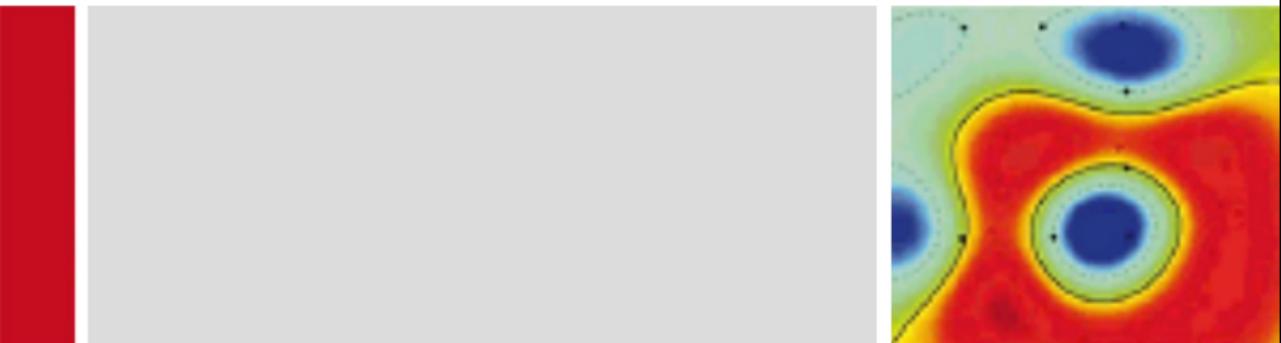




WiSe 2024/25

Machine Learning 1/1-X



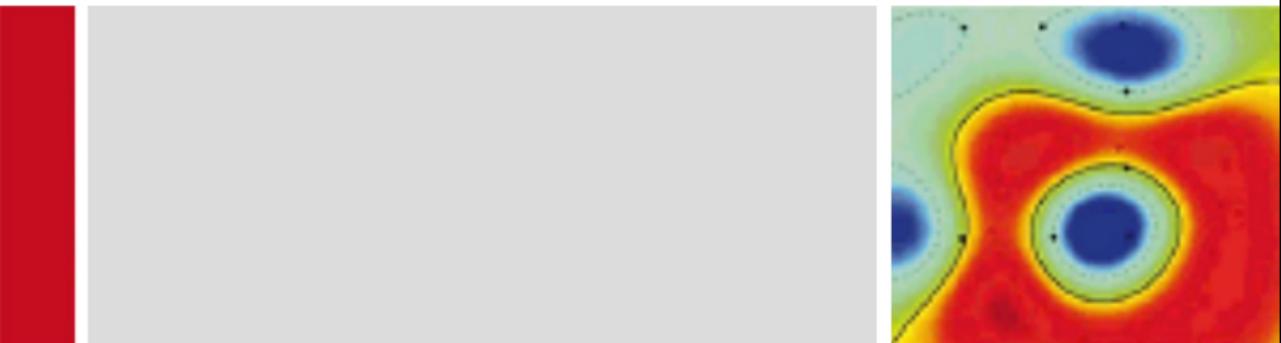
Lecture 2

Parameter Estimation



WiSe 2024/25

Machine Learning 1/1-X



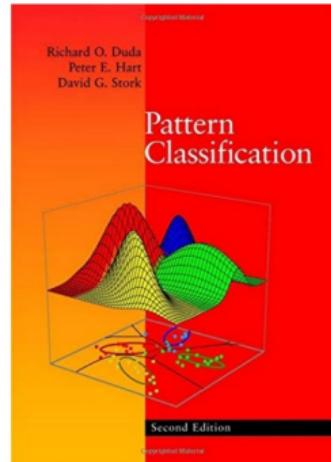
Lecture 2

Parameter Estimation

Textbook Reference

Duda et al. Pattern Classification, 2nd Edition
(2000)

- ▶ This week: Sections 3.1–3.5



Recap: Bayes Decision Theory

Recap:

- ▶ Knowing class priors and class-conditioned data densities, we can infer posterior probabilities using the Bayes theorem

$$P(\omega_j | x) = \frac{p(x | \omega_j) \cdot P(\omega_j)}{p(x)}$$

- ▶ From there, optimal classifiers can be built, e.g. the maximum accuracy classifier:

$$\begin{aligned} \arg \max_j [P(\omega_j | x)] \\ = \arg \max_j [\log p(x | \omega_j) + \log P(\omega_j)] \end{aligned}$$

Question:

- ▶ Can we assume that we know in advance the data densities $p(x | \omega_j)$, or should they be learned from the data?

