Foundations of Machine Learning
CentraleSupélec — Fall 2016

4. Bayesian decision theory

Chloé-Agathe Azencott
Centre for Computational Biology, Mines ParisTech
chloe-agathe.azencott@mines-paristech.fr

Practical matters...

- **Class representatives**
  - Abdelhak Lemkhenter
  - Nathan Vermeesch
- **Lecture handouts**

Learning objectives

After this lecture, you should be able to

- **Apply Bayes rule** for simple inference and decision problems;
- Explain the connection between **Bayes decision rule**, **empirical risk minimization**, **maximum a priori** and **maximum likelihood;**
- Use a graph to express **conditional independence** among random variables;
- Apply the **Naive Bayes** algorithm.

Let's start by tossing coins...

Probability and inference

- Result of **tossing a coin:** x in {heads, tails}
  - x = f(z)   z: **unobserved variables**
  - Replace f(z) (maybe deterministic but unknown) with the **random variable** X in {0, 1} drawn from a **probability distribution** P(X=x).

Probability and inference

- Result of **tossing a coin:** x in {heads, tails}
  - x = f(z)   z: **unobserved variables**
  - Replace f(z) (maybe deterministic but unknown) with the **random variable** X in {0, 1} drawn from a **probability distribution** P(X=x).
- **What's a good model for the probability distribution P?**

E.g: a complex physical function of the composition of the coin, the force that is applied to it, initial conditions, etc.

Probability and inference

- Result of **tossing a coin:** x in {heads, tails}
  - x = f(z)   z: **unobserved variables**
  - Replace f(z) (maybe deterministic but unknown) with the **random variable** X in {0, 1} drawn from a **probability distribution** P(X=x).
- **Bernouilli distribution**

$$P(X = x) = {p_0}^x (1 - p_0)^{(1-x)}$$

- We do not know P but a **sample** $X = \{x^i\}_{i=1, \dots, n}$
- Goal: **approximate P** (from which X is drawn)

  How can we achieve this?

Probability and inference

- Result of **tossing a coin:** x in {heads, tails}
  - x = f(z)   z: **unobserved variables**
  - Replace f(z) (maybe deterministic but unknown) with the **random variable** X in {0, 1} drawn from a **probability distribution** P(X=x).
- **Bernouilli distribution**

$$P(X=x) = {p_0}^x (1-p_0)^{(1-x)}$$

- We do not know P but a **sample** $X = \{x^i\}_{i=1, \ldots, n}$
- Goal: **approximate P** (from which X is drawn)

  p_0 = # heads / # tosses
- **What's the prediction rule for a new toss?**

Probability and inference

- Result of **tossing a coin:** x in {heads, tails}
  - x = f(**z**) z: **unobserved variables**
  - Replace f(z) (maybe deterministic but unknown) with the **random variable** X in {0, 1} drawn from a **probability distribution** P(X=x).
- **Bernouilli distribution**

$$P(X=x) = {p_0}^x (1-p_0)^{(1-x)}$$

- We do not know P but a **sample** $X = \{x^i\}_{i=1, \ldots, n}$
- Goal: **approximate P** (from which X is drawn)

  p_0 = # heads / # tosses
- **Prediction** of next toss:

  heads if $p_0 > 0.5$, tails otherwise

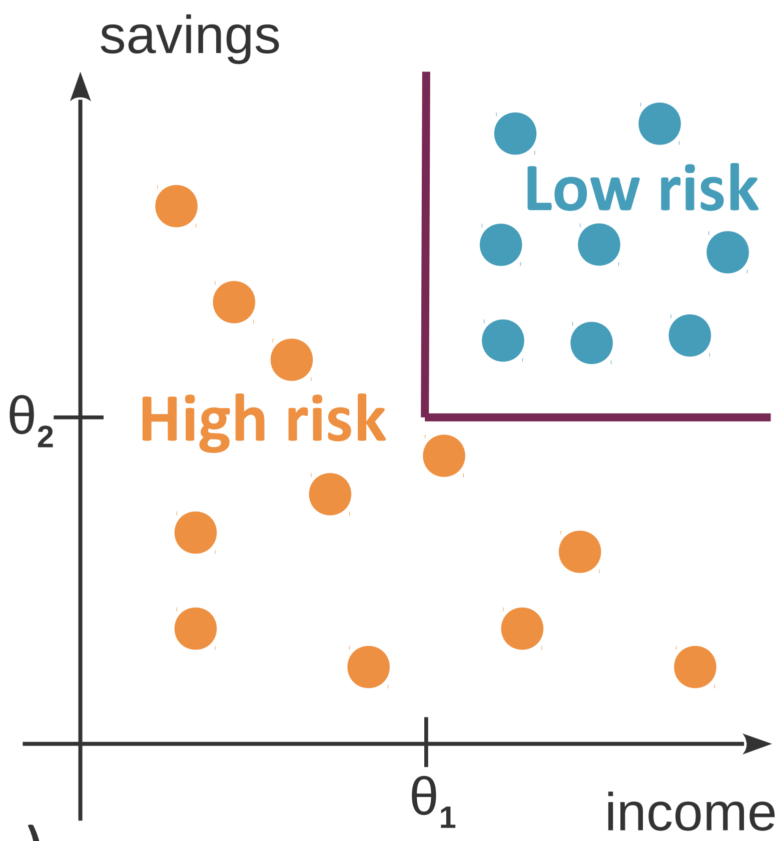
Classification

- Credit scoring:
  - Input = income (x_1), savings (x_2)
  - Output = {low-risk, high-risk}
- **Prediction:**
  - C = 1 if $P(C=1|x_1, x_2) > 0.5$

    C = 0 otherwise

    or
  - C = 1 if $P(C=1|x_1, x_2) > P(C=0|x_1, x_2)$

    C = 0 otherwise



Bayes rule

Reverend Thomas Bayes

170?-1761

… possibly

Bayes rule

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

Example: rare disease testing

- test is correct 99% of the time
- disease prevalence = 1 out of 10,000

What is the probability that a patient that tested positive actually has the disease?

99% ?   90% ?   10% ?   1% ?

Example: rare disease testing

- test is correct 99% of the time
- disease prevalence = 1 out of 10,000

What is the probability that a patient that tested positive actually has the disease?

$$P(d|t) = \frac{\overset{?}{P(d)}\,\overset{?}{P(t|d)}}{P(t)}$$

Example: rare disease testing

- test is correct 99% of the time $P(t|d) = P(\bar{t}|\bar{d}) = 0.99$
- disease prevalence = 1 out of 10,000 $P(d) = 10^{-4}$

What is the probability that a patient that tested positive actually has the disease?

$$P(d|t) = \frac{\overset{0.0001}{P(d)}\overset{0.99}{P(t|d)}}{\underset{?}{P(t)}}$$

Example: rare disease testing

- test is correct 99% of the time $P(t|d) = P(\bar{t}|\bar{d}) = 0.99$
- disease prevalence = 1 out of 10,000 $P(d) = 10^{-4}$

What is the probability that a patient that tested positive actually has the disease?

$$P(d|t) = \frac{\overbrace{P(d)}^{0.0001}\overbrace{P(t|d)}^{0.99}}{P(t)}$$

$$P(t) = \underbrace{P(t|d)}_{0.99}\underbrace{P(d)}_{0.0001} + \overbrace{P(t|\bar{d})}^{?}\underbrace{P(\bar{d})}_{?}$$

