Probability density estimation; Parametric methods

Václav Hlaváč

Czech Technical University in Prague
Czech Institute of Informatics, Robotics and Cybernetics
160 00 Prague 6, Jugoslávských partyzánů 1580/3, Czech Republic
http://people.ciirc.cvut.cz/hlavac, vaclav.hlavac@cvut.cz
also Center for Machine Perception, http://cmp.felk.cvut.cz

Courtesy: V. Franc, R. Gutierrez-Osuna.

Outline of the talk:

- Three taxonomies.
- Maximum likelihood (ML) estimation.
- Examples: Gaussian, discrete distribution.
- ML for pattern rec., a coin example.
- Logistic regression.
- Bayesian estimate, a coin example.

What if the statistical model is unknown?

- Bayesian and non-Bayesian methods allow designing the optimal classifier provided we have a statistical model describing the observation $x \in \mathcal{X}$ and the hidden state $y \in \mathcal{Y}$.
  - Bayesian methods require posterior probability $p(y|x)$.
  - Non-Bayesian methods require the class-conditional probability $p(x|y)$.
- In most practical situations, the statistical model is unknown and must be estimated from the training multiset of examples

$$\big\{(x_1, y_1), \ldots, (x_m, y_m)\big\} \quad \Rightarrow \quad \hat{p}(x, y)\,.$$

- The estimated $\hat{p}(x, y)$ replaces the true distribution $p(x, y)$.

Note: Estimation (in statistics) is also called learning (in pattern recognition and machine learning).

Taxonomy 1: according to the statistical model
Parametric $\times$ nonparametric

- Parametric: A particular form of the density function (e.g., Gaussian) is assumed to be known and only its parameters $\theta \in \Theta$ need to be estimated (e.g., the mean, the covariance).

  Sought: $p(x, \theta)$;

  Performed: $\{x_1, \ldots, x_n\} \rightarrow \hat{\theta}$.

- Nonparametric: No assumption on the form of the distribution. However, many more training examples are required in comparison with parametric methods.

  Sought: $p(x)$;

  Performed: $\{x_1, \ldots, x_n\} \rightarrow \hat{p}(x)$.

Taxonomy 2: according to the input information
Supervised, unsupervised & semi-supervised

The goal is to estimate the probability density $p(x, y)$ or $p(x)$.

The probability density estimation methods can be categorized according to the information available in the training multiset.

- Supervised methods: Completely labeled examples are available.
  Sought: $p(x, y; \theta)$;
  Performed: $\{(x_1, y_1) \ldots, (x_n, y_n)\} \to \hat{\theta}$.
- Unsupervised methods: Unlabeled examples of observations, used, e.g., for clustering.
  Sought: $p(x; \theta)$;
  Performed: $\{x_1, \ldots, x_n\} \to \hat{\theta}$.
- Semi-supervised methods: Both n labeled and $m - n$ unlabeled examples.
  Sought: $p(x, y; \theta)$;
  Performed: $\{(x_1, y_1), \ldots, (x_n, y_n), x_{n+1}, \ldots, x_{n+m}\} \to \hat{\theta}$.

Taxonomy 3: according to the estimation principle

- Maximum-Likelihood estimation:

  Sought: $p(x, y; \theta)$, where θ is fixed but unknown.

  Performed: $\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ p(\text{dataset}|\theta)$

  In words: The ML method seeks the solution, which is the best explanation of the dataset $X \in \mathcal{X}$ using the likelihood function, i.e. the class-conditional probability distribution.

- Bayesian estimation:

  Sought: $p(x, y; \theta)$. θ is a random variable with the known prior $p(\theta)$.

  In words: Bayesian method estimates the optimal parameter Θ of the given probability density, which maximizes the posterior probability distribution $p(\Theta|X)$.

- Minimax estimation: (suggested by M.I. Schlesinger, not explicated here)

  Sought: $p(x, y; \theta)$

  Performed: $\hat{\theta} = \arg\max_{\theta} \min_{i=1,\ldots,m} p(x_i, \theta)$

This lecture: supervised methods

- This lecture covers the supervised methods estimating a known probability distribution, i.e., parametric methods.
- There will be a separate lecture in this course on
  - non-parametric metods

    (as histogram-based; kernel density estimation; Parzen window estimation, k-Nearest Neighbors estimation);
  - unsupervised methods

    (as K-means clustering; EM-algorithm).

Maximum likelihood estimation

Assumptions

- The density function $p(x;\theta)$ is known up to a parameter $\theta \in \Theta$. The column vector of parameters $\theta = [\theta_1, \ldots, \theta_d]^\top \in \mathbb{R}^d$.
- i.i.d. assumption: A measured data (set of examples) $\mathcal{D}=\{x_1, \ldots, x_n\}$ independently drawn from the identical distribution $p(x;\theta^*)$ is available.
- The true parameter $\theta^* \in \Theta$ is unknown but it is fixed.

The probability $p(x;\theta)$ assumes that n examples $\mathcal{D}$ were generated (measured) for a given Θ is called the likelihood function

$$p(\mathcal{D};\theta) = \prod_{i=1}^{n} p(x_i;\theta) \; .$$

Maximum likelihood estimation $\hat{\theta}_{\mathrm{ML}} = \operatorname*{argmax}_{\theta\in\Theta} p(\mathcal{D};\theta)$ seeks for the parameter $\hat{\theta}_{\mathrm{ML}}$, which best explains the examples $\mathcal{D}$.