Learning $p(x | \omega_j)$: Histogram

理念:

- 在输入空间中构建一个网格，对于每个类别，构建一个直方图来统计属于每个箱子的观测数量。

问题:

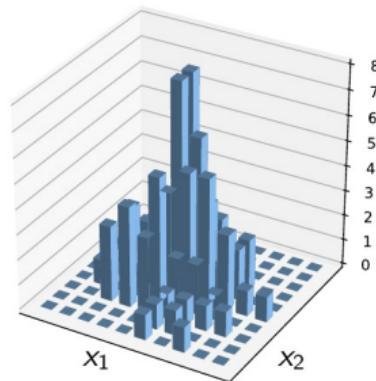
- 箱子的数量是 s^d ，其中 s 是每个维度上的步数， d 是维度的数量。
- 许多箱子将没有观测值，这并不是因为数据的概率真的为零，而是因为观测到的样本数量有限。

Idea:

- Build a grid in input space, and for each class build a histogram that counts the number of observations that belong to each bin.

Problem:

- The number of bins is s^d where s is the number of steps along each dimension and d is the number of dimensions.
- Many bins will have zero observations, not because the data probability is truly zero, but because finitely many examples have been observed.



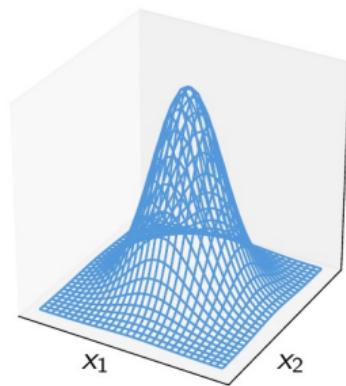
Learning $p(x | \omega_j)$: Model-Based

Idea:

- ▶ Assume that $p(x | \omega_j)$ is a known parametric form, e.g. $\mathcal{N}(\mu_j, \Sigma_j)$ where μ_j, Σ_j are the parameters of the distribution.
- ▶ Estimate the parameters of the distribution that best fit the few observations we have.

Advantage:

- ▶ With the model-based approach, we need to estimate a finite and typically small number of model parameters and not the whole data distribution.



Maximum Likelihood Estimation

Goal: Let $\{p(x | \theta), \theta \in \Theta\}$ be a set of density functions (e.g. Gaussian), where θ denotes a parameter vector (e.g. mean / covariance). We would like to find the parameter θ that is the most likely with respect to the data.

Maximum Likelihood (ML) Approach:

- ▶ Assume that we have a dataset $\mathcal{D} = (x_1, \dots, x_N)$. $X \in \mathbb{R}^{N \times d}$
- ▶ Assume that each example $x_k \in \mathbb{R}^d$ in the dataset has been generated independently and from the same density function $p(x | \theta)$. "i.i.d."
- ▶ In that case, the joint density function can be written as:

$$p(\mathcal{D} | \theta) = \prod_{k=1}^N p(x_k | \theta)$$

We also call this quantity the *likelihood* of θ w.r.t. the dataset \mathcal{D} .

- ▶ The best parameter is then given by $\hat{\theta} = \arg \max_{\theta} p(\mathcal{D} | \theta)$.

Max Likelihood, Simple Gaussian Case

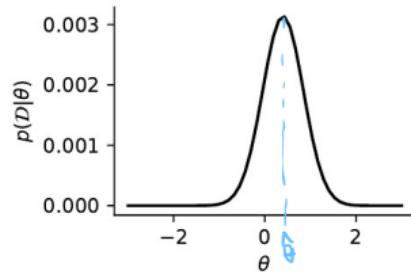
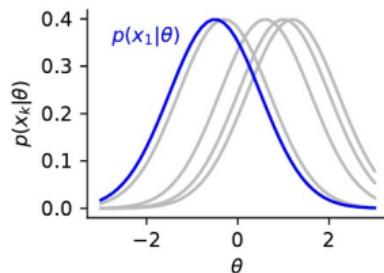
Assume the data density is modeled as a univariate Gaussian with unit variance and unknown mean θ . For a given data point x_k , the density function can be written as:

$$p(x_k | \theta) = \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{1}{2}(x_k - \theta)^2 \right]$$

