Bayesian Machine Learning. Introduction and Main Tools

Evgeny Burnaev

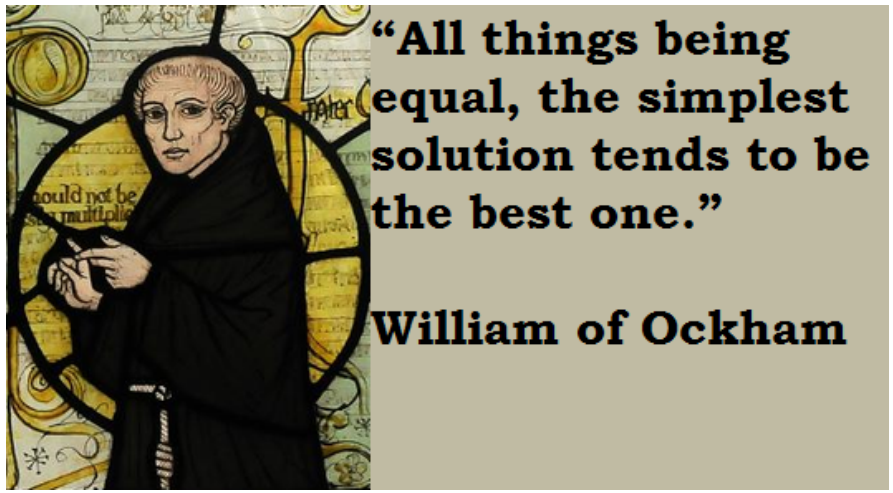
Skoltech, Moscow, Russia

Outline

Main Principles

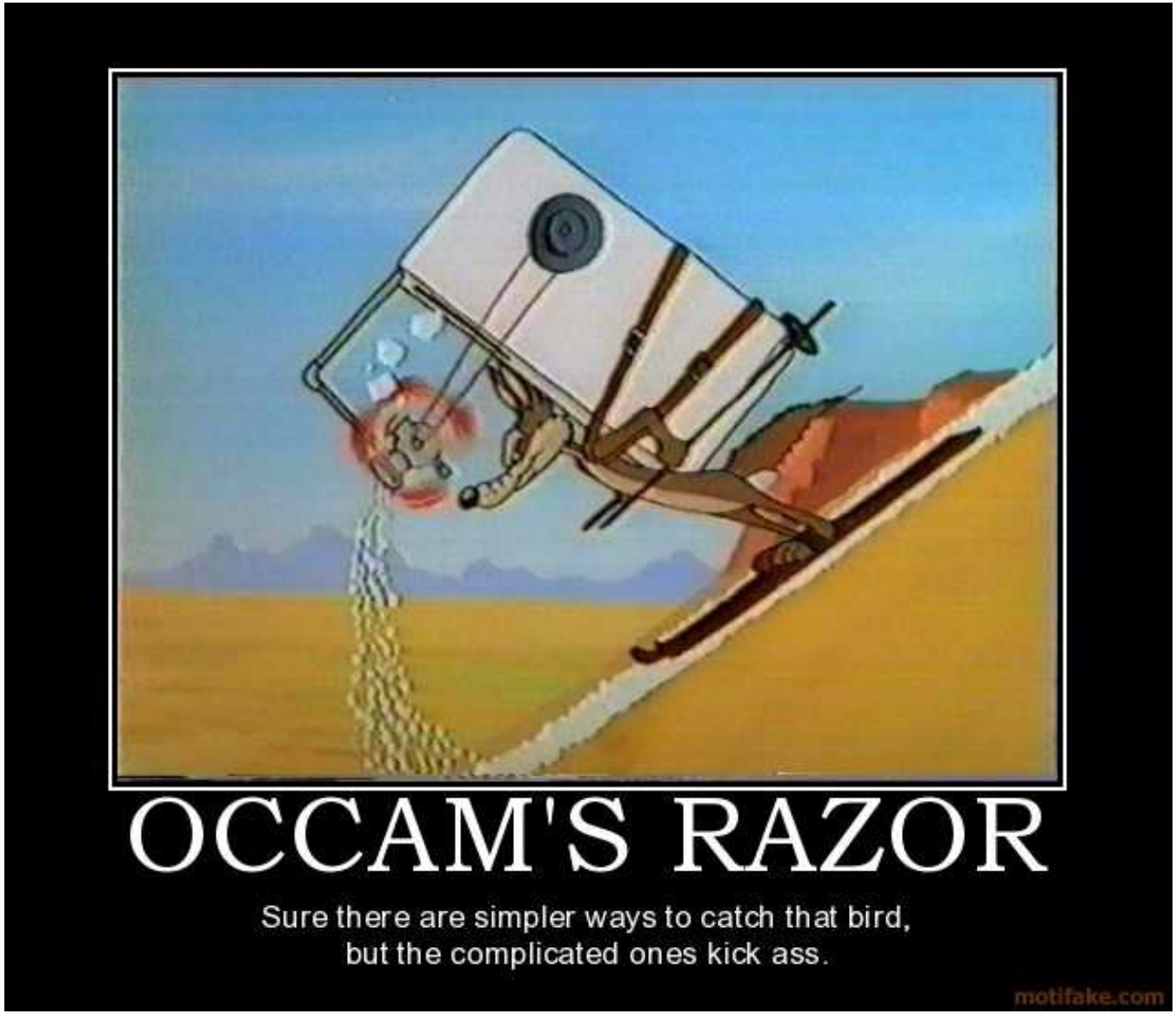
Thomas Bayes (c. 1701 – 7 April 1761) was an English statistician, philosopher and Presbyterian minister

$$\mathbb{P}(B|A) = \frac{\mathbb{P}(A|B)\mathbb{P}(B)}{\mathbb{P}(A)}$$



William of Ockham (c. 1287 – 1347) was an English Franciscan friar and scholastic philosopher and theologian

Occam's razor



Example: Polynomial Curve Fitting

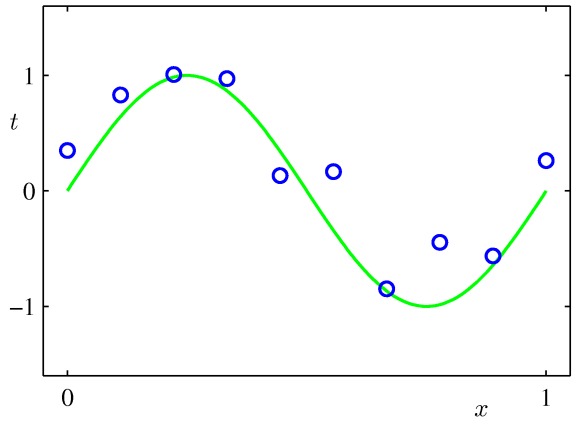


Figure : Plot of a training data

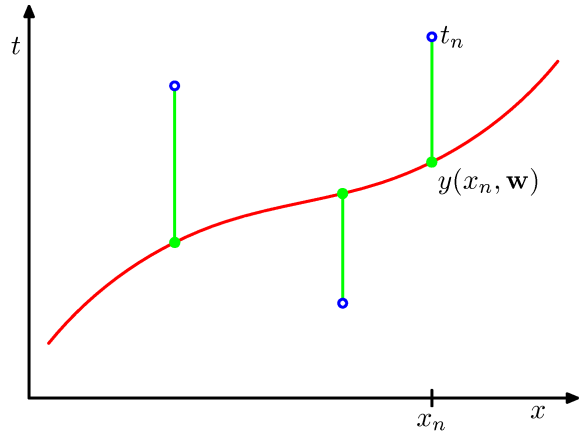


Figure : Residuals

- $S_n = \{\mathbf{X}_n, \mathbf{Y}_n\} = \{(x_i, y_i)\}_{i=1}^n$, where
  $y_i = \sin(2\pi x_i) + \varepsilon_i$, ε_i is a Gaussian white noise