Log-likelihood function $L(\theta)$

- It is convenient to maximize the log-likelihood function

$$L(\theta) = \log p(\mathcal{D}; \theta) = \log \prod_{i=1}^{n} p(x_i; \theta) = \sum_{i=1}^{n} \log p(x_i; \theta)$$

- Because the logarithm is monotonically increasing, it holds

$$\hat{\theta}_{\mathrm{ML}} = \operatorname*{argmax}_{\theta \in \Theta} p(\mathcal{D}; \theta) = \operatorname*{argmax}_{\theta \in \Theta} \underbrace{\sum_{i=1}^{n} \log\ p(x_i; \theta)}_{L(\theta)} = \operatorname*{argmax}_{\theta \in \Theta} L(\theta)$$

- Maximizing a sum of terms is always an easier task than maximizing a product; cf., the difficulty of expressing derivative of a long product of terms. In particular, the logarithm of a Gaussian is very easy.

ML estimate of the log-likelihood function

The (maximally likely) ML estimate can be often obtained by finding the stationary point of the log-likelihood function $\frac{\partial L(\theta)}{\partial \theta_j} = \nabla_\theta L = 0$.

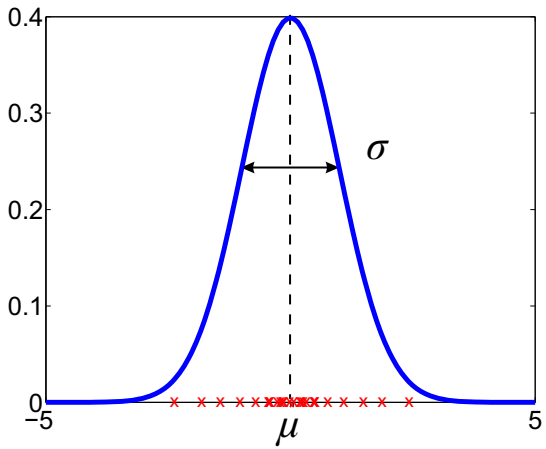
$$\begin{aligned}
\frac{\partial L(\theta)}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \sum_{i=1}^{n} \log\, p(x_i;\theta) \\
&= \sum_{i=1}^{n} \frac{\partial}{\partial \theta_j} \log\, p(x_i;\theta) \\
&= \sum_{i=1}^{n} \frac{1}{p(x_i;\theta)} \frac{\partial\, p(x_i;\theta)}{\partial \theta_j} = 0
\end{aligned}$$

Provided $L(\theta)$ is convex, the found solution is the ML estimate $\hat{\theta}_{ML}$.

Example: Normal distribution, ML estimate (1)

Input

- $\mathcal{D} = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^n$
- $p(x; \theta = (\mu, \sigma)) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma}\right)$
- $\theta = \{(\mu, \sigma) \mid \mu \in \mathbb{R},\, \sigma \in \mathbb{R}^+\}$.



Likelihood function $p(\mathcal{D}; \theta)$ and its logarithm, i.e. log-likelihood function $L(\theta)$

$$p(\mathcal{D}; \theta) = \prod_{i=1}^{n} p(x_i; \theta) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma}\right)$$

$$L(\theta) = \log p(\mathcal{D}; \theta) = \sum_{i=1}^{n} \left(-\log\sigma - \log\sqrt{2\pi} - \frac{(x_i-\mu)^2}{2\sigma^2}\right)$$

Example: Normal distribution, ML estimate (2)

To compute $\hat{\theta}_{\mathrm{ML}} = \operatorname{argmax}_\theta L(\theta)$, we will solve $\nabla_\theta L = \frac{\partial L(\theta)}{\partial \theta} = 0$.

$$L(\theta) = \log p(\mathcal{D};\theta) = \sum_{i=1}^{n} \left(-\log\sigma - \log\sqrt{2\pi} - \frac{(x_i-\mu)^2}{2\sigma^2} \right)$$

- Mean value $\hat{\mu}_{\mathrm{ML}}$:

$$\frac{\partial L}{\partial \mu} = \sum_{i=1}^{n} \left(\frac{-2(x_i-\mu)}{2\sigma^2} \right) = 0 \quad \Rightarrow \quad \hat{\mu}_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

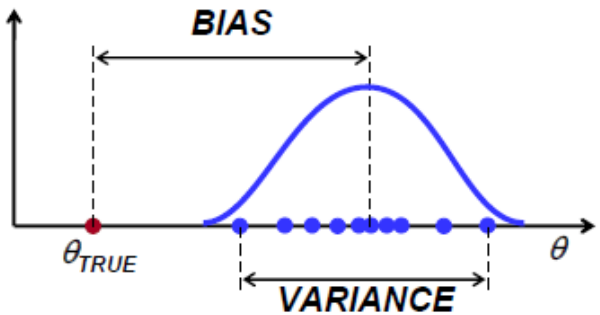
- Variance $\hat{\sigma}^2_{\mathrm{ML}}$:

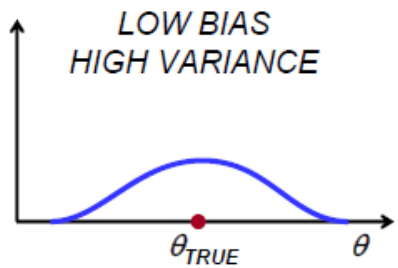
$$\frac{\partial L}{\partial \sigma} = \sum_{i=1}^{n} \left(\frac{-1}{\sigma} - \frac{(x_i-\mu)^2}{2} \left(\frac{-2}{\sigma^3} \right) \right) = 0 \quad \Big| \cdot \sigma^3$$

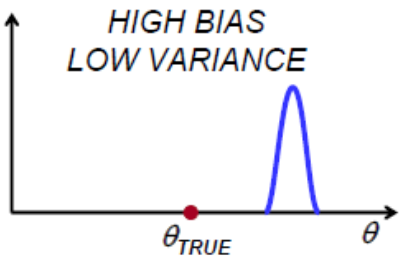
$$\sum_{i=1}^{n} \left(-\sigma^2 + (x_i-\mu)^2 \right) = 0 \quad \Rightarrow \quad \hat{\sigma}^2_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n} (x_i-\mu)^2$$

Properties of the estimates (1)

- $\hat{\theta}$ is a random variable. Consequently, we can talk about its mean value (the most probable value) and its variance.
- Bias – Tells how close is the estimate to the true value.
  Unbiased estimate: $E\left(\hat{\theta}\right) = \theta$.
- Variance – How much does the estimate change for different runs, e.g. for different datasets?
- Bias-variance tradeoff: In most cases, one can only decrease either the bias or the variance at the expense of the other.



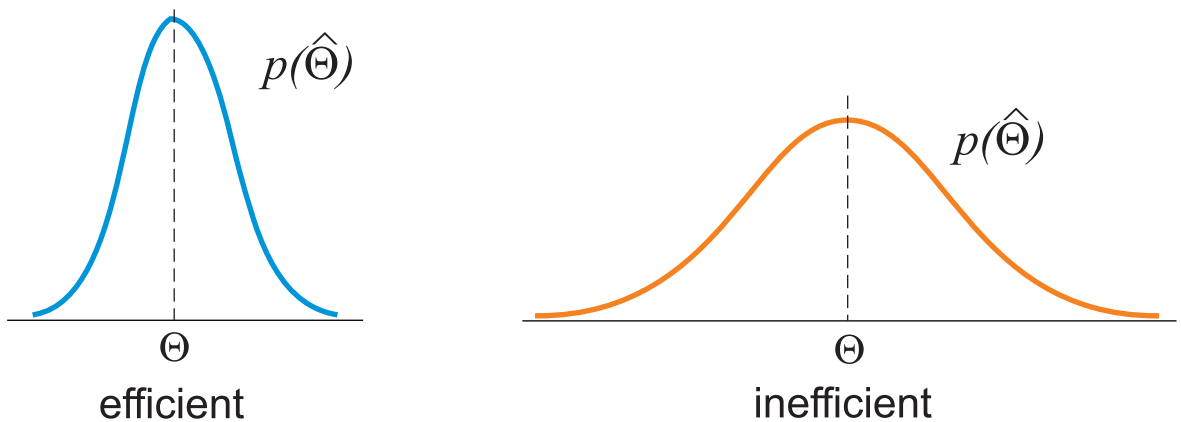




Properties of the estimates (2)

- Efficiency of the estimate: $\mathrm{var}\left(\hat{\theta}\right) = E\left(\hat{\theta} - \theta\right)^2$.

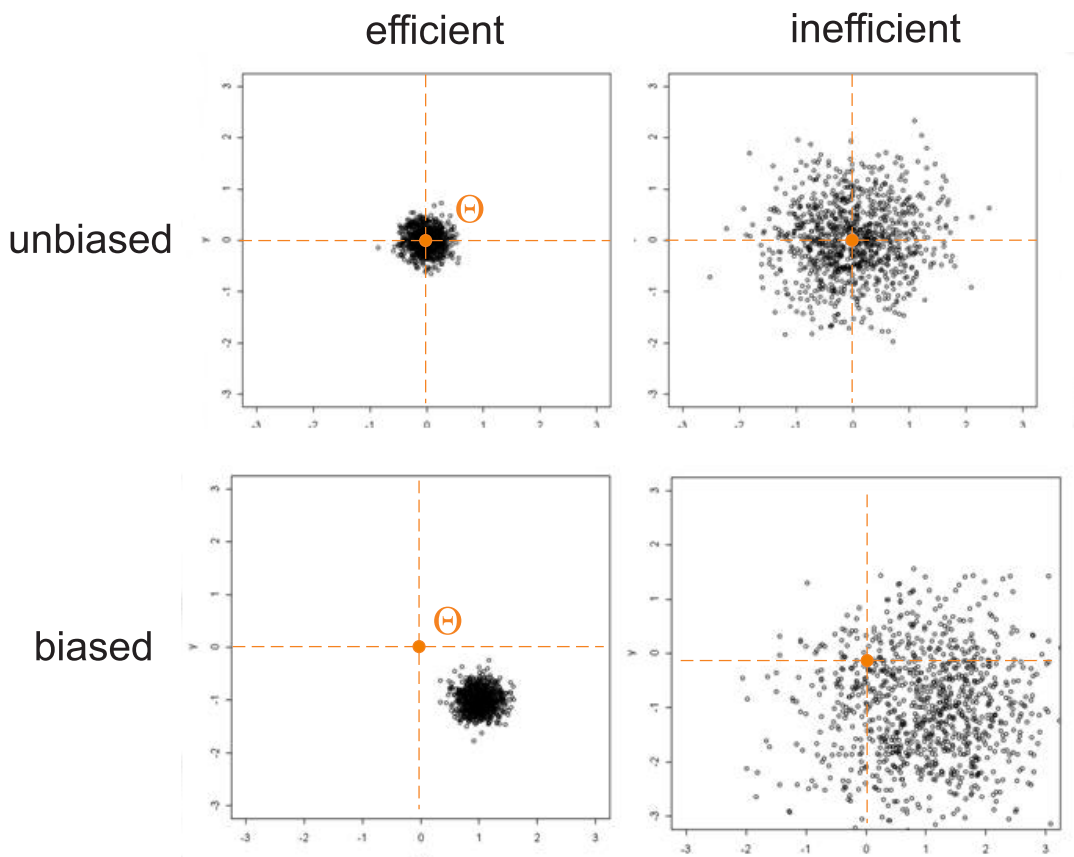
  The best estimate has the minimal dispersion.



