ECE 1513: Introduction to Machine Learning

Lecture 2: Probabilistic Modeling and Density Estimation

Ali Bereyhi

ali.bereyhi@utoronto.ca

Department of Electrical and Computer Engineering

University of Toronto

Winter 2025

Quick Recap: *ML General Recipe*

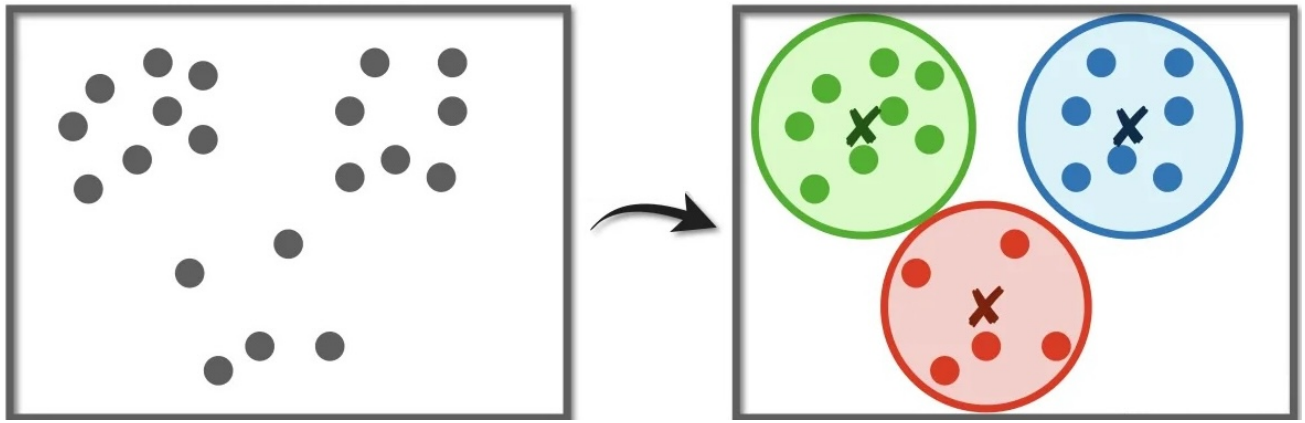
We defined ML as

> *the set of data-driven approaches that help us understand the environment and its behavior, and generalize it!*

Any learning task is accomplished by ML through *three major steps*

- Collect data
- Specify a model *that captures the pattern*
- Develop a learning algorithm

Quick Recap: *Clustering*



- *Data*
  - ↳ *Collection of samples* $\mathbb{D} = \{\boldsymbol{x}_n : n = 1, \ldots, N\}$
- *Model*
  - ↳ *A function mapping* $\boldsymbol{x}$ *to a cluster, e.g.,* K*-centroid model*
- *Learning algorithm*
  - ↳ *It takes* $\mathbb{D}$ *and returns a good clustering*

Quick Recap: *Clustering*

K-`Means()`:

1: Initiate $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
2: **while** $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ changing **do**
3: Set $\mathcal{C}_1, \ldots, \mathcal{C}_K \leftarrow$ `Cluster_Assignment`$(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$
4: Update $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \leftarrow$ `Centroid_Update`$(\mathcal{C}_1, \ldots, \mathcal{C}_K)$
5: **end while**
6: Return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

The optimal clustering algorithm should minimize average risk

$$\mathcal{J} = \frac{1}{N} \sum_{k=1}^{K} \sum_{n=1}^{N} r_{n,k} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

where $r_{n,k} = 1$ *if* $\boldsymbol{x}_n \in \mathcal{C}_k$ *and zero otherwise* $\rightsquigarrow$ *the smaller* $\mathcal{J}$*, the better*

Today's Agenda: *Density Estimation*

In today's lecture, we study another *unsupervised* learning problem, i.e.,

density estimation

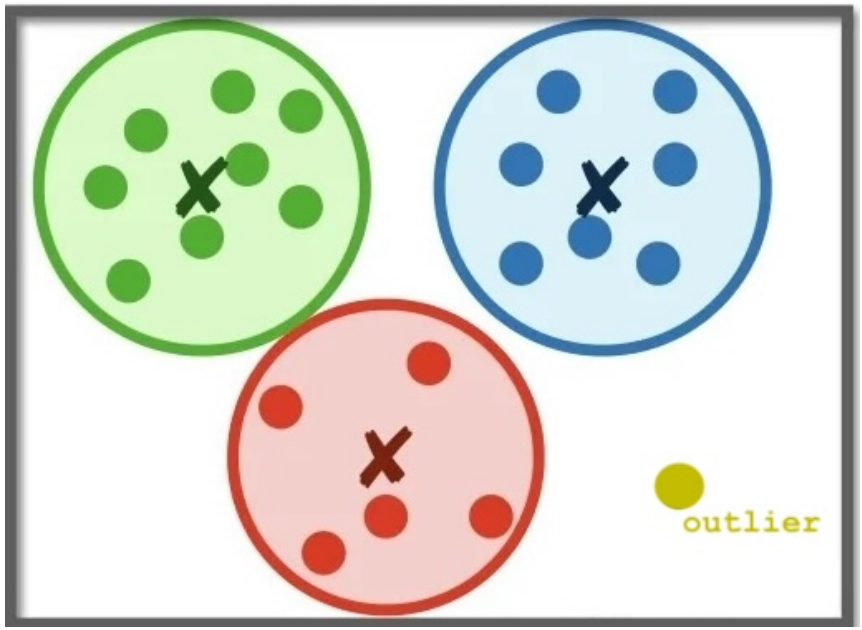
through the following steps

- *Probabilistic modeling of data*
  - ↳ *We review key notions from probability theory*
- *Maximum likelihood approach to density estimation*
  - ↳ *We look at Gaussian maximum likelihood*
- *Revisiting the clustering problem*
- *Few notes on how things extend beyond clustering*

We also talk about *validating a model* and *testing its generalization*

Motivation: *Outlier Detection*

In clustering, we talked about outliers



Let's look at alternative approach!

Outlier Detection: *Example*

We collect data from results of a game

$$\mathbb{D} = \{x_n : n = 1, \ldots, N\}$$

with x_n *being*

$$x_n = \begin{cases} 1 & \textit{Player } n \textit{ wins} \\ 0 & \textit{Player } n \textit{ loses} \end{cases}$$

For instance, we collect $\mathbb{D} = \{1, 0, 0, 1, 1, 0, 0, 0, 0, 0\}$ *in 10 days*

Outlier Detection

Is it typical (normal) to see in next 3 days the following observation?

$$\{1, 1, 1\}$$

Outlier Detection: *Learning Task*

This is a learning task

- *Data*
  - ↳ $\mathbb{D} = \{x_n : n = 1, \ldots, N\}$ *collected in* N *days*
- *Pattern* $\rightsquigarrow$ *Model*
  - ↳ *There is a certain winning chance in normal situation*
- *Learning Algorithm*
  - ↳ $\mathbb{D} \mapsto$ *Is what we see coming from a normal situation?*

Density Estimation

We can interpret this problem from stochastic viewpoint

- *We observe some data samples*

$$\mathbb{D} = \{\boldsymbol{x}_n : n = 1, \ldots, N\}$$

- *We know (assume) that they are coming from a random process*
  - ↳ *The model is the distribution of this random process*
- *We want to infer from $\mathbb{D}$ what the random process is*
  - ↳ *This is what the learning algorithm does*

Let's recap key notions in Probability Theory

Discrete Random Variables

Say $x \in \mathbb{X} = \{a_1, \ldots, a_I\}$; then, $x \sim P(x)$ implies that

- *Probability of* $x = a_i$ *is* $P(a_i)$
- $P(x)$ *is non-negative for any* $x \in \mathbb{X}$ *and less than 1*