Using the independence assumption, the joint density becomes:

$$p(\mathcal{D} | \theta) = \prod_{k=1}^N \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{1}{2}(x_k - \theta)^2 \right]$$

Question: How to find θ that maximizes the function $p(\mathcal{D} | \theta)$.



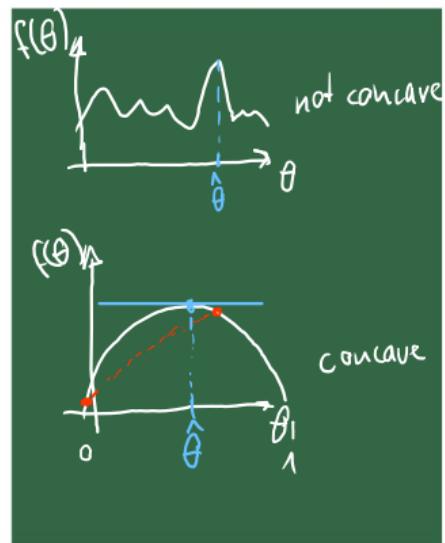
Finding the Maximum of a Function

For some function f of interest (e.g. the data likelihood) we would like to compute:

$$\hat{\theta} = \arg \max_{\theta} f(\theta)$$

When the function to optimize is **凹地** continuously differentiable and concave, the maximum is found at the point where the gradient is zero.

$$\nabla_{\theta} f(\theta) = \begin{bmatrix} \partial f / \partial \theta_1 \\ \partial f / \partial \theta_2 \\ \vdots \\ \partial f / \partial \theta_h \end{bmatrix} = \mathbf{0}$$



Max Likelihood, Simple Gaussian Case

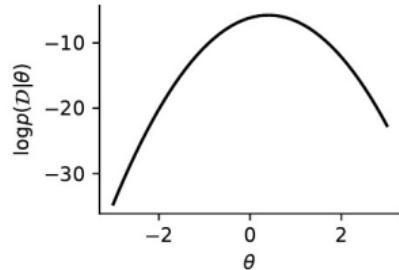
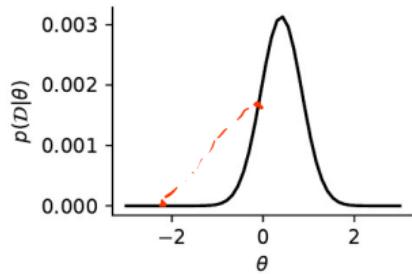
Observation: The function $p(\mathcal{D} | \theta)$ is not concave with θ .

Idea: Transform the function in a way that

- (i) doesn't change its argmax and
- (ii) makes it concave.

Applying the logarithm ensures the two properties above:

$$\begin{aligned}\log p(\mathcal{D} | \theta) &= \log \prod_{k=1}^N \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{1}{2}(x_k - \theta)^2 \right] \\ &= \sum_{k=1}^N \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2}(x_k - \theta)^2 \right]\end{aligned}$$



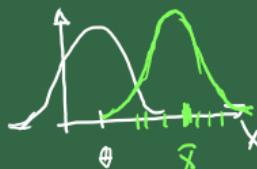
Max Likelihood, Simple Gaussian Case

Having found the log-likelihood w.r.t. \mathcal{D} to be

$$\log p(\mathcal{D} | \theta) = \sum_{k=1}^N \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2}(x_k - \theta)^2 \right],$$

the best parameter $\hat{\theta}$ can then be found by solving $\nabla_{\theta} \log P(\mathcal{D} | \theta) = 0$.

$$\begin{aligned}\nabla_{\theta} \log p(\mathcal{D} | \theta) &= \sum_{k=1}^N -\frac{1}{2} \cdot 2(x_k - \theta) \cdot (-1) \stackrel{!}{=} 0 \\ \Leftrightarrow \sum_{k=1}^N x_k &= N \cdot \theta \quad \Rightarrow \quad \hat{\theta} = \frac{1}{N} \sum_{k=1}^N x_k\end{aligned}$$