- Consistent estimate $E\left(\hat{\theta}\right) = \theta$ for $n \to \infty$; $\mathrm{var}\left(\hat{\theta}\right) = 0$ for $n \to \infty$.

  The growing amount of samples induces that statistical characteristics converge to the true value. (The consequence of the rule of large numbers.)

2D illustration of the estimate bias and efficiency



Compare with the terminology used in engineering measurements: precision, repeatibility.

ML estimate of the m-variate Gaussian

- One data sample $\mathbf{x}_i \in \mathbf{X} = (x_1, x_2, \ldots, x_m)^\top$.
- It can be derived similarly that the Maximum Likelihood parameter estimate yields the sample mean vector and the sample covariance matrix.
- Sample mean vector

$$\hat{\mu}_{\mathrm{ML}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$$

- Sample covariance matrix

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \hat{\mu})(\mathbf{x}_i - \hat{\mu})^\top$$

Example: Discrete distribution, ML estimate (1)

- $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$
- Discrete range (quantization levels)
  $x \in X = \{1, 2, \ldots, m\}$

$$p(x;\theta) = \begin{bmatrix} p(x=1) = \theta_1 \\ p(x=2) = \theta_2 \\ \vdots \\ p(x=m) = \theta_m \end{bmatrix}$$

Likelihood function; δ is Dirac function

$$p(x_1, \ldots, x_n; \theta) = \prod_{i=1}^{n} p(x_i) = \prod_{i=1}^{n} \sum_{x \in X} p(x) \cdot \delta(x_i = x) = \prod_{x \in X} p(x)^{n(x)}$$

Logarithm of the likelihood function

$$L(\theta) = \log p(x_1, \ldots, x_n; \theta) = \sum_{x \in X} n(x) \log p(x)$$

Example: Discrete distribution, ML estimate (2)

Logarithm of the likelihood function $L(\theta) = \sum_{x \in X} n(x) \log p(x)$ is minimized

under the condition $\sum_{x \in X} p(x) = 1\,, \quad p(x) \geq 0$

Lagrange multipliers method is used

$$F(\theta) = \sum_{x \in X} n(x) \log p(x) + \lambda \left(\sum_{x \in X} p(x) - 1 \right)$$

$$\frac{\partial F(\theta)}{\partial p(x)} = \frac{n(x)}{p(x)} + \lambda = 0\,, \quad \text{for } \forall x$$

Example: Discrete distribution, ML estimate (3)

Rewritten from the previous slide:

$$\frac{\partial F(\theta)}{\partial p(x)} = \frac{n(x)}{p(x)} + \lambda = 0\,, \text{ for } \forall x$$

$$n(x) = -\lambda\, p(x) \quad \Rightarrow \quad p(x) = \frac{n(x)}{-\lambda}$$

It follows from the constraint:

$$\sum_{x \in X} p(x) = 1 = \sum_{x \in X} \frac{n(x)}{-\lambda} = \frac{1}{-\lambda} \sum_{x \in X} n(x) = \frac{n}{-\lambda}$$

$$\lambda = -n$$

$$p(x) = \frac{-n(x)}{\lambda} = \frac{n(x)}{n}$$

Maximum likelihood estimate if there is no analytical solution

If the analytical solution to $\hat{\theta}_{\mathrm{ML}} = \mathrm{argmax}_\theta \, L(\theta)$ does not exist then the ML estimation task can be solved numerically:

1. Using standard optimization techniques like gradient methods or Newton methods.
2. Using the Expectation-Maximization algorithm when

$$p(x;\theta) = \sum_{h \in \mathcal{H}} p(x,h;\theta)$$

For example, in the case of the mixture models

$$p(x;\theta) = \sum_{h=1}^{H} p(x|h;\theta)\, p(h;\theta)$$

Properties of the ML estimate

Pros:

- ML has favorable statistical properties. It is asymptotically (i.e., for $n \to \infty$) unbiased, it has the smallest dispersion, and it is consistent.
- ML leads to simple analytical solution for many "simple distributions" as was demonstrated in the Gaussian example.

