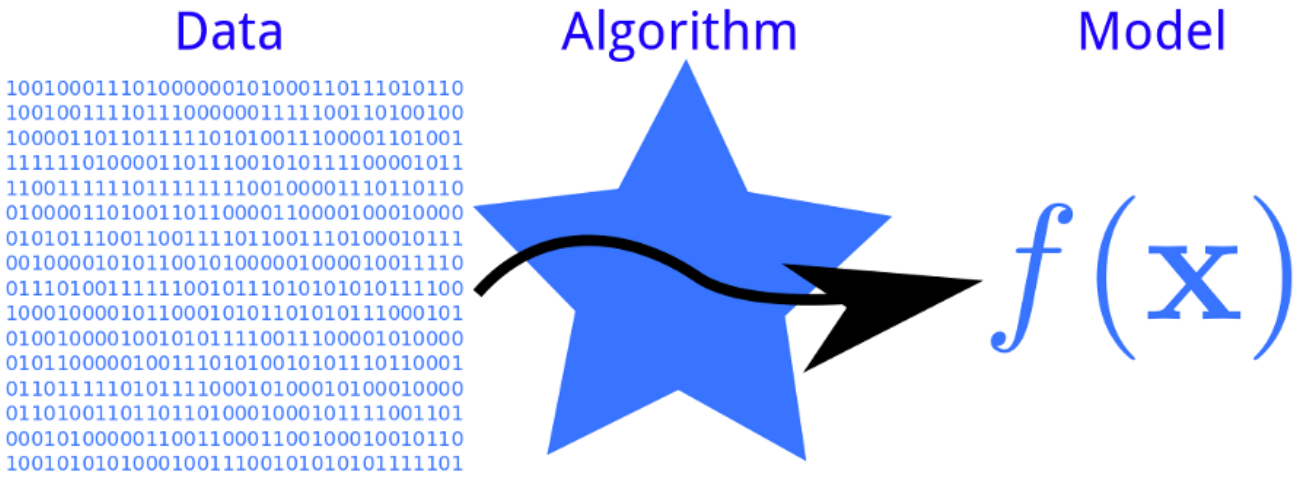
Introduction to Supervised Learning



Lecture 1: Introduction, fundamental concepts

Iasonas Kokkinos

Iasonas.kokkinos@gmail.com

University College London

Course Requirements and Grading

Lab exercises (8/20)

- **Matlab:** easy to get started
- Synthetic data (easy and fast to train)
- Hands-on experience with algorithms
- 3 deliverables, distributed evenly over the module
- 20-30% during practicals, 70-80% at home
- Code of honor: any copying results in zero credit for everyone.

Theory exercises (2/20)

- Close to the end (early December), getting you started for the exam.

Final exam (10)

- Theory questions (judgement-oriented)
- Simulate running algorithms by hand

Meeting hours

Office: 110B, 68-72 Gower street

Meetings hours: Tuesday, 5:00-6:00 pm

Please, use this time

If you need more time for meetings, just ask!

If anything is not clear during the lecture, just ask!

Course support

- TBD
- “You can search for COMPGI20 through Moodle and it should come up in the results and you should be able to access it. The password is ‘supervised’.”

Lecture outline

Introduction to the course

Introduction to Machine Learning

where you learn what you will learn about learning..

Linear Classifiers

Tentative Course Schedule

- 1st week: Introduction, fundamental concepts
- 2nd week: Linear Regression
- 3rd week: Logistic Regression
- 4th week: Support Vector Machines
- 5th week: Ensemble Models (Adaboost, Random Forests)
- 6th week: Unsupervised learning (K-means, PCA, Sparse Coding)
- 7th week: Deep Learning (neural networks, backpropagation, SGD)
- 8th week: Probabilistic modelling (hidden variable models, EM)
- 9th week: Intro to Structured Prediction (Random Fields, Graphical Models)
- 10th week: review and applications

No rush - stop me whenever something is not clear!

Machine learning

Principles, methods, and algorithms for learning and prediction on the basis of past evidence

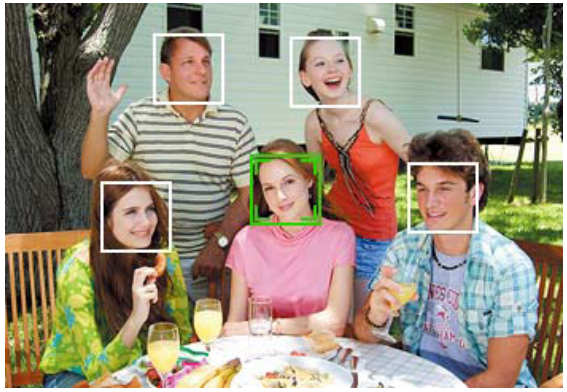
Goal: Machines that learn to perform a task from experience

Why?

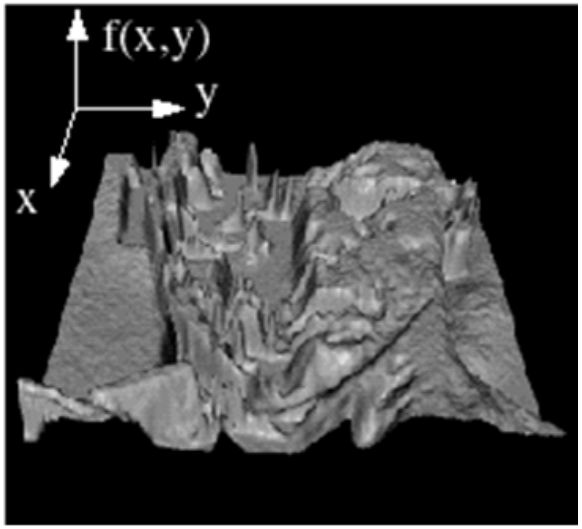
- Crucial component of every intelligent/autonomous system
- Important for a system's adaptability
- Important for a system's generalization capabilities
- Attempt to understand human learning

Computer vision example: face detection

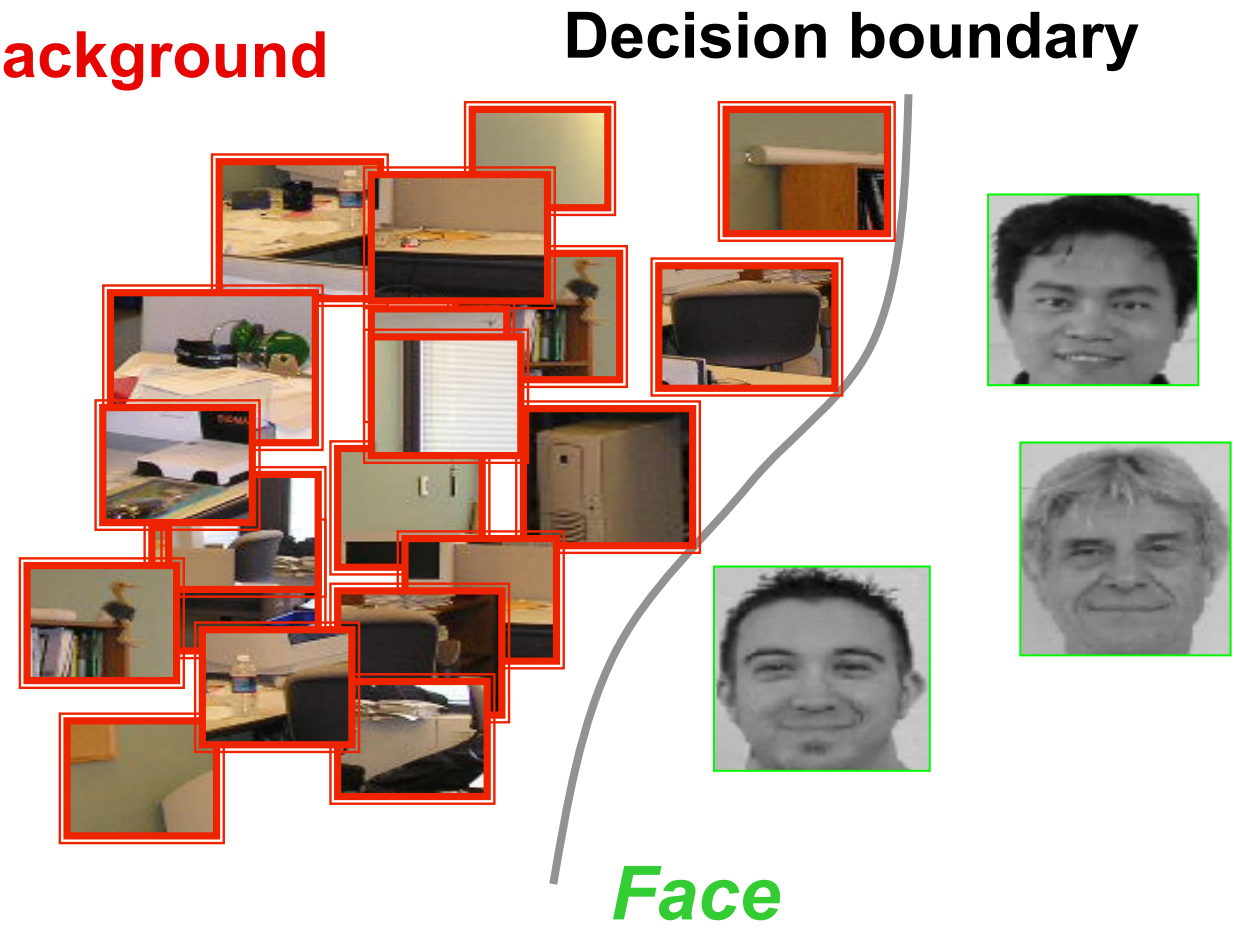
- How do digital cameras detect faces?



- Input to a digital camera: intensity at pixel locations



`Faceness function': classifier



Test time: deploy the learned function

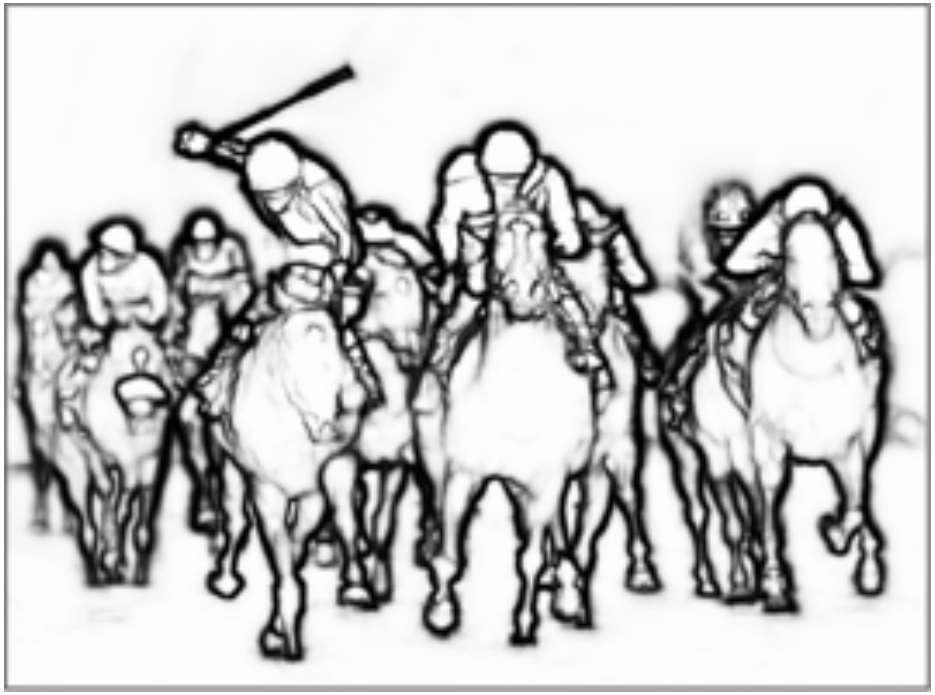
- Scan window over image
  - Multiple scales
  - Multiple orientations
- Classify window as either:
  - Face
  - Non-face

P. Viola & M. Jones: Rapid object detection using a boosted cascade of simple features, CVPR 2001
C. Papageorgiou & T. Poggio: A Trainable system for object detection, IJCV 2000
H. Rowley, S. Baluja & T. Kanade: Neural Network-based face detection, PAMI, 1998