Max Likelihood, Multivariate Case

The log-likelihood of a *multivariate Gaussian distribution* w.r.t. \mathcal{D} is given by

$$\log p(\mathcal{D} | \mu) = \sum_{k=1}^N -\frac{1}{2} \log [(2\pi)^d \det(\Sigma)] - \frac{1}{2} (\mathbf{x}_k - \mu)^\top \Sigma^{-1} (\mathbf{x}_k - \mu)$$

$\nabla_{\mu} \log p(\mathcal{D} | \mu) = -\Sigma^{-1} \Rightarrow$ concave ✓

Question: Assuming Σ is fixed, what is the optimal parameter vector μ ?

$$\nabla_{\mu} \log p(\mathcal{D} | \theta) = \sum_{k=1}^N 0 + \Sigma^{-1} (\mathbf{x}_k - \mu) \stackrel{!}{=} 0$$

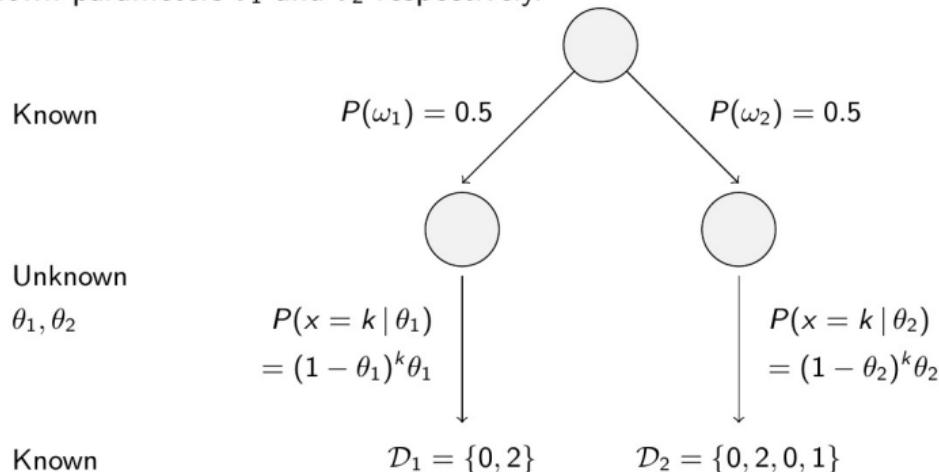
$$\Rightarrow \frac{1}{N} \sum_{k=1}^N \Sigma^{-1} \mathbf{x}_k - \Sigma^{-1} \mu \quad | \quad \Sigma(\cdot)$$

$$\Rightarrow \frac{1}{N} \sum_{k=1}^N \mathbf{x}_k = \hat{\mu}$$



Building a Classifier with ML

Consider a labeled dataset containing two examples for the first class, and four examples for the second class. Points are in \mathbb{N}_0 and we assume they are generated from classes ω_1, ω_2 following geometric probability distributions of unknown parameters θ_1 and θ_2 respectively.



Building a Classifier with ML

Recall:

$$P(\omega_1) = 0.5$$

$$P(\omega_2) = 0.5$$

$$P(x = k | \theta_1) = (1 - \theta_1)^k \theta_1$$

$$P(x = k | \theta_2) = (1 - \theta_2)^k \theta_2$$

$$\mathcal{D}_1 = \{0, 2\}$$

$$\mathcal{D}_2 = \{0, 2, 0, 1\}$$

$$\begin{aligned} p(\mathcal{D}_1 | \theta_1) &= p(x=0|\theta_1) \cdot p(x=2|\theta_1) \\ &= \theta_1 \cdot (1-\theta_1)^2 \theta_1 \\ &= \theta_1^3 \cdot (1-\theta_1)^2 \\ \log p(\mathcal{D}_1 | \theta_1) &= 2 \log \theta_1 + 2 \log (1-\theta_1) \\ \nabla_{\theta_1} \log p(\mathcal{D}_1 | \theta_1) &= \frac{2}{\theta_1} + \frac{2}{1-\theta_1} \cdot (-1) \stackrel{!}{=} 0 \\ \hat{\theta}_1 &= 1/2 \end{aligned}$$