Cons:

- There can be several solutions with the same quality.
- There is a single global solution (the value of the quality). However, it is difficult to find it.
- There is no analytical solution for the mixture of Gaussians. In this case, EM algorithm is used often *(and mentioned in this lecture later)*.

Note: The ML estimate has often mentioned properties. In general, however, these properties cannot be guaranteed because the estimate values need not be real numbers. Thus we cannot claim what their distribution was.

ML estimation in pattern recognition (1)

- $\mathcal{D} = \{(x_1, y_1), (x_2, y_2) \ldots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$
- $x \in \mathcal{X}$ ... discrete (a finite set) or continuous (an infinite set), typically $\mathcal{X} \in R^m$
- $y \in \mathcal{Y}$ ... discrete (a finite set)
- $p(x, y; \theta) = p(x|y; \theta_y)\, p(y; \theta_A)$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{|\mathcal{Y}|} \\ \theta_A \end{bmatrix}$$

The likelihood function $p(\mathcal{D}; \theta)$ reads

$$p\left(\mathcal{D}; \theta\right) = \prod_{j=1}^{n} p(x_j, y_j; \theta_y) = \prod_{j=1}^{n} p(x_j|y_j; \theta_{y_j})\, p(y_j; \theta_A)$$

ML estimation in pattern recognition (2)

$$L(\theta) = \log p\left(\mathcal{D};\theta\right) = \sum_{j=1}^{n} \log p(x_j|y_j;\theta_{y_j}) + \sum_{j=1}^{n} \log p(y_j;\theta_A)$$

$$= \sum_{y\in\mathcal{Y}} \sum_{j\in\{i|y_i=y\}} \log p(x_j|y;\theta_y) + \sum_{j=1}^{n} \log p(y_j;\theta_A)$$

The original maximization task decomposes into $|\mathcal{Y}|+1$ simpler independent tasks:

$$\hat{\theta}_y = \operatorname*{argmax}_{\theta_y} \sum_{j\in\{i|y_i=y\}} \log p(x_j|y;\theta_y)\,, \qquad \forall y \in \mathcal{Y}$$

$$\hat{\theta}_A = \operatorname*{argmax}_{\theta_A} \sum_{j=1}^{n} p(y_j|\theta_A)$$

Example: Is the coin fair? (ML case 1)

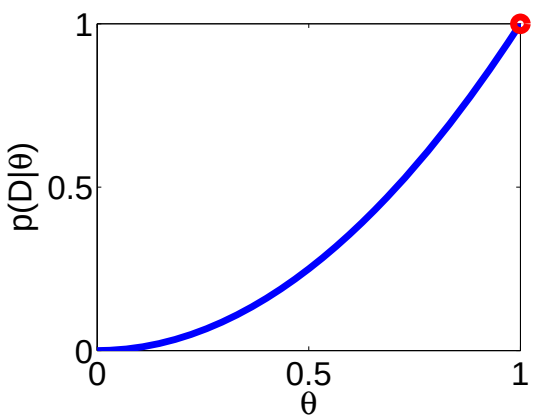
- Assume a coin flipping experiment with the training set $\mathcal{D} = \{\text{head}, \text{head}, \text{tail}, \text{head}, \ldots, \text{tail}\}$.
- Let n be the length of the experiment (number of tosses) and k the number of heads observed in experiments.
- The probabilistic model of the unknown coin is: $p(x = \text{head}|\theta) = \theta$ and $p(x = \text{tail}|\theta) = 1 - \theta$.
- The likelihood of $\theta \in \Theta = [0, 1]$ is given by Bernoulli distribution,

$$p(\mathcal{D}|\theta) = \prod_{j=1}^{n} p(x_j|\theta) = \theta^k (1-\theta)^{n-k}$$

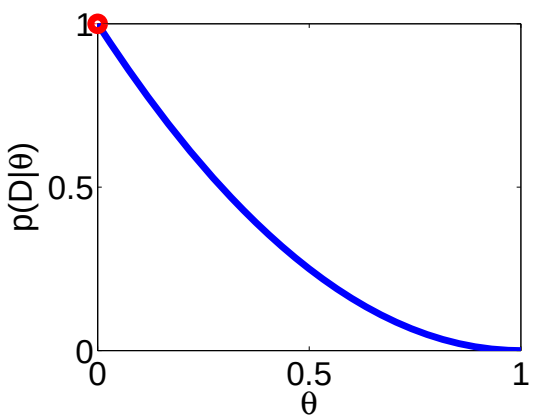
Example: Is the coin fair? (ML case 2)

- Repeated from previous slide: $p(\mathcal{D}|\theta) = \prod_{j=1}^{n} p(x_j|\theta) = \theta^k(1-\theta)^{n-k}$
- ML estimate $\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ p(\mathcal{D}|\theta) = \frac{k}{n}$.
- The optimal value is illustrated as the red dot.

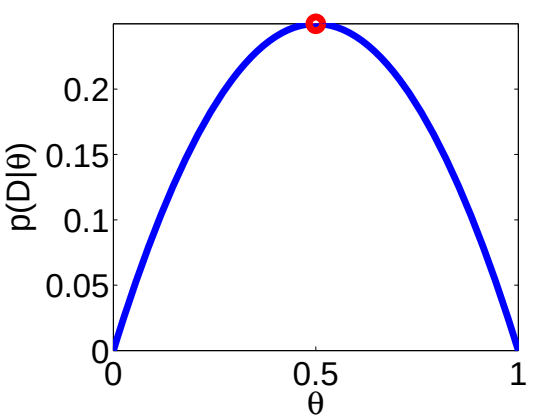
Example: What if only $m = 2$ samples are available?