Computer vision, 2016

input

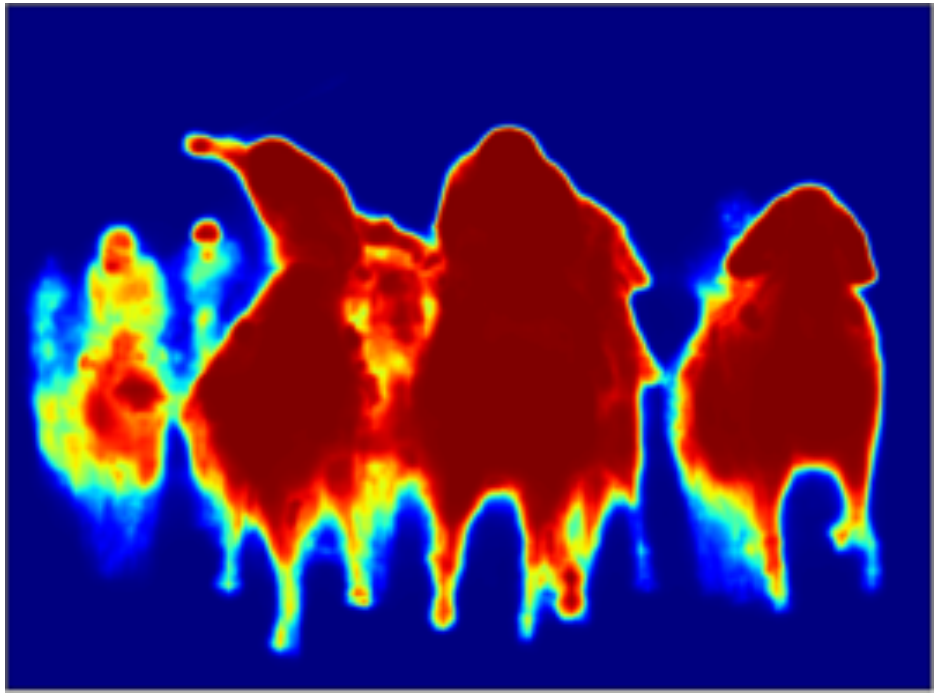
boundary detection



Computer vision, 2016

input

saliency estimation



Computer vision, 2016

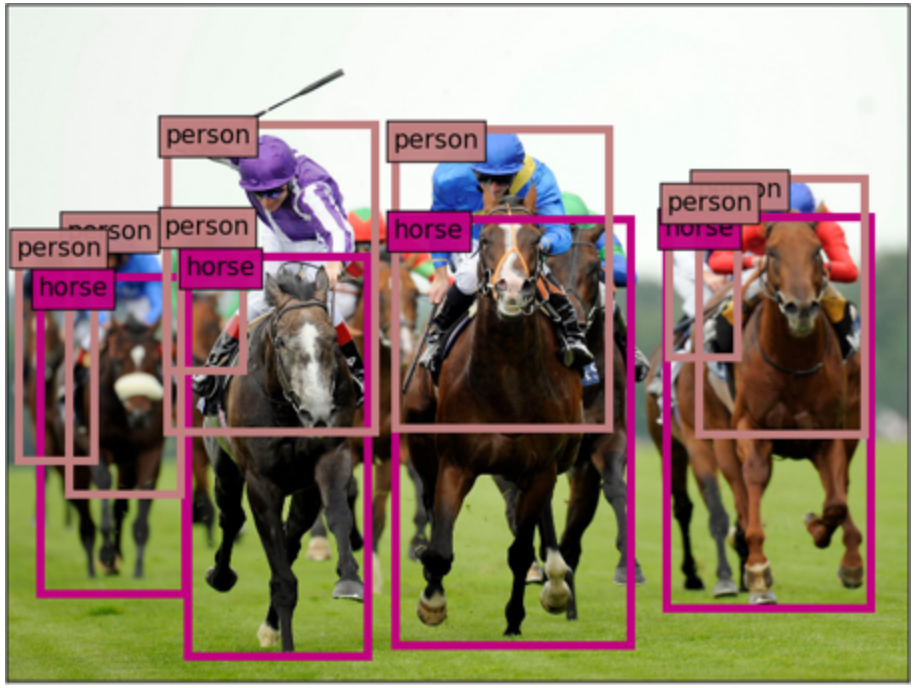
input

normal estimation

Computer vision, 2016

input

object detection



Computer vision, 2016

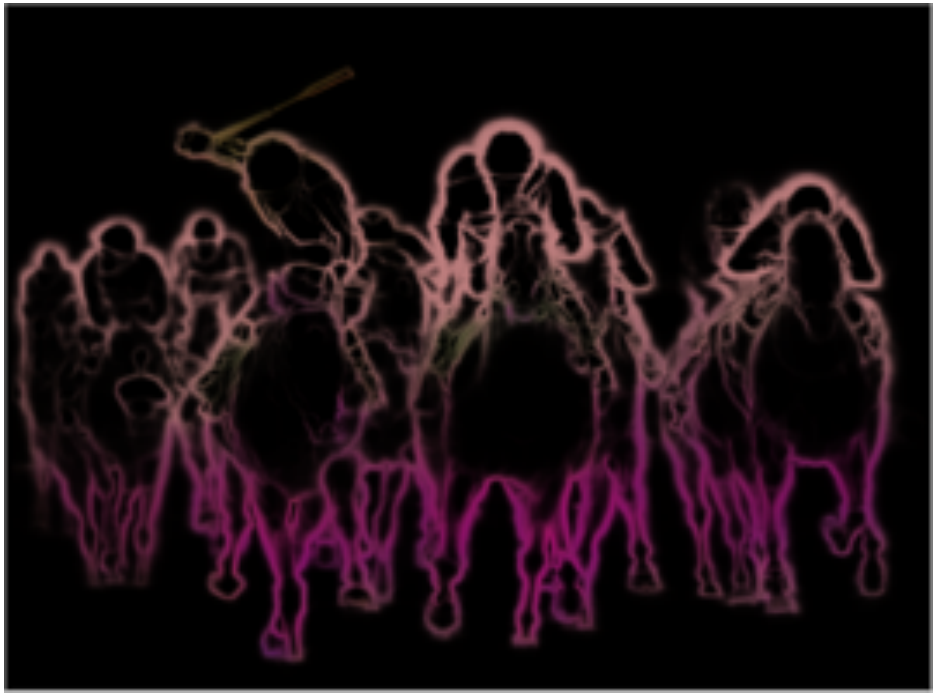
input

semantic segmentation

Computer vision, 2016

input

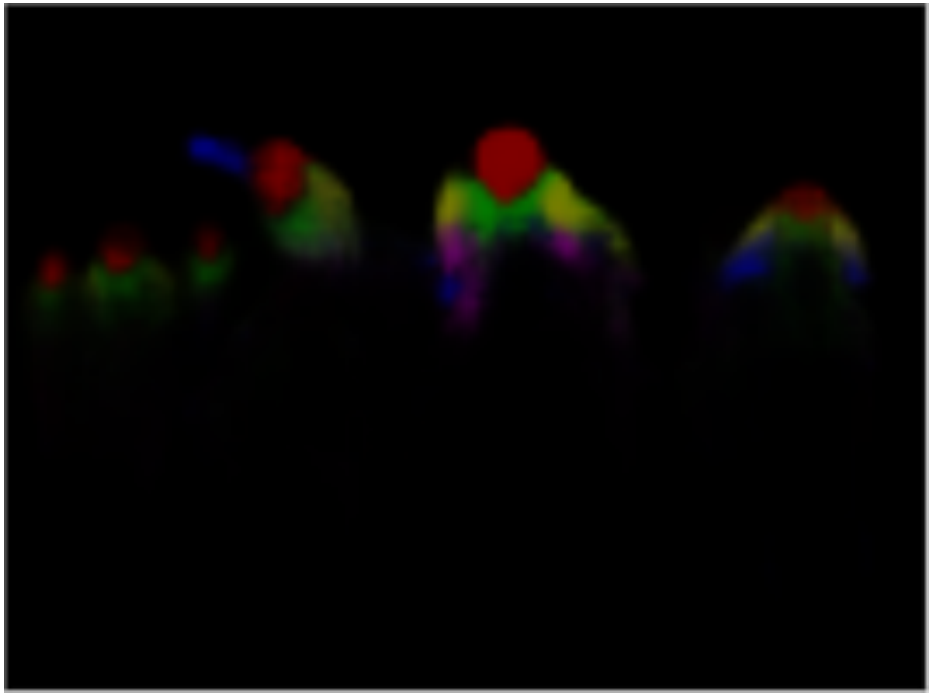
semantic boundary detection



Computer vision, 2016

input

human part detection



http://cvn.ecp.fr/ubernet/

I. Kokkinos, UberNet : Training a ‘Universal’ Convolutional Neural Network for Low-, Mid-, and High-Level Vision using Diverse Datasets and Limited Memory, arxiv, 2016

Machine Learning variants

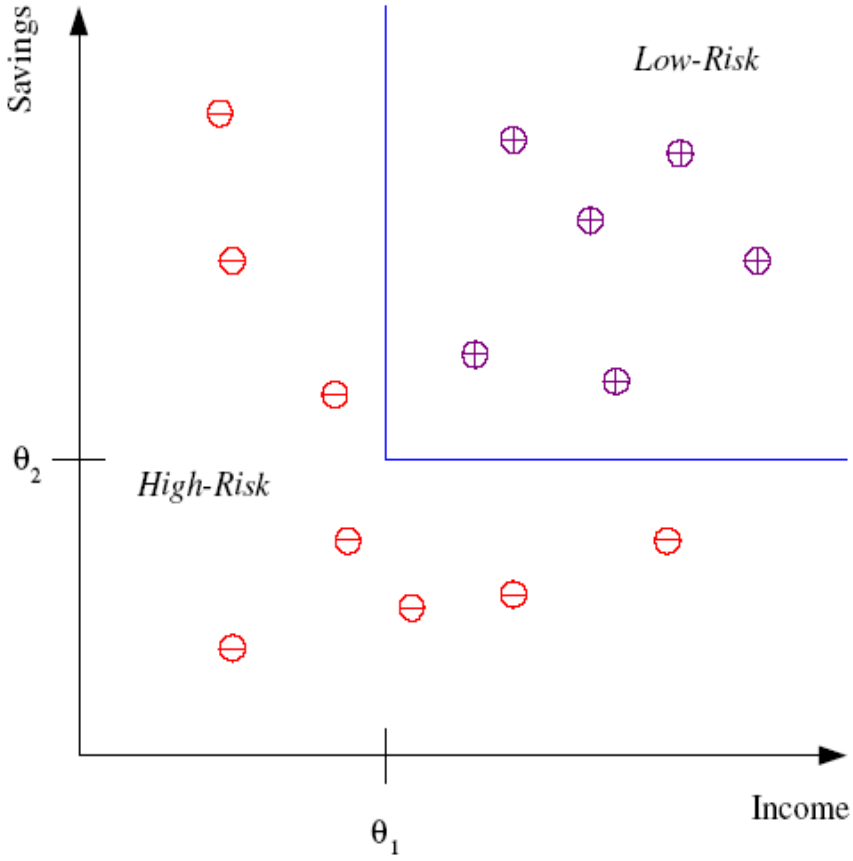
- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimensionality Reduction
- Weakly supervised

  Some data supervised, some unsupervised
- Reinforcement learning

  Supervision: reward for a sequence of decisions

Classification

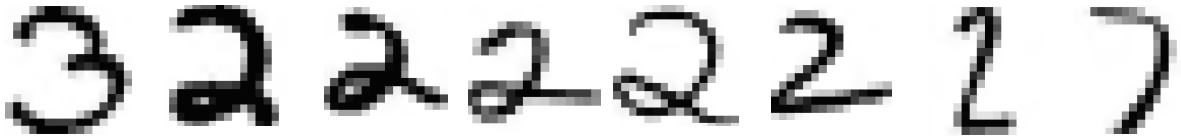
- Based on our experience, should we give a loan to this customer?
  - Binary decision: yes/no