Building a Classifier with ML

Recall:

$$P(\omega_1) = 0.5$$

$$P(\omega_2) = 0.5$$

$$P(x = k \mid \theta_1) = (1 - \theta_1)^k \theta_1$$

$$P(x = k \mid \theta_2) = (1 - \theta_2)^k \theta_2$$

$$\mathcal{D}_1 = \{0, 2\}$$

$$\mathcal{D}_2 = \{0, 2, 0, 1\}$$

$$P(D_2 \mid \theta_2) = \prod_{k=1}^N P(x_k \mid \theta_2)$$

$$= \theta_2^4 (1 - \theta_2)^3$$

:

:

$$\hat{\theta}_2 = 4/7$$

Building a Classifier with ML

Recall:

$$P(\omega_1) = 0.5$$

$$P(\omega_2) = 0.5$$

$$P(x = k | \theta_1) = (1 - \theta_1)^k \theta_1$$

$$P(x = k | \theta_2) = (1 - \theta_2)^k \theta_2$$

$$\mathcal{D}_1 = \{0, 2\}$$

$$\mathcal{D}_2 = \{0, 2, 0, 1\}$$

$$\arg \max_j p(\omega_j | x) = \arg \max_j p(x | \omega_j) p(\omega_j)$$

Classify $x = 1$:

$$\omega_1 : p(x = 1 | \theta_1) \stackrel{(x=1)}{=} (1 - \theta_1)^1 \cdot \theta_1 \cdot p(\omega_1)$$
$$= \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = 0.125$$
$$\omega_2 : p(x = 1 | \theta_2) \stackrel{(x=1)}{=} (1 - \theta_2)^1 \cdot \theta_2 \cdot p(\omega_2)$$
$$= \frac{3}{4} \cdot \frac{1}{3} \cdot \frac{1}{2}$$
$$= 0.125$$

\Rightarrow decide ω_1

Bayes Parameter Estimation

Assuming some dataset $\mathcal{D} = (x_1, \dots, x_N)$. We would like to model this dataset using some probability function $p(\mathcal{D}|\theta)$ with θ some unknown parameter that needs to be learned.

ML parameter estimation: Chose the parameter θ that maximises the data likelihood:

$$\hat{\theta} = \arg \max_{\theta} p(\mathcal{D}|\theta)$$

Bayes parameter estimation: Instead of learning a fixed estimate, build a posterior distribution over this parameter using the Bayes theorem:

$$\begin{aligned} p(\theta|\mathcal{D}) &= \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} \\ &= \frac{p(\mathcal{D}|\theta)p(\theta)}{\int_{\theta} p(\mathcal{D}|\theta)p(\theta)d\theta} \end{aligned}$$

This approach requires to define a *prior* distribution $p(\theta)$ which models our initial belief about the parameter θ before observing the data.

From ML to Bayes Classifiers

ML classifier: Class posteriors are given by:

$$P(\omega_i | \mathbf{x}) = \frac{p(\mathbf{x} | \omega_i; \hat{\theta}_i) P(\omega_i)}{\sum_i p(\mathbf{x} | \omega_i; \hat{\theta}_i) P(\omega_i)}$$

where $\hat{\theta}_i$ is the maximum likelihood parameter for the distribution of class ω_i .

Bayes classifier: Class posteriors are computed by bypassing the intermediate computation of the parameters $\hat{\theta}_i$, and instead conditioning directly on the data:

贝叶斯分类器：类后验概率通过绕过参数 $\hat{\theta}_i$ 的中间计算，直接基于数据进行条件化来计算：

$$\begin{aligned} P(\omega_i | \mathbf{x}, \mathcal{D}) &= \frac{p(\mathbf{x} | \omega_i, \mathcal{D}) P(\omega_i | \mathcal{D})}{\sum_i p(\mathbf{x} | \omega_i, \mathcal{D}) P(\omega_i | \mathcal{D})} \\ &= \frac{p(\mathbf{x} | \omega_i, \mathcal{D}_i) P(\omega_i)}{\sum_i p(\mathbf{x} | \omega_i, \mathcal{D}_i) P(\omega_i)} \end{aligned}$$

Bayes Classifiers (cont.)