Example: rare disease testing

- test is correct 99% of the time $P(t|d) = P(\bar{t}|\bar{d}) = 0.99$
- disease prevalence = 1 out of 10,000 $P(d) = 10^{-4}$

What is the probability that a patient that tested positive actually has the disease?

$$P(d|t) = \frac{\overset{0.0001}{P(d)}\overset{0.99}{P(t|d)}}{P(t)}$$

$$P(t) = \underset{0.99}{P(t|d)}\underset{0.0001}{P(d)} + \overset{(1\text{-}0.99)}{P(t|\bar{d})}\underset{(1\text{-}0.0001)}{P(\bar{d})}$$

Example: rare disease testing

- test is correct 99% of the time $P(t|d) = P(\bar{t}|\bar{d}) = 0.99$
- disease prevalence = 1 out of 10,000 $P(d) = 10^{-4}$

What is the probability that a patient that tested positive actually has the disease?

$$P(d|t) = \frac{\overset{0.0001}{P(d)}\,\overset{0.99}{P(t|d)}}{P(t)} \approx 0.0098.$$

$$P(t) = \underset{0.99}{P(t|d)}\,\underset{0.0001}{P(d)} + \overset{(1-0.99)}{P(t|\bar{d})}\,\underset{(1-0.0001)}{P(\bar{d})}$$

Bayes rule

prior

likelihood

$$P(C = c|\boldsymbol{x}) = \frac{P(C = c)p(\boldsymbol{x}|C = c)}{p(\boldsymbol{x})}$$

posterior

evidence

$$P(C = 0) + P(C = 1) = 1$$
$$P(C = 0|\boldsymbol{x}) + P(C = 1|\boldsymbol{x}) = 1$$

$$p(\boldsymbol{x}) = p(\boldsymbol{x}|C = 1)P(C = 1) + p(\boldsymbol{x}|C = 0)P(C = 0)$$

Bayes' decision rule:

$$C = \begin{cases} 1 & \text{if } P(C = 1|\boldsymbol{x}) > P(C = 0|\boldsymbol{x}) \\ 0 & \text{otherwise.} \end{cases}$$

Maximum A Posteriori criterion

- **MAP decision rule:**
  - pick the hypothesis that is most probable
  - i.e. **maximize the posterior** $P(C|\boldsymbol{x}) = \dfrac{P(C)p(\boldsymbol{x}|C)}{p(\boldsymbol{x})}$

$$\Lambda_{\mathrm{MAP}}(\boldsymbol{x}) = \frac{P(C=1|\boldsymbol{x})}{P(C=0|\boldsymbol{x})}$$

- **Decision rule:**

If $\Lambda_{MAP}(\mathbf{x}) > 1$

then choose C=1

else choose C=0.

$$C = \begin{cases} 1 & \text{if } P(C=1|\boldsymbol{x}) > P(C=0|\boldsymbol{x}) \\ 0 & \text{otherwise.} \end{cases}$$

Likelihood ratio test (LRT)

$$\Lambda_{\text{MAP}}(\boldsymbol{x}) = \frac{P(C=1|\boldsymbol{x})}{P(C=0|\boldsymbol{x})} \qquad \Lambda_{\text{MAP}}(\boldsymbol{x}) >^? 1 \qquad P(C|\boldsymbol{x}) = \frac{P(C)p(\boldsymbol{x}|C)}{p(\boldsymbol{x})}$$

$$\Lambda_{\text{MAP}}(\boldsymbol{x}) = \frac{P(C=1)p(\boldsymbol{x}|C=1)p(\boldsymbol{x})}{P(C=0)p(\boldsymbol{x}|C=0)p(\boldsymbol{x})}$$

p(**x**) does not affect the decision rule.

- **Likelihood ratio test:**

test whether the **likelihood ratio** Λ(**x**) is larger than $\frac{P(C=0)}{P(C=1)}$

$$\Lambda(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C=1)}{p(\boldsymbol{x}|C=0)}$$

decision rule:

$$\Lambda(\boldsymbol{x}) >^? \frac{P(C=0)}{P(C=1)}$$

Example: LRT decision rule

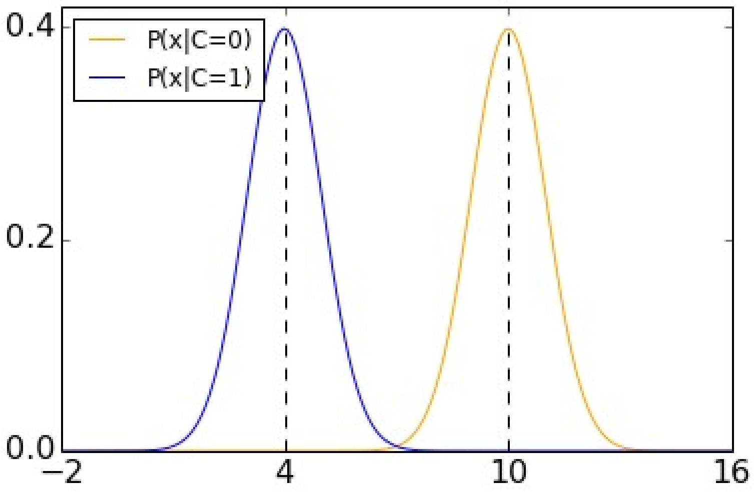
$$\Lambda(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C=1)}{p(\boldsymbol{x}|C=0)} >^{?} \frac{P(C=0)}{P(C=1)}$$

Assuming the likelihoods below <u>and equal priors,</u> derive a decision rule based on the LRT.

$$p(x|C=1) \sim \mathcal{N}(4,1) \qquad p(x|C=0) \sim \mathcal{N}(10,1)$$

$$Z \sim \mathcal{N}(\mu, \sigma^2):$$

$$f(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(z-\mu)^2/(2\sigma^2)}$$



- **Likelihood ratio:**

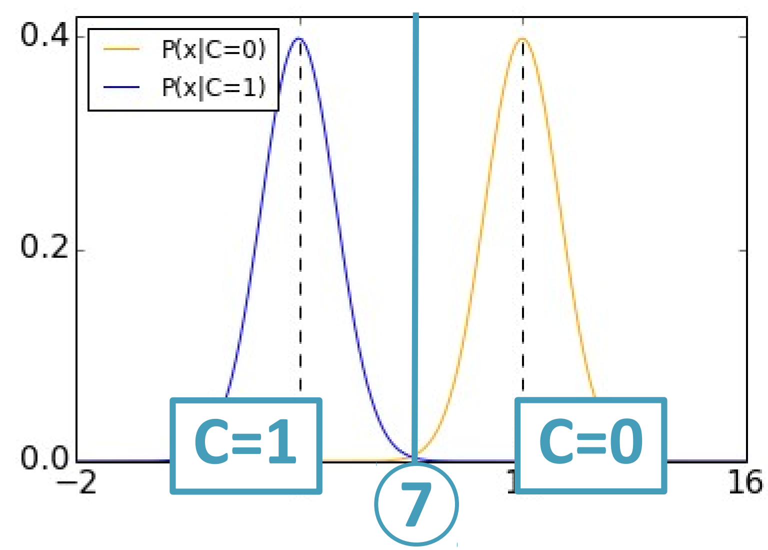
$$\Lambda(x) = \frac{(1/\sqrt{2\pi})e^{-(x-4)^2/2}}{(1/\sqrt{2\pi})e^{-(x-10)^2/2}}$$