Decision boundary

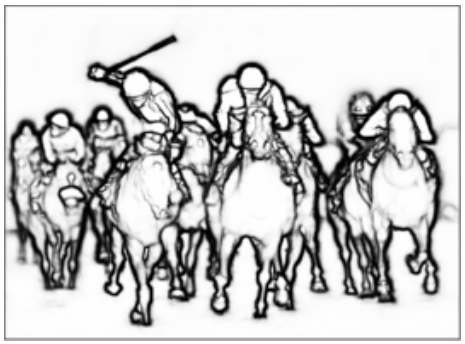
Classification examples

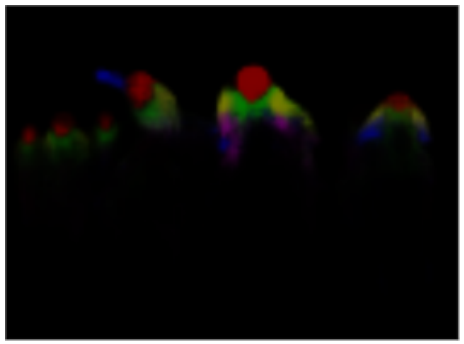
- Digit Recognition

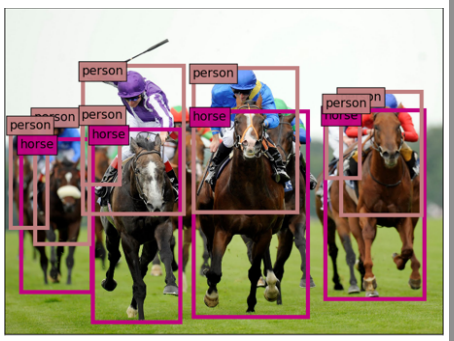


- Spam Detection

- Sliding window classifiers:







Machine Learning variants

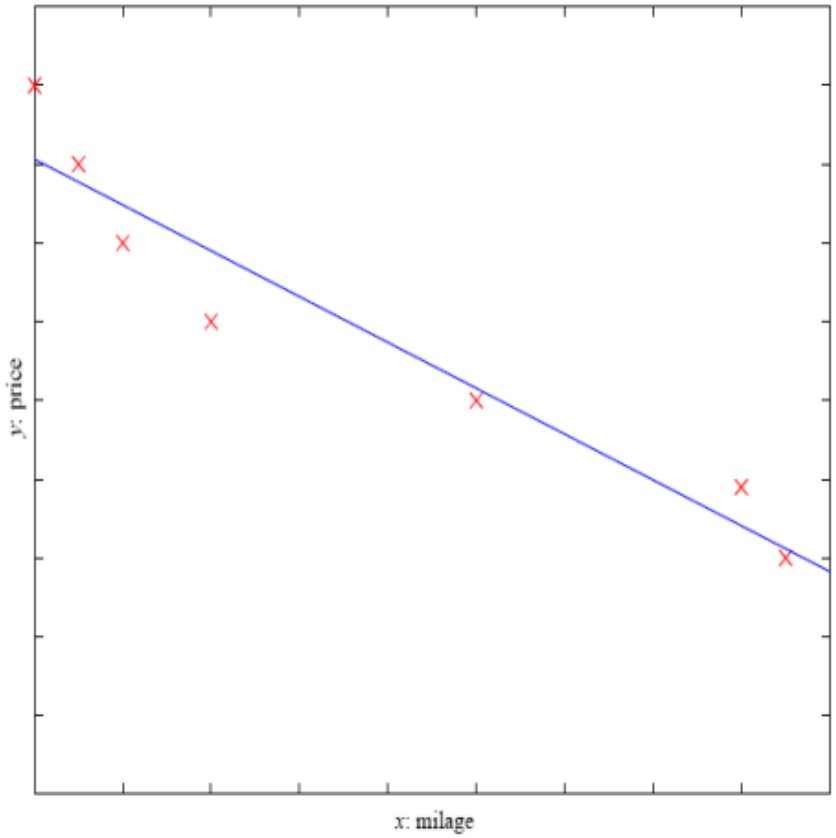
- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimensionality Reduction
- Weakly supervised

  Some data supervised, some unsupervised
- Reinforcement learning

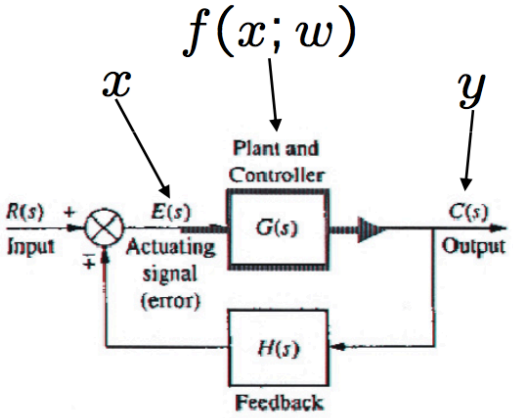
  Supervision: reward for a sequence of decisions

Regression

- Output: Continuous
  - E.g. price of a car based on years, mileage, condition,…

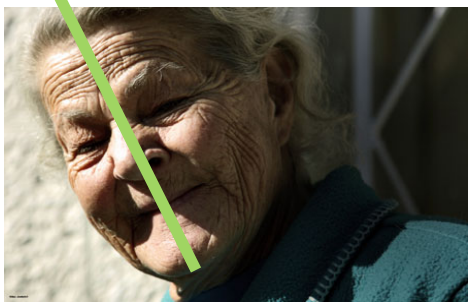


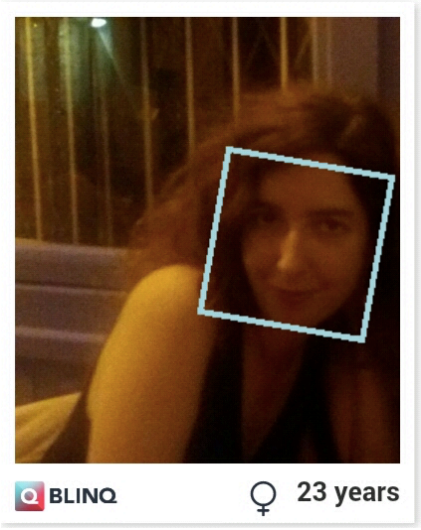
Regression examples



- Typical example: driving a car
  - Outputs: force on pedal, angle of steering wheel

- Estimate face orientation/age





- Surface normal estimation:

Machine Learning variants

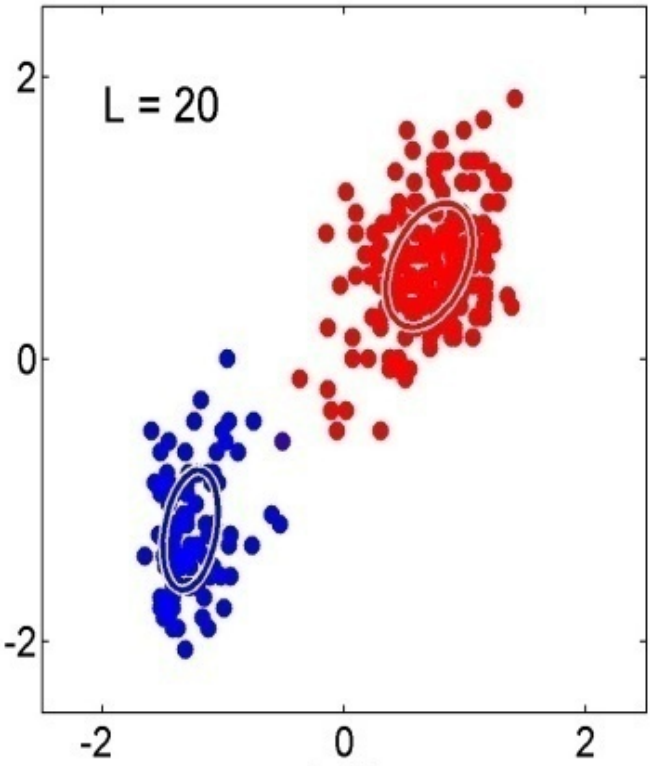
- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimensionality Reduction
- Weakly supervised

  Some data supervised, some unsupervised
- Reinforcement learning

  Supervision: reward for a sequence of decisions

Clustering

- Break a set of data into coherent groups
  - Labels are `invented'



Clustering examples

Text clustering

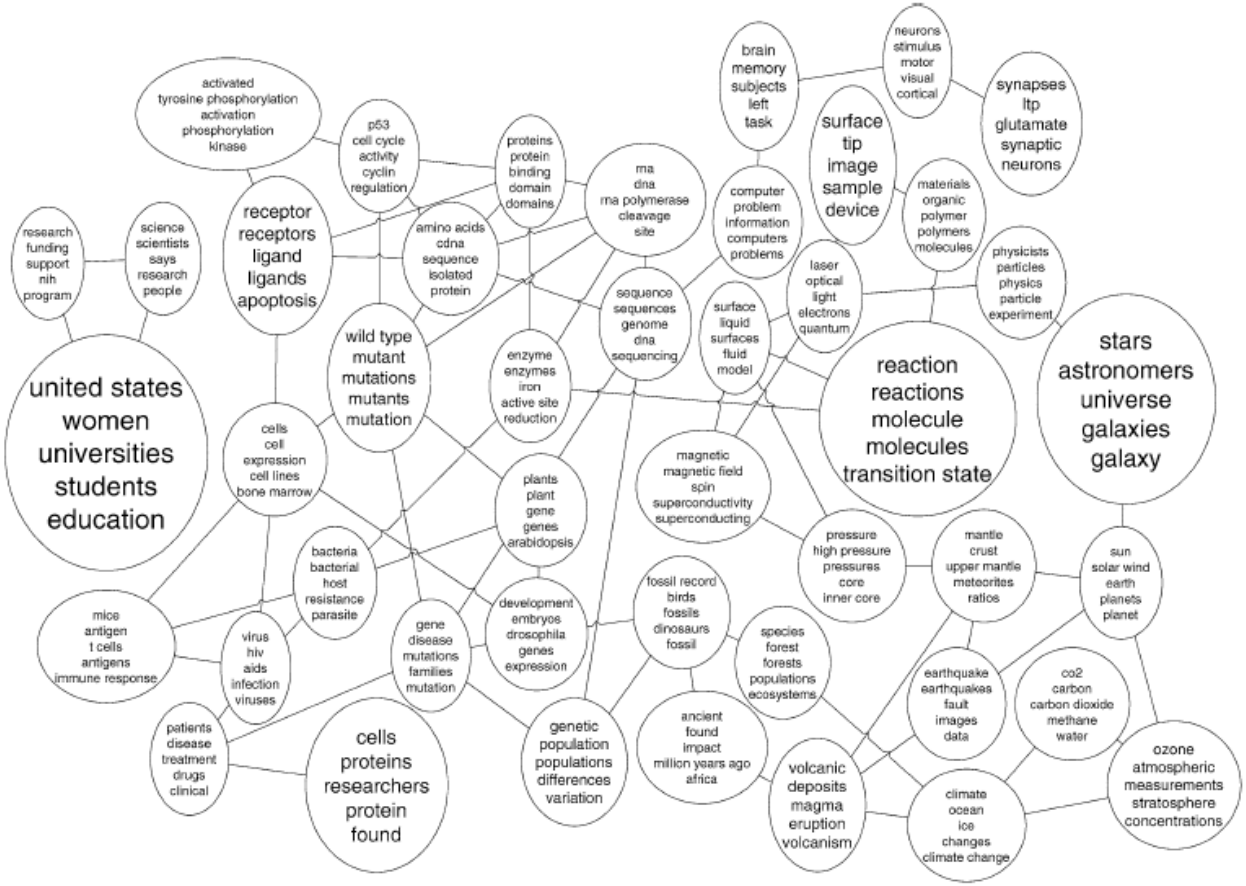


FIG. 2. *A portion of the topic graph learned from* 16,351 *OCR articles from Science* (1990–1999). *Each topic node is labeled with its five most probable phrases and has font proportional to its popularity in the corpus.* (*Phrases are found by permutation test.*) *The full model can be found in* http://www.cs.cmu.edu/~lemur/science/ *and on STATLIB.*