Example: Polynomial Curve Fitting

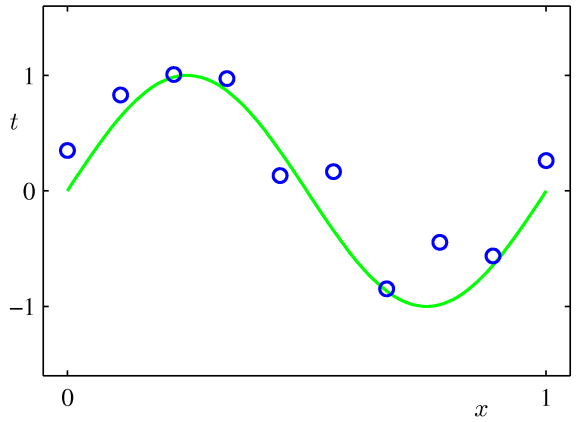


Figure : Plot of a training data

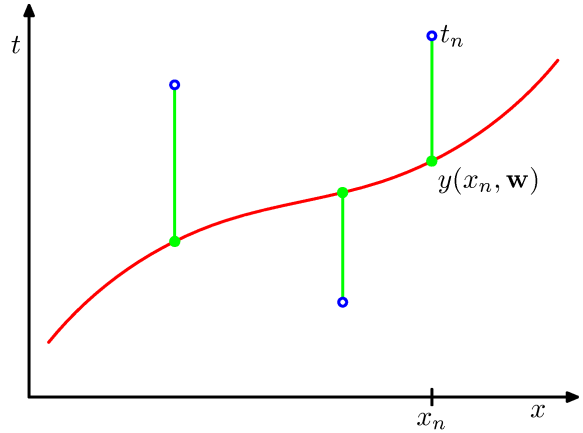


Figure : Residuals

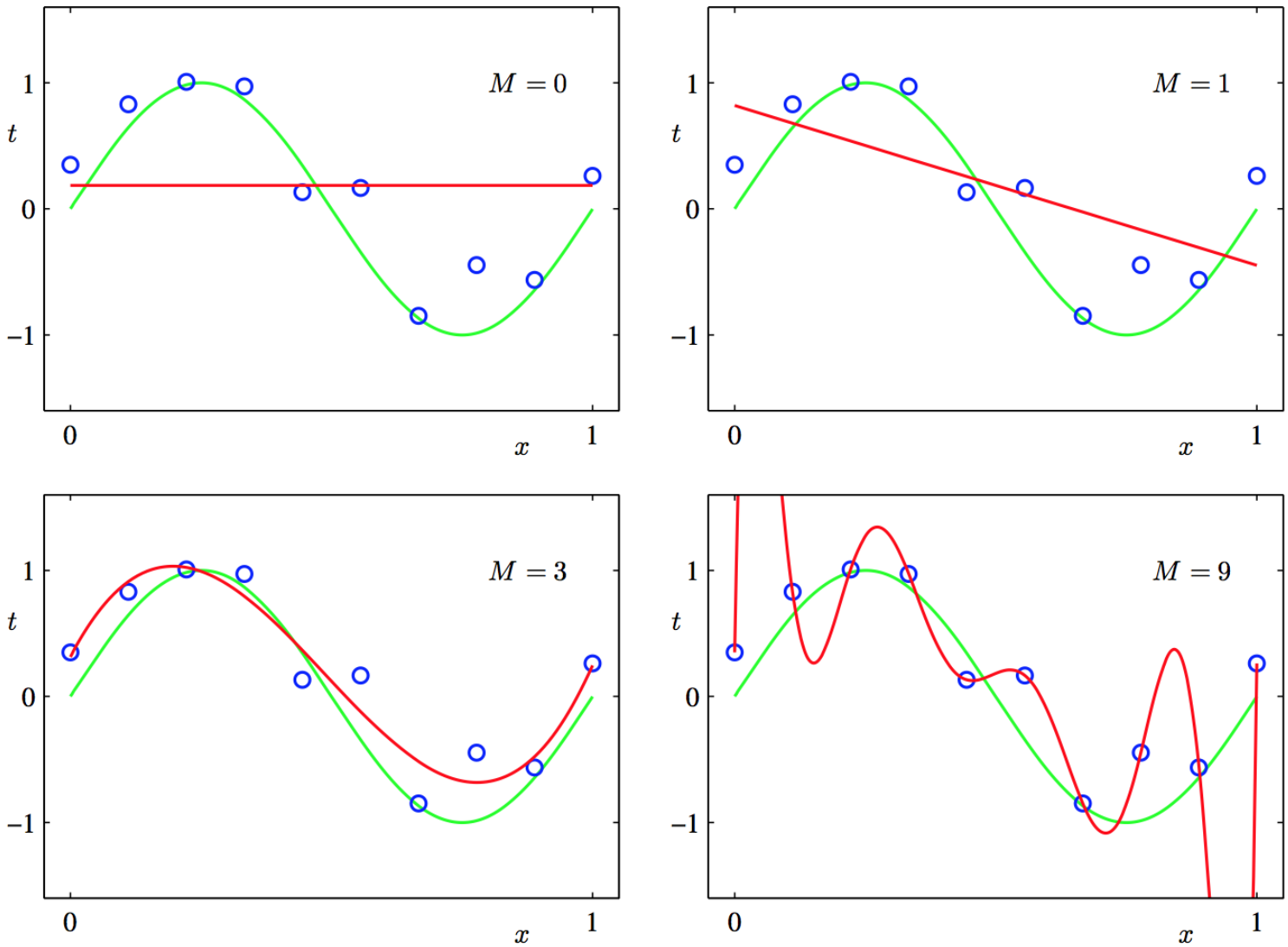
- We fit a model

$$y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j x^j,$$

by minimizing the error

$$E(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{n} \{y(x_i, \mathbf{w}) - y_i\}^2$$