- Simplifying the equation and taking the log:

$$\log(\Lambda(x)) = -(x-4)^2 + (x-10)^2$$

- **Equal priors** mean we're testing whether **log(LR) > 0**

  Hence: If x < 7 then assign C=1 else assign C=0



- **Likelihood ratio:**

$$\Lambda(x) = \frac{(1/\sqrt{2\pi})e^{-(x-4)^2/2}}{(1/\sqrt{2\pi})e^{-(x-10)^2/2}}$$

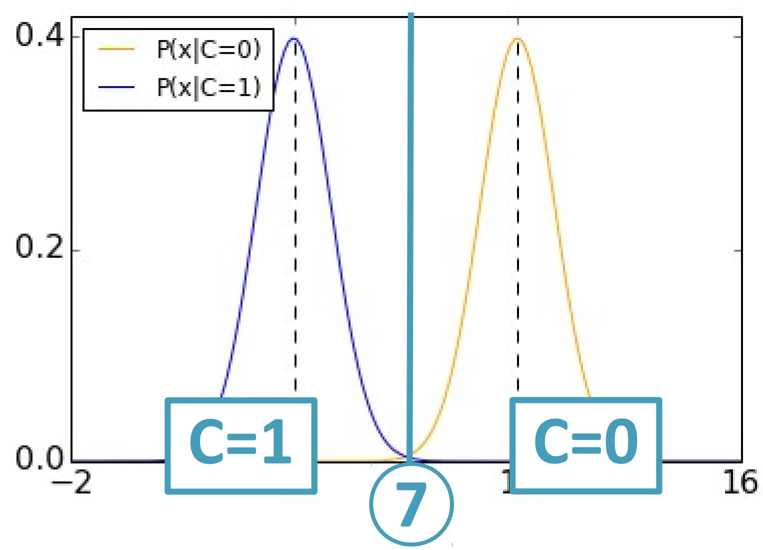
- Simplifying the equation and taking the log:

$$\log(\Lambda(x)) = -(x-4)^2 + (x-10)^2$$

- **Equal priors** mean we're testing whether **log(LR) > 0**

  Hence: If x < 7 then assign C=1 else assign C=0

How does the rule change if P(C=1) = 2 P(C=0)?



- **Likelihood ratio:**

$$\Lambda(x) = \frac{(1/\sqrt{2\pi})e^{-(x-4)^2/2}}{(1/\sqrt{2\pi})e^{-(x-10)^2/2}}$$

- Simplifying the equation and taking the log:

$$\log(\Lambda(x)) = -(x-4)^2 + (x-10)^2$$

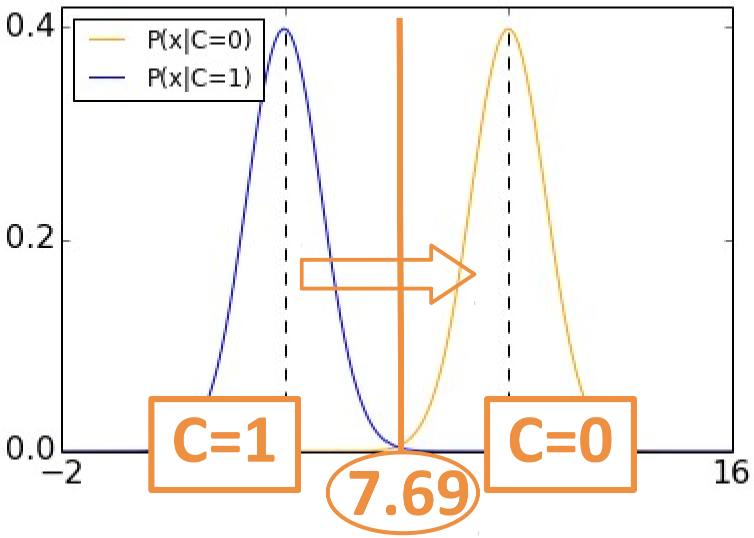
- **Equal priors** mean we're testing whether **log(LR) > 0**

  Hence: If x < 7 then assign C=1 else assign C=0

How does the rule change if P(C=1) = 2 P(C=0)?

x < 7 – log(1/2) ≈ 7.69

C=1 is more likely.



Bayes rule for K > 2

- **Bayes rule:**

$$P(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)P(C_k)}{\sum_{k=1}^{K} p(\boldsymbol{x}|C_k)P(C_k)}$$

- $P(C_k) \geq 0$ and $P(C_1) + P(C_2) + \ldots + P(C_K) = 1$
- **What is the decision rule?**

Bayes rule for K > 2

- **Bayes rule:**

$$P(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)P(C_k)}{\sum_{k=1}^{K} p(\boldsymbol{x}|C_k)P(C_k)}$$

- $P(C_k) \geq 0$ and $P(C_1) + P(C_2) + \ldots + P(C_K) = 1$
- **Decision:**

  Choose C_k if $P(C_k \mid \mathbf{x}) = \max_k P(C_k \mid \mathbf{x})$

Risk minimization

Losses and risks

- So far we've assumed all errors were **equally costly.**

  But misclassfying a cancer sufferer as a healthy patient is much more problematic than the other way around.

- **Action α_k:** assigining class C_k
- **Loss:** quantify the cost λ_{kl} of taking action α_k when the true class is C_l
- **Expected risk:**

$$R(\alpha_k|\boldsymbol{x}) = \sum_{l=1}^{K} \lambda_{lk} P(C_l|\boldsymbol{x})$$

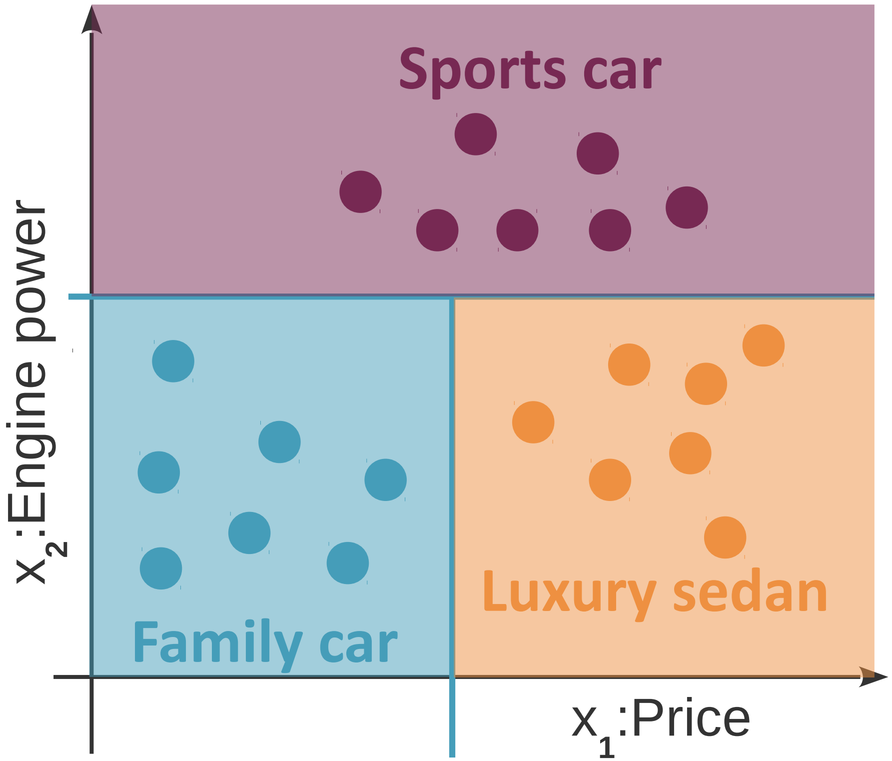
- **Decision (Bayes Classifier):** $\arg\min_k R(\alpha_k|\boldsymbol{x})$