Clustering examples

Image segmentation

Machine Learning variants

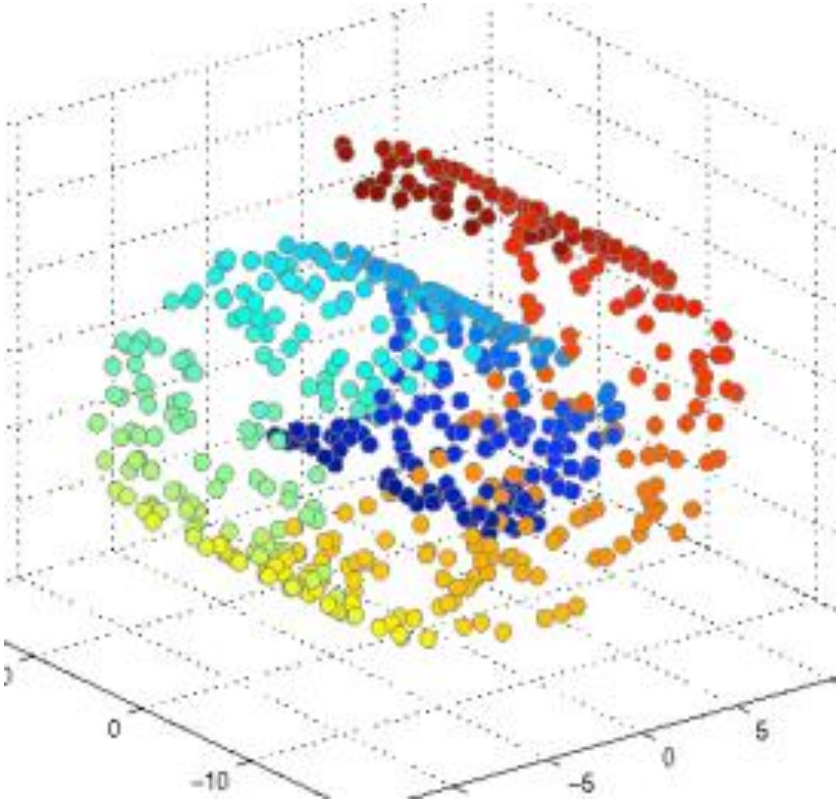
- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimensionality Reduction, Manifold Learning
- Weakly supervised

  Some data supervised, some unsupervised
- Reinforcement learning

  Supervision: reward for a sequence of decisions

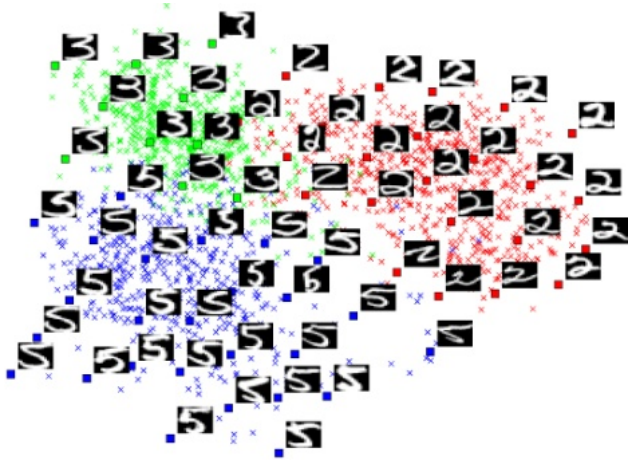
Dimensionality reduction & manifold learning

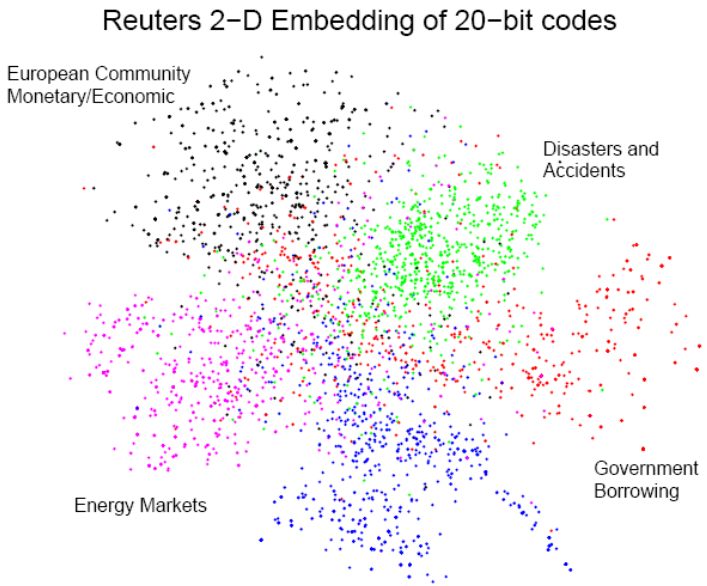
- Find a low-dimensional representation of high-dimensional data
  - Continuous outputs are `invented'
- 3D data, one dimensional embedding (hue of color)



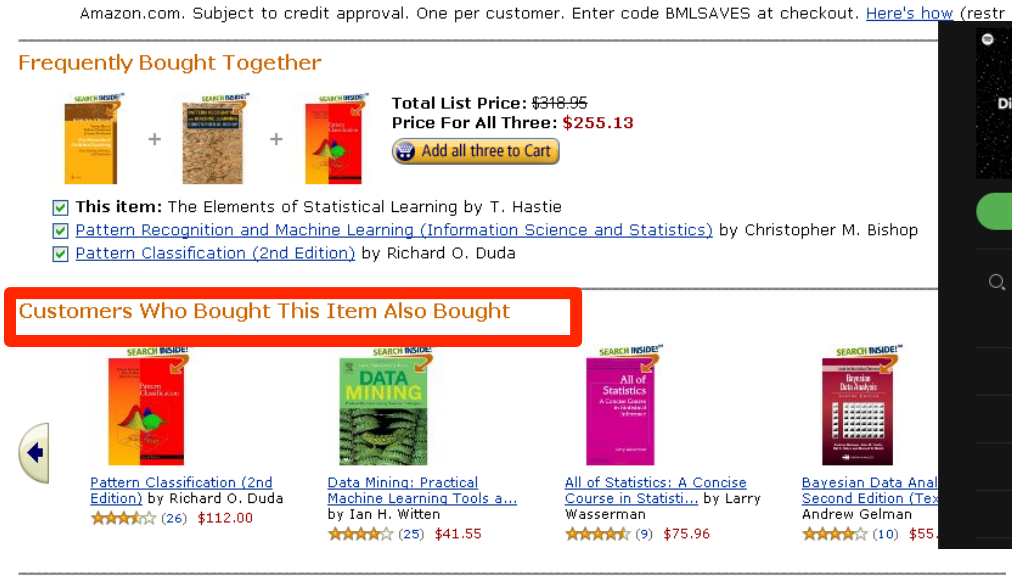
Dimensionality reduction & manifold learning examples

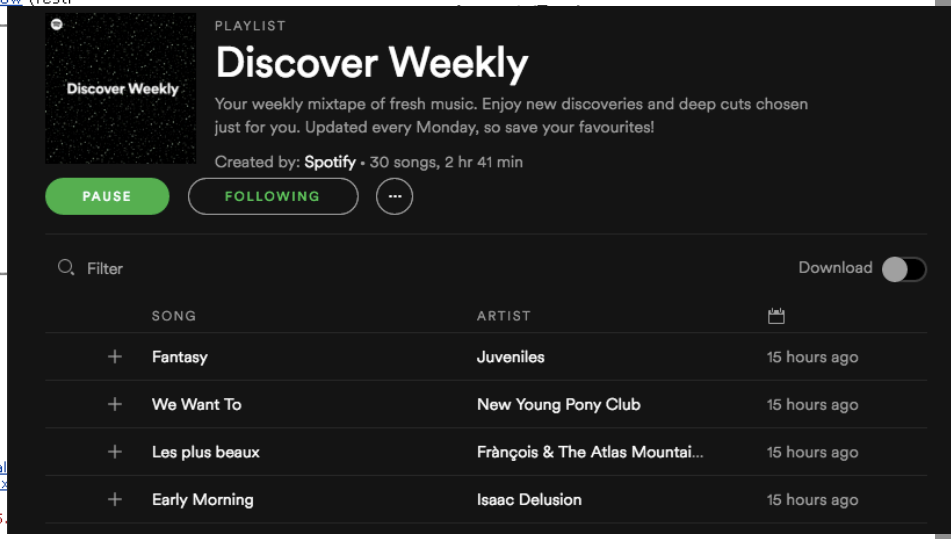
- Visualization, retrieval





- Recommendation systems





Focus of this course: supervised learning

- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimensionality Reduction, Manifold Learning
- Weakly supervised

  Some data supervised, some unsupervised
- Reinforcement learning

  Supervision: reward for a sequence of decisions

What we want to learn: a function

- Input-output mapping

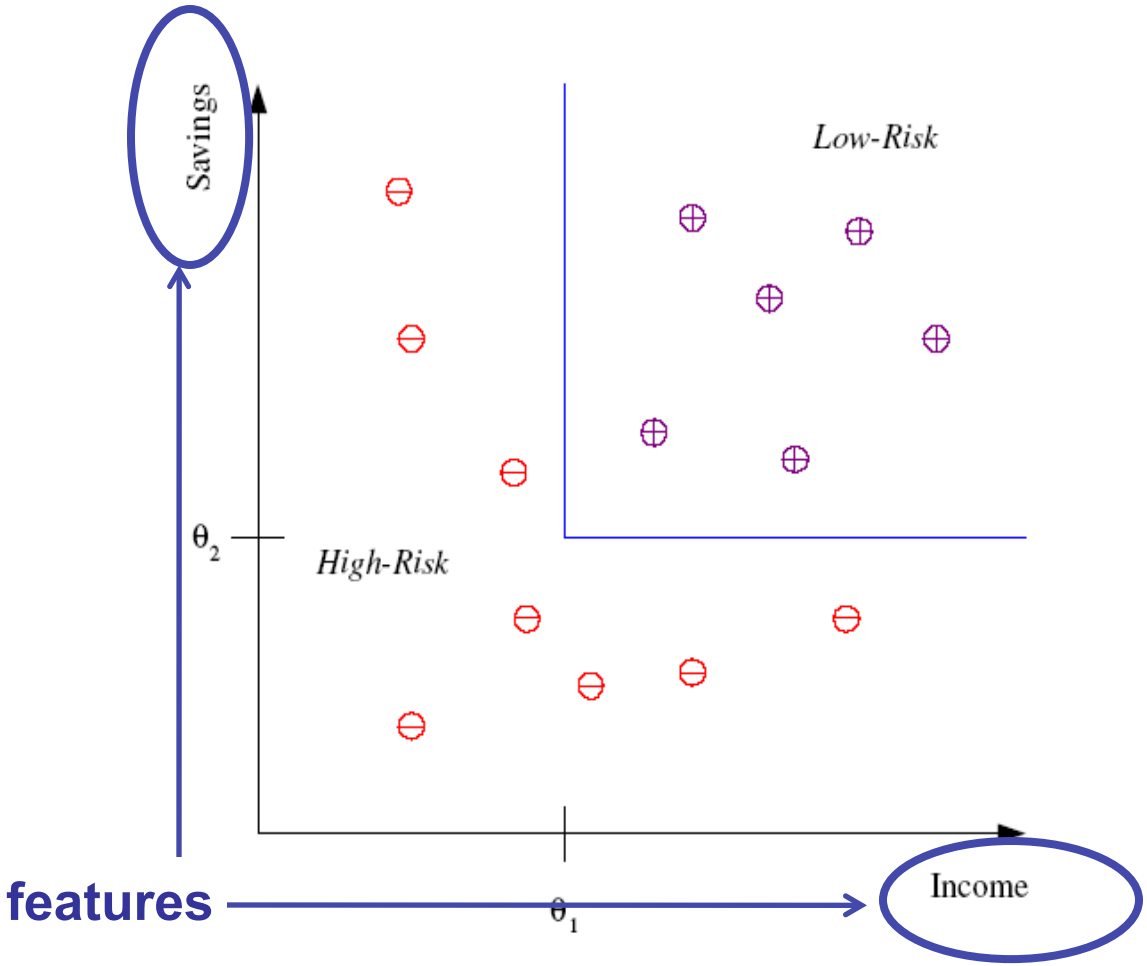
$$y = f_w(x) \quad (= f(x, w))$$

  - Output: y
  - Input: x
  - Method: f
  - Parameters: w

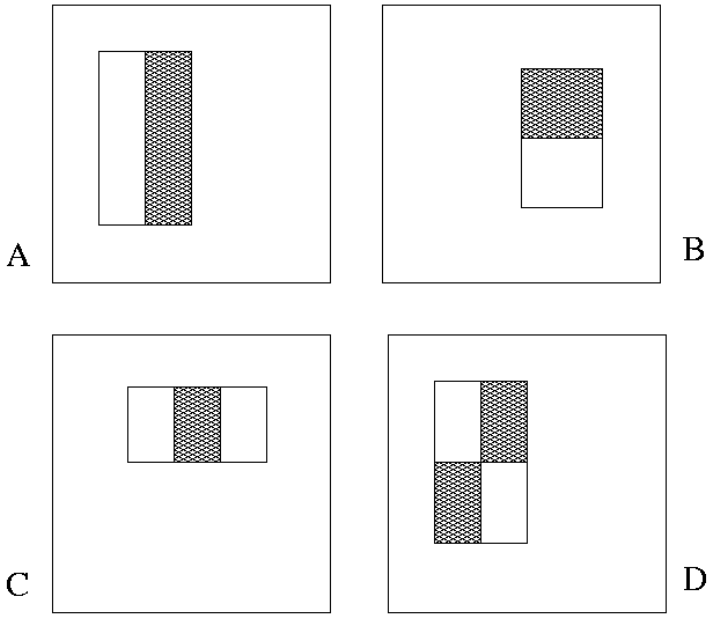
How to construct this function?

