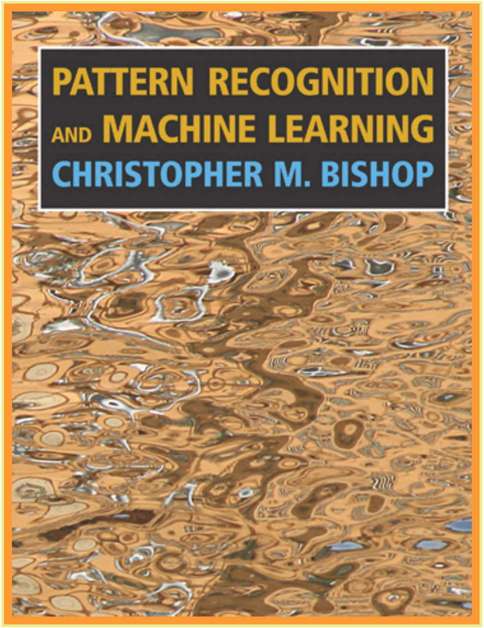
Cybernetics and Artificial Intelligence (2017)

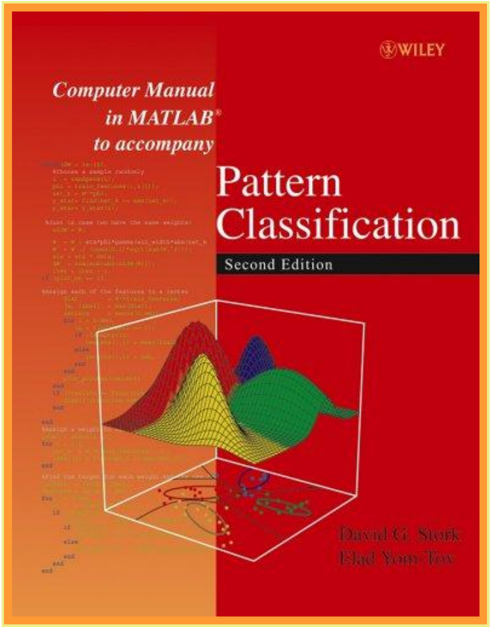
Probabilistic Classification and Decision Making

Dept. of Cybernetics
Czech Technical University in Prague
Matěj Hoffmann, Zdeněk Straka
Thanks to: Daniel Novák, Filip Železný

Literature, demos

- Duda, Hart, Stork: Pattern Classification `http://www.crc.ricoh.com/~stork/DHS.html`
- Ch. Bishop, Pattern Recognition and Machine Learning `http://research.microsoft.com/en-us/um/people/cmbishop/prml/`
- Kotek, Vysoký, Zdráhal: Kybernetika 1990
- Classification toolbox `http://stuff.mit.edu/afs/sipb.mit.edu/user/arolfe/matlab/`
- Statistical Pattern Recognition Toolbox `http://cmp.felk.cvut.cz/cmp/software/stprtool/`