$p\left(\begin{smallmatrix} m=2 \\ k=2 \end{smallmatrix} \mid \theta\right) = \theta^2$



$p\left(\begin{smallmatrix} m=2 \\ k=0 \end{smallmatrix} \mid \theta\right) = (1-\theta)^2$



$p\left(\begin{smallmatrix} m=2 \\ k=1 \end{smallmatrix} \mid \theta\right) = \theta(1-\theta)$

Use of ML in unsupervised learning (1)

- $\mathcal{D}_X = \{x_1, x_2, \ldots, x_n\}$ ... only observable states.
- $p(x, y; \theta) = p(x|y; \theta_y)\, p(y, \theta_A)$ ... the statistical model considers an unknown hidden state y.

$$p(\mathcal{D}_x; \theta) = \prod_{j=1}^{n} p(x_j; \theta) = \prod_{j=1}^{n} \sum_{y \in Y} p(x_j, y; \theta) = \prod_{j=1}^{n} \sum_{y \in Y} p(x_j|y; \theta)\, p(y; \theta_A)$$

$$\begin{aligned} L(\theta) &= \log p(\mathcal{D}; \theta) = \sum_{j=1}^{n} \log \sum_{y \in Y} p(x_j|y; \theta)\, p(y; \theta_A) \\ &= \sum_{j=1}^{n} \log \left(\sum_{y \in Y} p(x_j|y; \theta)\, p(y; \theta_A) \right) \end{aligned}$$

Use of ML in unsupervised learning (2)

The quality of the solution is copied from the previous page,

$$L(\theta) = \sum_{j=1}^{n} \log \left(\sum_{y \in Y} p(x_j|y;\theta)\, p(y;\theta_A) \right) \tag{1}$$

$\nabla_\theta L(\theta) = 0$ does not have the analytical solution!

Expectation-Maximization (EM) algorithm

- is often used for this unsupervised learning task. It is an iterative method, which can be used to find the ML estimate of Equation (1).
- It can be used if we are able to perform ML estimate from $\mathcal{D}_X$.
- One lecture is dedicated to EM algorithm later in this course.

Logistic regression (1)

- Given examples $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in (\mathbb{R}^n \times \{+1, -1\})^n$. Assuming the examples were i.i.d. drawn.
- The goal is to find the posterior model $p(y|x)$ needed for the classification.
- Let assume that the logarithm of the likelihood ratio is a linear function

$$\log \frac{p(y = +1|x)}{p(y = -1|x)} = \langle x, w \rangle + b$$

- Under this assumption, we get that the $p(y = 1|x)$ is the logical function (i.e., yielding the value TRUE or FALSE)

$$p(y|x; w) = \frac{1}{1 + \exp\big(-y(\langle x, w \rangle + b)\big)}$$

- The likelihood of the parameters $\theta = (w, b)$ given the examples $\mathcal{D}$ reads

$$p(\mathcal{D}; w, b) = \prod_{i=1}^{n} p(x_i, y_i; w, b) = \prod_{i=1}^{n} p(y_i|x_i; w, b)\ p(x_i)$$

Logistic regression (2)

- Bayes classifier does not require the model of $p(x)$ and hence

$$(\hat{w}, \hat{b}) = \operatorname*{argmax}_{w,b} \prod_{i=1}^{n} p(y_i|x_i; w, b)\; p(x) =$$

$$= \operatorname*{argmax}_{w,b} \left(\sum_{i=1}^{n} \log p(y_i|x_i; w, b) + \sum_{i=1}^{n} \log p(x_i) \right) = \operatorname*{argmax}_{w,b} \sum_{i=1}^{n} \log p(y_i|x_i; w, b)$$

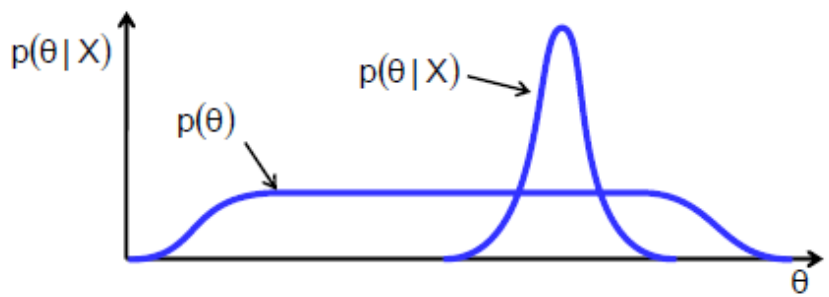
- The maximization task is concave in variables w, b. However, the task does not have the analytical solution.
- Bayes classifier minimizing the classification error leads to the linear rule

$$\log \frac{p(y=+1|x)}{p(y=-1|x)} = \langle x, w \rangle + b \qquad q(x) = \begin{cases} +1 & \text{if} \quad \langle x, w \rangle + b \geq 0 \\ -1 & \text{if} \quad \langle x, w \rangle + b < 0 \end{cases}$$

Bayesian estimate, the introduction

Our uncertainty about the parameters Θ is represented by the probability distribution in the Bayesian approach.