$$y = f_w(x)$$

- Step 1: Determine its inputs, x



Feature example: Haar wavelets (NOT part of our course)



Value* = ∑ *(pixels in white area)* – ∑ *(pixels in black area)

Why these features?
Extremely fast to compute
(4 pixel operations per box)

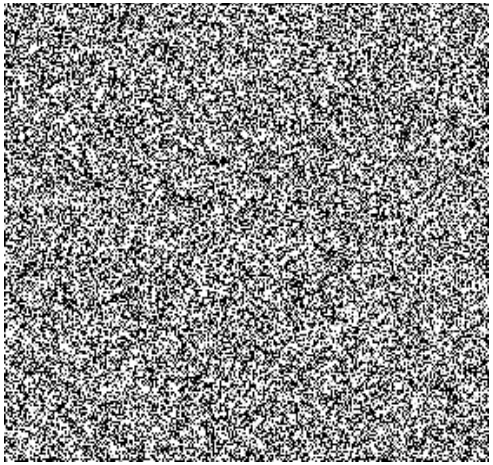
1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

input image

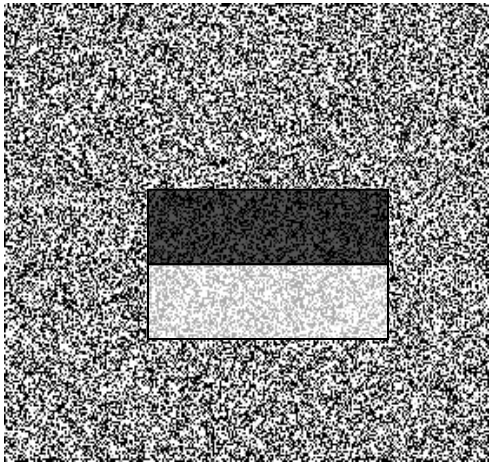
0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	74

integral image

One Haar wavelet feature

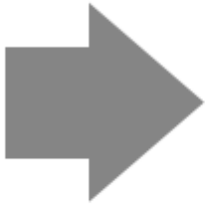


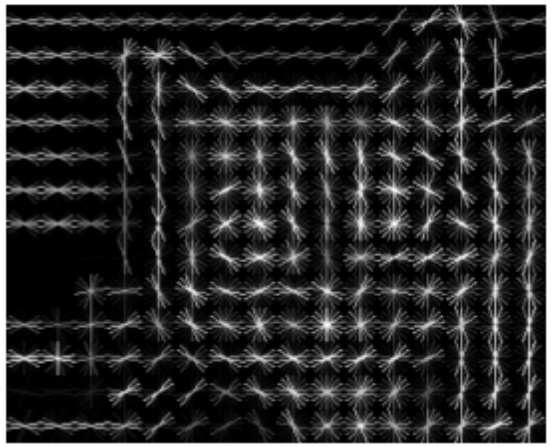
Source

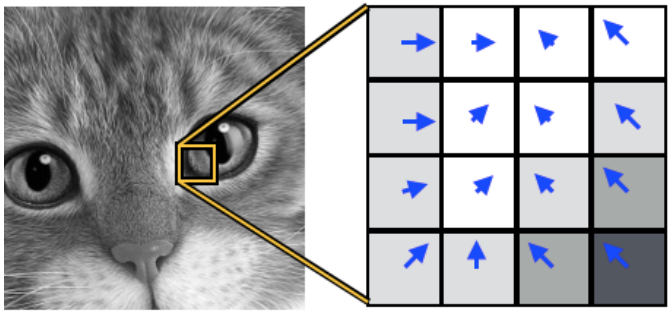


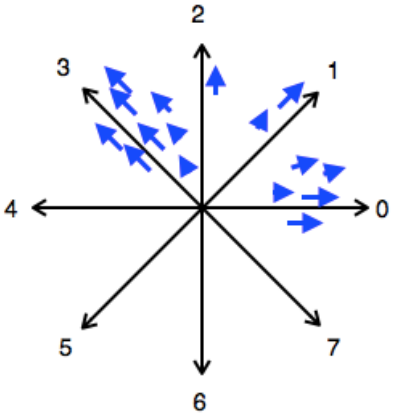
Result

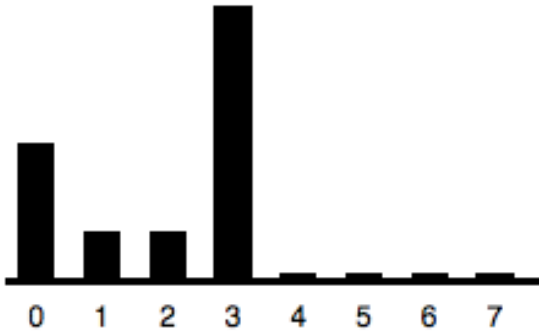
Feature example: Histogram-of-gradient features (NOT part of our course)











Feature example: Bag-of-word features (NOT part of our course)