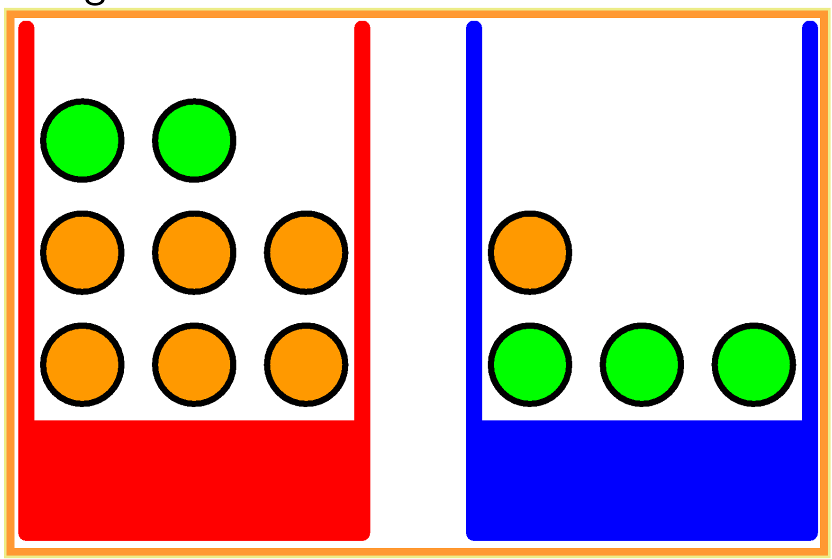


(Re)introducing probability

- Markov Decision Processes - uncertainty about outcome of actions
- Now: uncertainty may be also associated with states
  - Different states may have different *prior* probabilities
  - The states $s \in S$ may not be directly observable
    - They need to be inferred from features $x \in X$
- This is addressed by the rules of probability *(such as Bayes theorem)* and leads on to
  - Bayesian classification
  - Bayesian decision making

Example for illustration – picking fruits from boxes [Bishop (2006) - Ch. 1.2]

- red box: 2 apples, 6 oranges
- blue box: 3 apples, 1 orange



- Scenario: Pick a box (say red box in 40% cases), then pick a fruit at random
- Questions:
  - 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?

Rules of probability and notation

- random variables X, Y
- x_i where $i = 1, ..., M$ – values taken by variable X
- y_j where $j = 1, ..., L$ – values taken by variable Y
- $p(X = x_i, Y = y_i)$ – probability that X takes the value x_i, Y takes y_i
  - *joint probability*
- $p(X = x_i)$ – probability that X takes the value x_i
- *Sum rule of probability*:
  - $p(X = x_i) = \sum_{j=1}^{L} p(X = x_i, Y = y_j)$
  - $p(X = x_i)$ is sometimes called *marginal probability* – obtained by marginalizing / summing out the other variables
  - general rule, compact notation: $p(X) = \sum_Y p(X, Y)$
    - to denote a distribution over random variables

Rules of probability and notation 2

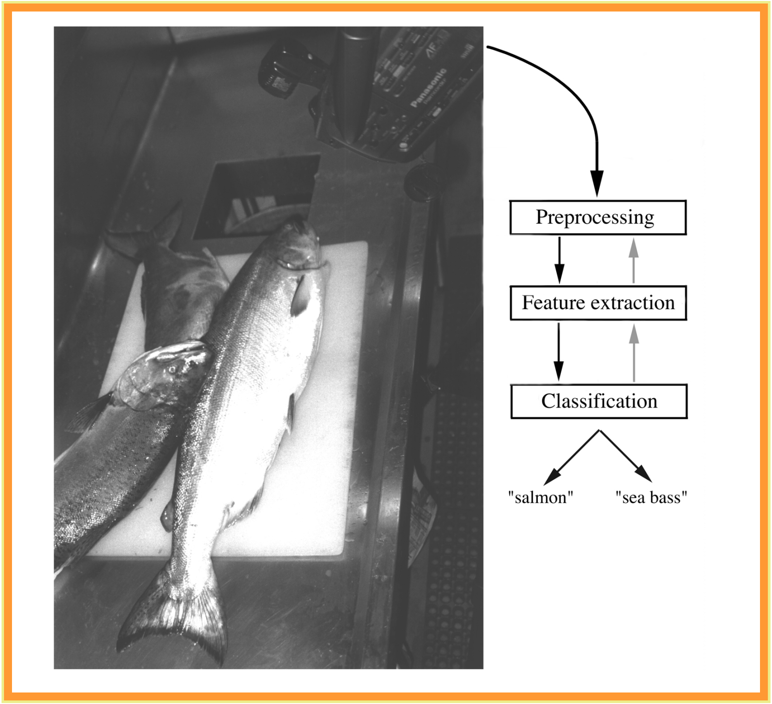
- *Conditional probability*: $p(Y = y_j | X = x_i)$
- *Product rule of probability*
  - $p(X = x_i, Y = y_i) = p(Y = y_j | X = x_i) p(X = x_i)$
  - general rule, compact notation: $p(X, Y) = p(Y|X)p(X)$
    - to denote a distribution over random variables
- *Bayes theorem*
  - from $p(X, Y) = p(Y, X)$ and product rule