Plots of polynomials having various orders M



Overfitting

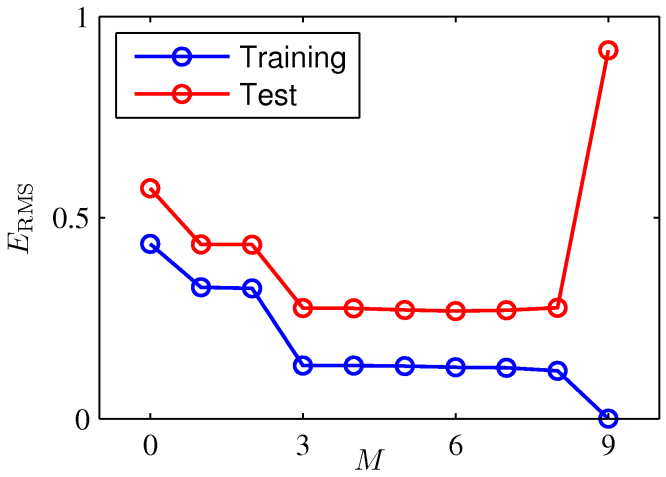


Figure : $E_{RMS} = \sqrt{2E(\mathbf{w}^*)/n}$

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Figure : Coefficients w^*

Overfitting vs. Sample size

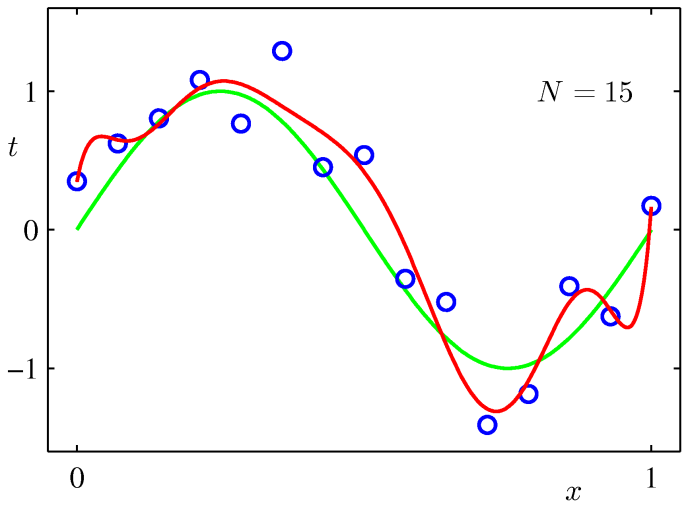


Figure : $M = 9, n = 15$

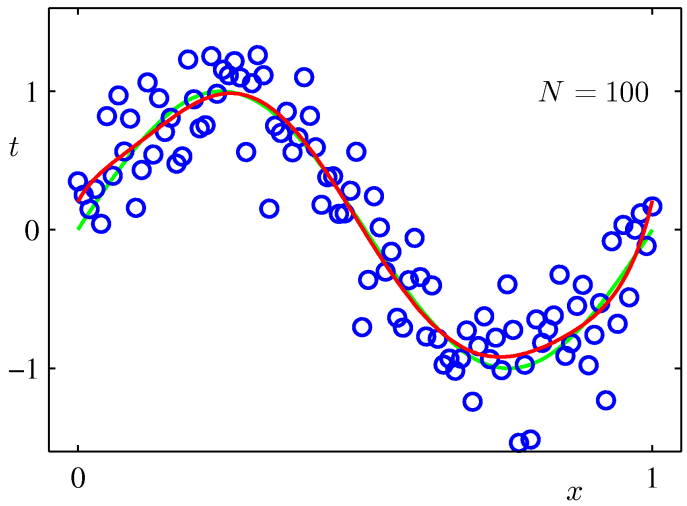


Figure : $M = 9, n = 100$

Overfitting vs. Regularization

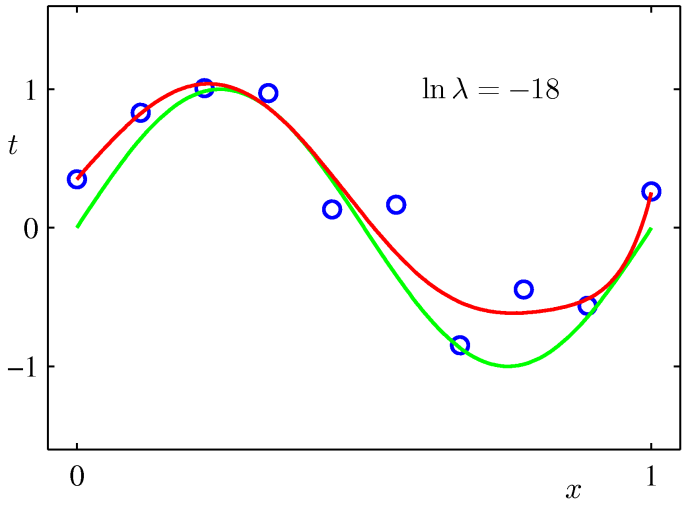


Figure : $\lambda = e^{-18} \approx 0$

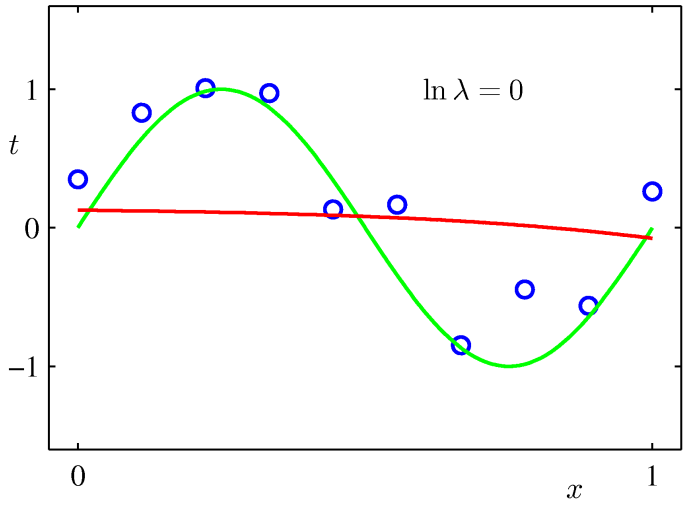


Figure : $\lambda = 1$

- Limit the number of parameters M w.r.t. the size of the available training set?
- Instead choose the complexity of the model (effective model parameters) according to the complexity of the problem!

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}\{y(x_i, \mathbf{w}) - y_i\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

Overfitting vs. Regularization

	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Figure : Dependence of w^* on λ

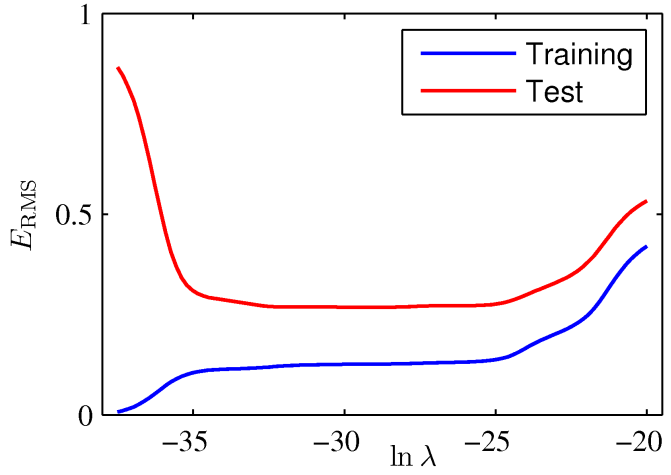


Figure : Dependence of E_{RMS} on λ

- We would have to find a way to determine a suitable value for the model complexity!
- Hold-out set to select a model complexity (either M or λ)? Too wasteful $\Rightarrow$ Bayesian Learning!

Probability Rules

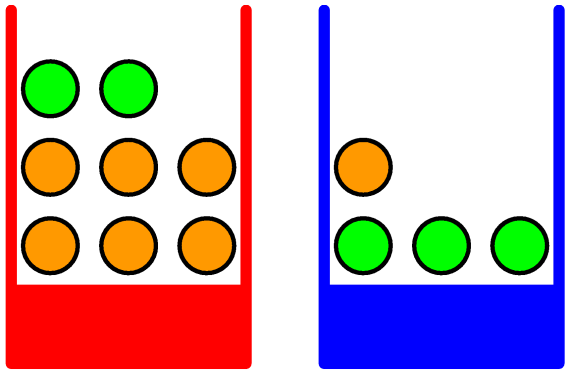


Figure : Two boxes with fruits (apples, oranges)

- We randomly pick red box 40% of the time and blue box 60% of the time
- From that box we randomly select an item of fruit equally likely in the box
- Having observed which sort of fruit it is we replace it in the box from which it came $\Rightarrow$ Repeat the experiment!
- $B \in \{r, b\}$ is a randomly selected box, $F \in \{a, o\}$ is a randomly selected fruit

Probability Rules

- $\mathbb{P}(B = r) = \frac{4}{10}$, $\mathbb{P}(B = b) = \frac{6}{10}$
- "what is the overall probability that the selection procedure will pick an apple?", "given that we have chosen an orange, what is the probability that the box we chose was the blue one?"
- More general example: two random variables (X, Y), $X \in \{x_1, \ldots, x_M\}$, $Y \in \{y_1, \ldots, y_L\}$

Probability Rules

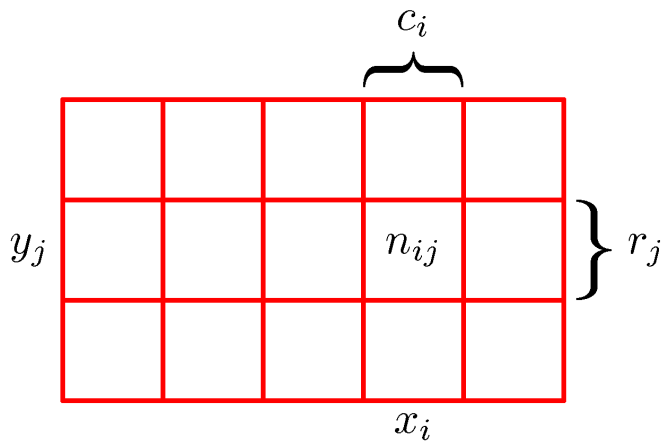


Figure : More general example

Natural definitions:

$$\mathbb{P}(X = x_i, Y = y_j) = \frac{n_{ij}}{n},$$

since $P(X = x_i) = \frac{c_i}{n}$, $c_i = \sum_j n_{ij}$, then

$$\mathbb{P}(X = x_i) = \sum_{j=1}^{L} \mathbb{P}(X = x_i, Y = y_j).$$

Probability Rules

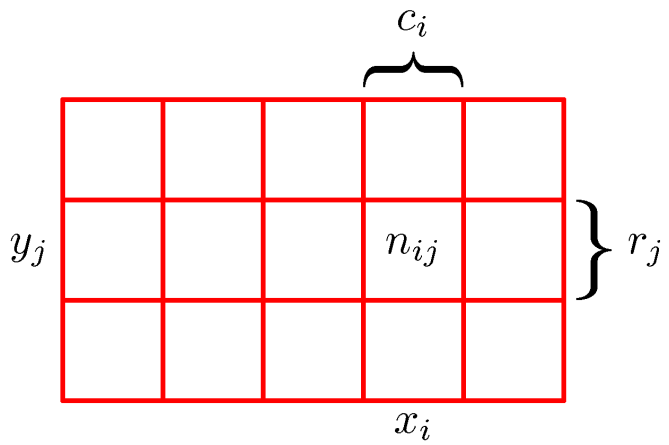


Figure : More general example

Natural definitions: since

$$\mathbb{P}(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i},$$

then

$$\mathbb{P}(X = x_i, Y = y_j) = \frac{n_{ij}}{n} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{n},$$
$$= \mathbb{P}(Y = y_j | X = x_i)\mathbb{P}(X = x_i)$$