$$0 \leqslant P(x) \leqslant 1$$

- $P(x)$ *adds up to 1*

$$\sum_{x \in \mathbb{X}} P(x) = \sum_{i=1}^{I} P(a_i) = 1$$

$P(x) \equiv$ *Probability Mass Function* $\rightsquigarrow$ *Distribution*

Discrete Random Variables: *Example*

Binary (Bernoulli) random variable $x \in \mathbb{X} = \{0, 1\}$

$$x \sim \text{Ber}(\theta)$$

is distributed as

$$P(x) = \begin{cases} \theta & x = 1 \\ 1 - \theta & x = 0 \end{cases}$$

Continuous Random Variables

Say $x \in \mathbb{R}$; then, $x \sim P(x)$ implies that

- *Probability of* $a < x \leqslant b$ *is*

$$\int_a^b P(x)\,\mathrm{d}x$$

- $P(x)$ *is non-negative for any* $x \in \mathbb{X}$

$$P(x) \geqslant 0$$

- $P(x)$ *adds up to 1*

$$\int_{-\infty}^{+\infty} P(x)\,\mathrm{d}x = 1$$

$P(x) \equiv$ *Probability Density Function* $\rightsquigarrow$ *Distribution*

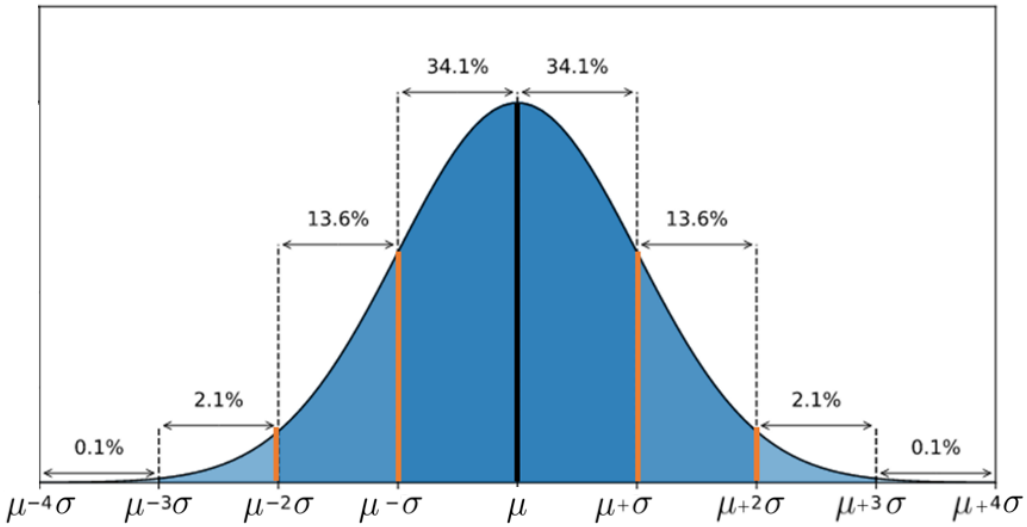
Continuous Random Variables: *Example*

Gaussian random variable with mean μ and variance σ^2

$$x \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

is distributed as

$$P\left(x\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{\left(x-\mu\right)^2}{2\sigma^2}\right\}$$



Independent Processes

Random variables $x \sim P(x)$ and $y \sim P(y)$ are independent if

$$P(x, y) = P(x) P(y)$$

i.i.d. Sequence

$x_1, \ldots, x_N$ are independent and identically distributed (i.i.d.) if they are independent and have the same distribution $\equiv$

independent samples of the same process

Example: *i.i.d. Bernoulli Samples*

We sample $\mathrm{Ber}(\theta)$ *5 times independently; then, the probability of observing*

$$x_1, x_2, x_3, x_4, x_5 = 1, 1, 0, 1, 0$$

is given by

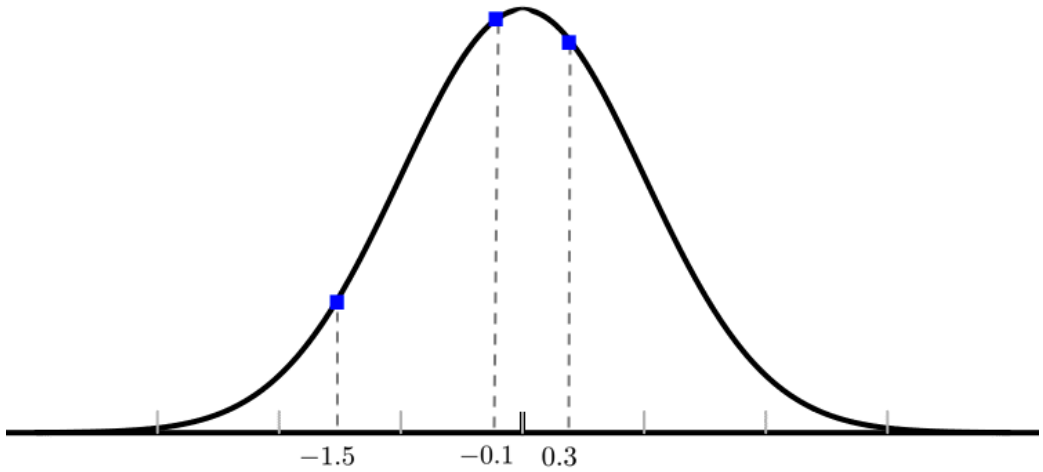
$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5) &= P(1, 1, 0, 1, 0) \\ &= \theta^3 (1-\theta)^2 \end{aligned}$$

Expectation: *Gaussian Example*

We sample $\mathcal{N}(0,1)$ 3 times independently; then, the density at

$$x_1, x_2, x_3 = 0.3, -1.5, -0.1$$

is given by



$$P(x_1, x_2, x_3) = \left(\frac{1}{\sqrt{2\pi}}\right)^3 \exp\{-0.045\} \exp\{-1.125\} \exp\{-0.005\}$$

Expectation: *Average in Asymptotic Sense*

Say $x \sim P(x)$; then,

$$\mathbb{E}\{f(x)\} = \begin{cases} \sum_{x \in \mathbb{X}} f(x) P(x) & x \text{ is discrete} \\ \int_{-\infty}^{+\infty} f(x) P(x) \, \mathrm{d}x & x \text{ is continuous} \end{cases}$$

Intuitive Meaning

Say $x_1, \ldots, x_N$ are i.i.d. samples from $P(x)$; then,

$$\frac{1}{N} \sum_{n=1}^{N} f(x_n) \approx \mathbb{E}\{f(x)\}$$

when N is large

Expectation: *Bernoulli Example*

Say $x \sim \mathrm{Ber}(\theta)$: *this means that*

$$\mathbb{E}\{x\} = \theta$$

Intuitive Meaning

Say $x_1, \ldots, x_N$ *are independent samples of* $\mathrm{Ber}(\theta)$*; then,*

$$\frac{1}{N} \sum_{n=1}^{N} x_n \approx \theta$$

when N *is large*

Expectation: *Gaussian Example*

Say $x \sim \mathcal{N}\left(\mu, \sigma^2\right)$: *this means that*

$$\mathbb{E}\left\{x\right\} = \mu$$

$$\mathbb{V}\mathrm{ar}\left\{x\right\} = \mathbb{E}\left\{(x-\mu)^2\right\} = \sigma^2$$

Intuitive Meaning

Say $x_1, \ldots, x_N$ *are independent Gaussian samples from* $\mathcal{N}\left(\mu, \sigma^2\right)$*; then,*

$$\frac{1}{N}\sum_{n=1}^{N} x_n \approx \mu$$

$$\frac{1}{N}\sum_{n=1}^{N} (x_n-\mu)^2 \approx \sigma^2$$

when N *is large*

Random *Variables* $\rightsquigarrow$ Random *Objects*

We can extend all these notions to higher dimensions

- $\boldsymbol{x}$ *is a random vector whose entries are random variables*
- $\mathbf{X}$ *is a random matrix whose entries are random variables*
- $\ldots$

We then write

$$\boldsymbol{x} \sim P(\boldsymbol{x})$$

and mean by $P(\boldsymbol{x})$ the joint distribution of all random entries

Density Estimation: *Formulation*

We can interpret this problem from stochastic viewpoint

- *We observe some data samples*

$$\mathbb{D} = \{\boldsymbol{x}_n : n = 1, \ldots, N\}$$

- *We know (assume) that they are coming from a random process*
  - ↳ *The model is* the distribution of this *random process*
- *We want to infer from $\mathbb{D}$ what the random process is*
  - ↳ *This is what the learning algorithm does*

Modeling Distributions

We observe some data samples

$$\mathbb{D} = \{\boldsymbol{x}_n : n = 1, \ldots, N\}$$

and may see a pattern in the data, e.g.,

- ↳ *Samples are binary*
- ↳ *Histogram of samples looks roughly Gaussian*

- ? *How to model this pattern?*
- ! *We assume $\boldsymbol{x}_n$'s are sampled from a distribution with some unknowns*

$$\boldsymbol{x}_n \sim P_{\boldsymbol{\theta}}(\boldsymbol{x}_n)$$

Modeling Distributions: *Binary Example*

We observe the results of a game: *samples are*

$$\mathbb{D} = \{x_n \in \{0, 1\} : n = 1, \ldots, N\}$$

If we know each sample is given by an independent player, we could assume

$$x_n \sim P_\theta\left(x_n\right) \equiv \mathrm{Ber}\left(\theta\right)$$

and hence the probability of whole dataset is

$$\begin{aligned}\mathbb{D} \sim P_\theta\left(x_1, \ldots, x_N\right) &= \prod_{n=1}^{N} P_\theta\left(x_n\right) \\ &= \theta^{\#\text{ of 1's}}\left(1-\theta\right)^{\#\text{ of 0's}}\end{aligned}$$