$$\boxed{p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}}$$

$$\boxed{posterior = \frac{likelihood \times prior}{evidence}}$$

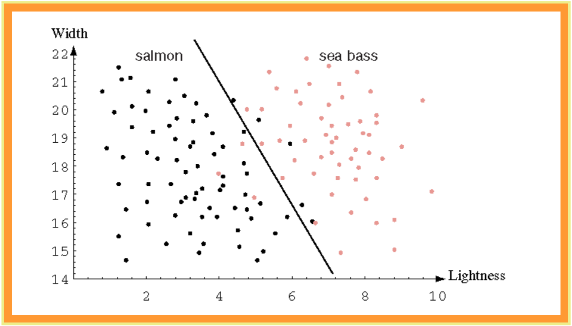
- back to fruits and boxes

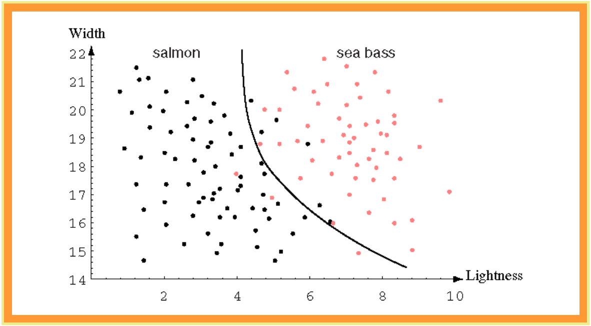
Motivation example – fish classification [Duda, Hart, Stork: Pattern Classification]

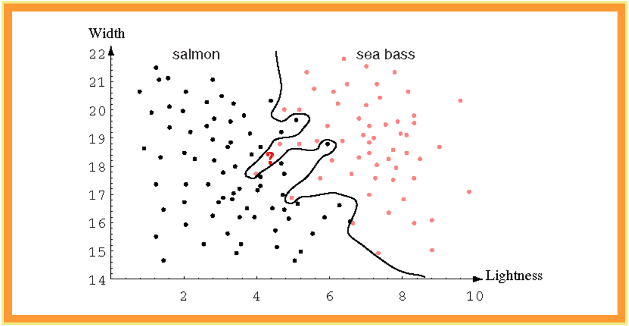


- Factory for fish processing
- 2 classes:
  - salmon
  - sea bass
- Features: length, width, lightness etc. from a camera