Discriminant functions

- Classification = find K **discriminant functions** f_k s.t. $\mathbf{x}$ is assigned class C_k if k = argmax $f_l(\mathbf{x})$
- Bayes classifier: $f_k(\boldsymbol{x}) = -R(\alpha_k|\boldsymbol{x})$

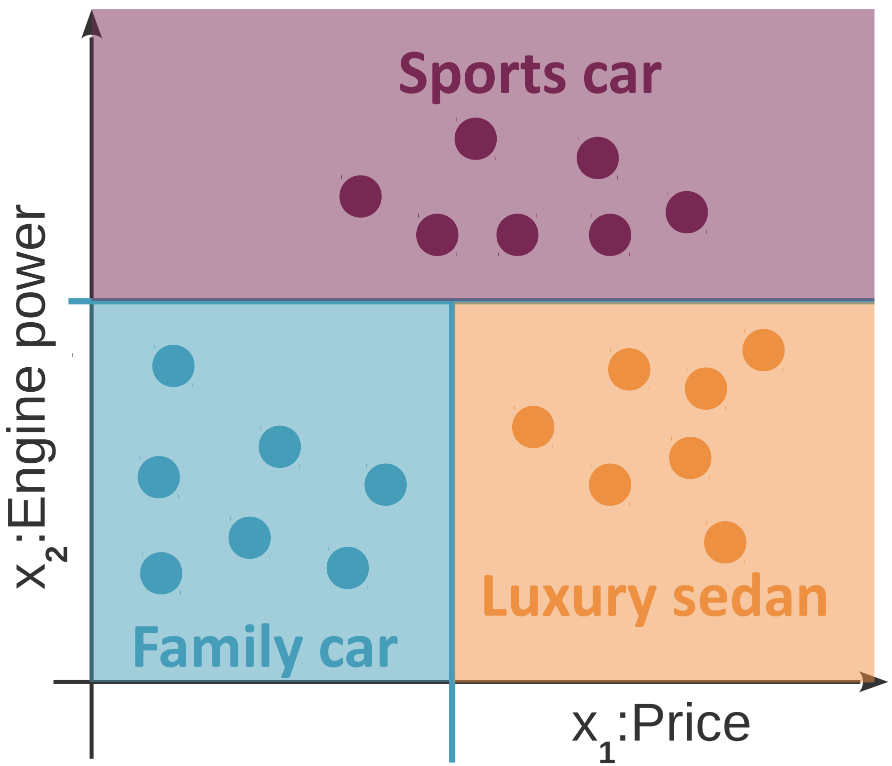
Discriminant functions

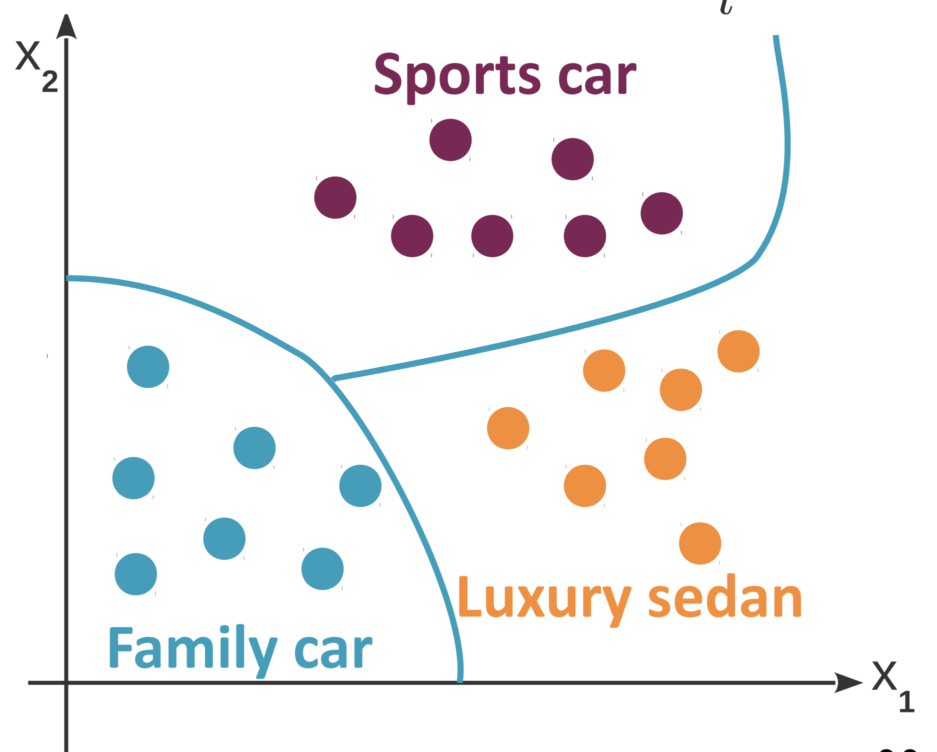
- Classification = find K **discriminant functions** f_k s.t. $\mathbf{x}$ is assigned class C_k if k = argmax $f_l(\mathbf{x})$
- Bayes classifier: $f_k(\boldsymbol{x}) = -R(\alpha_k|\boldsymbol{x})$
- Defines K **decision regions** $R_k = \{\boldsymbol{x} : f_k(\boldsymbol{x}) = \max_l f_l(\boldsymbol{x})\}$



Discriminant functions

- Classification = find K **discriminant functions** f_k s.t. $\mathbf{x}$ is assigned class C_k if $k = \text{argmax } f_l(\mathbf{x})$
- Bayes classifier: $f_k(\boldsymbol{x}) = -R(\alpha_k|\boldsymbol{x})$
- Defines K **decision regions** $R_k = \{\boldsymbol{x} : f_k(\boldsymbol{x}) = \max_l f_l(\boldsymbol{x})\}$





Bayes risk minimization

- **Bayes risk:** overall expected risk

$$R(\boldsymbol{x}) = \sum_{k=1}^{K} \sum_{l=1}^{K} \lambda_{lk} \; p(\boldsymbol{x} \in R_k | C_l) P(C_l)$$

- **Bayes decision rule:** use the discriminant functions that **minimize the Bayes risk.**

Bayes risk minimization

- **Bayes risk:** overall expected risk

$$R(\boldsymbol{x}) = \sum_{k=1}^{K} \sum_{l=1}^{K} \lambda_{lk} \; p(\boldsymbol{x} \in R_k | C_l) P(C_l)$$

- **Bayes decision rule:** use the discriminant functions that **minimize the Bayes risk.**
- This is also a LRT.