Modeling Distributions: *Gaussian Example*

We observe the a Gaussian random generator: *samples are*

$$\mathbb{D} = \{x_n : n = 1, \ldots, N\}$$

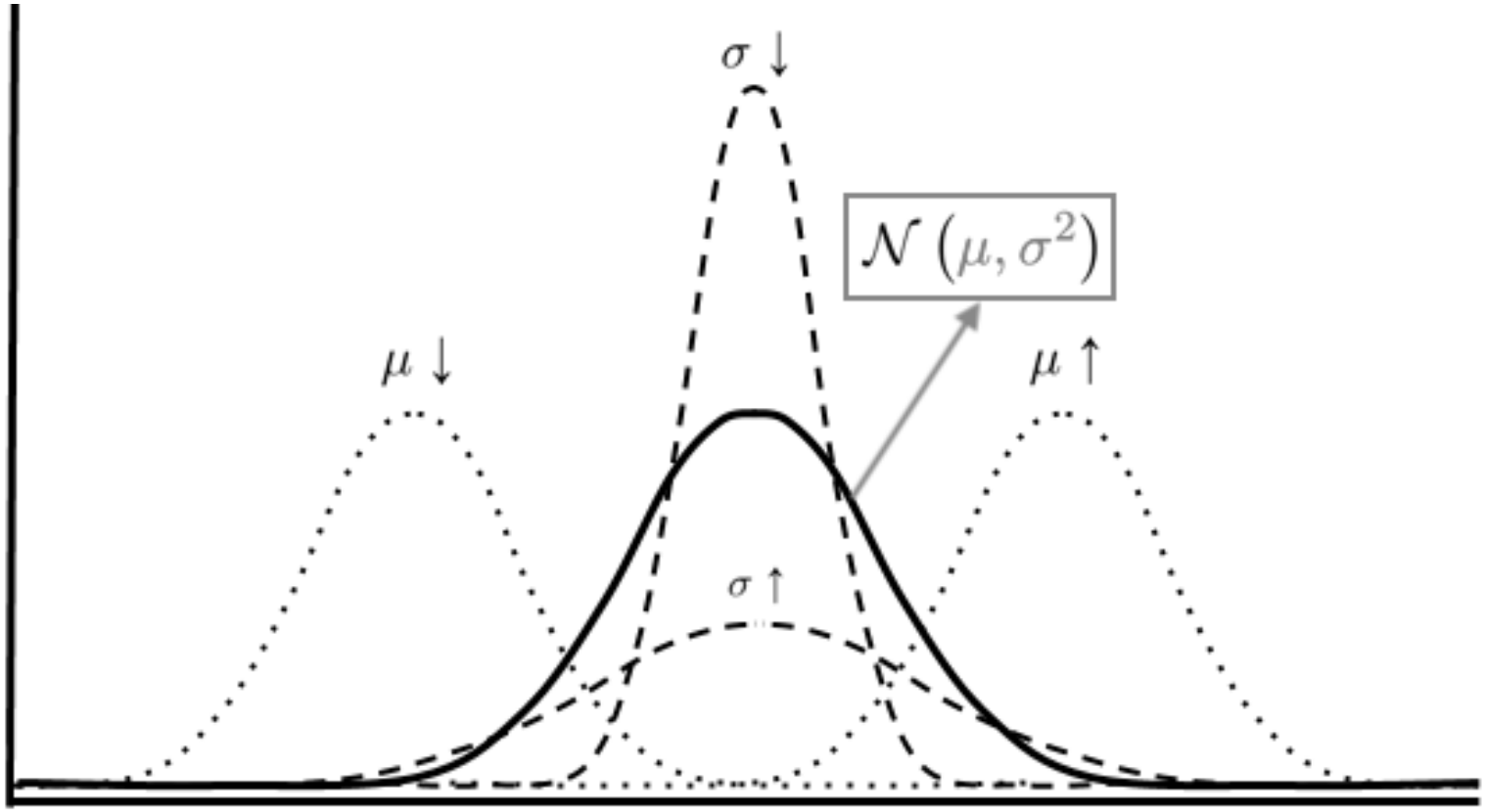
If we know samples are generated independently, we know

$$x_n \sim P_{\mu,\sigma}(x_n) \equiv \mathcal{N}\left(\mu, \sigma^2\right)$$

and hence the probability of whole dataset is

$$\begin{aligned}\mathbb{D} \sim P_{\mu,\sigma}(x_1, \ldots, x_N) &= \prod_{n=1}^{N} P_{\mu,\sigma}(x_n) \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2\right\}\end{aligned}$$

Modeling Distributions: *Gaussian Example*



Likelihood: *Key to Modeling Distributions*

We assume a model for the distribution as

$$\boldsymbol{x}_n \sim P_{\boldsymbol{\theta}}\left(\boldsymbol{x}_n\right)$$

Likelihood

The likelihood of the dataset is

$$\mathcal{L}_{\mathbb{D}}\left(\boldsymbol{\theta}\right) = P_{\boldsymbol{\theta}}\left(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\right)$$

Attention!

Likelihood is only a function of the unknown $\boldsymbol{\theta}$

Likelihood: *Binary Example*

We observe 5 sample results of a game

$$\mathbb{D} = \{1, 1, 0, 1, 0\}$$

Assuming them to be samples of $\mathrm{Ber}(\theta)$*, the likelihood is*

$$\begin{aligned}\mathcal{L}_{\mathbb{D}}(\theta) &= P_{\theta}(1, 1, 0, 1, 0) \\ &= \theta^3 (1-\theta)^2\end{aligned}$$

Attention!

Likelihood is only a function of the unknown θ

Likelihood: *Gaussian Example*

We observe three samples of a Gaussian random generator

$$\mathbb{D} = \{0.3, -1.5, -0.1\}$$

The likelihood is

$$\begin{aligned}\mathcal{L}_{\mathbb{D}}(\mu, \sigma) &= P_{\mu,\sigma}(0.3, -1.5, -0.1) \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^3 \exp\left\{-\frac{(\mu+0.1)^2 + (\mu-0.5)^2 + (\mu-1.5)^2}{2\sigma^2}\right\}\end{aligned}$$

Attention!

Likelihood is only a function of the unknowns μ and σ

Optimal Model $\equiv$ *Maximum Likelihood*

? *How can we find right choice for the unknown* θ

! *We find the unknown* θ *that maximizes the likelihood of dataset*

Notion of Optimality $\equiv$ *Likelihood*

This does the same job that distortion did for us in clustering

Maximum Likelihood Estimation

The optimal model is given by

$$\begin{aligned}\boldsymbol{\theta}^{\star} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathcal{L}_{\mathbb{D}}\left(\boldsymbol{\theta}\right) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, P_{\boldsymbol{\theta}}\left(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\right)\end{aligned}$$

Classic case is that we assume samples in dataset are i.i.d.

$$\begin{aligned}\boldsymbol{\theta}^{\star} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathcal{L}_{\mathbb{D}}\left(\boldsymbol{\theta}\right) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{n=1}^{N} P_{\boldsymbol{\theta}}\left(\boldsymbol{x}_n\right) \;=\; \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{n=1}^{N} \mathcal{L}_n\left(\boldsymbol{\theta}\right)\end{aligned}$$

Maximum Likelihood Estimation

The optimal model is given by

$$\boldsymbol{\theta}^{\star} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{n=1}^{N} \mathcal{L}_n\left(\boldsymbol{\theta}\right)$$

Log Trick

Assume $f\left(\boldsymbol{\theta}\right) \geqslant 0$; then, we can write

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, f\left(\boldsymbol{\theta}\right) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log f\left(\boldsymbol{\theta}\right)$$

Using log trick we have

$$\boldsymbol{\theta}^{\star} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{n=1}^{N} \log \mathcal{L}_n\left(\boldsymbol{\theta}\right) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{n=1}^{N} \mathcal{S}_n\left(\boldsymbol{\theta}\right)$$

Maximum Likelihood: *Binary Example*

We observe 5 sample results of a game

$$\mathbb{D} = \{1, 1, 0, 1, 0\}$$

Assuming them to be samples of $\mathrm{Ber}(\theta)$*, the likelihood is*

$$\mathcal{L}_{\mathbb{D}}(\theta) = \theta^3 (1-\theta)^2$$

Maximum likelihood estimate is

$$\theta^\star = \underset{\theta}{\operatorname{argmax}} \left[3 \log \theta + 2 \log (1-\theta)\right] = 0.6$$

Maximum Likelihood: *Gaussian Example*

We observe three samples of a Gaussian random generator $\mathcal{N}(\mu, 1)$

$$\mathbb{D} = \{0.3, -1.5, -0.1\}$$

The likelihood is

$$\mathcal{L}_{\mathbb{D}}(\mu) = \left(\frac{1}{\sqrt{2\pi}}\right)^3 \exp\left\{-\frac{(\mu+0.1)^2 + (\mu-0.5)^2 + (\mu-1.5)^2}{2}\right\}$$

Maximum likelihood estimate is

$$\begin{aligned}\theta^\star &= \operatorname*{argmax}_{\theta}\left[-\frac{3}{2}\log 2\pi - \frac{(\mu+0.1)^2 + (\mu-0.5)^2 + (\mu-1.5)^2}{2}\right] \\ &= 0.63\end{aligned}$$

