Chapter 10
Overview of Generative Models

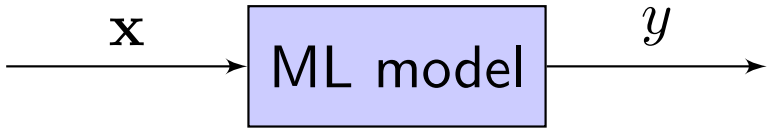
supplementary slides to
Machine Learning Fundamentals

published by Cambridge University Press

August 2020

Outline

Discriminative Models in ML: Review

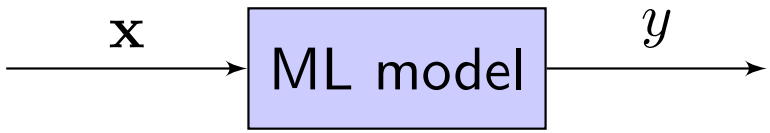


- input $\mathbf{x}$ is a random vector: $\mathbf{x} \sim p(\mathbf{x})$
- output y is generated by an unknown but *deterministic target* function $y = \bar{f}(\mathbf{x})$ for each input $\mathbf{x}$
- our goal: estimate $\bar{f}(\cdot)$ in a model space $\mathbb{H}$
- use a training set: $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \sim p(\mathbf{x})$ and $y_i = \bar{f}(\mathbf{x}_i)$
- choose a loss function $l(y, y')$, and minimize the empirical risk:

$$f^* = \arg\min_{f \in \mathbb{H}} R_{\text{emp}}(f|\mathcal{D}) = \arg\min_{f \in \mathbb{H}} \sum_{i=1}^{N} l(y_i, f(\mathbf{x}_i))$$

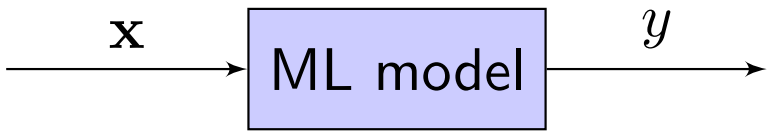
- finial performance depends on the generalization bound

Generative Models in ML



- input $\mathbf{x}$ and output y are random variables drawn from an unknown joint distribution, i.e. $(\mathbf{x}, y) \sim p(\mathbf{x}, y)$
- the relation $\mathbf{x} \rightarrow y$ is stochastic, solely relies on $p(y|\mathbf{x})$
- our goal: estimate $p(\mathbf{x}, y)$ using a probabilistic model $\hat{p}_\theta(\mathbf{x}, y)$
- use a training set: $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$, where $(\mathbf{x}_i, y_i) \sim p(\mathbf{x}, y)$
- the relation $\mathbf{x} \rightarrow y$ may be approximated by $\hat{p}_\theta(y|\mathbf{x})$
- final performance relies on the gap between $p(\mathbf{x}, y)$ and $\hat{p}_\theta(\mathbf{x}, y)$, e.g. $\mathrm{KL}\big(p(\cdot) \,||\, \hat{p}_\theta(\cdot)\big)$

Discriminative vs. Generative Models: Recap



the goal: to estimate an ML model to predict output y from input $\mathbf{x}$ based on some samples $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$

discriminative models

- data generation assumption: $\mathbf{x}_i \sim p(\mathbf{x})$ and $y_i = \bar{f}(\mathbf{x}_i)$
- $\mathbf{x} \rightarrow y$ is deterministic: $y = \bar{f}(\mathbf{x})$
- use $\mathcal{D}$ to estimate the target function: $y = \bar{f}(\mathbf{x})$

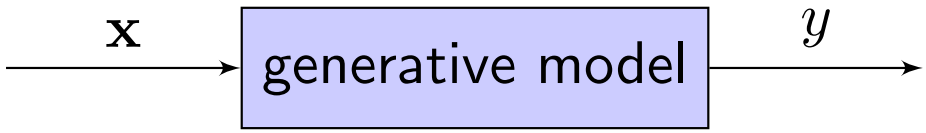
generative models

- data generation assumption: $(\mathbf{x}_i, y_i) \sim p(\mathbf{x}, y)$
- $\mathbf{x} \rightarrow y$ is stochastic: $p(y|\mathbf{x})$
- use $\mathcal{D}$ to estimate the joint distribution: $p(\mathbf{x}, y)$