The terms of the class posterior:

$$P(\omega_i | \mathbf{x}, \mathcal{D}) = \frac{p(\mathbf{x} | \omega_i, \mathcal{D}_i) P(\omega_i)}{\sum_i p(\mathbf{x} | \omega_i, \mathcal{D}_i) P(\omega_i)}$$

can be expressed with model parameters as:

$$\begin{aligned} p(\mathbf{x} | \omega_i, \mathcal{D}_i) &= \int p(\mathbf{x} | \theta_i, \omega_i, \mathcal{D}_i) p(\theta_i | \omega_i, \mathcal{D}_i) d\theta_i \\ &= \int p(\mathbf{x} | \theta_i) p(\theta_i | \mathcal{D}_i) d\theta_i \end{aligned}$$

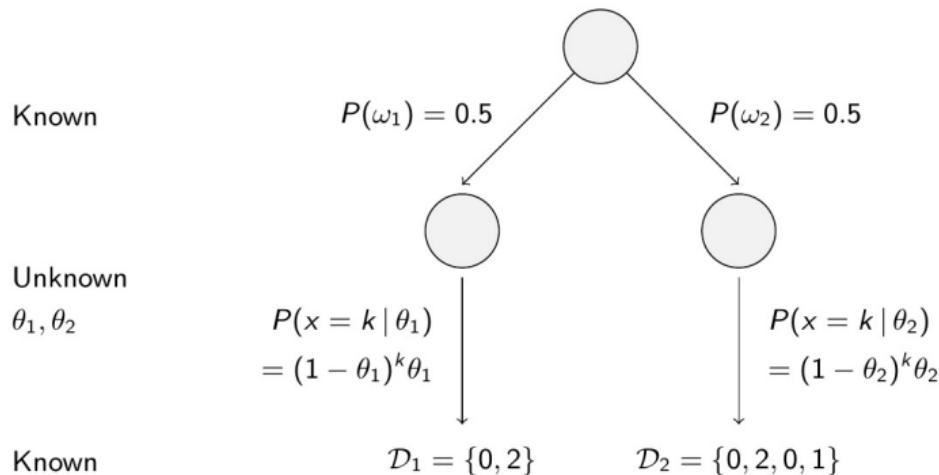
and

$$p(\theta_i | \mathcal{D}_i) = \frac{p(\mathcal{D}_i | \theta_i) p(\theta_i)}{\int p(\mathcal{D}_i | \theta_i) p(\theta_i) d\theta_i}.$$

are our Bayes parameter estimates.

Building a Classifier with Bayes

Recall: we consider the following data generating process



Building a Classifier with Bayes

Recall:

$$P(\omega_1) = 0.5$$

$$P(\omega_2) = 0.5$$

$$P(x = k | \theta_1) = (1 - \theta_1)^k \theta_1$$

$$P(x = k | \theta_2) = (1 - \theta_2)^k \theta_2$$

$$\mathcal{D}_1 = \{0, 2\}$$

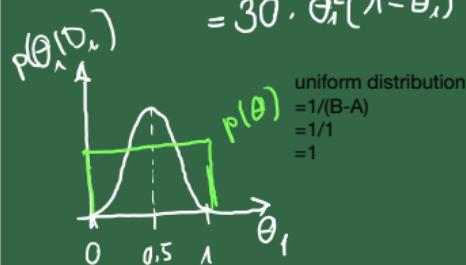
$$\mathcal{D}_2 = \{0, 2, 0, 1\}$$

... and we further set:

$$p(\theta_1) \sim \mathcal{U}(0, 1)$$

$$p(\theta_2) \sim \mathcal{U}(0, 1)$$

$$\begin{aligned} P(\theta_1 | \mathcal{D}_1) &= \frac{p(\mathcal{D}_1 | \theta_1) p(\theta_1)}{\int p(\mathcal{D}_1 | \theta_1) p(\theta_1) d\theta_1} \\ &= \frac{\theta_1^2 (1 - \theta_1)^2 \cdot 1}{\int \theta_1^2 (1 - \theta_1)^2 \cdot 1 d\theta_1} \\ &= 30 \cdot \theta_1^2 (1 - \theta_1)^2 \end{aligned}$$