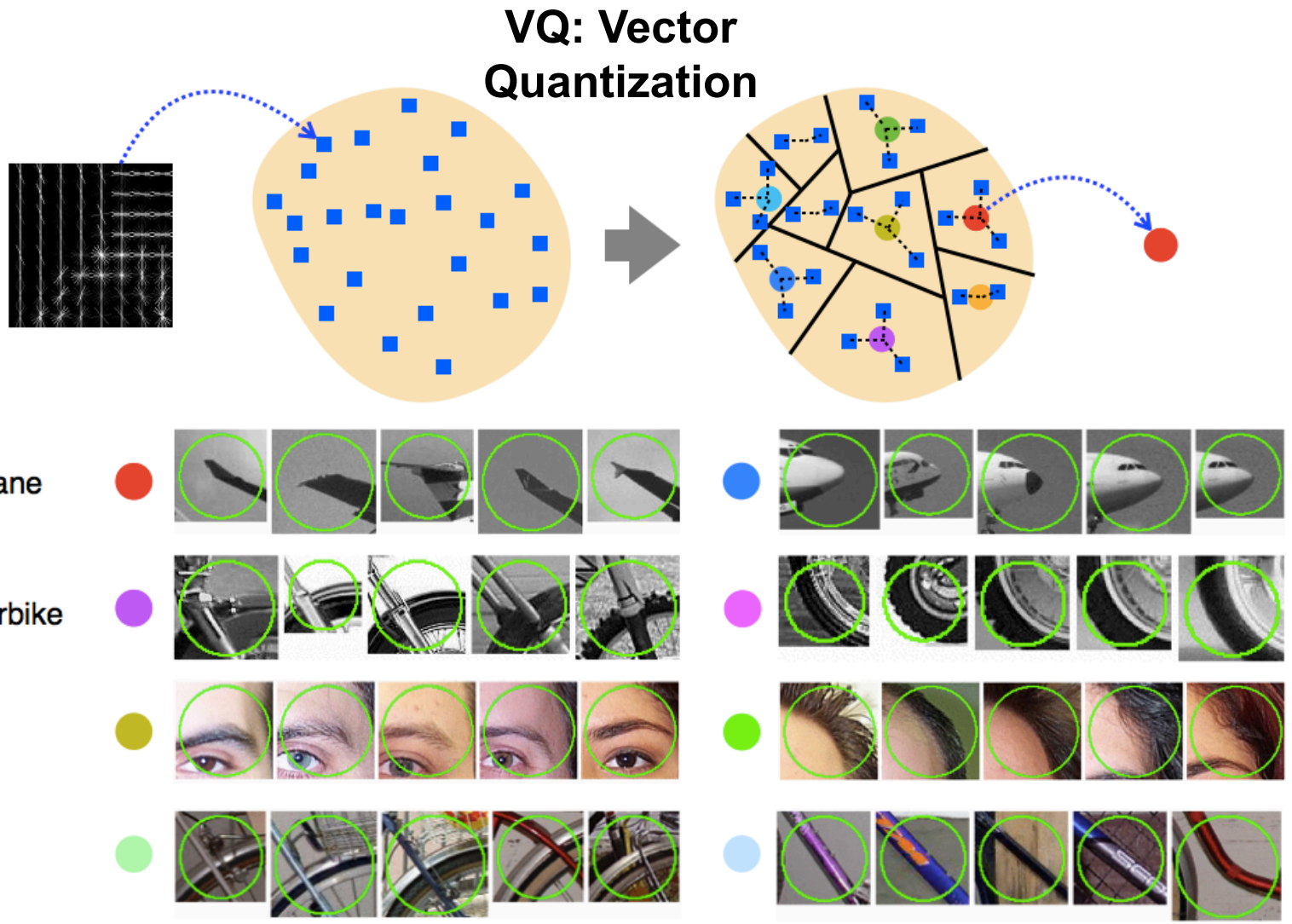
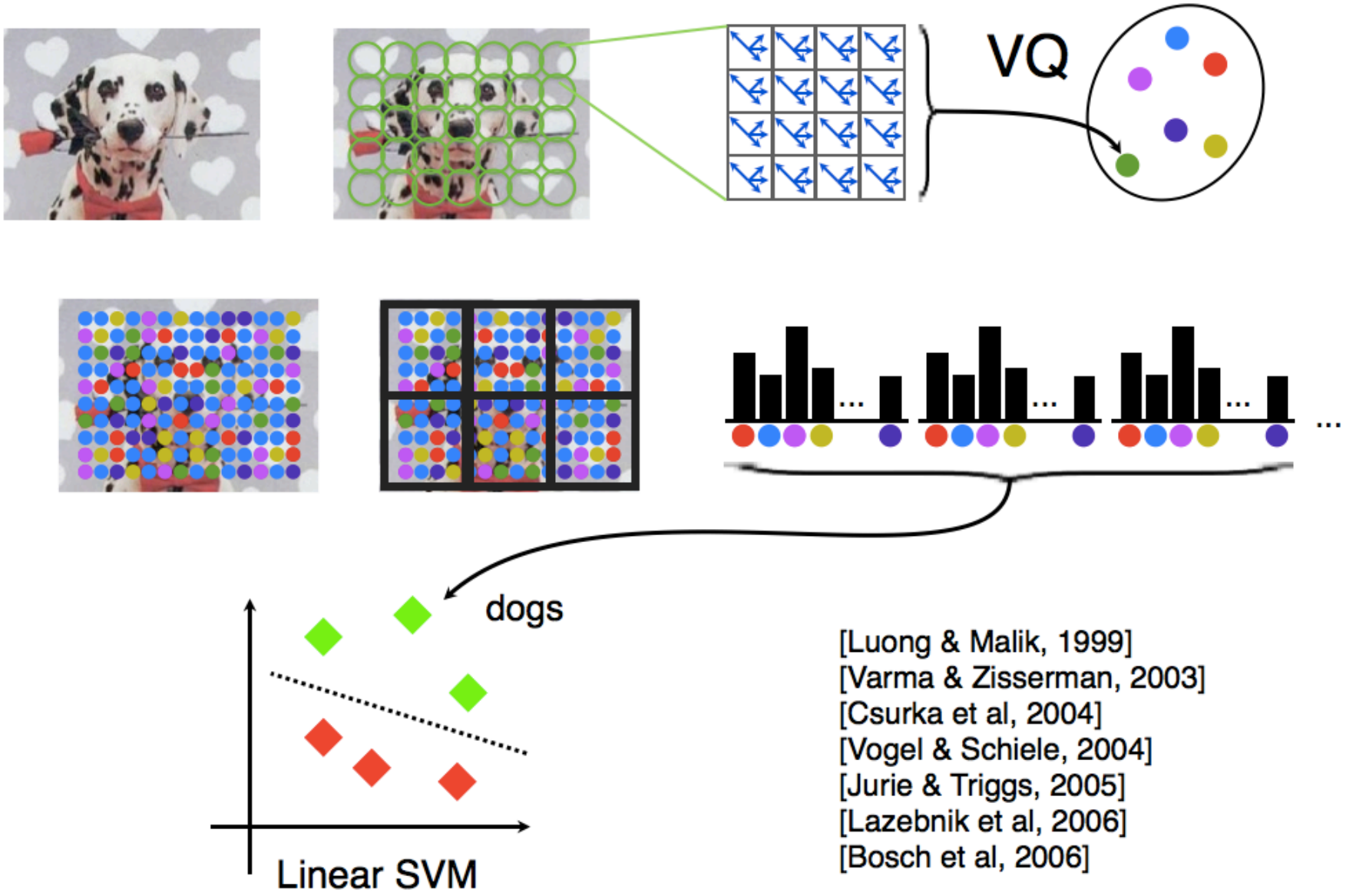
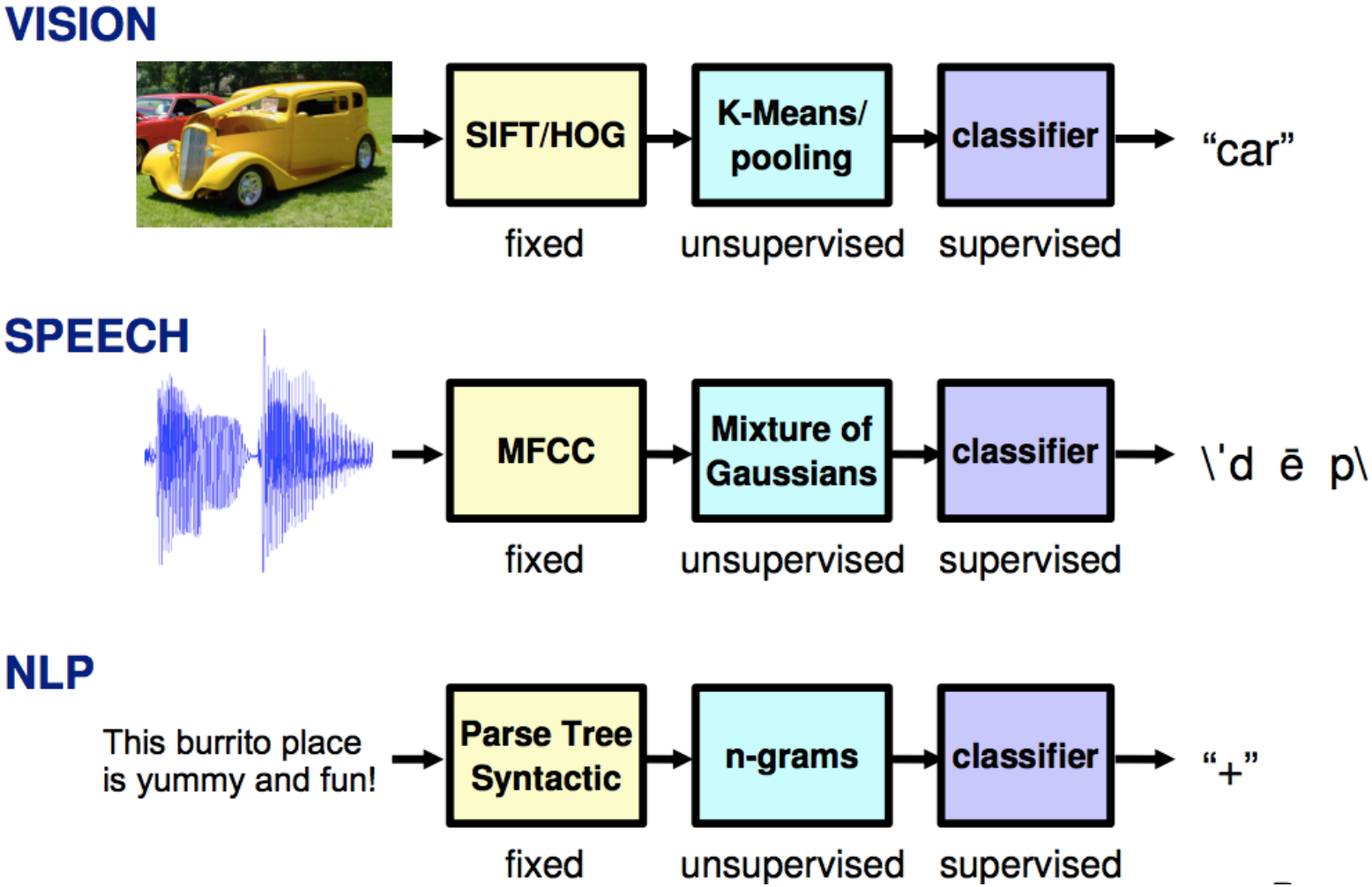


Image classification in a nutshell (NOT part of our course)

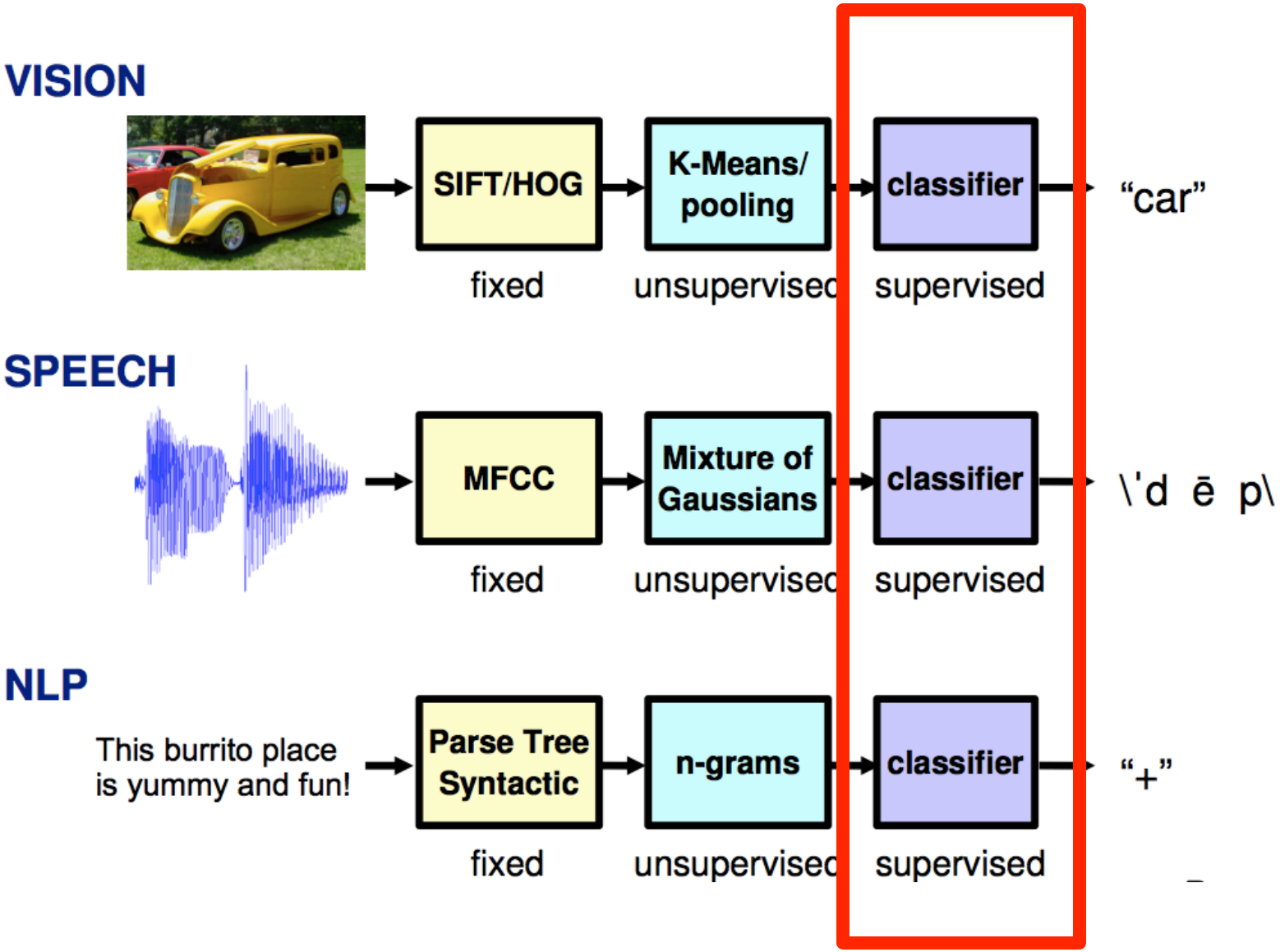


[Luong & Malik, 1999]
[Varma & Zisserman, 2003]
[Csurka et al, 2004]
[Vogel & Schiele, 2004]
[Jurie & Triggs, 2005]
[Lazebnik et al, 2006]
[Bosch et al, 2006]

Machine Learning for X: features for X + ML



Our course


VISION
SIFT/HOG
K-Means/ pooling
classifier
"car"
fixed
unsupervised
supervised
SPEECH
MFCC
Mixture of Gaussians
classifier
\'d ē p\
fixed
unsupervised
supervised
NLP
This burrito place is yummy and fun!
Parse Tree Syntactic
n-grams
classifier
"+"
fixed
unsupervised
supervised

Clarification

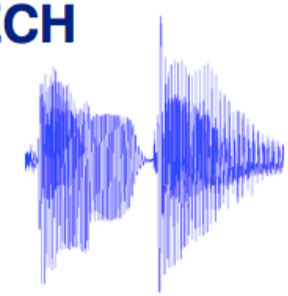
- Feature extraction:
  - Complicated!
  - No need to worry about it!
  - It would take another course to explain it
  - This course: we focus on the general problem and talk about ‘features’
  - Deep learning: no hand-engineered features needed!

Deep Learning: learn the features!