Deterministic vs. Stochastic

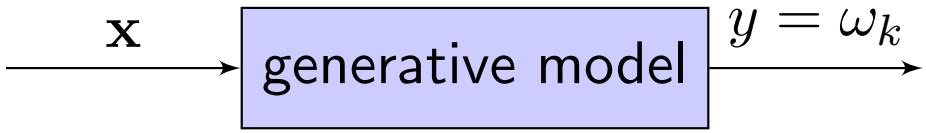
- **deterministic**: given the same input $\mathbf{x}$, the output y is always the same as $y = \bar{f}(\mathbf{x})$
- **stochastic**: given the same input $\mathbf{x}$, the output y is still a random variable following $p(y|\mathbf{x})$
- stochasticity may come from noises, parameter variations, etc.
- discriminative models focus on function estimation
- generative models focus on density estimation
- generative models are a more generic setting but also more challenging to learn in general

Generative Models for Classification



- input $\mathbf{x}$: feature vectors (continuous or discrete)
- output $y = \{\omega_1, \omega_2, \cdots, \omega_K\}$: **discrete**, called class labels
- the joint distribution $p(\mathbf{x}, y) = p(y)p(\mathbf{x}|y)$ breaks down to:
  - prior probabilities: $p(y = \omega_k) \triangleq \Pr(\omega_k)$ $(\forall k = 1, 2, \cdots, K)$
  - class-conditional distributions: $p(\mathbf{x}|y = \omega_k) \triangleq p(\mathbf{x}|\omega_k)$ $(\forall k = 1, 2, \cdots, K)$
- probabilistic distribution constraints:
  - priors satisfy $\sum_{k=1}^{K} Pr(\omega_k) = 1$
  - if $\mathbf{x}$ is continuous, $\int_{\mathbf{x}} p(\mathbf{x}|\omega_k)d\mathbf{x} = 1 \quad (\forall k = 1, 2, \cdots, K)$
  - if $\mathbf{x}$ is discrete, $\sum_{\mathbf{x}} p(\mathbf{x}|\omega_k) = 1 \quad (\forall k = 1, 2, \cdots, K)$

Bayesian Decision Theory (I): Classification



- given any $\mathbf{x}$, determine the best class in $\{\omega_1, \cdots \omega_K\}$
- any decision rule: $\mathbf{x} \longmapsto g(\mathbf{x}) \in \{\omega_1, \cdots \omega_K\}$
- Bayesian decision theory: the best decision is

$$\begin{aligned} g^*(\mathbf{x}) &= \arg\max_k \; p(\omega_k|\mathbf{x}) = \arg\max_k \; \frac{\Pr(\omega_k)p(\mathbf{x}|\omega_k)}{p(\mathbf{x})} \\ &= \arg\max_k \; \Pr(\omega_k) \cdot p(\mathbf{x}|\omega_k) \end{aligned}$$

  a.k.a. the **maximum a posterior (MAP) rule** or Bayes decision rule.

- why is this optimal?

Optimality of the MAP rule (I)

Theorem 1

Assume $p(\mathbf{x}, \omega)$ is known, when $\mathbf{x}$ is used to predict ω, the MAP rule leads to the lowest expected risk (using 0-1 loss).

- the 0-1 loss function: $l(\omega, \omega') = \begin{cases} 0 & \text{when } \omega = \omega' \\ 1 & \text{otherwise} \end{cases}$
- the expected risk of any rule $\mathbf{x} \longmapsto g(\mathbf{x}) \in \{\omega_1, \cdots \omega_K\}$:

$$R(g) = \mathbb{E}_{p(\mathbf{x},\omega)}\Big[l\big(\omega, g(\mathbf{x})\big)\Big] = \int_{\mathbf{x}} \sum_{k=1}^{K} l\big(\omega_k, g(\mathbf{x})\big) p(\mathbf{x}, \omega_k) d\mathbf{x}$$

$$= \int_{\mathbf{x}} \underbrace{\Big[\sum_{k=1}^{K} l\big(\omega_k, g(\mathbf{x})\big) p(\omega_k|\mathbf{x})\Big]}_{\sum_{\omega_k \neq g(\mathbf{x})} p(\omega_k|\mathbf{x})} p(\mathbf{x}) d\mathbf{x}$$