Building a Classifier with Bayes

Recall:

$$P(\omega_1) = 0.5$$

$$P(\omega_2) = 0.5$$

$$P(x = k | \theta_1) = (1 - \theta_1)^k \theta_1$$

$$P(x = k | \theta_2) = (1 - \theta_2)^k \theta_2$$

$$\mathcal{D}_1 = \{0, 2\}$$

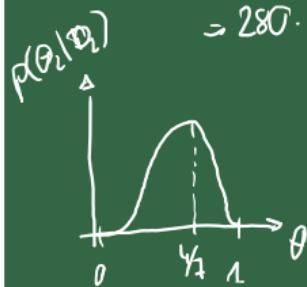
$$\mathcal{D}_2 = \{0, 2, 0, 1\}$$

... and we further set:

$$p(\theta_1) \sim \mathcal{U}(0, 1)$$

$$p(\theta_2) \sim \mathcal{U}(0, 1)$$

$$\begin{aligned} p(\theta_2 | \mathcal{D}_1) &= \frac{p(\mathcal{D}_1 | \theta_2) p(\theta_2)}{\int p(\mathcal{D}_1 | \theta_1) p(\theta_1) d\theta_1} \\ &= \frac{\theta_2^4 (1 - \theta_2)^2 \cdot 1}{\int \theta_1^4 (1 - \theta_1)^2 d\theta_1} \\ &= 280 \cdot \theta_2^4 (1 - \theta_2)^2 \end{aligned}$$



Building a Classifier with Bayes

Recall:

$$P(\omega_1) = 0.5$$

$$P(\omega_2) = 0.5$$

$$P(x = k | \theta_1) = (1 - \theta_1)^k \theta_1$$

$$P(x = k | \theta_2) = (1 - \theta_2)^k \theta_2$$

$$\mathcal{D}_1 = \{0, 2\}$$

$$\mathcal{D}_2 = \{0, 2, 0, 1\}$$

... and we further set:

$$p(\theta_1) \sim \mathcal{U}(0, 1)$$

$$p(\theta_2) \sim \mathcal{U}(0, 1)$$

$$\arg \max_j P(\omega_j | x, D) = \arg \max_j P(x | \omega_j, D) p(\omega_j)$$

Classify $x = 1$

$$\omega_1 : P(x = 1 | \omega_1, D) = \int_0^1 P(x = 1 | \theta_1) p(\theta_1 | D_1) d\theta_1$$
$$= \int_0^{1/3} (1 - \theta_1)^3 \cdot 30 d\theta_1$$
$$= \frac{3}{14} \approx 0.214$$
$$\omega_2 : P(x = 1 | \omega_2, D) = \int_0^1 P(x = 1 | \theta_2) p(\theta_2 | D_2) d\theta_2$$
$$= \int_0^{1/2} (1 - \theta_2)^4 \cdot 280 d\theta_2$$
$$= 2/9 \approx 0.222$$

Building a Classifier with Bayes

Recall:

$$P(\omega_1) = 0.5$$

$$P(\omega_2) = 0.5$$

$$P(x = k | \theta_1) = (1 - \theta_1)^k \theta_1$$

$$P(x = k | \theta_2) = (1 - \theta_2)^k \theta_2$$

$$\mathcal{D}_1 = \{0, 2\}$$

$$\mathcal{D}_2 = \{0, 2, 0, 1\}$$

... and we further set:

$$p(\theta_1) \sim \mathcal{U}(0, 1)$$

$$p(\theta_2) \sim \mathcal{U}(0, 1)$$

$$\underset{i}{\operatorname{argmax}} p(x | \omega_i, D) p(\omega_i)$$

$$\omega_1 = 0.216 \cdot \underline{0.5}$$

$$\omega_2 = 0.211 \cdot 0.5$$

\Rightarrow decide ω_2

ML vs. Bayes Classifiers

Observations:

- ▶ ML and Bayes classifiers do not always produce the same decisions
- ▶ Bayes classifiers are influenced by the prior distribution and are consequently less sensitive to the data.
- ▶ Bayes classifiers will tend to favor the outcome that is supported by a larger amount of data.