For 2 classes, let us show that Bayes decision rule is equivalent to:

$$\Lambda(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C=1)}{p(\boldsymbol{x}|C=0)} >^{?} \frac{(\lambda_{10} - \lambda_{00}) P(C=0)}{(\lambda_{01} - \lambda_{11}) P(C=1)}$$

0/1 Loss

- All misclassifications are **equally costly.**
- λ_{kl} = 0 if k=l and 1 otherwise

$$\begin{aligned} R(\alpha_k|\boldsymbol{x}) &= \sum_{l=1}^{K} \lambda_{lk} P(C_l|\boldsymbol{x}) \\ &= \sum_{l \neq k} P(C_l|\boldsymbol{x}) \\ &= 1 - P(C_k|\boldsymbol{x}) \end{aligned}$$

- **Minimizing the risk:**
  - choose the most probable class (MAP)
  - this is equivalent to the Bayes decision rule.

$$\Lambda(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C=1)}{p(\boldsymbol{x}|C=0)} >^{?} \frac{(\lambda_{10} - \lambda_{00})P(C=0)}{(\lambda_{01} - \lambda_{11})P(C=1)}$$

Reject

- Add an artificial "reject" class (K+1) for **refusing to take a decision.**

  E.g. Zip code detection.

- $$\lambda_{kl} = \begin{cases} 0 \text{ if } k = k \\ \lambda \text{ if } k = K+1 \\ 1 \text{ otherwise} \end{cases}$$

$$R(\alpha_k|\boldsymbol{x}) = \sum_{l \neq k} P(C_l|\boldsymbol{x}) = 1 - P(C_k|\boldsymbol{x})$$

$$R(\alpha_{K+1}|\boldsymbol{x}) = \sum_{l=1}^{K} \lambda P(C_l|\boldsymbol{x}) = \lambda$$

- **Decision:**

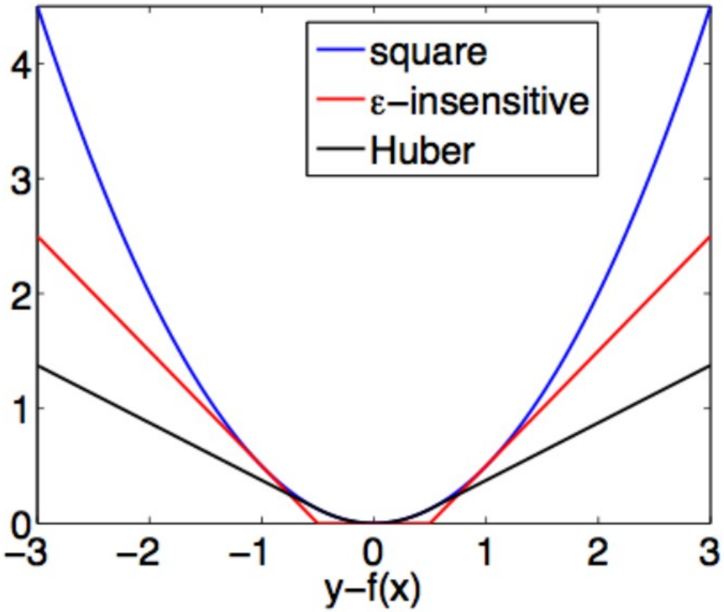
  Ck if P(Ck|x) > P(Cl|x) for all l ≠ k **and P(Ck|x) > 1-λ**

  else reject.

  Only meaningful if 0 < λ < 1

Losses for regression

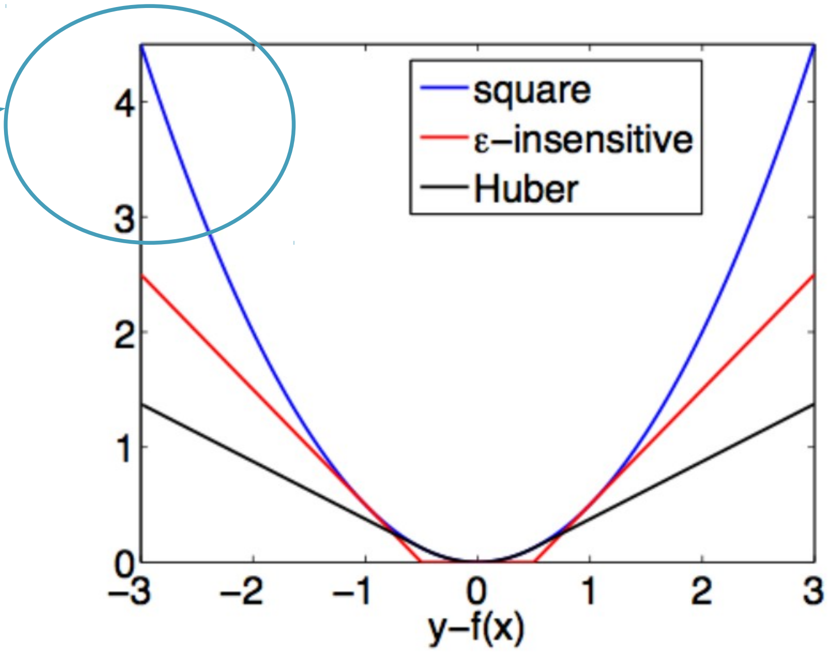
- **Square loss:** $L(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$



Losses for regression

- **Square loss:** $L(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$

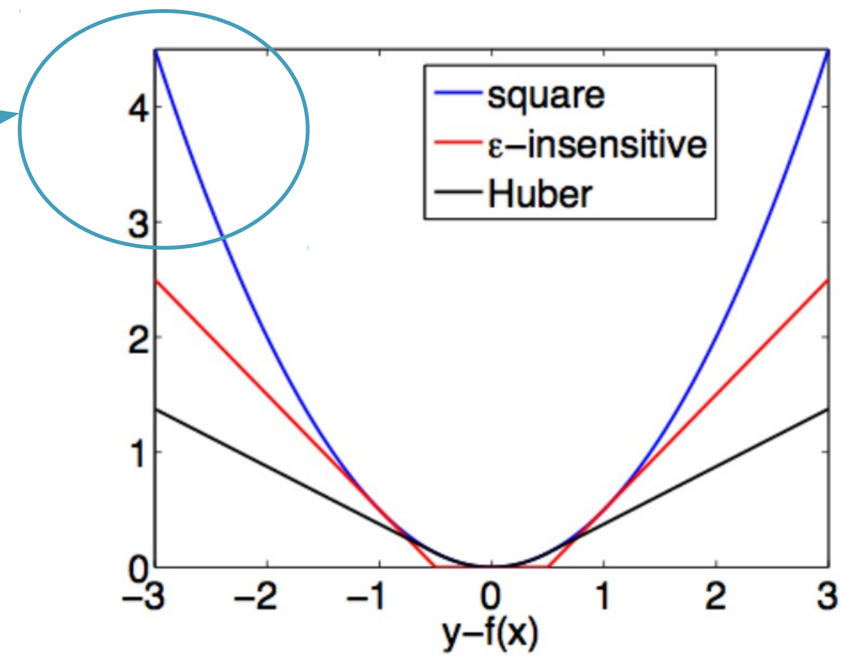
square loss: dominated by outliers



Losses for regression

- **Square loss:** $L(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$
- **ε-insensitive loss:** $L(f(\mathbf{x}), y) = (|f(\mathbf{x}) - y| - \varepsilon)_+$