Optimality of the MAP rule (II)

- due to $\sum_{k=1}^{K} p(\omega_k|\mathbf{x}) = 1$, we have

$$\sum_{\omega_k \neq g(\mathbf{x})} p(\omega_k|\mathbf{x}) = 1 - p\big(g(\mathbf{x})|\mathbf{x}\big)$$

- we have

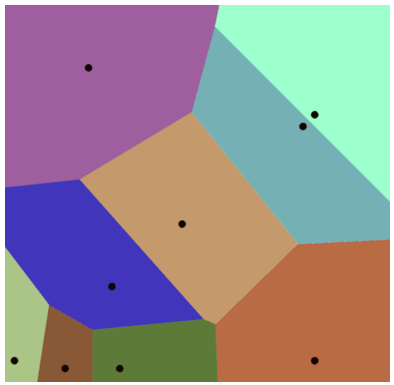
$$R(g)\downarrow \Longrightarrow \forall \mathbf{x}, \Big[1 - p\big(g(\mathbf{x})|\mathbf{x}\big)\Big]\downarrow \Longrightarrow \forall \mathbf{x}, p\big(g(\mathbf{x})|\mathbf{x}\big)\uparrow$$

- since $g(\mathbf{x}) \in \{\omega_1, \cdots \omega_K\}$, we choose:

$$g^*(\mathbf{x}) = \arg\max_k \; p(\omega_k|\mathbf{x})$$

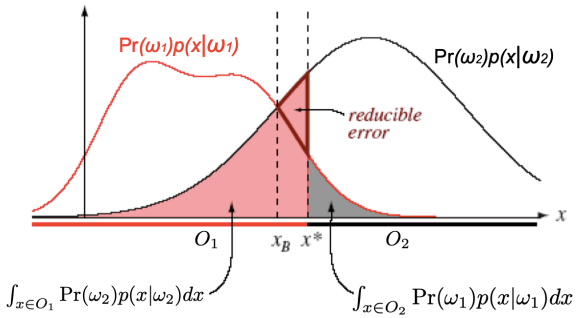
Classification Error Probability

- any rule $\mathbf{x} \mapsto g(\mathbf{x}) \in \{\omega_1, \cdots \omega_K\}$ partitions input space into K regions, i.e. $O_1, O_2, \cdots, O_K$
  $\mathbf{x} \in O_k \implies g(\mathbf{x}) = \omega_k$



- the expected risk is the probability of classification error:

$$\begin{aligned} R(g) &= \Pr(\text{error}) = 1 - \Pr(\text{correct}) \\ &= 1 - \sum_{k=1}^{K} \Pr(\mathbf{x} \in O_k, \omega_k) \\ &= 1 - \sum_{k=1}^{K} \Pr(\omega_k) \int_{\mathbf{x} \in O_k} p(\mathbf{x}|\omega_k) d\mathbf{x} \end{aligned}$$



- the Bayes error: $R(g^*)$ of the MAP rule (the lowest possible error)

Example: the MAP rule for independent binary features

- 2-class (ω_1 and ω_2) classification: $\Pr(\omega_1)$ and $\Pr(\omega_2)$
- use d independent binary features $\mathbf{x} = \left[x_1, x_2, \cdots, x_d\right]^{\mathsf{T}}$, where $x_i \in \{0, 1\} \quad \forall i = 1, 2, \cdots, d$
- denote $\alpha_i \triangleq \Pr(x_i = 1|\omega_1)$ and $\beta_i \triangleq \Pr(x_i = 1|\omega_2)$, we have: $p(\mathbf{x}|\omega_1) = \prod_{i=1}^{d} \alpha_i^{x_i}(1-\alpha_i)^{1-x_i} \quad p(\mathbf{x}|\omega_2) = \prod_{i=1}^{d} \beta_i^{x_i}(1-\beta_i)^{1-x_i}$
- the MAP rule: given $\mathbf{x}$, classify as ω_1 if $\Pr(\omega_1) \cdot p(\mathbf{x}|\omega_1) \geq \Pr(\omega_2) \cdot p(\mathbf{x}|\omega_2)$, otherwise ω_2.
- take logarithm to derive a **linear** decision boundary:

$$g(\mathbf{x}) = \sum_{i=1}^{d} \lambda_i x_i + \lambda_0 = \begin{cases} \geq 0 & \Longrightarrow \omega_1 \\ < 0 & \Longrightarrow \omega_2 \end{cases}$$

where $\lambda_i = \ln \frac{\alpha_i(1-\beta_i)}{\beta_i(1-\alpha_i)}$ and $\lambda_0 = \sum_{i=1}^{d} \ln \frac{1-\alpha_i}{1-\beta_i} + \ln \frac{\Pr(\omega_1)}{\Pr(\omega_2)}$

Generative Models for Regression

- input: n-dimensional vector $\mathbf{x} \in \mathbb{R}^n$; output: $y \in \mathbb{R}$
- the joint distribution $p(\mathbf{x}, y)$ is known
- $\mathbf{x}$ is used to predict y as $y = g(\mathbf{x})$
- what is the best decision rule $g(\mathbf{x})$ for $\mathbf{x} \mapsto y$?
- Bayesian decision theory suggests the best rule as:

$$g^*(\mathbf{x}) = \mathbb{E}(y|\mathbf{x}) = \int_y y \cdot p(y|\mathbf{x}) dy$$

Theorem 2

Assume $p(\mathbf{x}, y)$ is known, the conditional mean $\mathbb{E}(y|\mathbf{x})$ leads to the lowest expected risk (using mean square loss).

Optimality of Conditional Mean for Regression

Proof:

- The expected risk of any rule $\mathbf{x} \rightarrow g(\mathbf{x}) \in \mathbb{R}$:

$$R(g) = \mathbb{E}_{p(\mathbf{x},y)}\Big[l\big(\omega, g(\mathbf{x})\big)\Big] = \int_{\mathbf{x}}\int_{y}\Big(y - g(\mathbf{x})\Big)^2 p(\mathbf{x}, y)d\mathbf{x}dy$$

$$= \int_{\mathbf{x}} \underbrace{\Big[\int_{y}\Big(y - g(\mathbf{x})\Big)^2 p(y|\mathbf{x})dy\Big]}_{Q(g|\mathbf{x})} p(\mathbf{x})d\mathbf{x}$$

- Take partial derivative w.r.t. g as:

$$\frac{\partial Q(g|\mathbf{x})}{\partial g(\cdot)} = 0 \implies \int_{y}\Big(g(\mathbf{x}) - y\Big)p(y|\mathbf{x})dy = 0$$

$$\implies g^*(\mathbf{x}) = \int_{y} y \cdot p(y|\mathbf{x})dy = \mathbb{E}(y|\mathbf{x}) \quad \blacksquare$$