Probability Rules

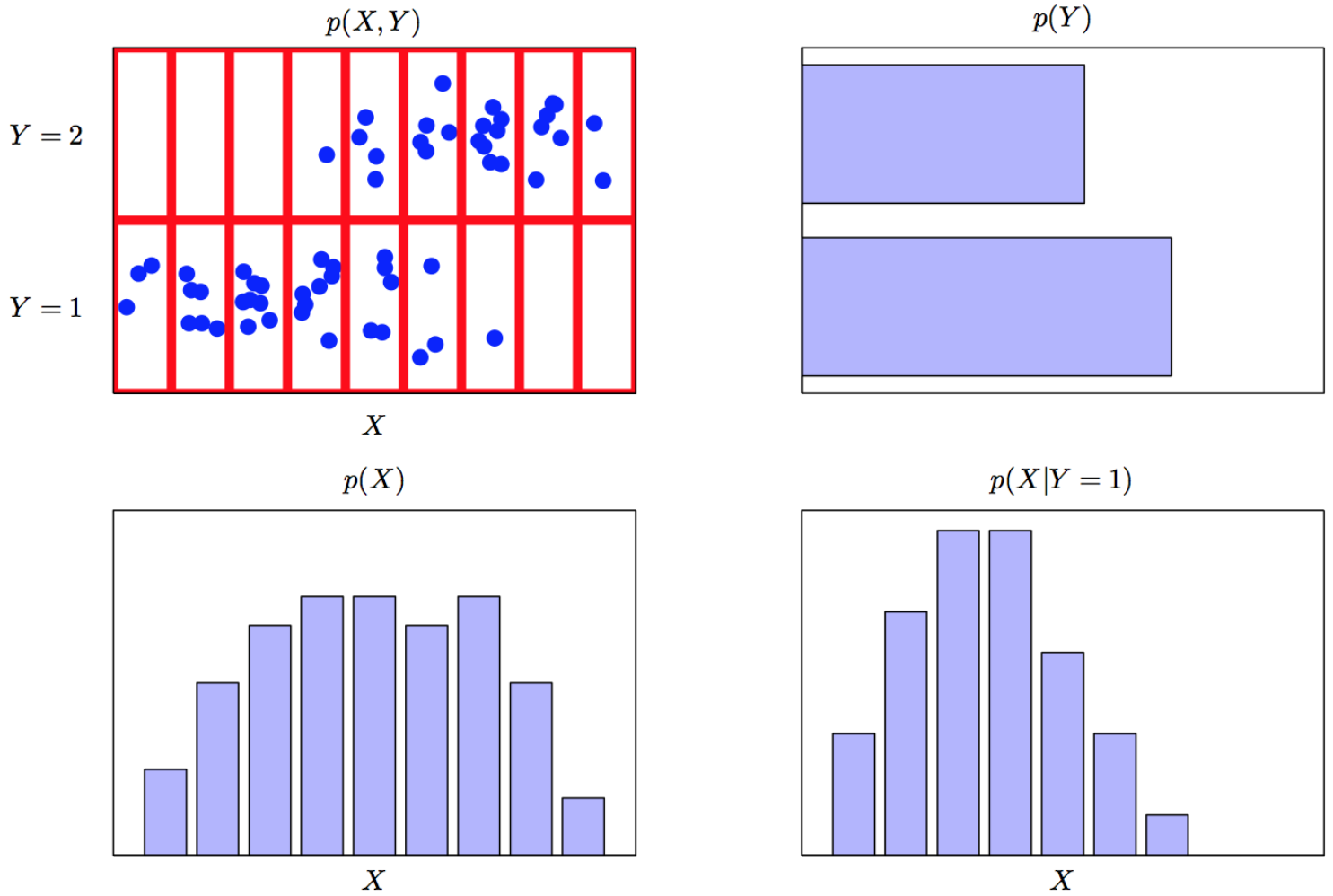


Figure : Example of a joint distribution

Probability Rules

$$\mathbb{P}(X) = \sum_Y \mathbb{P}(X, Y)$$

$$\mathbb{P}(X, Y) = \mathbb{P}(Y|X)\mathbb{P}(X)$$

$$\mathbb{P}(X, Y) = \mathbb{P}(Y, X)$$

$$\mathbb{P}(Y|X) = \frac{\mathbb{P}(X|Y)\mathbb{P}(Y)}{\mathbb{P}(X)}$$

$$\mathbb{P}(X) = \sum_Y \mathbb{P}(X|Y)\mathbb{P}(Y)$$

$$\mathbb{P}(F = a) = \mathbb{P}(F = a|B = r)\mathbb{P}(B = r) + \mathbb{P}(F = a|B = b)\mathbb{P}(B = b)$$
$$= \frac{1}{4}\frac{4}{10} + \frac{3}{4}\frac{6}{10} = \frac{11}{20}$$

$$\mathbb{P}(B = r|F = o) = \frac{\mathbb{P}(F = o|B = r)\mathbb{P}(B = r)}{\mathbb{P}(F = o)} = \frac{3}{4}\frac{4}{10}\frac{20}{9} = \frac{2}{3}$$

Probability densities

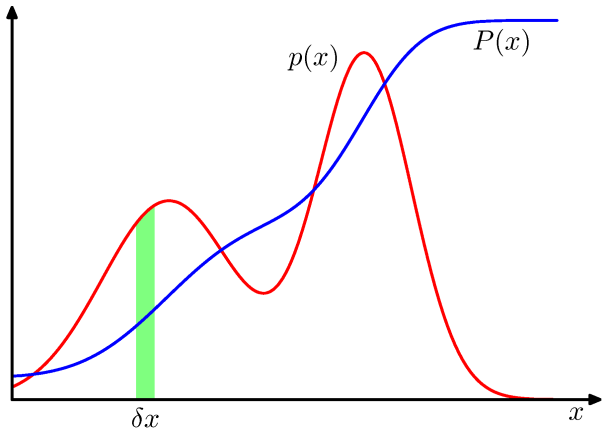


Figure : Probability density: $\mathbb{P}(X \in (x, x + \delta x)) = p(x)\delta x$

$$p(\mathbf{x}) \geq 0, \int_{\mathbb{R}} p(\mathbf{x})d\mathbf{x} = 1$$

$$F(\mathbf{z}) = \mathbb{P}(X_1 \leq z_1, \ldots, X_M \leq z_M)$$

$$= \int_{-\infty}^{z_1} \cdots \int_{-\infty}^{z_M} p(\mathbf{x})d\mathbf{x} \quad \text{(cumulative distr. function)}$$

$$p(x) = \int p(x, y)dy, \; p(x, y) = p(y|x)p(x)$$

Expectations and covariances

$$\mathbb{E}[f] = \sum_x p(x) f(x)$$

$$\mathbb{E}[f] = \int_x p(x) f(x) dx$$

$$\mathbb{E}[f] \approx \frac{1}{n} \sum_{i=1}^{n} f(x_i), \quad x_i \sim p(\cdot)$$

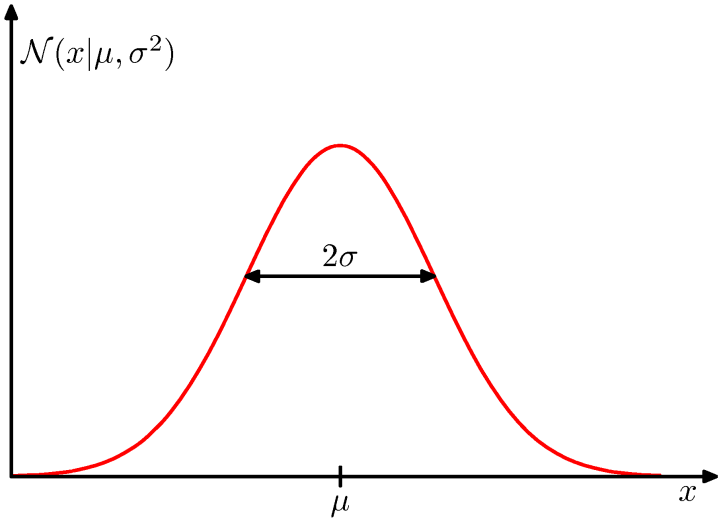
$$\mathbb{E}_x[f|y] = \sum_x p(x|y) f(x)$$

$$\mathrm{var}[f] = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2] = \mathbb{E}[(f(x))^2] - (\mathbb{E}[f(x)])^2$$

$$\mathrm{cov}[x, y] = \mathbb{E}_{x,y}[(x - \mathbb{E}[x])(y - \mathbb{E}[y])] = \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y]$$

$$\mathrm{cov}[\mathbf{x}, \mathbf{y}] = \mathbb{E}_{\mathbf{x},\mathbf{y}}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{y} - \mathbb{E}[\mathbf{y}])^{\mathrm{T}}] = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbf{x}\mathbf{y}^{\mathrm{T}}] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^{\mathrm{T}}]$$