Multivariate Gaussian Distribution

A basic approach to model the data is to fit it into a Gaussian distribution

Multivariate Gaussian Distribution

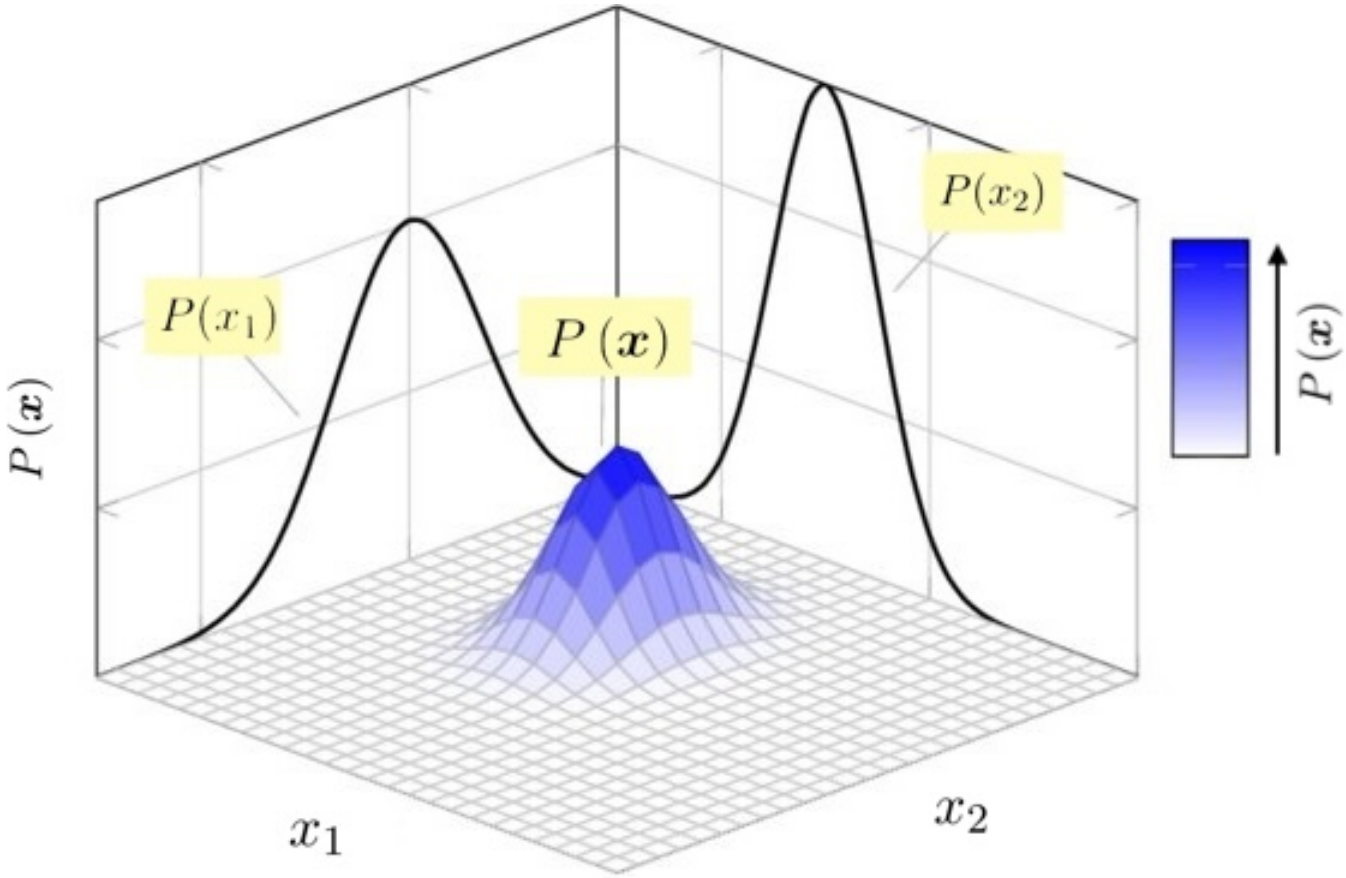
Random vector $\boldsymbol{x} \in \mathbb{R}^d$ is Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ and noted as $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ if

$$P(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}$$

Note that

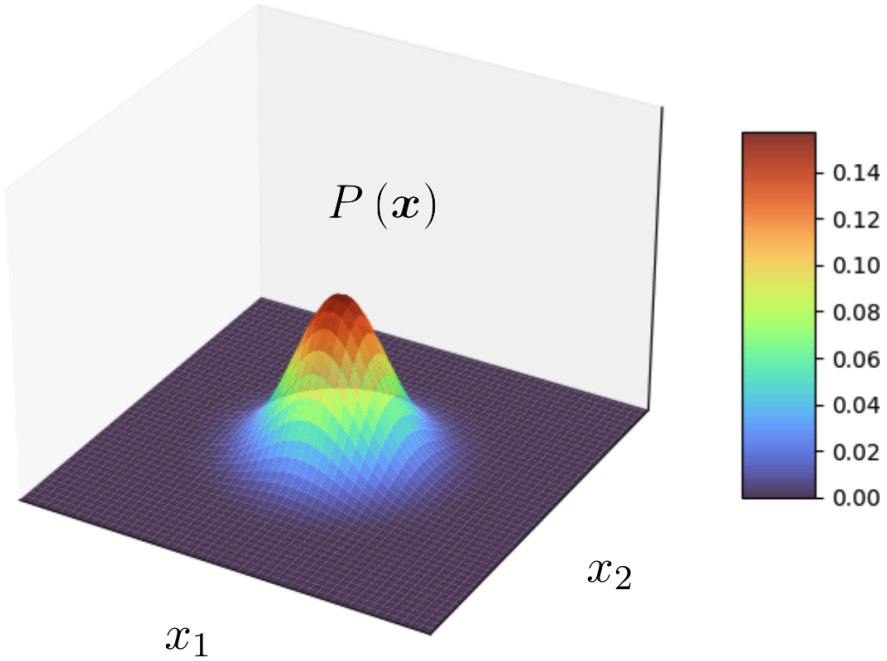
$$\boldsymbol{\mu} = \mathbb{E}\{\boldsymbol{x}\} \qquad \boldsymbol{\Sigma} = \mathbb{E}\left\{(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^\top\right\}$$

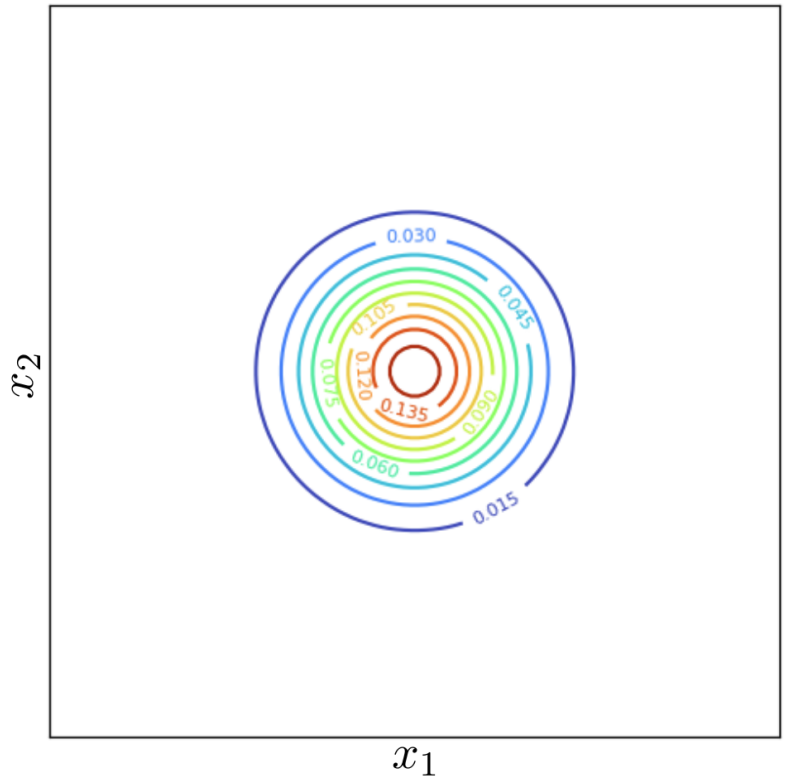
Example: *Bivariate Gaussian Distribution*



Example: *Bivariate Gaussian Distribution*

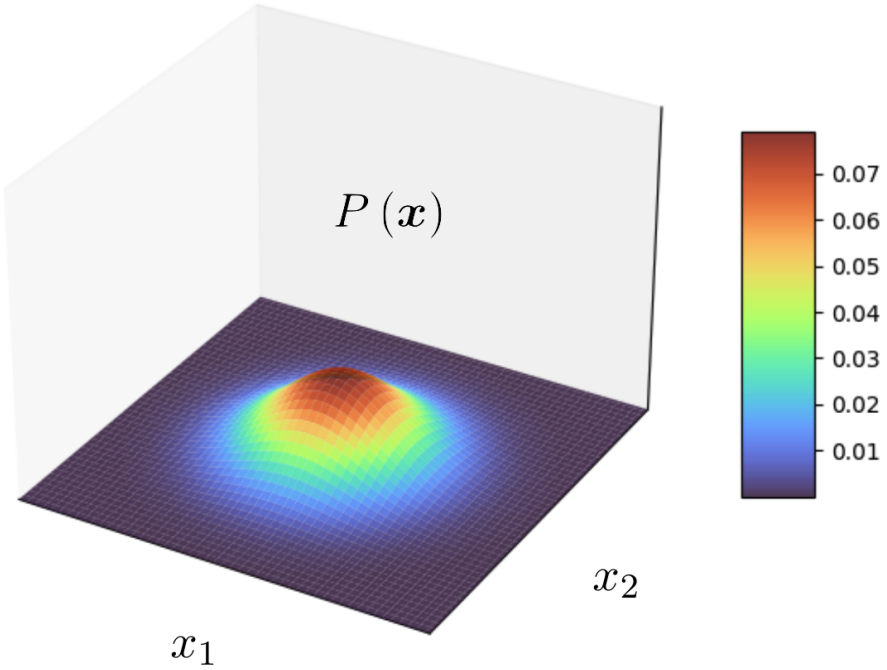
$$\boldsymbol{\Sigma} = \mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