square loss: dominated by outliers

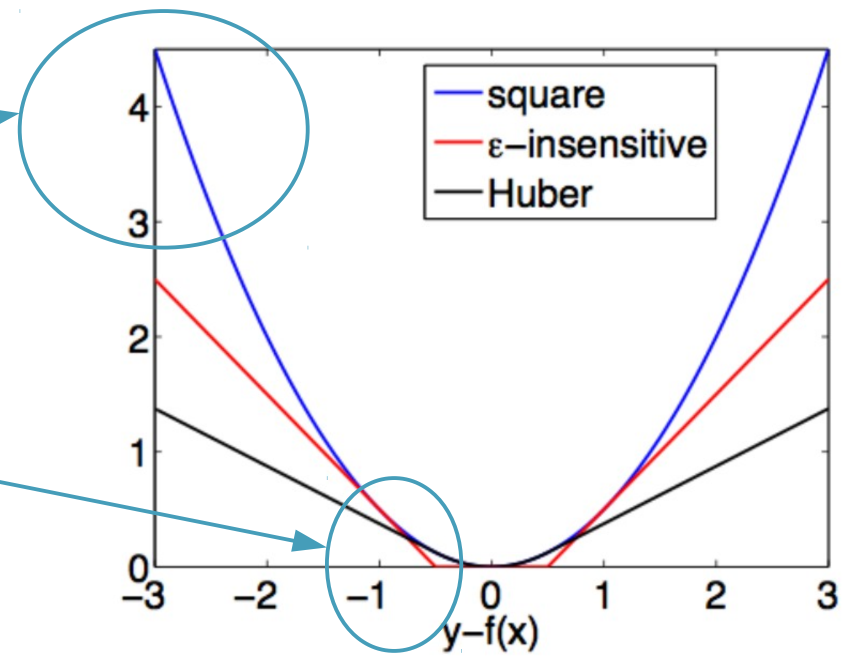


Losses for regression

- **Square loss:** $L(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$
- **ε-insensitive loss:** $L(f(\mathbf{x}), y) = (|f(\mathbf{x}) - y| - \varepsilon)_+$

square loss:
dominated by outliers

ε-insensitive loss:
non smooth



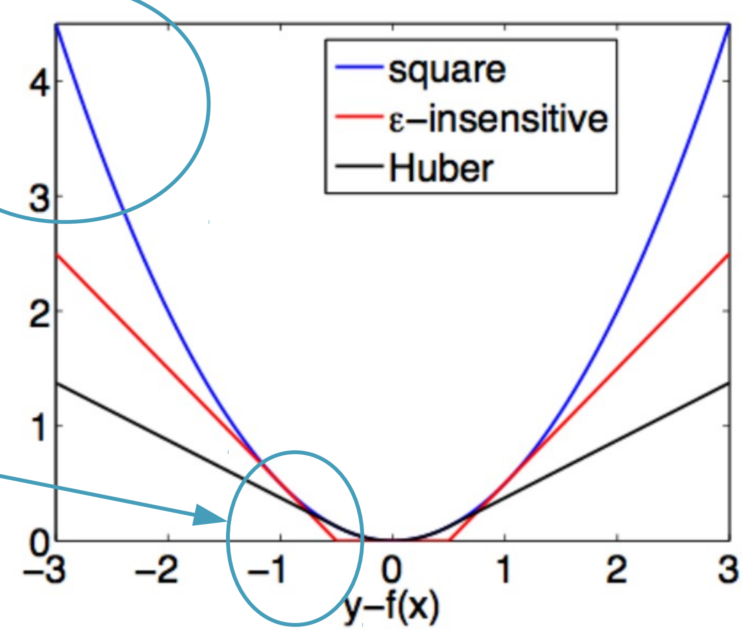
Losses for regression

- **Square loss:** L(f(**x**), y) = (f(**x**) – y)²
- **ε-insensitive loss:** L(f(**x**), y) = (|f(**x**) – y|– ε)₊
- **Huber loss:** mix of linear and quadratic

$$L_\delta(f(\boldsymbol{x}), y) = \begin{cases} \frac{1}{2}\left(y - f(\boldsymbol{x})\right)^2 & \text{if } |y - f(\boldsymbol{x})| \le \delta \\ \delta |y - f(\boldsymbol{x})| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

square loss:
dominated by outliers

ε-insensitive loss:
non smooth



Empirical risk minimization (ERM)

- **Loss:** L(f(**x**), y) small when f(**x**) predicts y well
- **Expected risk:**

$$R = \mathbb{E}[L(f(\boldsymbol{x}), y)]$$

- **Empirical risk:**

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n} L(f(\boldsymbol{x}^i), y^i)$$

- The **ERM estimator** of the functional class F is the solution, when it exists, of:

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} R_n(f)$$

Solving ERM

- There can sometimes be an **explicit analytical solution**
- Otherwise: **convex optimization** (if the loss function is convex in f)
- **Limits of ERM:**
  - **ill-posed**
  - **not statistically consistent**

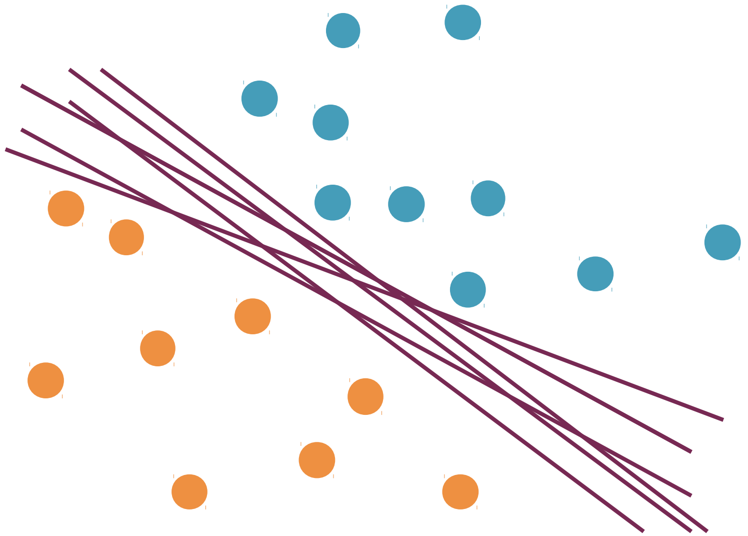
  This is particularly true in **high dimension.**

ERM is ill-posed

- **Well-posed problems** (Hadamard):

  Mathematical models of physical phenomena such that
  - a solution exists;
  - the solution is unique;
  - the solution's behavior changes continuously with the initial conditions.

- It can be that **an infinite number of solutions minimize the empirical risk** to zero.



ERM is not statistically consistent

- **Statistical consistency:** Estimator θ_N of θ that converges in probability towards θ as N increases.

$$\forall \epsilon > 0 \quad \lim_{N \to \infty} Pr(|\theta_N - \theta| \geq \epsilon) = 0$$

- From the **law of large numbers**,

$$\forall f \in \mathcal{F}, \quad R_N(f) \xrightarrow[N \to \infty]{} R(f)$$

  but this isn't enough to guarantee that minimizing $R_N(f)$ gives a good estimator of the minimizer of R(f).

- Vapnik showed that this is only true if the capacity of F is “not too large”.

Maximum likelihood criterion

- Consider **equal priors** P(C=1) = P(C=0)
- Consider the **0/1 loss function**

?