Gaussian distribution



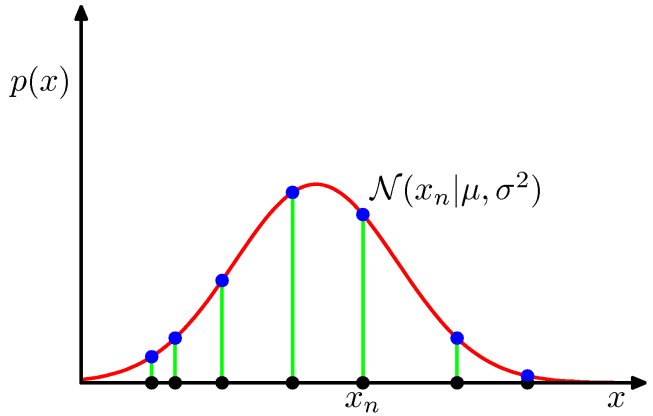
Gaussian distribution of $x \in \mathbb{R}^1$ with $\mathbb{E}[x] = \mu$, $\mathrm{var}[x] = \sigma^2$

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

Multivariate Gaussian distribution of $\mathbf{x} \in \mathbb{R}^d$ with $\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$, $\mathrm{cov}[\mathbf{x}] = \boldsymbol{\Sigma}$

$$\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Gaussian MLE



Likelihood of an i.i.d. data sample $\mathbf{X}_n = \{x_1, \ldots, x_n\}$ having gaussian distribution

$$p(\mathbf{X}|\mu,\sigma^2) = \prod_{i=1}^{n} \mathcal{N}(x_i;\mu,\sigma^2)$$

Log-likelihood is equal to

$$\log p(\mathbf{X}|\mu,\sigma^2) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2 - \frac{n}{2}\log\sigma^2 - \frac{n}{2}\log(2\pi)$$

Gaussian MLE

MLE is equal to

$$\mu_{ML} = \frac{1}{n}\sum_{i=1}^{n} x_i,\ \sigma^2_{ML} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_{ML})^2$$

Properties:

$$\mathbb{E}[\mu_{ML}] = \mu,\ \mathbb{E}[\sigma^2_{ML}] = \left(\frac{n-1}{n}\right)\sigma^2$$

Bias correction:

$$\tilde{\sigma}^2 = \frac{n}{n-1}\sigma^2_{ML} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_{ML})^2$$

Uncertainty

- Repeatable events $\Rightarrow$ classical (frequentist) interpretation of probability
- Bayesian view: probabilities provide a quantification of uncertainty
- Consider an uncertain (non-repeatable) event:
  - "whether the Arctic ice cap will have disappeared by the end of the century?"
  - we can generally have some idea how quickly we think the polar ice is melting
  - we obtain fresh data: e.g. from an Earth observation satellite we may revise our opinion on the rate of ice loss
  - we need to quantify our expression of uncertainty and make precise revisions of uncertainty in the light of new data

Probability vs. Uncertainty

- Let us use numerical values to represent degrees of belief
- Let us assume that a simple set of axioms encoding common sense properties of such beliefs is imposed
- Cox (1946) showed that this uniquely to a set of rules for manipulating degrees of belief that are equivalent to the sum and product rules of probability!
- Thus, probability theory could be regarded as an extension of Boolean logic to situations involving uncertainty

Example: curve fitting problem

- Data model: $y = f(x, \mathbf{w}) + \varepsilon$, so random values of y can be considered in the frequentist setting
- Quantify uncertainty about the appropriate choice for the model parameters $\mathbf{w}$?
- Prior $p(\mathbf{w})$ captures our assumptions about $\mathbf{w}$ before observing the data!

Probability vs. complexity:

— According to a Kolmogorov complexity theory: it is almost impossible to predict random rare events, so its description will be very long $\Rightarrow$ complex

— $\mathbf{w}$ defines "complexity" of the model

— $p(\mathbf{w})$ quantifies this complexity

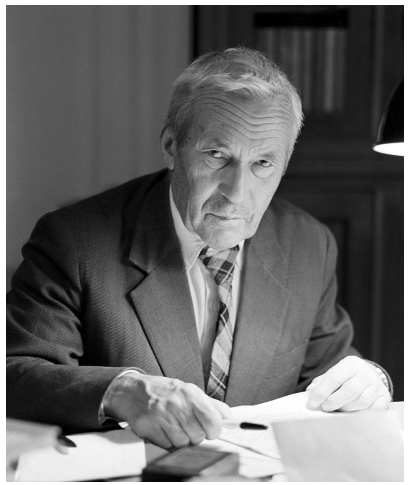


Figure : Kolmogorov A.N. (1903-1987)

Example: curve fitting problem

- The effect of the observed data $\mathcal{D} = \{y_1, \ldots, y_n\}$ is expressed through the conditional probability $p(\mathbf{w}|\mathcal{D})$:

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$$

- $p(\mathcal{D}|\mathbf{w})$ is a likelihood function (how probable the observed data set is for different settings of the parameter vector $\mathbf{w}$)
- General form:

$$\text{posterior} \sim \text{likelihood} \times \text{prior}$$

- Normalization constant (evience)

$$p(\mathcal{D}) = \int p(\mathcal{D}|\mathbf{w})p(\mathbf{w})d\mathbf{w}$$

Bayesian vs. Frequentist

- Frequentist setting: $\mathbf{w}$ is considered to be a fixed parameter, and error bars on its estimates obtained by considering the distribution of possible data sets $\mathcal{D}$
- Bayesian setting: here is only a single (observed) data set, and the uncertainty in the parameters is expressed through a probability distribution over $\mathbf{w}$
- The inclusion of prior knowledge arises naturally
- MLE estimate:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \log p(\mathcal{D}|\mathbf{w})$$

- MAP (Maximum posterior) estimate $\equiv$ regularized MLE:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} [\log p(\mathcal{D}|\mathbf{w}) + \log p(\mathbf{w})]$$

Bayesian vs. Frequentist

- Prior selection problem: the prior distribution is often selected on the basis of mathematical convenience rather than as a reflection of any prior beliefs
- Different approaches to construct priors including noninformative priors (to reduce the dependence on the prior)
- Bayesian methods are computationally intensive: output is a distribution, not just a point estimate! That is the reason why they became popular not long ago

Benefits of being a Bayesian

- Model selection
  - $\{\mathcal{M}_i\}_{i=1}^{M}$ is a set of models
  - Posterior

$$p(\mathcal{M}_i|\mathcal{D}) \sim p(\mathcal{M}_i)p(\mathcal{D}|\mathcal{M}_i), \ i = 1, \ldots, M$$

  - Model evidence

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w}, \mathcal{M}_i)p(\mathbf{w}|\mathcal{M}_i)d\mathbf{w}$$

- Prediction intervals
- Combining Bayesian models using probability rules without loss of information