Plug-in MAP Decision Rule for classification

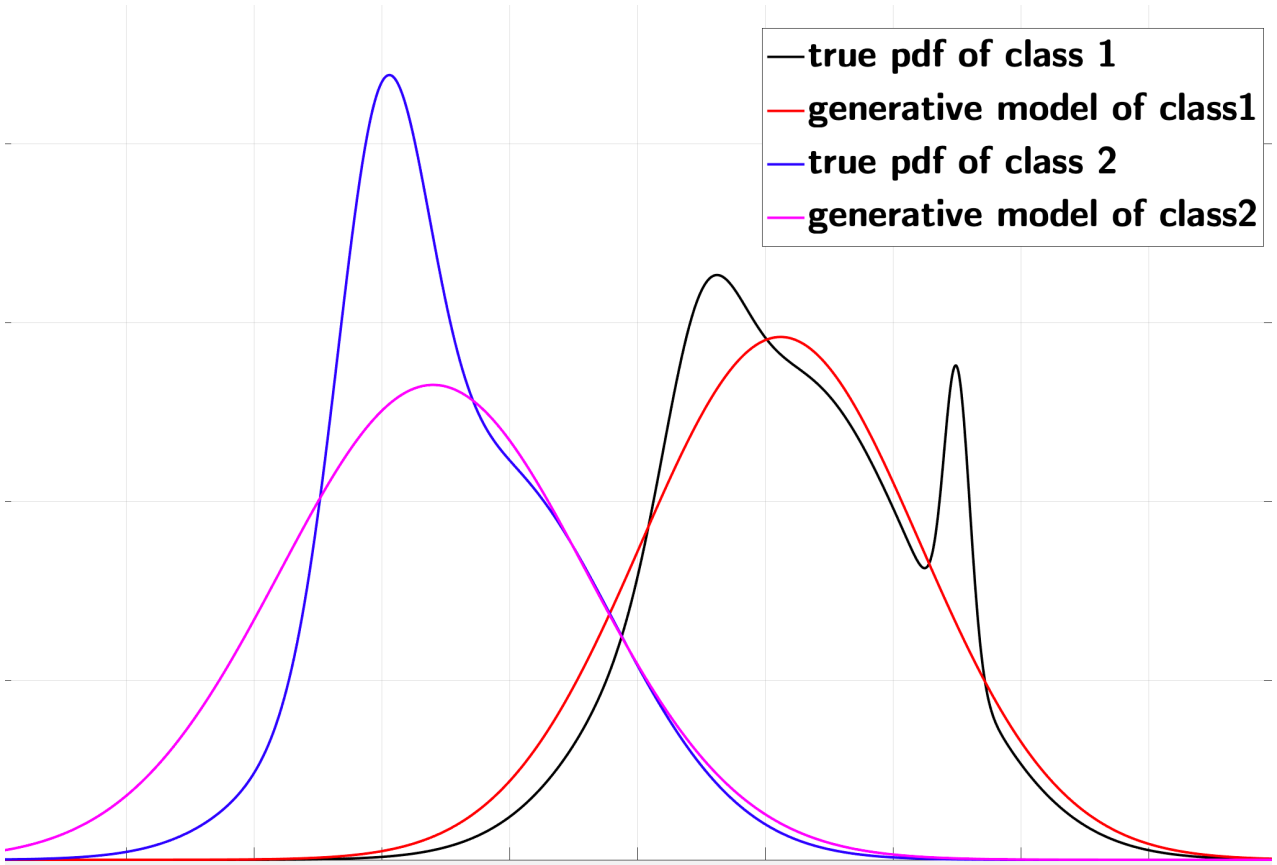
- since the true distributions $\Pr(\omega_k)$ and $p(\mathbf{x}|\omega_k)$ are unknown, the optimal MAP decision rule is not feasible in practice
- given training data: $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$
- choose two probabilistic models:
  - $\hat{p}_\lambda(\omega_k)$ to approximate $\Pr(\omega_k)$
  - $\hat{p}_{\theta_k}(\mathbf{x})$ to approximate $p(\mathbf{x} \mid \omega_k) \quad (\forall k = 1, 2, \cdots, K)$
- parameter estimation: estimate $\{\lambda, \theta_1, \cdots, \theta_K\}$ using $\mathcal{D}$
- the optimal MAP rule in theory:

$$\omega^* = \arg\max_k \ \Pr(\omega_k) \cdot p(\mathbf{x}|\omega_k)$$

- the plug-in MAP decision rule in pratice:

$$\omega^* = \arg\max_k \ \hat{p}_\lambda(\omega_k) \cdot \hat{p}_{\theta_k}(\mathbf{x})$$

Plug-in MAP Decision Rule



Statistical Data Modeling

Assume we have collected some training samples:

$$\mathcal{D} = \left\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\right\}$$

where each $(\mathbf{x}_i, y_i) \sim p(\mathbf{x}, y) \quad (\forall i = 1, 2, \cdots, N)$.