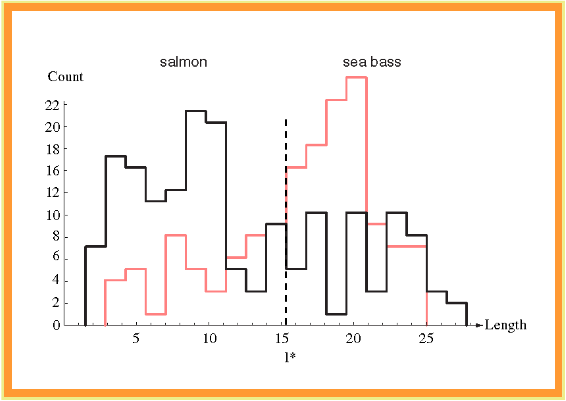
Fish classification in feature space

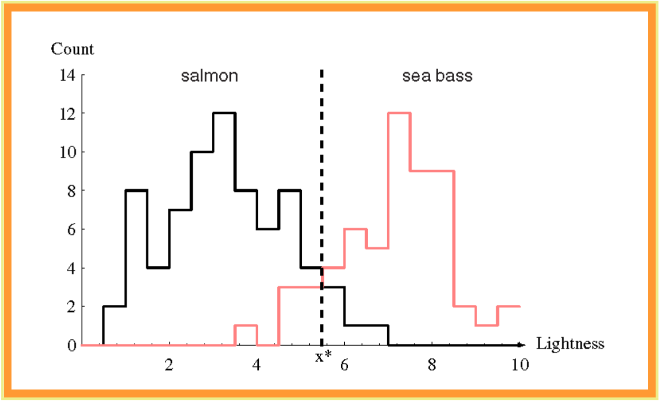






- Linear, quadratic, k-nearest neighbor classifier (next lecture)





- Feature frequency per class shown using histograms
- Classification errors due to histogram overlap

Fish – classification using probability

- direct classification in feature space may not be optimal
  - ignoring statistical dependendencies that may be available
- E.g., what if 95% of fish are salmon?
  - Prior may become more relevant than features
- Notation for classification problem
  - Classes $s_i \in S$ (e.g., salmon, sea bass)
  - Features $x_i \in X$ or feature vectors ($\vec{x}_i$) (also called attributes)
- Optimal classification of $\vec{x}$:

$$\delta^*(\vec{x}) = \arg\max_j p(s_j|\vec{x})$$

- We thus choose the most probable class for a given feature vector.
- Both likelihood and prior are taken into account – recall Bayes rule:

$$\boxed{p(s_j|x) = \frac{p(x|s_j)p(s_j)}{p(x)}}$$

Bayes classification in practice

- Usually we **are not given** $P(s|\vec{x})$
- It has to be estimated from already classified examples – training data
- For discrete $\vec{x}$, **training examples** $(\vec{x}_1, s_1), (\vec{x}_2, s_2), \ldots (\vec{x}_l, s_l)$
  - so-called i.i.d (independent, identically distributed) multiset
  - every $(\vec{x}_i, s)$ is drawn independently from $P(\vec{x}, s)$
- Without knowing anything about the distribution, a non-parametric estimate:

$$P(s|\vec{x}) \approx \frac{\#\text{ examples where } \vec{x}_i = \vec{x} \text{ and } s_i = s}{\#\text{ examples where } \vec{x}_i = \vec{x}}$$

- This is hard in practice:
  - To reliably estimate $P(s|\vec{x})$, the number of examples grows exponentially with the number of elements of $\vec{x}$.
    - e.g. with the number of pixels in images
    - curse of dimensionality
    - denominator often 0
  - Bayes classification provides a lower bound on classification error, but that is usually not achievable because $P(s|\vec{x})$ is not known.

Naïve Bayes classification

- For efficient classification we must thus rely on additional assumptions.
- In the **exceptional case** of **statistical independence** between $\vec{x}$ components for each class s it holds

$$\boxed{P(\vec{x}|s) = P(x(1)|s) \cdot P(x(2)|s) \cdot \ldots}$$

- Use simple Bayes law and maximize:

$$P(s|\vec{x}) = \frac{P(\vec{x}|s)P(s)}{P(\vec{x})} = \frac{P(s)}{P(\vec{x})}P(x(1)|s) \cdot P(x(2)|s) \cdot \ldots =$$

- No combinatorial curse in estimating $P(s)$ and $P(x(i)|s)$ separately for each i and s.
- No need to estimate $P(\vec{x})$. (Why?)
- N.B. $P(s)$ may be provided apriori.
- **naïve** $=$ when used despite statistical dependence btw. $x(i)$'s.

Decision making under uncertainty

- An important feature of intelligent systems
  - make the best possible decision
  - in **uncertain** conditions.
- **Example**: Take a tram OR subway from A to B?
  - Tram: timetables imply a quicker route, but adherence uncertain.
  - Subway: longer route, but adherence almost certain.
- **Example**: where to route a letter with this ZIP?

  - 15700? 15706? 15200? 15206?
- What is the **optimal decision**?
- Both examples fall into the same framework.