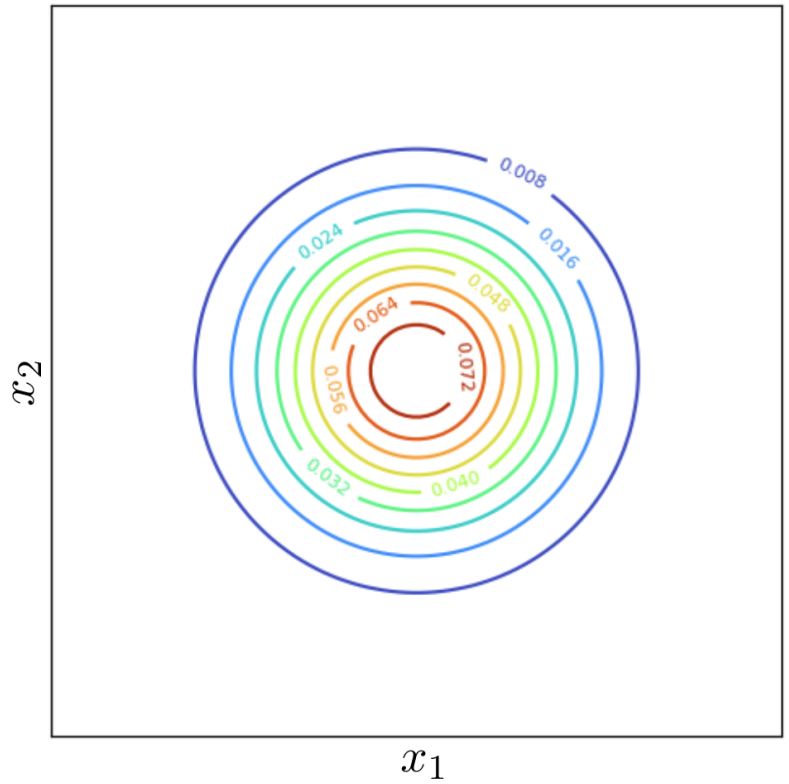




Example: *Bivariate Gaussian Distribution*

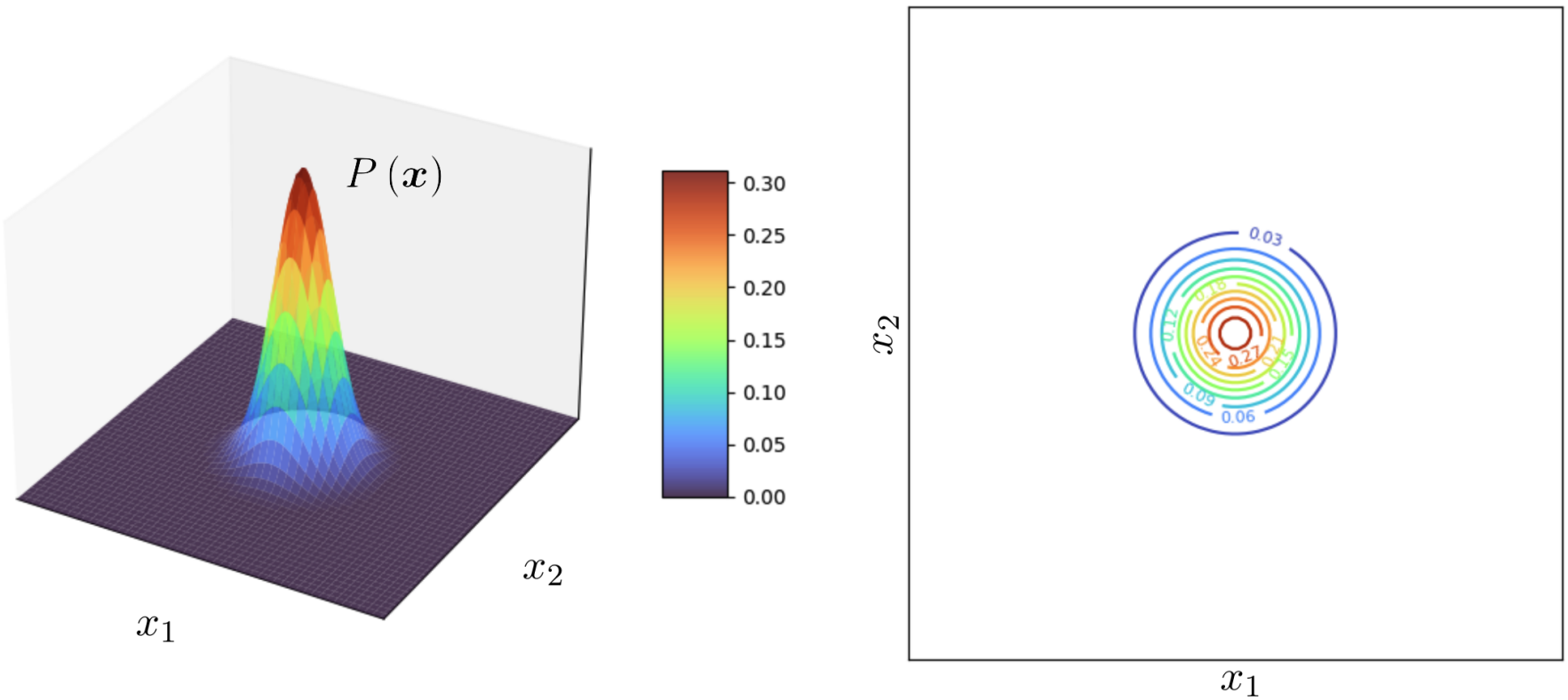
$$\boldsymbol{\Sigma} = 2\mathbf{I} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$





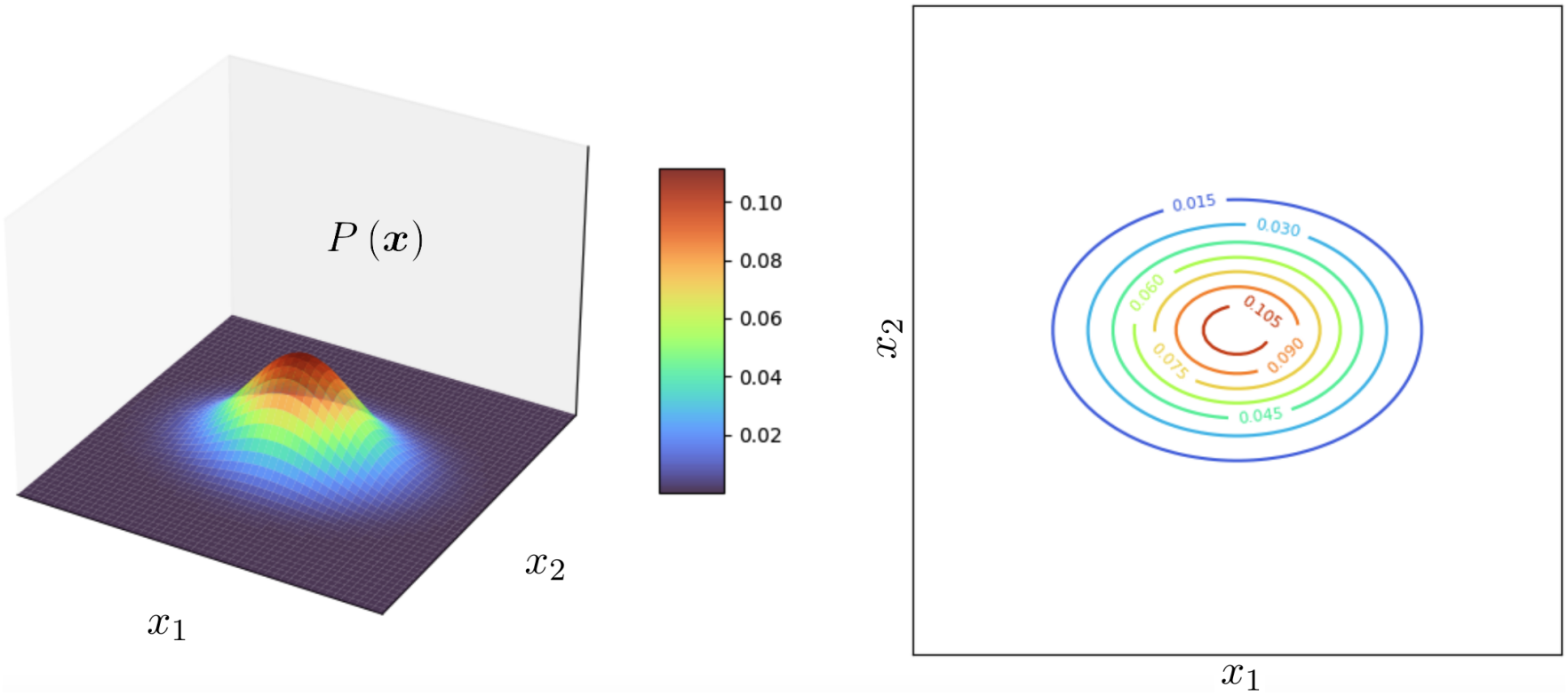
Example: *Bivariate Gaussian Distribution*

$$\boldsymbol{\Sigma} = 0.5\mathbf{I} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$