1. choose some probabilistic models:

$$\Pr(\omega_k) \approx \hat{p}_{\boldsymbol{\lambda}}(\omega_k)$$

$$p(\mathbf{x}|\omega_k) \approx \hat{p}_{\boldsymbol{\theta}_k}(\mathbf{x}) \quad (\forall k = 1, 2, \cdots, K)$$

2. estimate the model parameters:

$$\mathcal{D} \longrightarrow \left\{\boldsymbol{\lambda}, \boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K\right\}$$

3. apply the plug-in MAP rule:

$$\hat{g}(\mathbf{x}) = \arg\max_k \ \hat{p}_{\boldsymbol{\lambda}}(\omega_k) \cdot \hat{p}_{\boldsymbol{\theta}_k}(\mathbf{x})$$

Maximum Likelihood Estimation (I)

- generative models for classification $\{\omega_1, \cdots, \omega_K\}$:
  - prior probabilities: $\Pr(\omega_k) \quad (k = 1, \cdots, K)$
  - class-conditional distribution: $p(\mathbf{x}|\omega_k) \quad (k = 1, \cdots, K)$
- collect training data for each class: $\mathcal{D}_k \sim p(\mathbf{x}|\omega_k)$
- **density estimation**: estimate the probability distribution from a finite number of samples
- select probabilistic models: $\hat{p}_{\theta_k}(\mathbf{x}) \approx p(\mathbf{x}|\omega_k)$
- **maximum likelihood estimation (MLE)**: learn $\hat{p}_{\theta_k}(\mathbf{x})$ to maximize the probability of observing the training data $\mathcal{D}_k$

$$\theta_k^* = \arg\max_{\theta_k} \hat{p}_{\theta_k}(\mathcal{D}_k) \quad (k = 1, \cdots, K)$$

- MLE: fit data best; best interpret the observed data

Maximum Likelihood Estimation (II)

- drop index k and $\hat{p}(\cdot) \rightarrow p(\cdot)$, MLE turns to be:

$$\theta_{\text{MLE}} = \arg\max_{\theta} p_{\theta}(\mathcal{D}) = \arg\max_{\theta} p_{\theta}(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)$$

  where $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$

- assume all data are *i.i.d.* (independent and identically distributed), i.e., all samples are drawn independently from the same distribution:

$$p_{\theta}(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N) = \prod_{i=1}^{N} p_{\theta}(\mathbf{x}_i)$$

- why called maximum *likelihood* (not probability)?
  - $p_{\theta}(\mathbf{x})$: data distribution of various $\mathbf{x}$ if θ is given (fixed)
  - $p_{\theta}(\mathbf{x})$: likelihood function of θ if $\mathbf{x}$ is given (fixed)

Maximum Likelihood Estimation (III)

- in many cases, it is more convenient to work with the logarithm of the likelihood rather than the likelihood itself
- denote the log-likelihood function $l(\theta) = \ln p_\theta(\mathcal{D})$, we have

$$\theta_{\text{ML}} = \arg\max_\theta \ l(\theta) = \arg\max_\theta \ \sum_{i=1}^{N} \ln p_\theta(\mathbf{x}_i)$$

- optimization methods for ML estimation:
  - differential calculus for simple models, e.g., single univariate/multivariate Gaussian, ...
  - Lagrange optimization for models with constraints, e.g., multinomial, markov chain, ...
  - Expectation-Maximization (EM) method for mixture models, e.g., Gaussian mixture models (GMM), hidden Markov models

MLE Example: Univariate Gaussian Models

- the training set: $\mathcal{D} = \{x_1, x_2, \cdots, x_N\}$ $(\forall x_i \in \mathbb{R})$
- choose a univariate Gaussian approximate the unknown distribution:

$$p_\theta(x) = \mathcal{N}(x \,|\, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- the log-likelihood function:

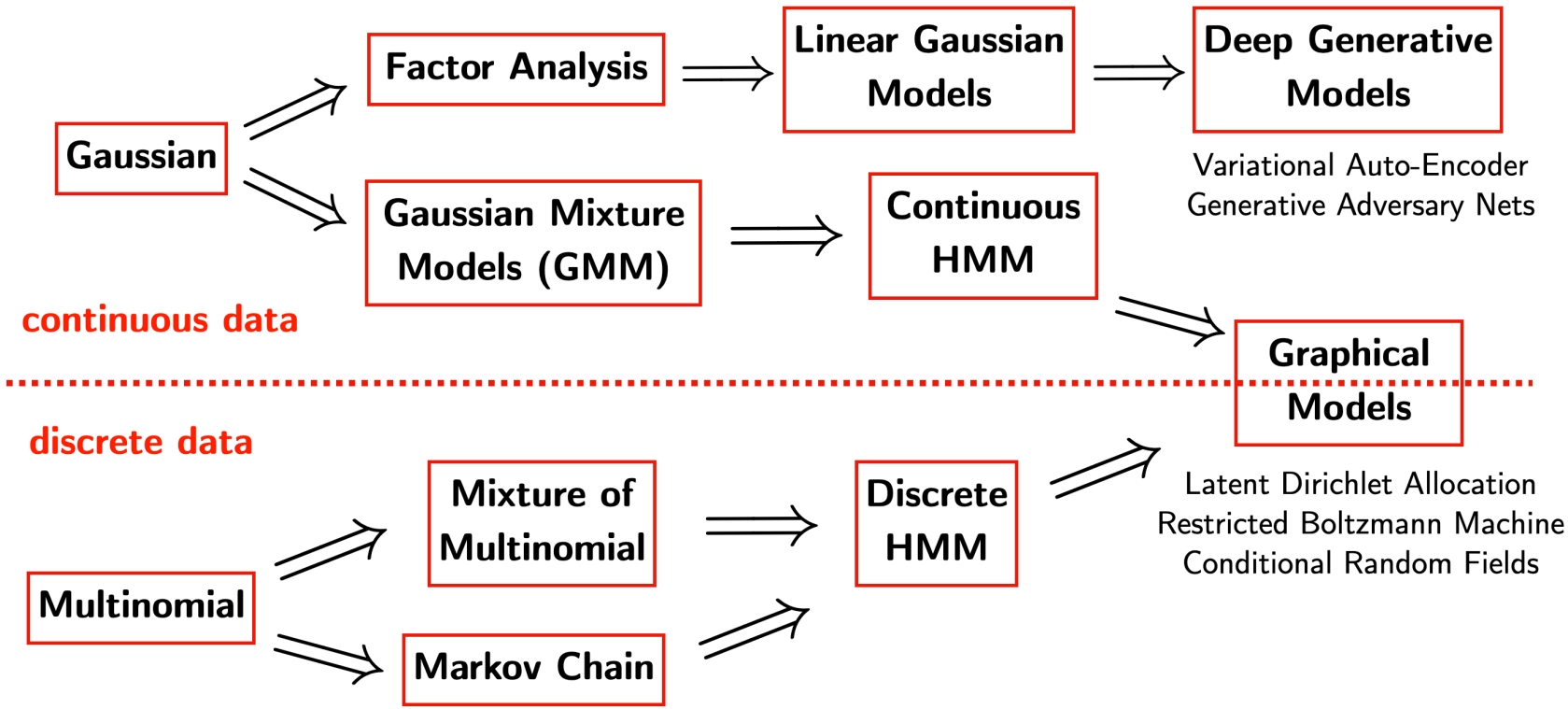
$$l(\mu, \sigma^2) = \sum_{i=1}^{N} \ln p_\theta(x_i) = \sum_{i=1}^{N} \Big[-\frac{\ln(2\pi\sigma^2)}{2} - \frac{(x_i - \mu)^2}{2\sigma^2} \Big]$$

- the MLE of the unknown Gaussian mean and variance:

$$\frac{\partial l(\mu, \sigma^2)}{\partial \mu} = 0 \implies \mu_{\text{MLE}} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

$$\frac{\partial l(\mu, \sigma^2)}{\partial \sigma^2} = 0 \implies \sigma^2_{\text{MLE}} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu_{\text{MLE}})^2$$

Roadmap of Generative Models



Generative Models (in a nutshell)

- Gaussian-derived generative models for continuous data
- multinomial-derived generative models for discrete data
- **unimodal models**: Gaussian, multinomial, Markov chains, generalized linear models, etc.
- **mixture models**: Gaussian mixture models, hidden Markov models, etc.
- **entangled models**: factor analysis, linear Gaussian models, deep generative models (e.g. VAE, GAN)
- **graphical models**: naive Bayes, latent Dirichlet allocation, restricted Boltzmann machine, conditional random fields, etc.
- Bayesian learning: treat model parameters as random variables