Example [Kotek, Vysoký, Zdráhal: Kybernetika 1990]

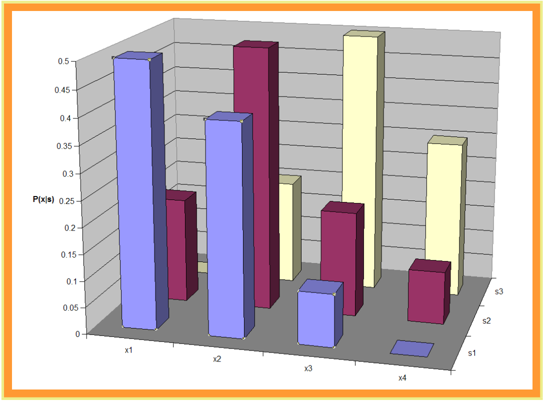
- *Wife coming back from work. Husband pondering what to cook for dinner.*
- 3 dishes (**decisions)** in his repertoire:
  - *nothing* . . . **don't bother cooking** $\Rightarrow$ no work but makes wife upset
  - *pizza* . . . **microwave a frozen pizza** $\Rightarrow$ not much work but won't impress
  - *g.T.c.* . . . **general Tso's chicken** $\Rightarrow$ will make her day, but very laborious.
- Husband quantifies the degree of hassle incurred by the individual options. This depends on how his wife is feeling on her way home. Her state of mind is an **uncertain state**. Let us distinguish her mood:
  - *good* . . . wife is feeling **good**.
  - *average* . . . wife **average** mooded.
  - *bad* . . . wife **bad** mooded.
- For each of the 9 possible situation (3 possible decisions $\times$ 3 possible states) the hassle is quantified by a **loss function** $l(d, s)$:

$l(s,d)$	$d =$ *nothing*	$d =$ *pizza*	$d =$ *g.T.c.*
$s =$ *good*	0	2	4
$s =$ *average*	5	3	5
$s =$ *bad*	10	9	6

Example (cont'd)

- Husband tries to estimate wife's state of mind through an experiment. He tells her he accidentally overtaped their wedding video and observes her reaction
- Anticipates 4 possible reactions:
  - *mild* . . . all right, we keep our memories.
  - *irritated* . . . how many times do I have to tell you....
  - *upset* . . . Why did I marry this guy?
  - *alarming* . . . silence
- The reaction is a measurable **attribute** ("feature") of the state of mind.
- From experience, the husband knows how individual reactions are probable in each state of mind; this is captured by the joint distribution $P(x, s)$.

$P(x,s)$	$x =$ *mild*	$x =$ *irritated*	$x =$ *upset*	$x =$ *alarming*
$s =$ *good*	0.35	0.28	0.07	0.00
$s =$ *average*	0.04	0.10	0.04	0.02
$s =$ *bad*	0.00	0.02	0.05	0.03



Decision strategy

- **Decision strategy**: a rule selecting a decision for any given value of the measured attribute(s).
- i.e. function $d = \delta(x)$.
- Example of husband's possible strategies:

$\delta(x)$	$x =$ *mild*	$x =$ *irritated*	$x =$ *upset*	$x =$ *alarming*
$\delta_1(x) =$	*nothing*	*nothing*	*pizza*	*g.T.c.*
$\delta_2(x) =$	*nothing*	*pizza*	*g.T.c.*	*g.T.c.*
$\delta_3(x) =$	*g.T.c.*	*g.T.c.*	*g.T.c.*	*g.T.c.*
$\delta_4(x) =$	*nothing*	*nothing*	*nothing*	*nothing*

- Overall, $3^4 = 81$ possible strategies (3 possible decisions for each of the 4 possible attribute values).
- How to define which strategy is best? How to sort them by quality?
- Define the **risk of a strategy** as a mean loss value.

$$\boxed{r(\delta) = \sum_x \sum_s l(s, \delta(x)) P(x, s)}$$

Calculating $r(\delta)$

- Bayes criterion: From two strategies, one with a lower mean risk is better.