Example: Bayesian decentralized learning

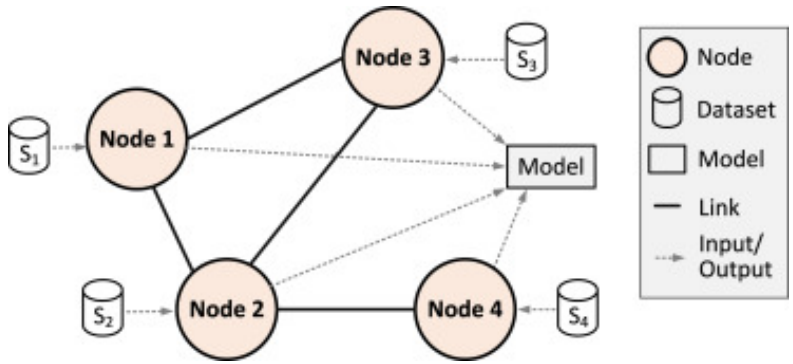
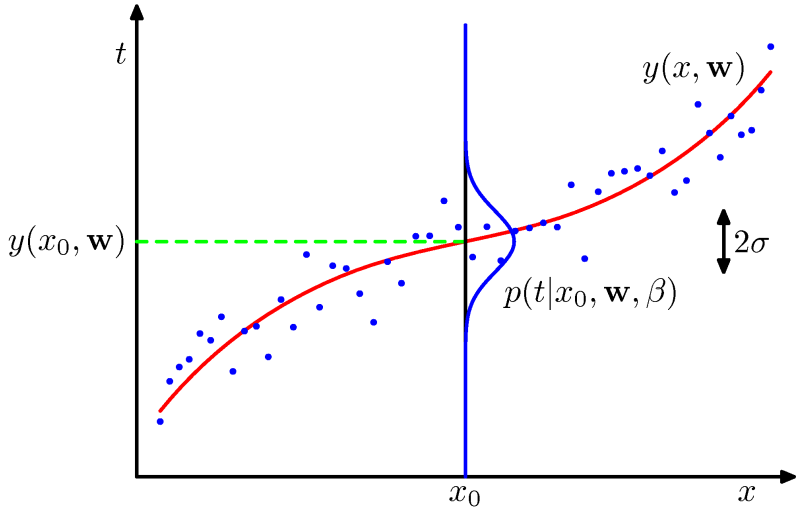


Figure : A decentralized learning scheme

- Nodes contain private data
- We can not gather all data on a server to learn a single model
- We learn locally on nodes using subsamples and construct Bayesian models
- We obtain a final model by aggregating local models using Bayes rules without loss of information and privacy

Curve-fitting



- Sample $S_n = \{\mathbf{X}_n, \mathbf{Y}_n\} = \{(x_i, y_i)\}_{i=1}^n$
- Probabilistic model $p(y|x, \mathbf{w}, \beta) = \mathcal{N}(y|y(x, \mathbf{w}), \beta^{-1})$, where
  - the mean is given by a polynomial $y(x, \mathbf{w})$
  - the noise precision is given by the parameter $\beta^{-1} = \sigma^2$
- Likelihood

$$p(\mathbf{Y}_n|\mathbf{X}_n, \mathbf{w}, \beta) = \prod_{i=1}^{n} \mathcal{N}(y_i|y(x_i, \mathbf{w}), \beta^{-1})$$

Curve-fitting

- Log-likelihood

$$\log p(\mathbf{Y}_n|\mathbf{X}_n,\ \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{i=1}^{n}(y(x_i, \mathbf{w})-y_i)^2+\frac{n}{2}\log\beta-\frac{n}{2}(2\pi)$$

- MLE of β

$$\frac{1}{\beta_{ML}} = \frac{1}{n}\sum_{i=1}^{n}(y(x_i, \mathbf{w}_{ML}) - y_i)^2$$

- Predictive distribution

$$p(y|x, \mathbf{w}_{ML}, \beta_{ML}) = \mathcal{N}(y|y(x, \mathbf{w}_{ML}), \beta_{ML}^{-1})$$

Curve-fitting revisited

- A prior distribution over the polynomial coefficients $\mathbf{w}$

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\boldsymbol{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right\}$$

- Posterior

$$p(\mathbf{w}|\mathbf{X}_n, \mathbf{Y}_n, \alpha, \beta) \sim p(\mathbf{Y}_n|\mathbf{X}_n, \mathbf{w}, \beta)p(\mathbf{w}|\alpha)$$

- Maximum posterior

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left[\frac{\beta}{2}\sum_{i=1}^{n}(y(x_i, \mathbf{w}) - y_i)^2 + \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right]$$

Bayesian curve-fitting

- Given the training data $\mathbf{X}_n$ and $\mathbf{Y}_n$, and a new test point x, our goal is to predict the value of y
- We would like to evaluate the predictive distribution $p(y|x, \mathbf{X}_n, \mathbf{Y}_n)$
- The predictive distribution

$$p(y|x, \mathbf{X}_n, \mathbf{Y}_n) = \int p(y|x, \mathbf{w}) p(\mathbf{w}|\mathbf{X}_n, \mathbf{Y}_n) d\mathbf{w}$$

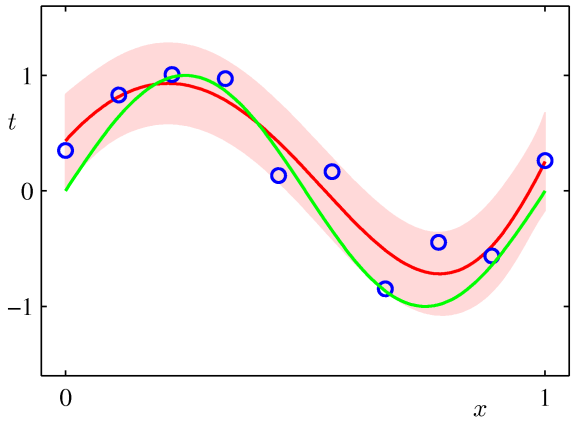


Figure : The predictive distribution for a polynomial with $M = 9$, parameters $\alpha = 5 \times 10^{-3}$ and $\beta = 11.1$ (known noise variance) are fixed

Bayesian curve-fitting

In case of linear models with an i.i.d. Gaussian noise and Gaussian prior

$$p(y|x, \mathbf{X}_n, \mathbf{Y}_n) = \mathcal{N}(y|m(x), s^2(x)),$$

where the mean and variance

$$m(x) = \beta\{\boldsymbol{\phi}(x)\}^{\mathrm{T}}\mathbf{S}\sum_{i=1}^{n}\boldsymbol{\phi}(x_i)y_i,$$

$$s^2(x) = \beta^{-1} + \{\boldsymbol{\phi}(x)\}^{\mathrm{T}}\mathbf{S}\boldsymbol{\phi}(x),$$

$$\mathbf{S}^{-1} = \alpha\mathbf{I} + \beta\sum_{i=1}^{n}\boldsymbol{\phi}(x_i)\boldsymbol{\phi}(x)^{\mathrm{T}},$$

$\phi_j(x) = x^j$, $j = 0, 1, \ldots, M-1$

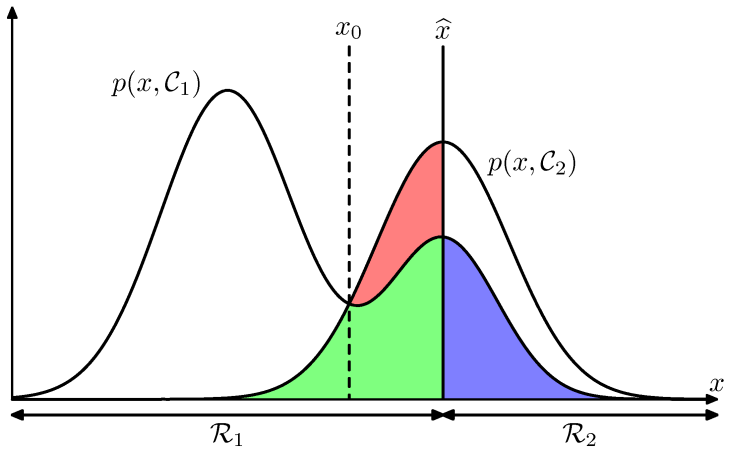
Decision theory

- Determination of $p(\mathbf{x}, y)$ from a set of training data is called *inference* problem
- Assume that $y \in \{0, 1\}$, i.e. there are two classes: C_1 and C_2, say "cancer/no cancer"
- Thus, we need to model $p(\mathbf{x}, C_k)$, $k = 1, 2$
- Decision: for a new x we have to decide to which C_k object $\mathbf{x}$ belongs to
- Bayesian decision

$$P(C_k \mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})}.$$

Here $p(C_k)$ is a prior probability to "have/not to have" cancer

Minimizing the misclassification rate

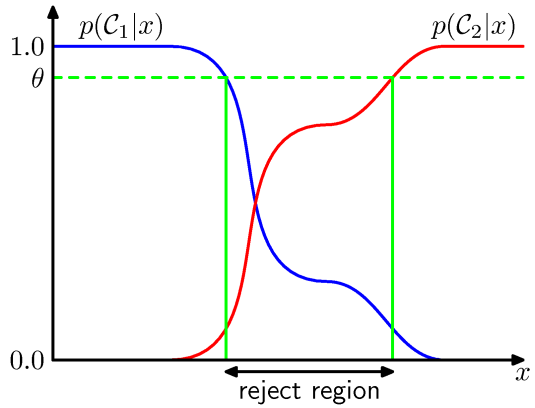


- R_k a decision region, i.e. if $\mathbf{x} \in R_k$, then we choose C_k
- The probability of a mistake is

$$\begin{aligned} p(\text{mistake}) &= p(\mathbf{x} \in R_1, C_2) + p(\mathbf{x} \in R_2, C_1) \\ &= \int_{R_1} p(\mathbf{x}, C_2) d\mathbf{x} + \int_{R_2} p(\mathbf{x}, C_1) d\mathbf{x} \end{aligned}$$