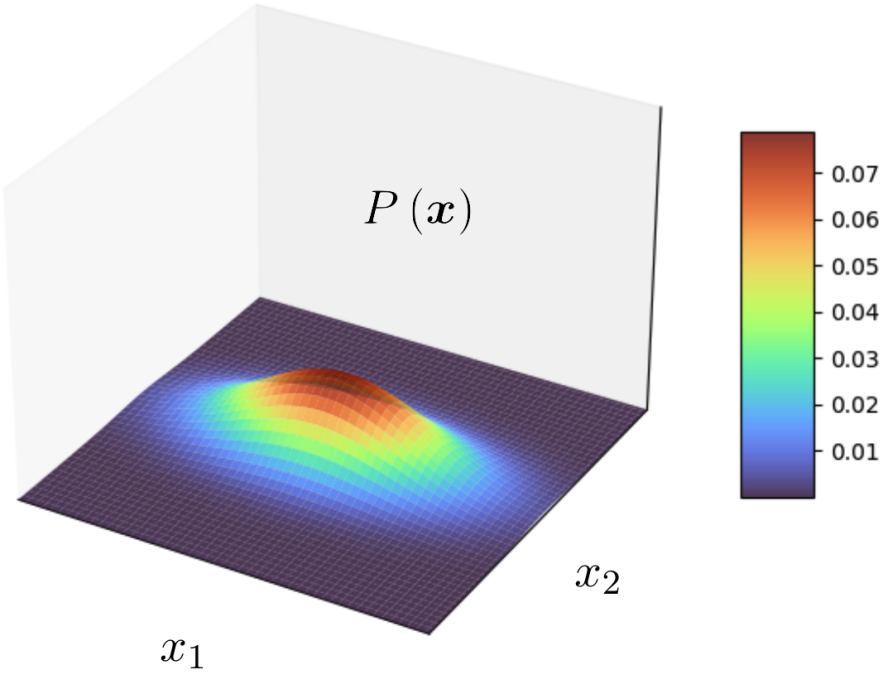
Example: *Bivariate Gaussian Distribution*

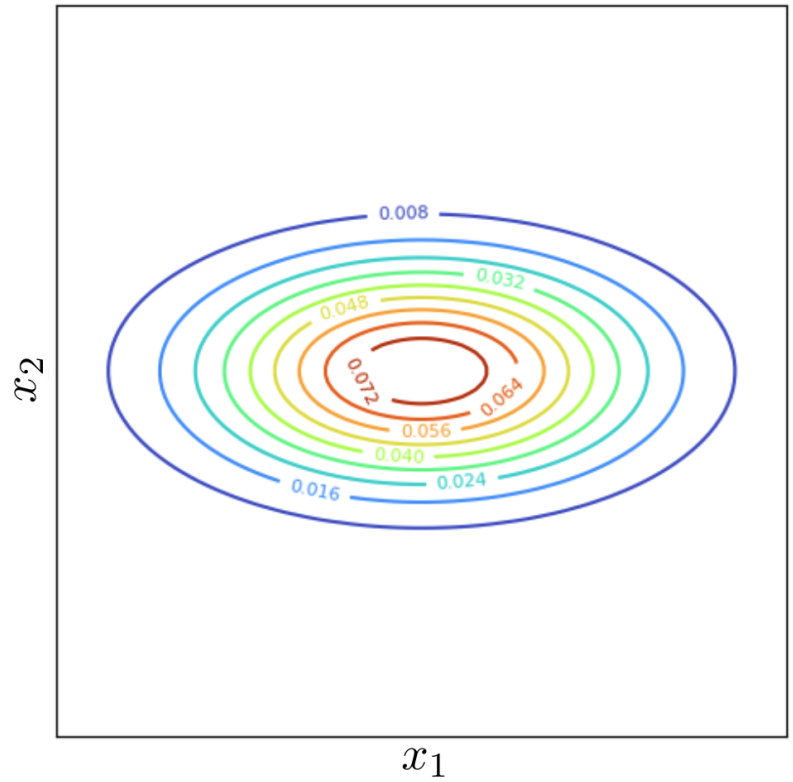
$$\boldsymbol{\Sigma} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$



Example: *Bivariate Gaussian Distribution*

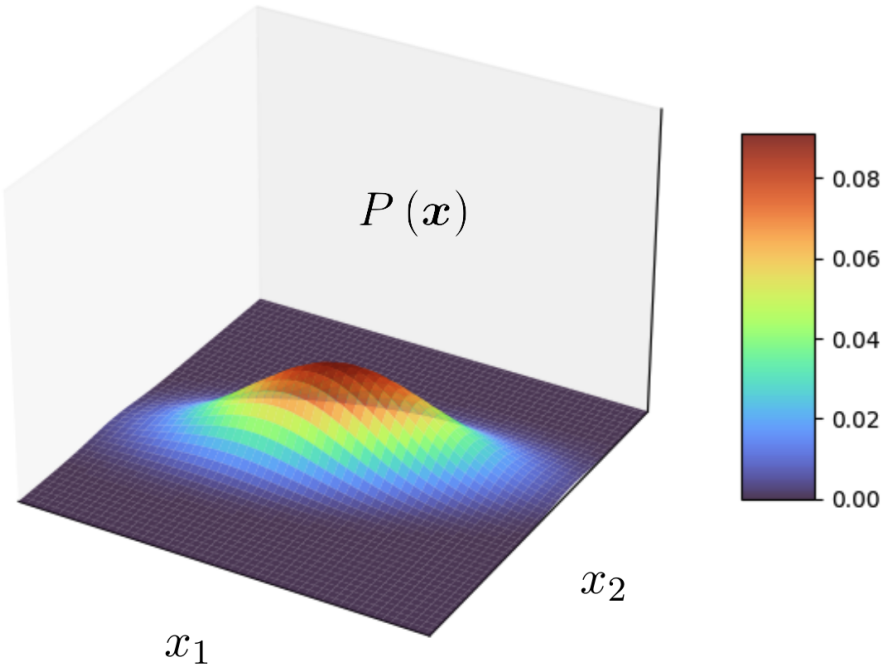
$$\boldsymbol{\Sigma} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

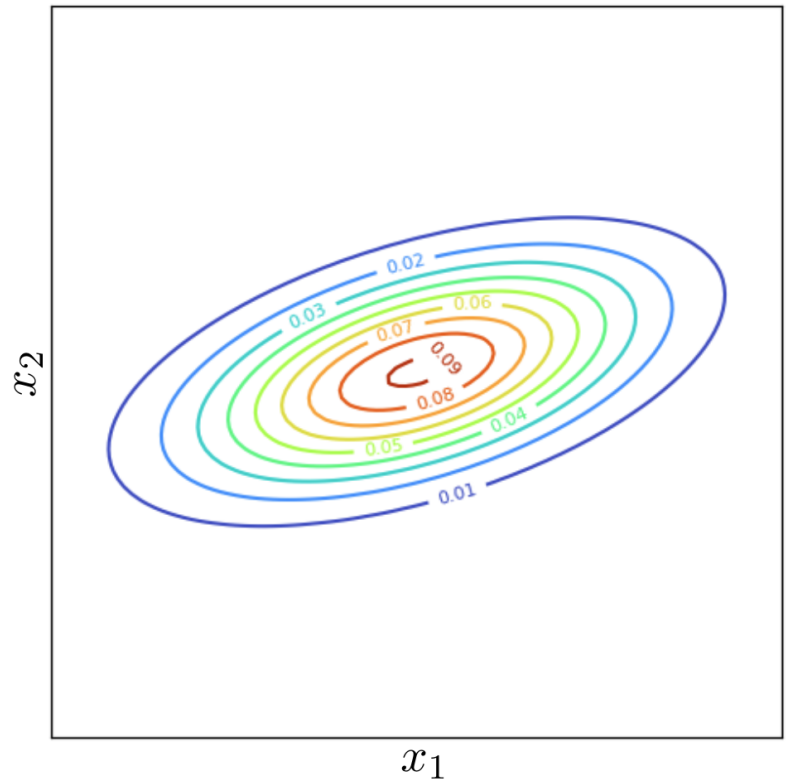




Example: *Bivariate Gaussian Distribution*

$$\boldsymbol{\Sigma} = \begin{bmatrix} 4 & 1 \\ 1 & 1 \end{bmatrix}$$





Multivariate Gaussian Distribution

A Gaussian model assumes the data is sampled from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Gaussian Model

Points $\boldsymbol{x}_n \in \mathbb{D}$ are sampled from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$; thus,

$$P_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{x}_n) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu})\right\}$$

Log-likelihood is then

$$\mathcal{S}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log\det(\boldsymbol{\Sigma}) - \frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu})$$

Why Gaussian?