VISION

$x \rightarrow \max(0, W^1 x) \rightarrow h^1 \rightarrow \max(0, W^2 h^1) \rightarrow h^2 \rightarrow W^3 h^2 \rightarrow o$

SPEECH



$x \rightarrow \max(0, W^1 x) \rightarrow h^1 \rightarrow \max(0, W^2 h^1) \rightarrow h^2 \rightarrow W^3 h^2 \rightarrow o$

NLP

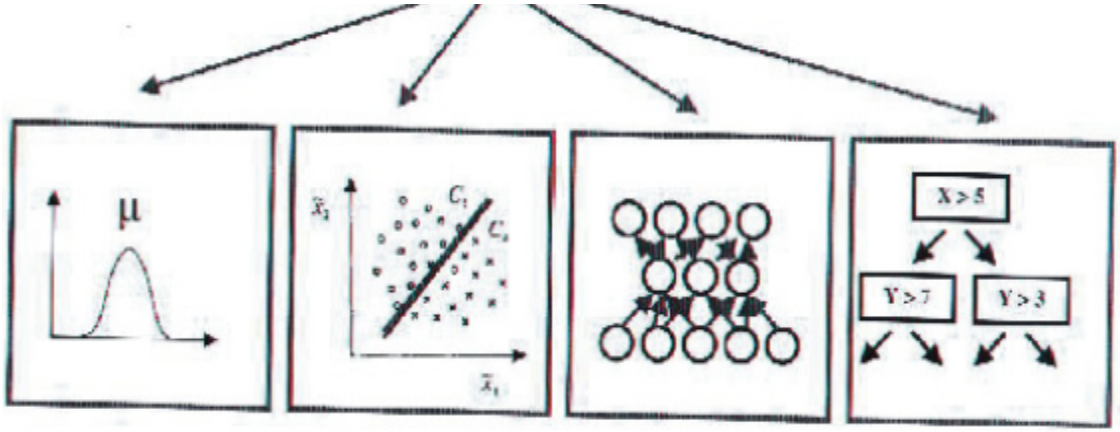
This burrito place is yummy and fun!

$x \rightarrow \max(0, W^1 x) \rightarrow h^1 \rightarrow \max(0, W^2 h^1) \rightarrow h^2 \rightarrow W^3 h^2 \rightarrow o$

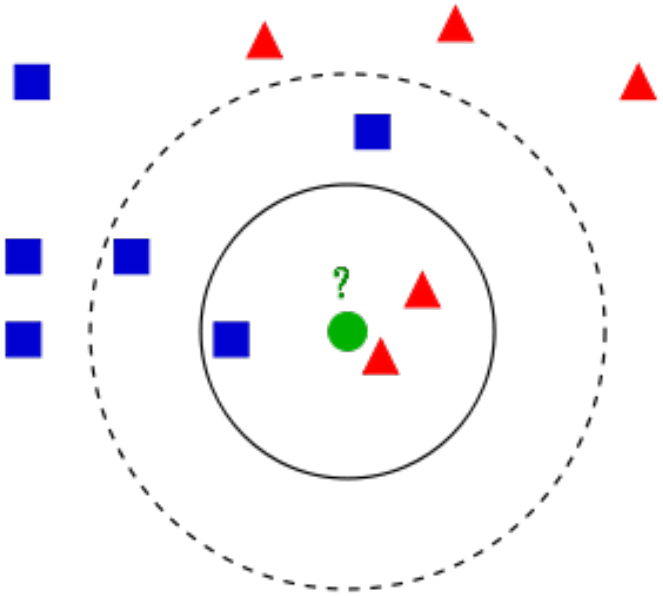
How to construct this function? $y = f_w(x)$

- Step 2: Determine the method (i.e. form of classifier)

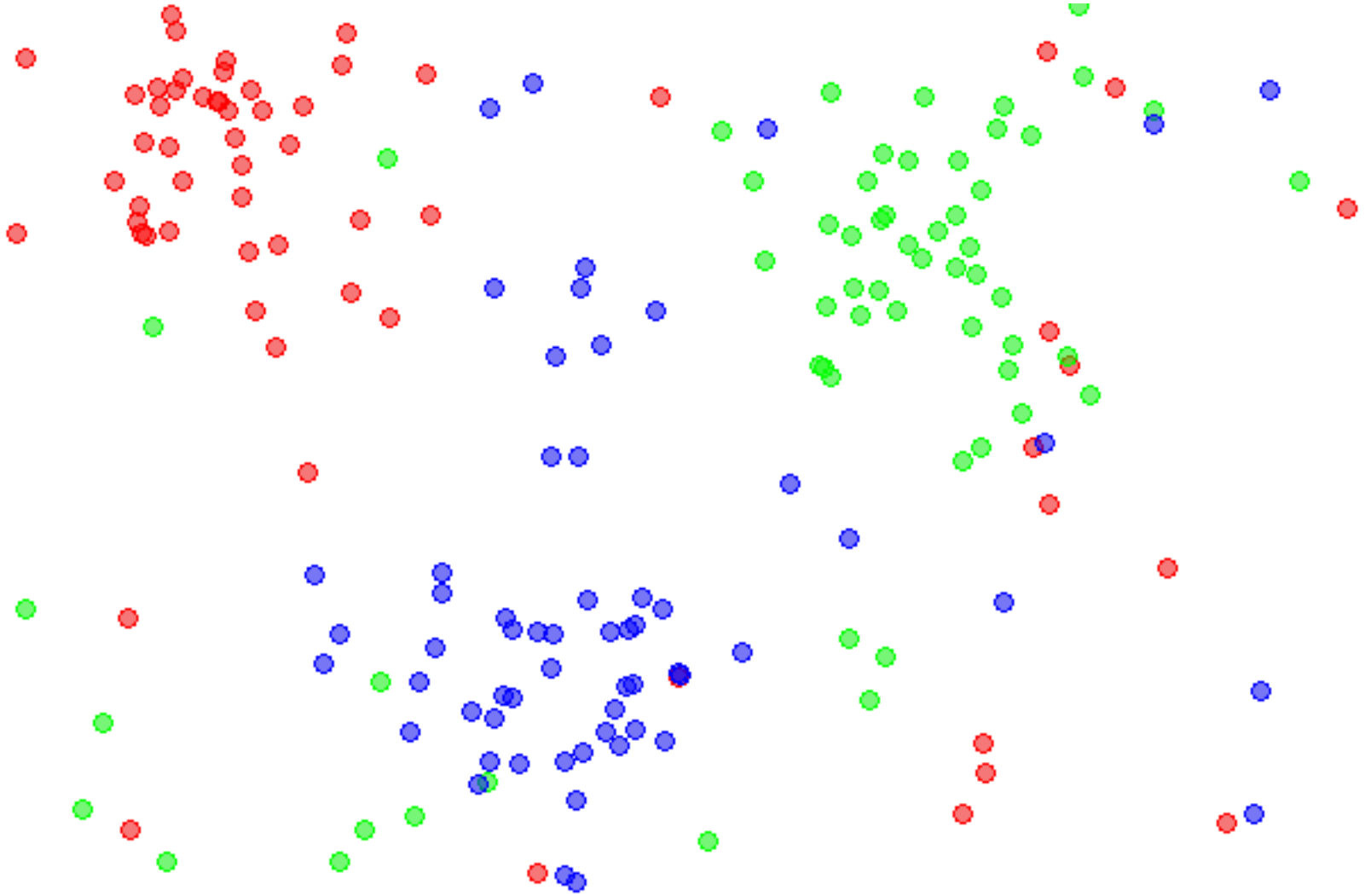
$$y = f_w(x) \quad (= f(x, w))$$



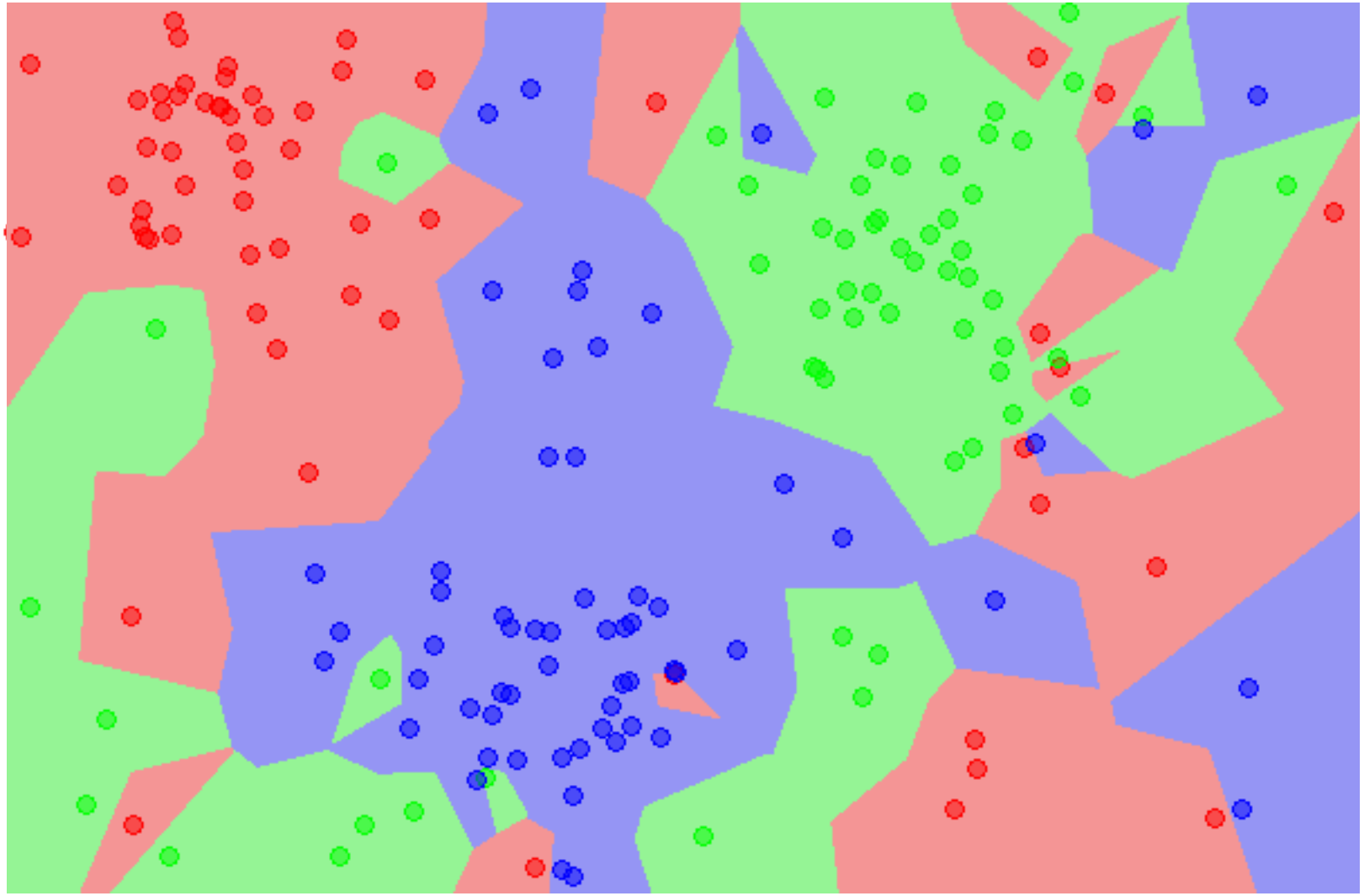
Method example: k-nearest-neighbor classifier