图1: 观察结果

- 最大似然 (ML) 分类器和贝叶斯分类器并不总是做出相同的决策。
- 贝叶斯分类器受到先验分布的影响，因此对数据的敏感性较低。
- 贝叶斯分类器倾向于选择由更多数据支持的结果。

ML vs. Bayes: Gaussian Case

Consider the simple data density $p(x | \mu) = \mathcal{N}(\mu, \sigma^2)$ with unknown parameter μ . We would like to compare the ML and Bayes approaches to estimate the parameter μ from some dataset $\mathcal{D} = \{x_1, \dots, x_n\}$.

ML approach:

- ▶ The maximum likelihood estimate is given by $\hat{\mu} = \frac{1}{n} \sum_{k=1}^n x_k$, i.e. the empirical mean (cf. previous slides).

Bayes approach:

- ▶ Assuming some prior distribution $p(\mu) = \mathcal{N}(\mu_0, \sigma_0^2)$, the posterior distribution can be computed as:

$$p(\mu | \mathcal{D}) = \frac{p(\mathcal{D} | \mu)p(\mu)}{p(\mathcal{D})} = \alpha \prod_{k=1}^n p(x_k | \mu)p(\mu)$$

where α is a normalizing factor.

ML vs. Bayes: Gaussian Case

Bayes approach (continued):

- The posterior distribution can be expanded as:

$$\begin{aligned} p(\mu|\mathcal{D}) &= \alpha \prod_{k=1}^n \overbrace{\frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{1}{2} \left(\frac{x_k - \mu}{\sigma} \right)^2 \right]}^{p(x_k|\mu)} \overbrace{\frac{1}{\sqrt{2\pi}\sigma_0} \exp \left[-\frac{1}{2} \left(\frac{\mu - \mu_0}{\sigma_0} \right)^2 \right]}^{p(\mu)} \\ &= \alpha' \exp \left[-\frac{1}{2} \left(\sum_{k=1}^n \left(\frac{\mu - x_k}{\sigma} \right)^2 + \left(\frac{\mu - \mu_0}{\sigma_0} \right)^2 \right) \right] \\ &= \alpha'' \exp \left[-\frac{1}{2} \left[\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \right) \mu^2 - 2 \left(\frac{1}{\sigma^2} \sum_{k=1}^n x_k + \frac{\mu_0}{\sigma_0^2} \right) \mu \right] \right], \end{aligned}$$

which corresponds to a new Gaussian distribution $\mathcal{N}(\mu_n, \sigma_n^2)$ with parameters

$$\mu_n = \frac{\sigma_n^2}{\sigma^2/n} \hat{\mu} + \frac{\sigma_n^2}{\sigma_0^2} \mu_0 \quad \text{and} \quad \sigma_n^2 = \left(\frac{1}{\sigma^2/n} + \frac{1}{\sigma_0^2} \right)^{-1}$$

- We observe that (1) the mean estimate μ_n is now pulled towards the prior if too little data is available, and (2) the mean estimate comes with a variance term σ_n^2 that can be useful for confidence estimation.

图1: 观察到的现象

1. 如果数据量太少, 均值估计 μ_n 会被拉向先验。
2. 均值估计具有方差项 σ_n^2 , 这对于置信度估计非常有用。



Summary

- ▶ In practice, parameters of the class-conditioned distributions are not known and they must be inferred from some finite dataset.
- ▶ Two main approaches for learning these parameters: (1) maximum likelihood estimation and (2) Bayesian inference.
- ▶ Bayesian inference is often more difficult than maximum likelihood estimation (both analytically and computationally), because it requires integration.
- ▶ Bayesian inference readily incorporates interesting functionalities (inclusion of priors, construction of confidence estimates).

图2：实际中的参数估计方法

- 在实际应用中，类条件分布的参数是未知的，必须从有限的数据集中推断出来。
- 学习这些参数的两种主要方法是：(1) 最大似然估计 和 (2) 贝叶斯推断。
- 贝叶斯推断通常比最大似然估计更难（无论在分析上还是计算上），因为它需要进行积分计算。
- 贝叶斯推断可以轻松地包含一些有趣的功能（例如包含先验、构造置信度估计）。