?

$$\Lambda(x) = \frac{p(x|C=1)}{p(x|C=0)} >^? \boxed{\frac{(\lambda_{10} - \lambda_{00})}{(\lambda_{01} - \lambda_{11})}} \boxed{\frac{P(C=0)}{P(C=1)}}$$

Maximum likelihood criterion

- Consider **equal priors** P(C=1) = P(C=0)
- Consider the **0/1 loss function**

$$\Lambda(x) = \frac{p(x|C=1)}{p(x|C=0)} >^{?} \overset{=1\ \text{(0/1 loss)}}{\boxed{\frac{(\lambda_{10}-\lambda_{00})}{(\lambda_{01}-\lambda_{11})}}}\ \overset{=1\ \text{(equal priors)}}{\boxed{\frac{P(C=0)}{P(C=1)}}}$$

Maximum likelihood criterion

- Consider **equal priors** P(C=1) = P(C=0)
- Consider the **0/1 loss function**
- Bayes decision rule seeks to maximize P(x|C=c) and is hence called the **Maximum Likelihood criterion**

Decision rule:

If $\Lambda_{\mathbf{ML}}(x) > 1$ then choose C=1 else choose C=0

$$\Lambda_{\mathrm{ML}}(x) = \frac{p(x|C=1)}{p(x|C=0)}$$

=1 (equal priors)

=1 (0/1 loss)

$$\Lambda(x) = \frac{p(x|C=1)}{p(x|C=0)} >^{?} \boxed{\frac{(\lambda_{10} - \lambda_{00})}{(\lambda_{01} - \lambda_{11})}}\boxed{\frac{P(C=0)}{P(C=1)}}$$

Multivariate classification: Naive Bayes

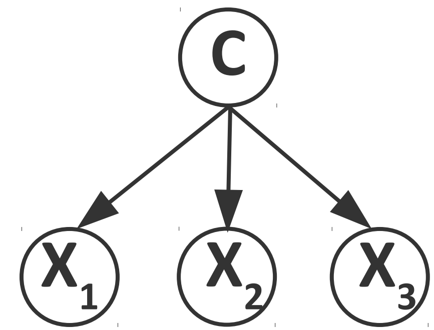
Naive Bayes

- Multivariate classification: **x** is multidimensional
- Assume the variables $x_1, x_2, \ldots x_p$ are **conditionally independent**: $p(x_{j_1}|C_k, x_{j_2}) = p(x_{j_1}|C)$

Graphical representation

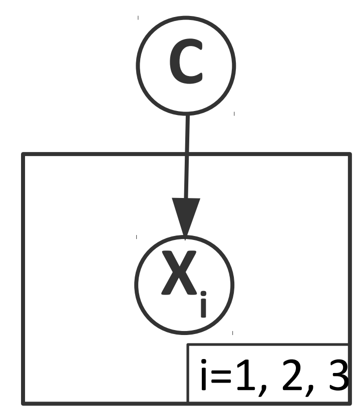
- We can use a graph to represent **conditional independence**:
  - arc from C to X_j means the distribution of X_j **depends** on C
  - no arc between X_{j1} and X_{j2} means that X_{j1} and X_{j2} are **independent given C**:

$$p(x_{j_1}|C_k, x_{j_2}) = p(x_{j_1}|C).$$



- A **plate** represents repeated structure:

  all X_j inside the same plate follow the same probability distribution.



Naive Bayes

- Multivariate classification: x is multidimensional
- Assume the variables x_1, x_2, … x_p are **conditionally independent**: $p(x_{j_1}|C_k, x_{j_2}) = p(x_{j_1}|C)$

$$P(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)P(C_k)}{\sum_{k=1}^{K} p(\boldsymbol{x}|C_k)P(C_k)}$$

$$p(x_1, \ldots, x_p|C_k) = p(x_1|C_k)p(x_2|C_k)\ldots p(x_p|C_k)$$

Hence:

$$P(C_k|x_1, \ldots, x_p) = \frac{1}{Z}P(C_k)p(x_1|C_k)p(x_2|C_k)\ldots p(x_p|C_k)$$

scaling factor, independent of C_k

Maximum a posteriori estimation

- **MAP decision rule:** pick the hypothesis that is most probable
- For Naive Bayes:

$$\hat{y} = \arg \max_{k=1,\ldots,K} p(C_k) \prod_{i=1}^{n} p(\boldsymbol{x}^i | C_k)$$

Naive Bayes spam filtering

- Input: email

  bag of words

  $(x_1, x_2, \ldots, x_p) = (0, 1, \ldots, 0)$

  rich
  CLICK
  viagra

- Output: spam / ham
- Naive Bayes assumption:

  conditional independence

$$P(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)P(C_k)}{\sum_{k=1}^{K} p(\boldsymbol{x}|C_k)P(C_k)}$$

Your Mail-Box has exceeded its storage
Limit CLICK=HERE FILL and Click on FINISH
for to get more space or you wont be able to
send Mail

SPAM

Dear Dr Azencott,

We obtained your contact information from your excellent papers, and would like to know if our company could serve you. Does your current work require the generation of custom monoclonal antibodies? If so, we would be glad to perform this tedious and time-consuming task on your behalf.

SPAM

Dear Dr Azencotte,

Thank you very much for your review of manuscript CHIN-D-15-00031

We greatly appreciate your assistance.

Best wishes,

Samuel Winthrop
Journal of Cheminformatics

NOT SPAM

- **$P(spam|(x_1, x_2, \ldots, x_p))$**

  $= 1/Z\ p(spam)\ p(x_1|spam)\ p(x_2|spam) \ldots p(x_p|spam)$

- **$P(ham|(x_1, x_2, \ldots, x_p))$**

  $= 1/Z\ p(ham)\ p(x_1|ham)\ p(x_2|ham) \ldots p(x_p|ham)$

- **Decision:**

  If $P(spam|(x_1, x_2, \ldots, x_p)) > P(ham|(x_1, x_2, \ldots, x_p))$ then spam else ham

- **Inference:** we need to determine

  $p(spam)$, $p(ham)$, $p(x_j|spam)$, $p(x_j|ham)$

  What are p(spam) and p(ham)?

- **$P(spam|(x_1, x_2, \ldots, x_p))$**

  $= 1/Z\ p(spam)\ p(x_1|spam)\ p(x_2|spam) \ldots p(x_p|spam)$

- **$P(ham|(x_1, x_2, \ldots, x_p))$**

  $= 1/Z\ p(ham)\ p(x_1|ham)\ p(x_2|ham) \ldots p(x_p|ham)$

- **Decision:**

  If $P(spam|(x_1, x_2, \ldots, x_p)) > P(ham|(x_1, x_2, \ldots, x_p))$ then spam else ham

- **Inference:** we need to determine

  $p(spam)$, $p(ham)$, $p(x_j|spam)$, $p(x_j|ham)$

frequency of spam in the training data

- **$P(spam|(x_1, x_2, \ldots, x_p))$**

  $= 1/Z\ p(spam)\ p(x_1|spam)\ p(x_2|spam) \ldots p(x_p|spam)$
- **$P(ham|(x_1, x_2, \ldots, x_p))$**

  $= 1/Z\ p(ham)\ p(x_1|ham)\ p(x_2|ham) \ldots p(x_p|ham)$
- **Decision:**

  If $P(spam|(x_1, x_2, \ldots, x_p)) > P(ham|(x_1, x_2, \ldots, x_p))$ then spam else ham
- **Inference:** we need to determine