Many processes in nature are approximately Gaussian

Central Limit Theorem – *Informal*

Sum of large enough independent random objects tends to a Gaussian object

---

Superposition of Gaussians can describe a large class of distributions

$$\mathcal{Q} \equiv \sum_{m=1}^{M} \pi_m \mathcal{N}\left(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\right)$$

↳ *We mention it again later*

Gaussian Maximum Likelihood

We assume

$$\mathbb{D} = \{\boldsymbol{x}_n : n = 1, \ldots, N\}$$

contains samples of $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

No harm if we further normalize

$$\mathcal{S}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{S}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$= -\frac{d}{2} \log 2\pi - \frac{1}{2} \log \det(\boldsymbol{\Sigma}) - \frac{1}{2N} \sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu})$$

Gaussian Maximum Likelihood: *Identity Covariance*

Assume we know that $\boldsymbol{\Sigma} = \mathbf{I}$

$$\mathcal{S}(\boldsymbol{\mu}) = -\frac{d}{2}\log 2\pi - \frac{1}{2N}\sum_{n=1}^{N}\|\boldsymbol{x}_n - \boldsymbol{\mu}\|^2$$

Maximum likelihood estimate is at the center of cluster

$$\boldsymbol{\mu}^\star = \underset{\boldsymbol{\mu}}{\operatorname{argmax}}\,\mathcal{S}(\boldsymbol{\mu}) = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n$$

! *That makes sense!*

Gaussian Maximum Likelihood: *Unknown Covariance*

For general $\boldsymbol{\Sigma}$, we have

$$\mathcal{S}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log\det\left(\boldsymbol{\Sigma}\right) - \frac{1}{2N}\sum_{n=1}^{N}\left(\boldsymbol{x}_n - \boldsymbol{\mu}\right)^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}_n - \boldsymbol{\mu}\right)$$

Let's first think of one-dimensional case $\boldsymbol{\mu} = \mu$ and $\boldsymbol{\Sigma} = \sigma^2$

$$\mathcal{S}\left(\mu, \sigma^2\right) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log\sigma^2 - \frac{1}{2N\sigma^2}\sum_{n=1}^{N}\left(x_n - \mu\right)^2$$

? *How to find the optimal μ and σ^2?*

! *Set the partial derivatives to zero*

Gaussian Maximum Likelihood: *Unknown Covariance*

For general μ and σ^2, we have

$$\mathcal{S}\left(\mu, \sigma^2\right) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log \sigma^2 - \frac{1}{2N\sigma^2}\sum_{n=1}^{N}\left(x_n - \mu\right)^2$$

Take partial derivative w.r.t. μ

$$\frac{\partial \mathcal{S}}{\partial \mu} = \frac{1}{N\sigma^2}\sum_{n=1}^{N}\left(x_n - \mu\right)$$

and set to zero

$$\frac{\partial \mathcal{S}}{\partial \mu} = 0 \rightsquigarrow \mu^\star = \frac{1}{N}\sum_{n=1}^{N} x_n$$

Gaussian Maximum Likelihood: *Unknown Covariance*

For general μ and σ^2, we have

$$\mathcal{S}\left(\mu, \sigma^2\right) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log \sigma^2 - \frac{1}{2N\sigma^2}\sum_{n=1}^{N}\left(x_n - \mu\right)^2$$

Take partial derivative w.r.t. σ^2

$$\frac{\partial \mathcal{S}}{\partial \sigma^2} = -\frac{1}{2\sigma^2} + \frac{1}{N\left(\sigma^2\right)^2}\sum_{n=1}^{N}\left(x_n - \mu\right)^2$$

and set to zero

$$\frac{\partial \mathcal{S}}{\partial \sigma^2} = 0 \rightsquigarrow \sigma^{\star 2} = \frac{1}{N}\sum_{n=1}^{N}\left(x_n - \mu^{\star}\right)^2$$

Gaussian Maximum Likelihood: *Unknown Covariance*

In vector form, we do the same with gradients: *set gradient w.r.t.* $\boldsymbol{\mu}$ *to zero*

$$\nabla_{\boldsymbol{\mu}} \mathcal{S}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = \mathbf{0} \rightsquigarrow \boldsymbol{\mu}^{\star} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n = \mathtt{avg}\left(\mathbb{D}\right)$$

And then gradient w.r.t. $\boldsymbol{\Sigma}$ *to zero*

$$\nabla_{\boldsymbol{\Sigma}} \mathcal{S}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = \mathbf{0} \rightsquigarrow \boldsymbol{\Sigma}^{\star} = \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{\mu}^{\star})(\boldsymbol{x}_n - \boldsymbol{\mu}^{\star})^{\mathsf{T}} = \mathtt{cov}\left(\mathbb{D}\right)$$

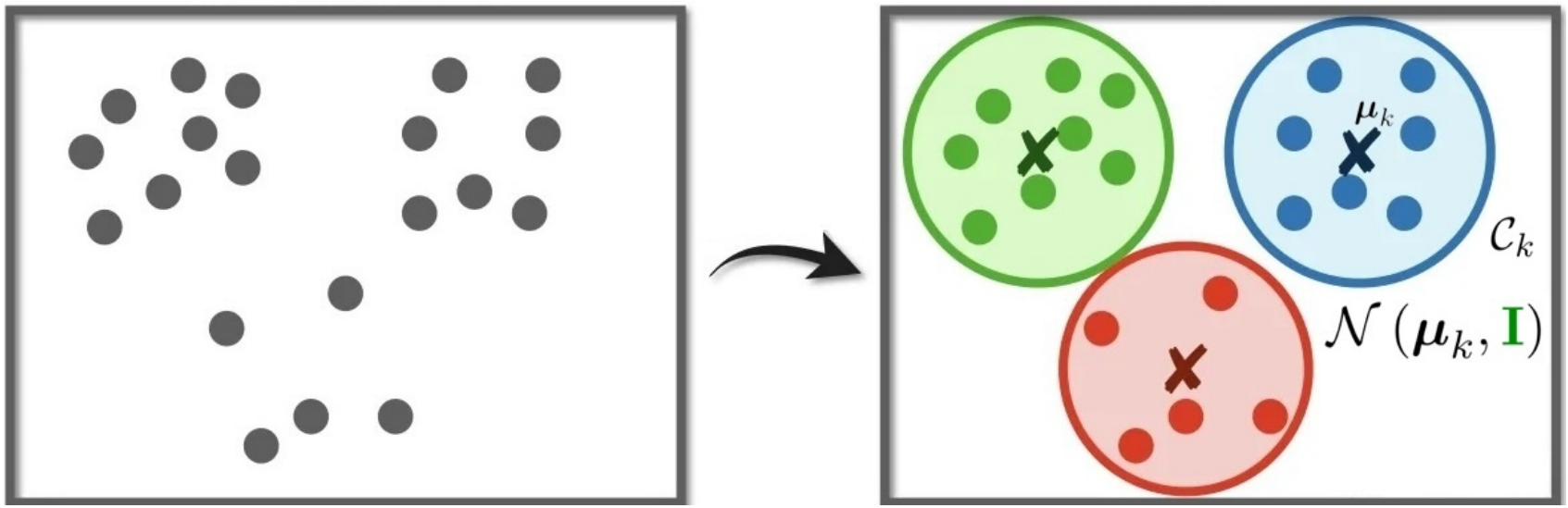
Attention

No worries if it's new for you! We will discuss gradient descent and function optimization in the next weeks!

