- Each component requires $3d + 3$ parameters (3 weight vectors + 3 biases)
- Total for C components: $3C(d + 1)$ parameters per conditional distribution

This parameterization enables flexible, context-dependent mixture distributions that adapt based on previous inputs $x_{<i}$ through the hidden state h_i.