- We assign $\mathbf{x}$ to such C_k, for which $p(C_k, \mathbf{x})$ is the highest. Since $p(C_k, \mathbf{x}) = p(C_k|\mathbf{x})p(\mathbf{x})$, this is the same as to consider $p(C_k|\mathbf{x})$

Minimizing the expected loss



- Expected loss

$$\mathbb{E}[L] = \sum_k \sum_j \int_{R_j} L_{kj} p(\mathbf{x}, C_k) d\mathbf{x}$$

- The decision rule should provide the smallest $\sum_k L_{kj} p(\mathbf{x}, C_k)$. Since $p(\mathbf{x}, C_k) = p(C_k|\mathbf{x})p(\mathbf{x})$, then

$$y = \arg\min_j \sum_k L_{kj} p(C_k|\mathbf{x})$$

- The reject option: avoid making decisions on difficult cases

Inference and decisions

- Generative modeling:
  - Solve the inference problem to determine the class-conditional densities $p(\mathbf{x}|C_k)$
  - Infer the prior class probabilities $p(C_k)$
  - Calculate

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})},$$

$$p(\mathbf{x}) = \sum_k p(\mathbf{x}|C_k)p(C_k)$$

  - Use decision theory to assign each new $\mathbf{x}$ to one of the classes
- In case of new costs L_{kj} we can easily recalculate decision using obtained $p(C_k|\mathbf{x})$
- Sampling from $p(\mathbf{x})$ we can generate new objects
- We can use $p(\mathbf{x})$ for outlier (anomaly) detection (one-class classification)

Inference and decisions

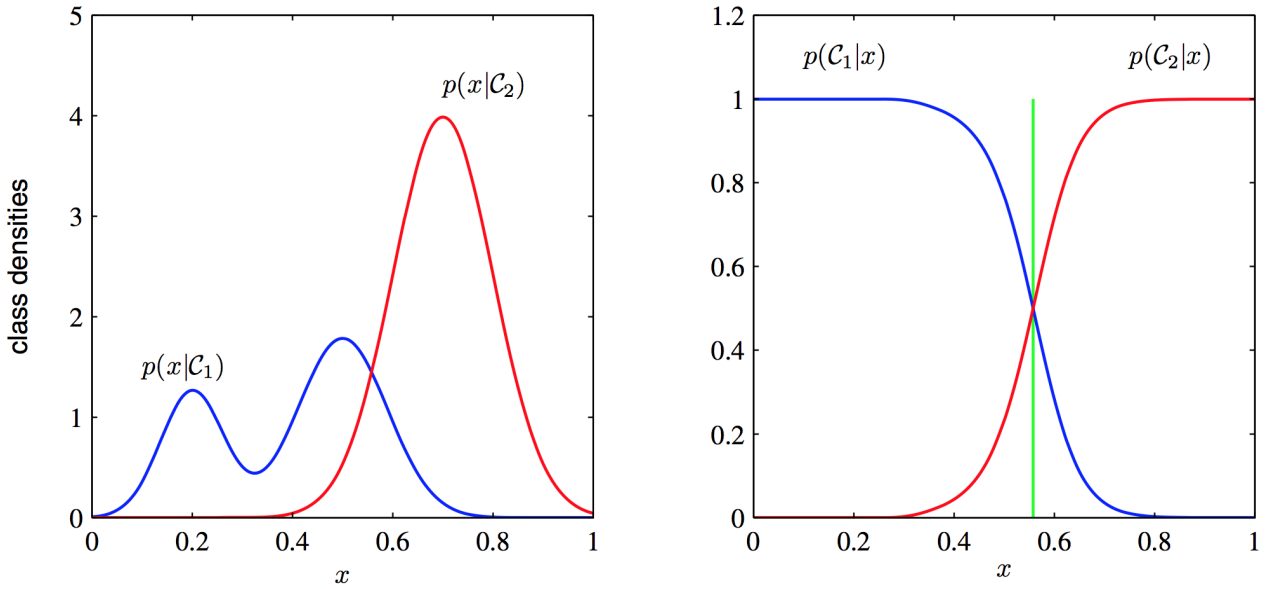


Figure : The class-conditional densities may contain a lot of structure that has little effect on the posterior probabilities

- Discriminative modeling:
  - Perform inference to determine $p(C_k|\mathbf{x})$
  - Use decision theory to assign each new $\mathbf{x}$ to one of the classes
- Discriminant function: find a function $f(\mathbf{x})$ which maps each input $\mathbf{x}$ directly onto a class label

Benefits using posterior probabilities

- Minimizing risk with taking into account real losses
- Reject option
- Compensating for class priors
  - Cancer detection from X-ray images
  - Cancer is rare $\Rightarrow$ classification problem is imbalanced and we should balance a sample
  - Using posteriors we will be able to easily compensate for the effects of modifications to the training data: re-weight posteriors w.r.t. to real distribution of classes
- Combining models. Naive Bayes:
  - Two tests: X-ray $\mathbf{x}_I$ and blood test $\mathbf{x}_B$
  - Given conditional independence property we get

$$p(C_k|\mathbf{x}_I, \mathbf{x}_B) \sim p(\mathbf{x}_I, \mathbf{x}_B|C_k)p(C_k) \sim p(\mathbf{x}_I|C_k)p(\mathbf{x}_B|C_k)p(C_k)$$
$$\sim \frac{p(C_k|\mathbf{x}_I)p(C_k|\mathbf{x}_B)}{p(C_k)}$$

Information

- Let us consider a discrete random variable x. How much information is received hwne we observe a specific value for this variable?
- The amount of information can be viewed as the "degree of surprise" on learning the value of x
- A highly improbable event has just occurred $\Rightarrow$ we will have received more information
- If some very likely event has just occurred, and if we knew that the event was certain to happen $\Rightarrow$ we would receive no information

Information

- We look for a quantity $h(x)$ that is a monotonic function of the probability $p(x)$
- If we have two events x and y that are unrelated, then it is reasonable to expect that $h(x, y) = h(x) + h(y)$
- Since $p(x, y) = p(x)p(y)$ for unrelated events, then it can be proved that

$$h(x) = -\log_2 p(x)$$

Entropy

- Suppose that a sender wishes to transmit the value of a random variable to a receiver
- The average amount of information that they transmit in the process is

$$H[x] = -\sum_x p(x) \log_2 p(x)$$

- Consider a r.v. x having 8 possible states, each of which is equally likely. In order to communicate the value of x to a receiver, we would need to transmit a message of length 3 bits, i.e.

$$H[x] = -8 \times \frac{1}{8} \log_2 \frac{1}{8} = 3 \,\text{bits}$$

Entropy

Now consider an example of a variable having 8 possible states $\{a, b, c, d, e, f, g, h\}$ with probabilities $\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\right)$. Then

$$H[x] = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{16}\log_2\frac{1}{16} - \frac{4}{64}\log_2\frac{1}{64} = 2 \text{ bits}$$

We see that the nonuniform distribution has a smaller entropy than the uniform one!

Entropy

- Let us consider how we would transmit the identity of the variable's state to a receiver
- We could do this, as before, using a 3-bit number
- Or we can take advantage of the nonuniform distribution: shorter codes for the more probable events, longer codes for the less probable events; we hope to get a shorter average code length
- E.g. we can represent the states $\{a, b, c, d, e, f, g, h\}$ with a set of code strings: 0, 10, 110, 1110, 111100, 111101, 111110, 111111.