- Before we observe the data, the parameters are described by a prior density $p(\Theta)$, which is typically very broad reflecting the fact that we know little about its true value.
- Once we obtain data, we use the Bayes theorem to find the posterior $p(\Theta|X)$. Ideally, we want the data to sharpen the posterior $p(\Theta|X)$. Said differently, we want to reduce our uncertainty about the parameters.



- However, it has to be kept in mind that our goal is to estimate the density $p(x)$ or, more precisely, the density $p(x|X)$, the density given the evidence provided by the dataset X.

Bayesian estimate of the probability density parameters (1)

- $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$ ... observed data.
- $p(\mathcal{D}, \theta)$ ... joint probability distribution of $\mathcal{D}$ and θ.
- $\theta \in \Theta$ ... is understood as a realization of the random variable (in the contrast to ML estimate) with the known prior distribution $p(x|\theta)$.
- The likelihood function $p(\mathcal{D}|\theta) = \sum_{j=1}^{n} p(x_j|\theta)$.
- Bayesian estimation seeks the posterior distribution of the parameter θ best explaining the observed data $\mathcal{D}$. It is obtained by Bayes formula: $p(\mathcal{D}|\theta) = \prod_{j=1}^{n} p(x_j|\theta)$.

Use of Bayesian estimation

What is the posterior distribution $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)\ p(\theta)}{p(\mathcal{D})}$ good for?

1. Marginalize over the parameter θ (numerically expensive)

$$p(x|\mathcal{D}) = \int_{\theta\in\Theta} p(x,\theta|\mathcal{D})\, d\theta = \int_{\theta\in\Theta} p(x|\theta)\ p(\theta|\mathcal{D})\, d\theta$$

2. Compute the point estimate by minimizing the expected risk

$$\hat{\theta} = \operatorname*{argmin}_{\theta'} \int_{\theta\in\Theta} W(\theta',\theta)\, p(\theta,\mathcal{D})\, d\theta\,,$$

where $W\colon \Theta\times\Theta \to \mathbb{R}$ is a penalty function penalizing incorrectly estimated parameters.

Bayesian estimation: Examples of loss functions

- Zero-one penalty function

$$W(\theta', \theta) = \begin{cases} 0 & \text{if} \quad \theta' = \theta \\ 1 & \text{if} \quad \theta' \neq \theta \end{cases}$$

leads to the Maximum Aposteriori (MAP) estimate $\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(\theta|\mathcal{D})$.

Note: ML estimate $\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(\mathcal{D}|\theta)$ *can be seen as the MAP estimate with an uninformative prior* $p(\theta)$.

- Quadratic penalty function $W(\theta', \theta) = (\theta' - \theta)^2$ leads to the estimate of the expected value of the parameter

$$\hat{\theta} = \int_{\theta \in \Theta} \theta \; p(\theta|\mathcal{D}) \; d\theta$$

Bayesian estimate of the probability density parameters (2)

- The estimate $\hat{\theta} = \Psi(\mathcal{D})$ is formulated as in Bayesian decision with the penalty function $W : \Theta \times \Theta \to \mathbb{R}$.
- Bayesian risk

$$R(\Psi) = \sum_{\mathcal{D}} \sum_{\theta} p(\mathcal{D}, \theta)\, W(\theta, \Psi(\mathcal{D}))$$

- The optimal decision function (strategy, rule)

$$\Psi^*(\mathcal{D}) = \operatorname*{argmin}_{\Psi} R(\Psi) = \operatorname*{argmin}_{\psi} \sum_{\theta} p(\mathcal{D}, \theta)\, W(\theta, \Psi(\mathcal{D}))$$

Bayesian estimate for the $W(\theta, \Psi(\mathcal{D})) = (\theta - \psi(\mathcal{D}))^2$

After the substitution $\Psi^*(\mathcal{D}) = \underset{\Psi}{\operatorname{argmin}} \sum_\theta p(\mathcal{D}, \theta)\,(\theta - \Psi)^2$, it must hold for the minimal loss

$$\frac{\partial}{\partial \Psi}\left(\sum_\theta p(\mathcal{D}, \theta)\,(\theta - \Psi)^2\right) = 0$$

$$\Psi^* \sum_\theta p(\mathcal{D}, \theta) = \sum_\theta \theta\, p(\mathcal{D}, \theta)$$

$$\Psi^*(\mathcal{D}) = \frac{\sum_\theta \theta\, p(\mathcal{D}, \theta)}{p(\mathcal{D})} = \frac{\sum_\theta \theta\, p(\theta|\mathcal{D})\, p(\mathcal{D})}{p(\mathcal{D})} = \frac{\cancel{p(\mathcal{D})} \sum_\theta \theta\, p(\theta|\mathcal{D})}{\cancel{p(\mathcal{D})}}$$

$$\Psi^*(\mathcal{D}) = \sum_\theta \theta\, p(\theta|\mathcal{D})$$

Example: Is the coin fair? (Bayesian case 1)