$P(x,s)$	$x =$ *mild*	$x =$ *irritated*	$x =$ *upset*	$x =$ *alarming*
$s =$ *good*	0.35	0.28	0.07	0.00
$s =$ *average*	0.04	0.10	0.04	0.02
$s =$ *bad*	0.00	0.02	0.05	0.03

$\delta(x)$	$x =$ *mild*	$x =$ *irritated*	$x =$ *upset*	$x =$ *alarming*
$\delta_1(x) =$	*nothing*	*nothing*	*pizza*	*g.T.c.*
$\delta_2(x) =$	*nothing*	*pizza*	*g.T.c.*	*g.T.c.*
$\delta_3(x) =$	*g.T.c.*	*g.T.c.*	*g.T.c.*	*g.T.c.*
$\delta_4(x) =$	*nothing*	*nothing*	*nothing*	*nothing*

$l(s,d)$	$d =$ *nothing*	$d =$ *pizza*	$d =$ *g.T.c.*
$s =$ *good*	0	2	4
$s =$ *average*	5	3	5
$s =$ *bad*	10	9	6

Bayes optimal strategy

- The **Bayes optimal strategy**: one minimizing mean risk. That is

$$\boxed{\delta^* = \arg\min_{\delta} r(\delta)}$$

- From $P(x, s) = P(s|x)P(x)$ (Bayes rule), we have

$$r(\delta) = \sum_x \sum_s l(s, \delta(x))P(x, s) = \sum_s \sum_x l(s, \delta(x))P(s|x)P(x)$$

$$= \sum_x P(x) \underbrace{\sum_s l(s, \delta(x))P(s|x)}_{\text{Conditional risk}}$$

- The optimal strategy is obtained by minimizing the conditional risk separately for each x:

$$\boxed{\delta^*(x) = \arg\min_{d} \sum_s l(s, d)P(s|x)}$$

Statistical decision making: wrapping up

- **Given**:
  - A set of possible **states**: $\mathcal{S}$
  - A set of possible **decisions**: $\mathcal{D}$
  - A **loss function** $l : \mathcal{D} \times \mathcal{S} \to \Re$
  - The range $\mathcal{X}$ of the **attribute**
  - Distribution $P(x, s)$, $x \in \mathcal{X}, s \in \mathcal{S}$.
- **Define**:
  - **Strategy**: function $\delta : \mathcal{X} \to \mathcal{D}$
  - **Risk of strategy** δ : $r(\delta) = \sum_x \sum_s l(s, \delta(x)) P(x, s)$
- **Bayes problem**:
  - Goal: find the optimal strategy $\delta^* = \arg\min_{\delta \in \Delta} r(\delta)$
  - Solution: $\delta^*(x) = \arg\min_d \sum_s l(s, d) P(s|x)$

A special case - Bayes classification

- Bayes classification is a special case of statistical decision theory:
  - Attribute vector $\vec{x} = (x_1, x_2, \dots)$: pixels # 1, 2, ...
  - **State set $\mathcal{S}$ = decision set $\mathcal{D} = \{0, 1, \dots 9\}$.**
  - State = actual class, Decision = recognized class.
  - Loss function:

$$l(s,d) = \begin{cases} 0, & d = s \\ 1, & d \neq s \end{cases}$$

Mathematical derivation:

$$\delta^*(\vec{x}) = \arg\min_d \sum_s \underbrace{l(s,d)}_{0 \text{ if } d=s} P(s|\vec{x}) = \arg\min_d \sum_{s \neq d} P(s|\vec{x}) = \arg\min_d [1 - P(d|\vec{x})] = \arg\max_d P(d|\vec{x})$$

  - We used equation

$$\sum_{s \neq d} P(s|\vec{x}) + P(d|\vec{x}) = 1$$

- Mean risk = mean classification **error**.