  $p(spam)$, $p(ham)$, $p(x_j|spam)$, $p(x_j|ham)$

frequency of spam in the training data

- **Bernouilli Naive Bayes:**
  - Each email is the outcome of p Bernouilli trials
  - **Naive assumption:** the trials are independent

    word co-occurences in a category aren't independent

    still, independence assumptions can give good results

$$p(x_j|\text{spam}) = {p_j}^{x_j}(1 - p_j)^{(1-x_j)}$$

- S = # spams in train set
- Sj = # spams containing word j in train set

  - **Direct estimate of p_j:** p_j = Sj / S
  - **What happens if a word is never seen?**

- **Bernouilli Naive Bayes:**
  - Each email is the outcome of p Bernouilli trials
  - **Naive assumption:** the trials are independent

    word co-occurences in a category aren't independent

    still, independence assumptions can give good results

$$p(x_j|\text{spam}) = {p_j}^{x_j}(1 - p_j)^{(1-x_j)}$$

- S = # spams in train set
- Sj = # spams containing word j in train set

  - **Direct estimate of p_j:** p_j = Sj / S
  - **Laplace-smoothed estimate of p_j:** p_j = (Sj + 1) / (S + 2)

For a word that's not in the training set now $p_{j=}0.5$ instead of 0

- $P(\text{spam} | (x_1, x_2, \ldots, x_p))$

  $= 1/Z\ p(\text{spam})\ p(x_1 | \text{spam})\ p(x_2 | \text{spam}) \ldots p(x_p | \text{spam})$
- $P(\text{ham} | (x_1, x_2, \ldots, x_p))$

  $= 1/Z\ p(\text{ham})\ p(x_1 | \text{ham})\ p(x_2 | \text{ham}) \ldots p(x_p | \text{ham})$
- **Decision:**

  If $P(\text{spam} | (x_1, x_2, \ldots, x_p)) > P(\text{ham} | (x_1, x_2, \ldots, x_p))$ then spam else ham
- **Inference:**

  $p(\text{spam})$, $p(\text{ham})$, $p(x_j | \text{spam})$, $p(x_j | \text{ham})$

frequency of spam in the training data

Bernouilli Naive Bayes: $p_j^{x_j}(1-p_j)^{(1-x_j)}$

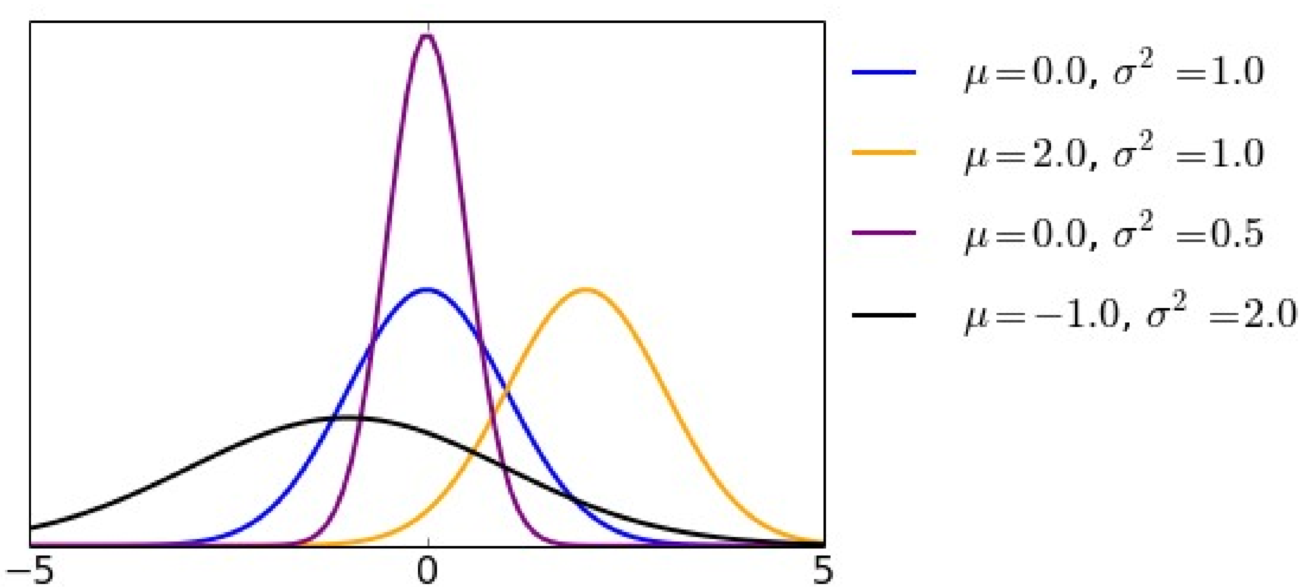
p_j = $(1 + Sj) / (2 + S)$
S = # spams in train set
Sj = # spams with word j in train set

Gaussian naive Bayes

- Assume

  $p(x_j|C_k)$ **univariate Gaussian**

$$p(x_j|C_k) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x_j-\mu)^2/(2\sigma^2)}$$



Bayesian model selection

- **Priors on model:** p(model)

$$p(\text{model}|\text{data}) = \frac{p(\text{data}|\text{model})p(\text{model})}{p(\text{data})}$$

- Regularization ≡ prior that favors simpler models.
- Take the log

$$\log p(\text{model}|\text{data}) = \underbrace{\log p(\text{data}|\text{model})}_{\equiv \text{ training error}} + \underbrace{\log p(\text{model})}_{\equiv \text{ model complexity}} - c$$

- MAP similar to minimizing

E' = empirical error + **λ** model complexity

Summary

prior likelihood

$$P(C|\boldsymbol{x}) = \frac{P(C)\,p(\boldsymbol{x}|C)}{p(\boldsymbol{x})}$$

posterior evidence

- **Bayes decision rule ≡ likelihood ratio test**

  choose the most probable class, given evidence (data) and prior belief.

- Equivalent to **minimizing Bayes risk**

  usually achieved approximately through **empirical risk minimization** (not equivalent!!)

- For the 0/1 loss, equivalent to **maximizing the posterior.**

- For the 0/1 loss and equal priors (uniform prior), equivalent to **maximizing the likelihood.**

Further reading

- Ghahramani, Z. (2015). **Probabilistic machine learning and artificial intelligence.** *Nature* 521, 452-459.
- Paul Graham, **A plan for spam** `http://www.paulgraham.com/spam.html`

Jupyter

- **Web application**
- **Notebooks:** webpages that contain
  - text (explanations, comments, conclusions...)
  - live code
  - equations
  - visualizations.
- Instructions for labs:
  - Get a **local version** of the notebook
  - Open the .ipynb file in Jupyter

```
> cd ma2823_2016/lab_notebooks
> jupyter notebook
```

GitHub

- **Version control**
  - Multiple people use/edit the same file(s) at the same time
  - Grownup version of `mydoc_v2_chloe_new.txt`
- **For our labs**
  - Instead of downloading the latest version of `ma_2823`, making sure not to overwrite work from the previous weeks…

    ```
    > git pull
    ```

    automatically updates the files that need updating.
  - **Fork:** to version control your own work

    `https://help.github.com/articles/fork-a-repo/`

`https://github.com/chagaz/ma2823_2016/blob/master/lab_notebooks/Lab%202%202016-09-21%20Introduction%20to%20scikit-learn.ipynb`