Entropy

- The average length of the code is

$$\text{average code length} = \frac{1}{2}\times 1+\frac{1}{4}\times 2+\frac{1}{8}\times 3+\frac{1}{16}\times 4+4\times\frac{1}{64}\times 6 = 2\,\text{bits}$$

- Shorter code strings cannot be used because it must be possible to disambiguate a concatenation of strings into its component parts
- E.g. 11001110 decodes uniquely into the state sequence c, a, d
- The noiseless coding theorem (Shannon, 1948) states that the entropy is a lower bound on the number of bits needed to transmit the state of a random variable

Alternative definition of Entropy

- Entropy $\Leftrightarrow$ the average amount of information needed to specify the state of a random variable
- We consider a set of n identical objects. We divide objects into a set of bins, each containing n_i objects
- The total number of ways to allocate n objects is

$$W = \frac{n!}{\prod_i n_i!}$$

- The entropy is equal to

$$H = \frac{1}{n} \log W = \frac{1}{n} \log n! - \frac{1}{n} \sum_i \log n_i!$$

Alternative definition of Entropy

- The entropy is equal to

$$H = \frac{1}{n} \log W = \frac{1}{n} \log n! - \frac{1}{n} \sum_i \log n_i!$$

- We consider the limit $n \to \infty$, apply Stirling's approximation $\log n! = n \log n - n$ and get

$$H = - \lim_{n \to \infty} \sum_i \left(\frac{n_i}{n} \right) \log \left(\frac{n_i}{n} \right) = - \sum_i p_i \log p_i$$

- Physics: the specific arrangements of objects in the bins is called a microstate, and the overall distribution of occupation numbers, expressed through the ratios $\frac{n_i}{n}$, is called a macrostate. The multiplicity W is also known as the weight of the macrostate

Maximal Entropy

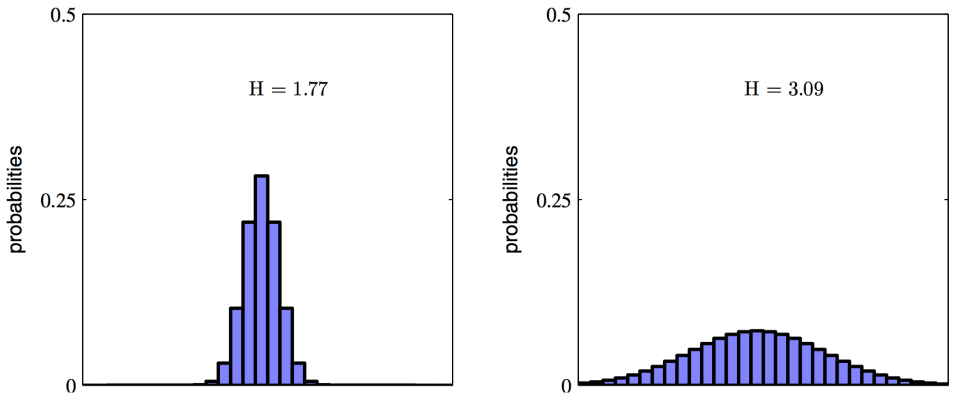


Figure : The higher value of H is for the broader distr.

- Let us consider an optimization problem

$$H[p] = -\sum_{i=1}^{M} p(x_i)\log p(x_i) \to \max_{p(\cdot)\geq 0:\, \sum_{i=1}^{M} p(x_i)=1}$$

- Using the Lagrange multipliers method we get

$$p^*(x_i) = \frac{1}{M}, i = 1, \ldots, M$$

Maximal entropy

- This result can be generalized to the case of a general density functions $p(\mathbf{x})$

$$H[p] = -\int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x}$$

- For d-dimensional $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$H[p] = \frac{1}{2} \log \left[(2\pi e)^d |\boldsymbol{\Sigma}| \right]$$

Conditional entropy

- Suppose we have a joint distribution $p(\mathbf{x}, y)$
- If a value of $\mathbf{x}$ is already known, then the additional information needed to specify value of y is $-\log p(y|\mathbf{x})$
- Conditional entropy is equal to

$$H[y|\mathbf{x}] = -\int\int p(y, \mathbf{x}) \log p(y|\mathbf{x}) d\mathbf{x}$$

- It holds that

$$H[\mathbf{x}, y] = H[y|\mathbf{x}] + H[\mathbf{x}]$$

Relative entropy

- $p(\mathbf{x})$ is an unknown distribution. We approximate it using an approximating distribution $q(\mathbf{x})$
- Suppose we use $q(\mathbf{x})$ to construct a coding scheme to transmit $\mathbf{x}$ to a receiver
- The average additional amount of information (in nats) required to specify $\mathbf{x}$ as a result of using $q(\mathbf{x})$ instead of the true $p(\mathbf{x})$ is

$$KL(p \,\|\, q) = -\int p(\mathbf{x}) \log q(\mathbf{x}) d\mathbf{x} - \left(-\int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} \right)$$
$$= -\int p(\mathbf{x}) \log \left\{ \frac{q(\mathbf{x})}{p(\mathbf{x})} \right\} d\mathbf{x}$$

This is relative entropy or Kullback-Leibler divergence

Relative entropy

- Non-symmetrical quantity: $KL(p \,\|\, q) \not\equiv KL(q \,\|\, p)$
- Since for any convex function

$$f(\lambda a + (1-\lambda)b) \leq \lambda f(a) + (1-\lambda) f(b),$$

i.e. $f(\mathbb{E}[x]) \leq \mathbb{E}[f(x)]$ (Jensen's inequality), then

$$KL(p \,\|\, q) = -\int p(\mathbf{x}) \log \left\{ \frac{q(\mathbf{x})}{p(\mathbf{x})} \right\} d\mathbf{x} \geq -\log \int q(\mathbf{x}) d\mathbf{x} = 0$$

with equality if only if $p \equiv q$

Relative entropy

- Data is generated i.i.d. from $p(\mathbf{x})$
- We approximate $p(\mathbf{x})$ using some parametric distribution $q(\mathbf{x}|\boldsymbol{\theta})$
- To tune $\boldsymbol{\theta}$ we minimize the Kullback-Leibler divergence between $p(\mathbf{x})$ and $q(\mathbf{x}|\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$
- Suppose that we observe $\mathbf{x}_i, i = 1, \ldots, n$, drawn from $p(\mathbf{x})$. Then we can approximate $KL(p \,\|\, q)$ by

$$KL(p \,\|\, q) \approx \sum_{i=1}^{n} \{-\log q(\mathbf{x}_i|\boldsymbol{\theta}) + \log p(\mathbf{x}_i)\}$$

- Since $\log p(\mathbf{x}_i)$ does not depend on $\boldsymbol{\theta}$, minimization of this approximation is equivalent to maximization of log-likelihood

Mutual Information

- $p(\mathbf{x}, y)$ is a joint distribution. In case $\mathbf{x}$ and y are independent, we get that $p(\mathbf{x}, y) = p(\mathbf{x})p(y)$
- Otherwise evaluating the Kullback-Leibler divergence between $p(\mathbf{x}, y)$ and $p(\mathbf{x})p(y)$ we can measure the extent to which these variables are independent
- Mutual Information

$$\begin{aligned} I[\mathbf{x}, y] &\equiv KL(p(\mathbf{x}, y) \,\|\, p(\mathbf{x})p(y)) \\ &= -\int\int p(\mathbf{x}, y) \log \left(\frac{p(\mathbf{x})p(y)}{p(\mathbf{x}, y)} \right) d\mathbf{x}dy \end{aligned}$$

- The mutual information is related to the conditional entropy through

$$I[\mathbf{x}, y] = H[\mathbf{x}] - H[\mathbf{x}|y] = H[y] - H[y|\mathbf{x}]$$

Lectures

- Introduction to probability; Introduction to Bayesian statistics; Decision theory, probability distributions
- Bayesian linear models for regression; Bayesian linear models for classification; Bayesian PCA
- Variational inference; Expectation propagation; MCMC
- Latent variables; Graphical models; Markov chains inference
- Gaussian process regression; Gaussian process based ML methods; Dirichlet processes for clustering
- Bayesian Neural Networks; Bayesian Deep Neural Networks; Bayesian VAE

Seminars

- Introduction to Python and Theano: linear and logistic regression; PyMC3 - probabilistic programming library and model construction with PyMC3; Prior selection: objective and subjective approach
- EM algorithm; Bayesian Linear regression and Generalized linear regression; Bayesian PCA
- Approximate inference: variational inference and EP; Approximate inference: approaches to sampling; Approximate inference: MCMC
- Latent variables: Gaussian mixture; Inference in graphical models; Bayesian model selection
- Gaussian processes for regression; Gaussian processes for other ML problems; Dirichlet processes for clustering
- Bayesian neural networks; Bayesian neural networks with Lasagne; Bayesian VAE