- *(The same slide as in the ML case. The slide No. 23 is repeated here.)*
- Assume a coin flipping experiment with the training set $\mathcal{D} = \{\text{head}, \text{head}, \text{tail}, \text{head}, \ldots, \text{tail}\}$.
- Let n be the length of the experiment (number of tosses) and k the number of heads observed in experiments.
- The probabilistic model of the unknown coin is: $p(x = \text{head}|\theta) = \theta$ and $p(x = \text{tail}|\theta) = 1 - \theta$.
- The likelihood of $\theta \in \Theta = [0, 1]$ is given by Bernoulli distribution,

$$p(\mathcal{D}|\theta) = \prod_{j=1}^{n} p(x_j|\theta) = \theta^k (1-\theta)^{n-k}$$

Example: Is the coin fair? (Bayesian case 2)

- Assume the uniform prior $p(\Theta) = \begin{cases} 1 & \text{for } \theta \in [0,1], \\ 0 & \text{otherwise.} \end{cases}$

$$\hat{\theta}_{\mathrm{B}} = p(x = \text{head}|\mathcal{D}) \quad = \quad \Psi^*(\mathcal{D}) = \frac{\int_0^1 \theta\, p(\mathcal{D}|\theta)\, p(\theta)\, \mathrm{d}\theta}{p(\mathcal{D})}$$

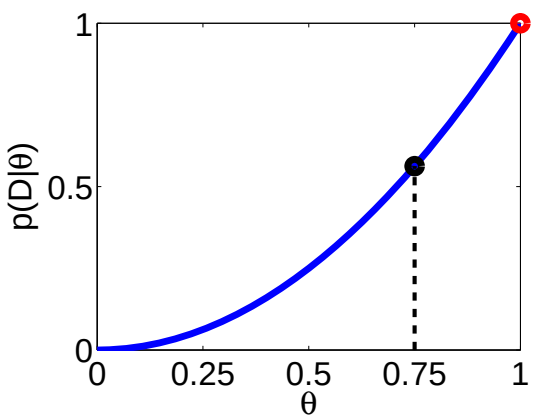
$$= \quad \frac{\int_0^1 \theta^{k+1}(1-\theta)^{n-k}\, \mathrm{d}\theta}{\int_0^1 \theta^{k}(1-\theta)^{n-k}\, \mathrm{d}\theta} = \frac{k+1}{n+2}\,.$$

- Note 1: $\hat{\theta}_{\mathrm{ML}} = \frac{k}{n}$.
- Note 2: For $n \to \infty$, it also holds $\hat{\Theta}_{\mathrm{B}} \to \hat{\Theta}_{\mathrm{ML}}$.

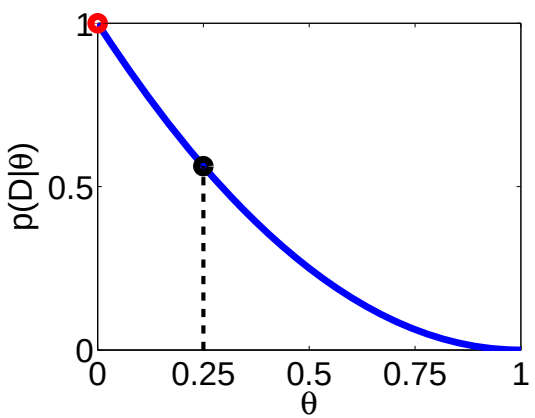
Example: Is the coin fair? (Bayesian case 3)

ML estimate $\hat{\theta}_{ML} = \frac{k}{n}$ (red dot) versus Bayesian estimate $\hat{\theta}_{\mathrm{B}} = \frac{k+1}{n+2}$ (black dot).

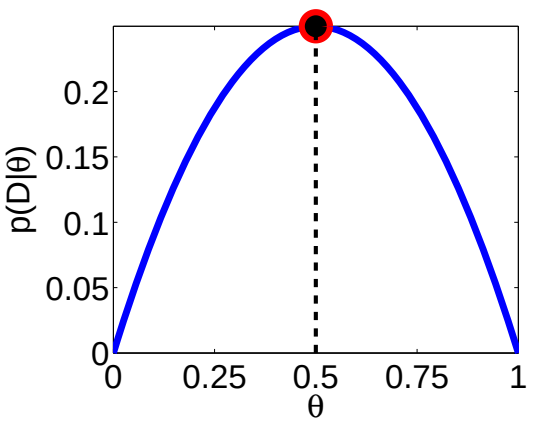
Example: What if only $m = 2$ samples are available?



$$p\left(\begin{smallmatrix} m=2 \\ k=2 \end{smallmatrix} \mid \theta\right) = \theta^2$$



$$p\left(\begin{smallmatrix} m=2 \\ k=0 \end{smallmatrix} \mid \theta\right) = (1-\theta)^2$$



$$p\left(\begin{smallmatrix} m=2 \\ k=1 \end{smallmatrix} \mid \theta\right) = \theta(1-\theta)$$

Relationship between the Maximal likelihood and Bayesian estimation

- By definition, the parameter-conditional probability $p(X|\Theta)$ peaks at the ML estimate. If this peak is relatively sharp and the prior is broad then the outcome will be dominated by the region around the ML estimate.

  Thus, the Bayesian estimate will approximate the ML solution.

- When the number of available data increases, the posterior $p(\Theta|X)$ tends to sharpen.
  - Therefore, the Bayesian estimate of $p(x)$ will approach the ML estimate for number of samples $N \to \infty$.
  - In practice, when we have a limited number of observations, the two approaches will yield different results.