Gaussian Model for Clustering

For clustering, assume *each cluster is sampled from a Gaussian distribution*

$$\boldsymbol{x}_n \in \mathcal{C}_k \rightsquigarrow \boldsymbol{x}_n \sim \mathcal{N}\left(\boldsymbol{\mu}_k, \mathbf{I}\right)$$



Maximum Likelihood: *Estimating Cluster Set*

? *Given K centroids, to which cluster belongs $\boldsymbol{x}_n$?*

Log-likelihood of $\boldsymbol{x}_n \in \mathcal{C}_k$ is

$$\mathcal{S}_n\left(k\right) = \log P_k\left(\boldsymbol{x}_n\right) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

The maximum likelihood estimate is thus given by

$$k^\star \leftarrow \underset{k}{\operatorname{argmax}}\, \mathcal{S}_n\left(k\right) = \underset{k}{\operatorname{argmin}} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

$$\equiv \texttt{Cluster\_Assignment}\left(\right)$$

Maximum Likelihood: *Estimating Centroids*

? *Given samples of a cluster $\mathcal{C}_k$, what is best $\boldsymbol{\mu}_k$?*

Log-likelihood of $\boldsymbol{\mu}_k$ is

$$\mathcal{S}_{\mathcal{C}_k}\left(\boldsymbol{\mu}_k\right) = \frac{1}{|\mathcal{C}_k|}\log P_{\boldsymbol{\mu}_k}\left(\mathcal{C}_k\right) = -\frac{d}{2}\log 2\pi - \frac{1}{2|\mathcal{C}_k|}\sum_{\boldsymbol{x}_n \in \mathcal{C}_k}\left\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\right\|^2$$

The maximum likelihood is thus given by

$$\boldsymbol{\mu}_k^\star \leftarrow \underset{\boldsymbol{\mu}_k}{\operatorname{argmax}}\, \mathcal{S}_{\mathcal{C}_k}\left(\boldsymbol{\mu}_k\right) = \frac{1}{|\mathcal{C}_k|}\sum_{\boldsymbol{x}_n \in \mathcal{C}_k}\left\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\right\|^2$$

$$\equiv \texttt{Centroid\_Update}\left(\right)$$

Conclusion: K-Means Maximizes Likelihoods

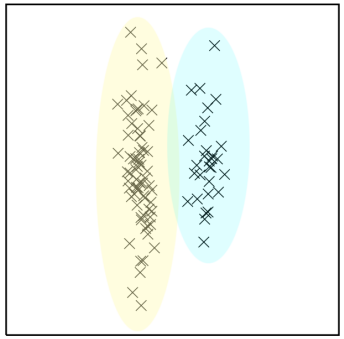
K-Means():

1: Initiate $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
2: **while** $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ changing **do**
3: Set $\mathcal{C}_1, \ldots, \mathcal{C}_K \leftarrow$ Cluster_Assignment$(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$
4: Update $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \leftarrow$ Centroid_Update$(\mathcal{C}_1, \ldots, \mathcal{C}_K)$
5: **end while**
6: Return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

$$K\text{-Means Clustering} \equiv \text{Iterative Maximum Likelihood}$$

Possible Extensions

- *In clustering, we could also assume $\boldsymbol{\Sigma}$ to be unknown*



- *In general, we can assume a distribution $\mathcal{Q}$ that is mixture of Gaussians*

$$\mathcal{Q} \equiv \sum_{m=1}^{M} \pi_m \mathcal{N}\left(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\right) \text{ with } \sum_{m=1}^{M} \pi_m = 1$$

↳ *Gaussian Mixture Model (GMM)*
↳ *We estimate $\pi_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m$ for $m = 1, \ldots, M$ all together*
↳ *Larger M makes it better, but needs more computation!*

Data Generation

Say $\boldsymbol{x}$ collects all pixel values of an image

$$\mathbb{D} = \{\boldsymbol{x}_n = \texttt{cat image} : n = 1, \ldots, N\}$$

We learn the distribution by finding a good estimate $P_{\boldsymbol{\theta}}(\boldsymbol{x})$

$$\boldsymbol{x}_{new} \sim P_{\boldsymbol{\theta}}(\boldsymbol{x}) \approx P_{data}(\boldsymbol{x})$$

$\boldsymbol{x}_{new}$ *is a new cat image!*

In practice, we need more sophisticated method to estimate $P_{data}(\boldsymbol{x})$

Further Read

- Bishop
  - ↳ Chapter 1: *Section 1.2* — ***Probability Review***
  - ↳ Chapter 2: *Sections 2.1 and 2.3* — ***Gaussian Maximum Likelihood***
  - ↳ Chapter 9: *Section 9.2* — ***Mixture of Gaussians***
- MacKay
  - ↳ Chapter 22 — ***Maximum Likelihood vs Clustering***
  - ↳ Chapter 23 — ***More on Probability***
- Goodfellow
  - ↳ Chapter 3 — ***Probability Review***
  - ↳ Chapter 5: *Section 5.5* — ***Maximum Likelihood***
- ESL
  - ↳ Chapter 14: *Section 14.3.7* — ***Mixture of Gaussians***

Testing A Model

? *How can we know if our model has learned the pattern?*

! *We can test it*

Collect a new set of samples

$$\mathbb{T} = \{\boldsymbol{x}_i : i = 1, \ldots, I\}$$

Use the trained model $\boldsymbol{\theta}^\star$ to find the likelihood

$$\mathcal{S}_{\mathbb{T}}\left(\boldsymbol{\theta}^\star\right) = \log P_{\boldsymbol{\theta}^\star}\left(\mathbb{T}\right)$$

↳ *If* $\mathcal{S}_{\mathbb{T}}\left(\boldsymbol{\theta}^\star\right) \approx \mathcal{S}_{\mathbb{D}}\left(\boldsymbol{\theta}^\star\right) \rightsquigarrow$ *the model generalizes*

↳ *If* $\mathcal{S}_{\mathbb{T}}\left(\boldsymbol{\theta}^\star\right) \ll \mathcal{S}_{\mathbb{D}}\left(\boldsymbol{\theta}^\star\right) \rightsquigarrow$ *the model does not generalize!*

↳ *Maybe we need more complicated or simpler model!*

Validating the Model

We typically reserve some samples to validate

$$\mathbb{V} = \{\boldsymbol{x}_j : i = 1, \ldots, J\}$$

We train the model $\boldsymbol{\theta}^\star$ on $\mathbb{D}$; then, check

$$\mathcal{S}_{\mathbb{V}}\left(\boldsymbol{\theta}^\star\right) = \log P_{\boldsymbol{\theta}^\star}\left(\mathbb{V}\right)$$

- *If* $\mathcal{S}_{\mathbb{V}}\left(\boldsymbol{\theta}^\star\right) \approx \mathcal{S}_{\mathbb{D}}\left(\boldsymbol{\theta}^\star\right) \rightsquigarrow$ *the model is valid*
  - *We now check generalization on the test set* $\mathbb{T}$
- *If* $\mathcal{S}_{\mathbb{V}}\left(\boldsymbol{\theta}^\star\right) \ll \mathcal{S}_{\mathbb{D}}\left(\boldsymbol{\theta}^\star\right) \rightsquigarrow$ *the model is not valid!*
  - *We change some hyperparameters in the model and repeat*

Test and Validation: *Example*

Consider clustering problem

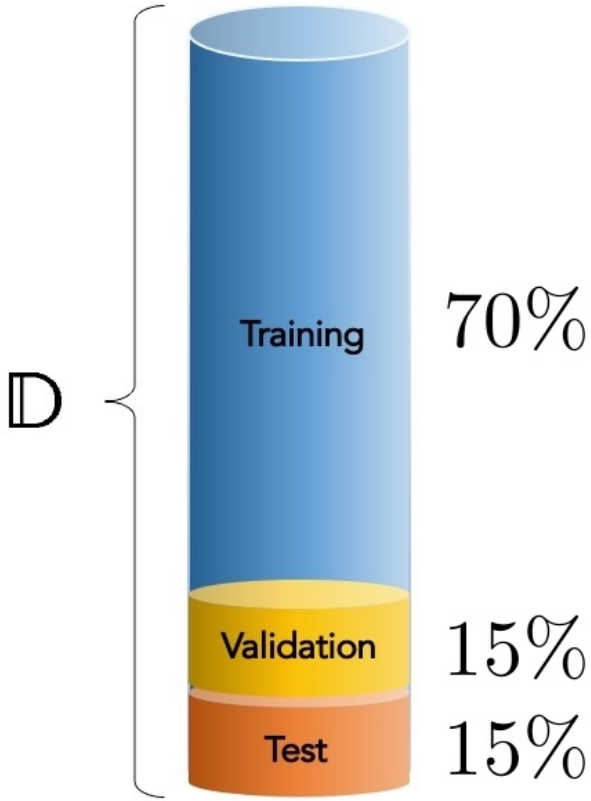
1. *Choose some hyperparameter K*
2. *Train via K-means clustering on $\mathbb{D}$*
   - ↳ *Find $\mu_1, \ldots, \mu_K$ and save the final risk*

$$\mathcal{J}_{\text{train}} = \frac{1}{N} \sum_{k=1}^{K} \sum_{n=1}^{N} r_{n,k} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

3. *Compute the risk on the set $\mathbb{V}$ and call it $\mathcal{J}_{\text{val}}$*
   - ↳ *If $\mathcal{J}_{\text{val}} \approx \mathcal{J}_{\text{train}}$ $\rightsquigarrow$ the model is valid*
     - ↳ *We now test if $\mathcal{J}_{\text{test}}$ on $\mathbb{T}$ is also close*
   - ↳ *If $\mathcal{J}_{\text{val}} \gg \mathcal{J}_{\text{train}}$ $\rightsquigarrow$ the model is not valid!*
     - ↳ *We change K to higher/lower value and get back to (1)*

Training vs Test Data

In practice, we split our collected data



Further Read

- Bishop
  - ↳ Chapter 1: *Section 1.3* — ***Validation and Test***
- Goodfellow
  - ↳ Chapter 5: *Sections 5.2 and 5.3* — ***Generalization***