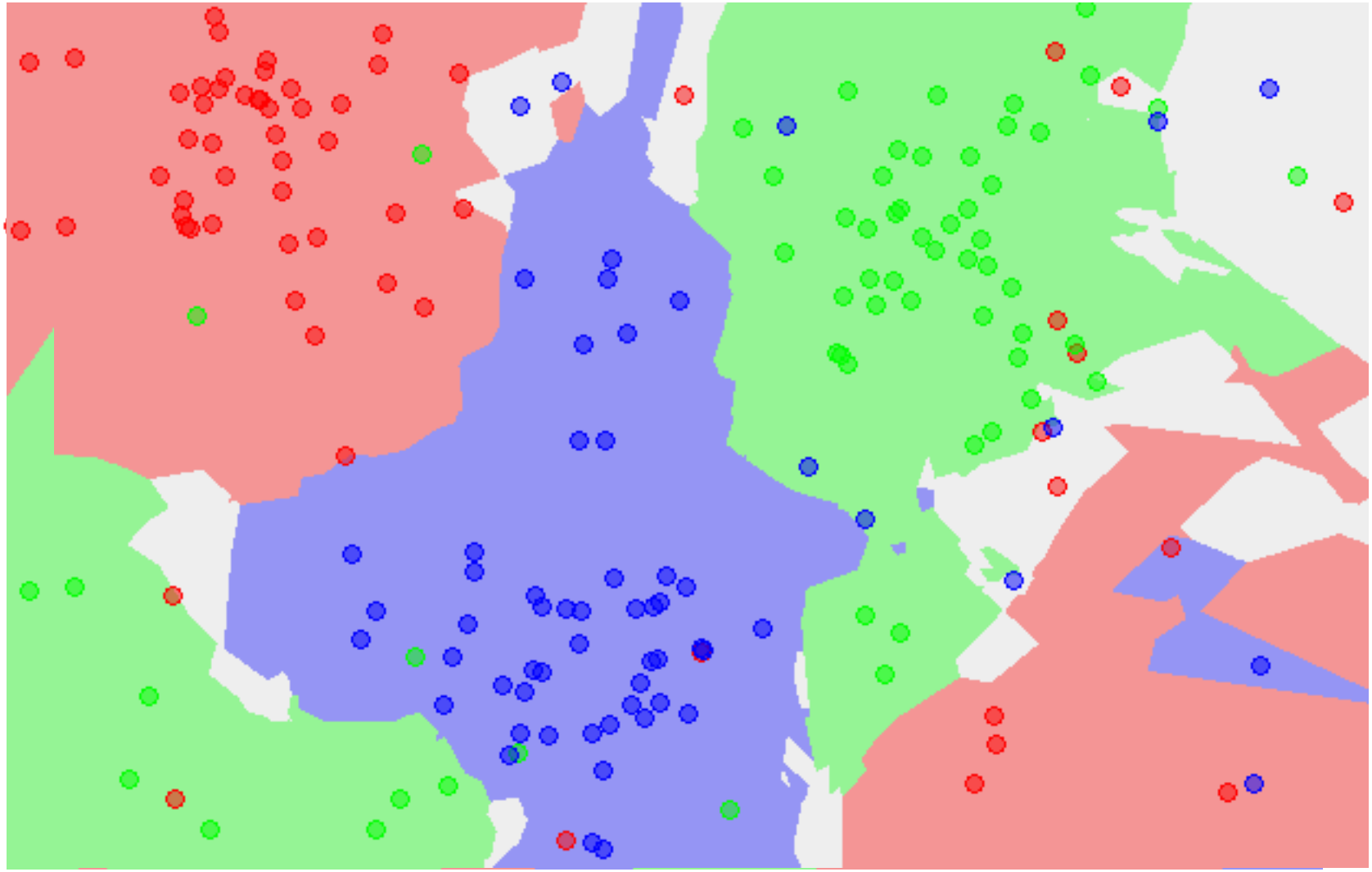
Training data for nearest-neighbor (nn) classifier



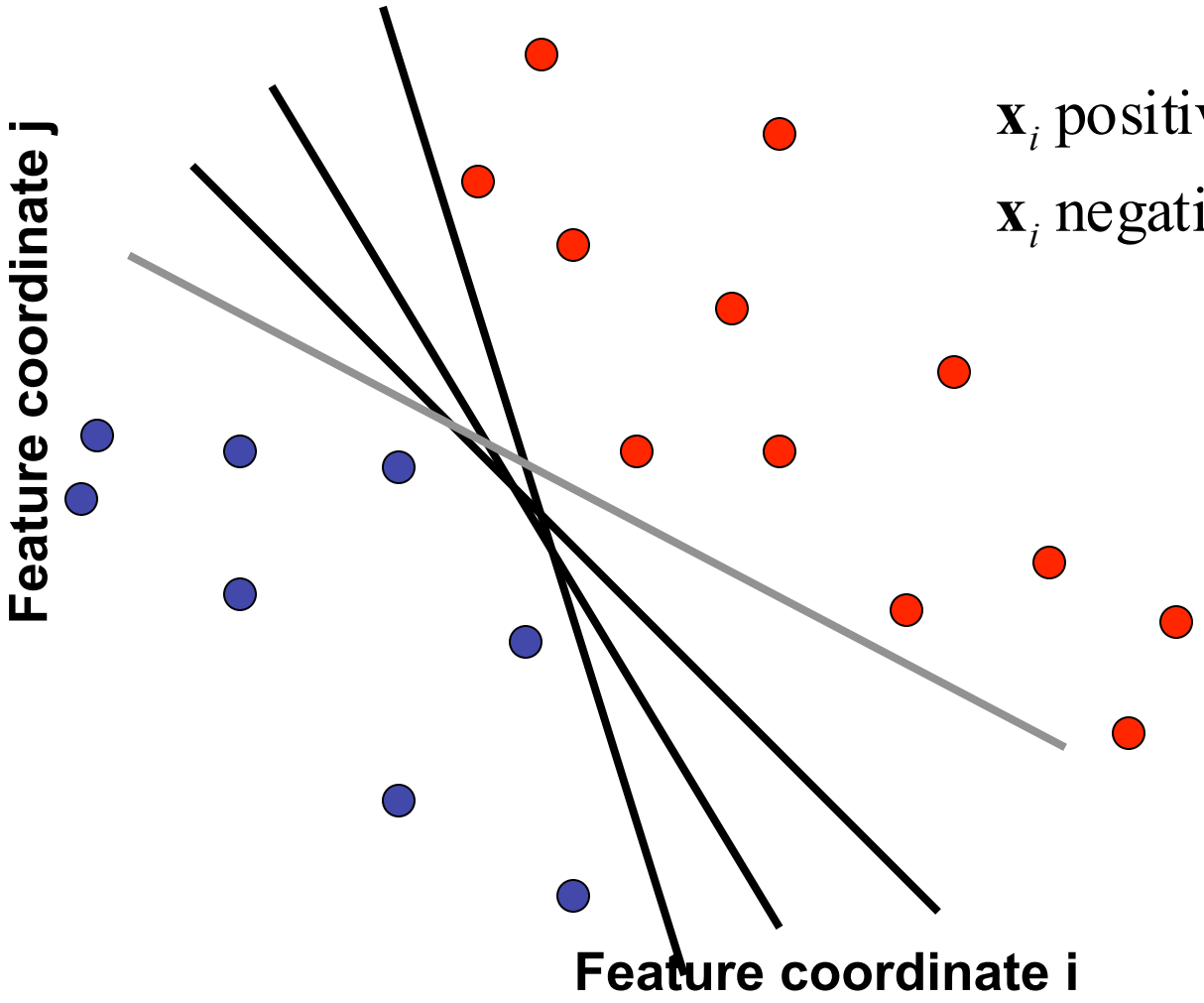
1-nn classifier prediction



3-nn classifier prediction



Method example: linear classifier



$\mathbf{x}_i$ positive: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 0$

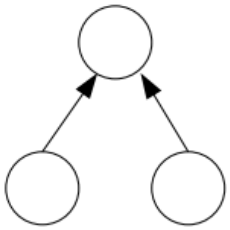
$\mathbf{x}_i$ negative: $\mathbf{x}_i \cdot \mathbf{w} + b < 0$

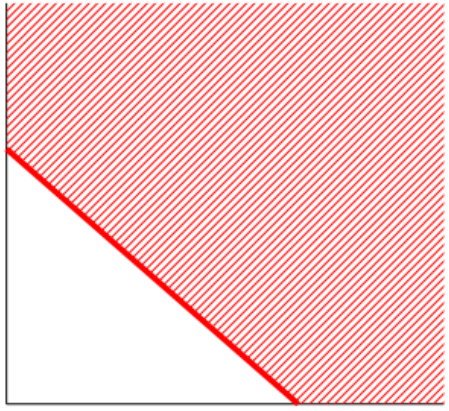
Each data point has a class label:

$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$

Method example: neural network

1 layer of trainable weights

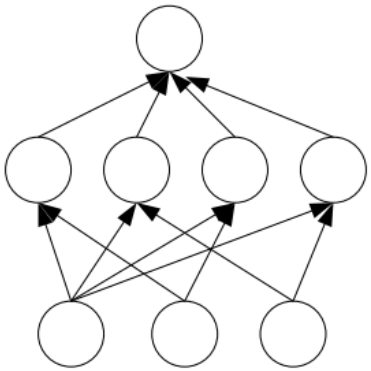


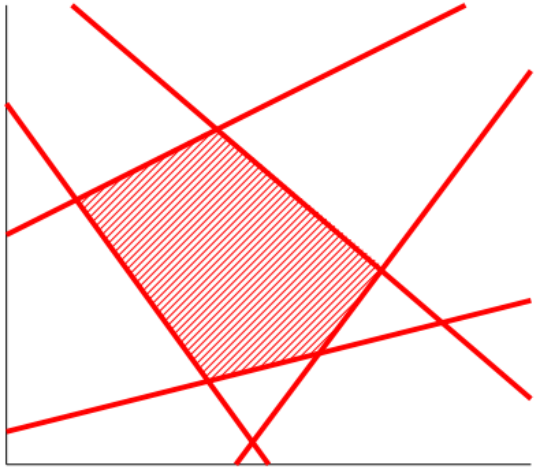


separating hyperplane

Method example: neural network

2 layers of trainable weights

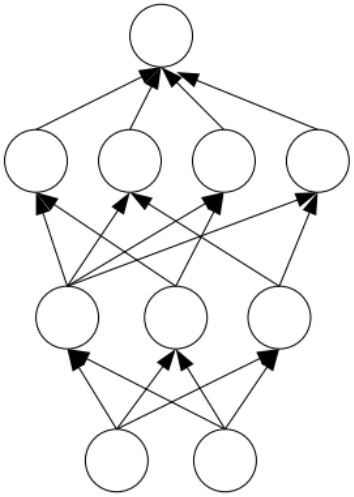


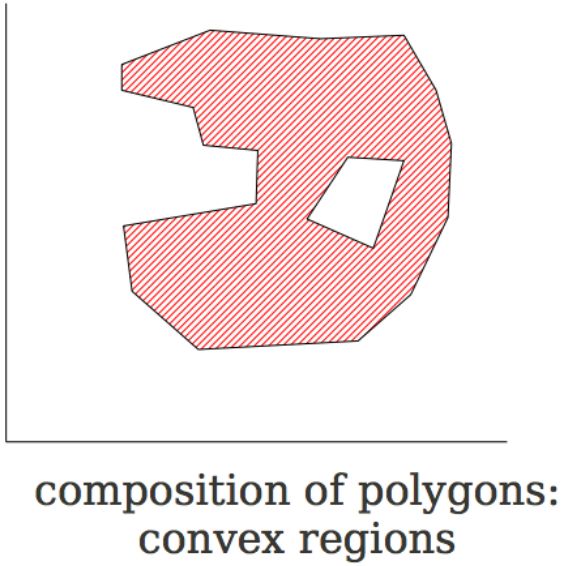


convex polygon region

Method example: neural network

3 layers of trainable weights





composition of polygons: convex regions

How to construct this function? $y = f_w(x)$

- Step 3: determine w
  - Step 3.1: quantify performance
  - Step 3.2: optimize with respect to w

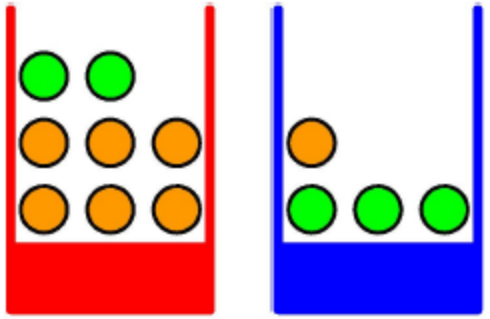
- What is the right performance measure for a classifier?
  - what do we expect to use this classifier for?

1st Lecture Layout

- Introduction to Pattern Recognition
- Introduction to Classification
  - Probability theory review
  - Decision Theory
  - Discriminative and Generative modelling

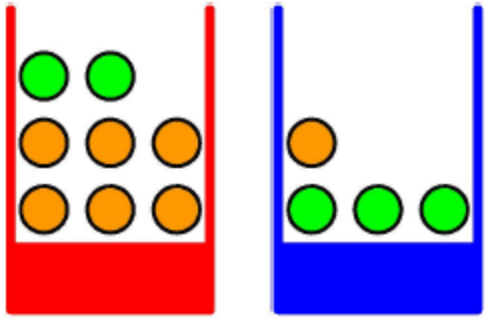
Probability Review -I

- Example: apples and oranges
  - We have two boxes to pick from.
  - Each box contains both types of fruit.
  - What is the probability of picking an apple?



Probability Review -I

- Example: apples and oranges
  - We have two boxes to pick from.
  - Each box contains both types of fruit.
  - What is the probability of picking an apple?



- Formalization
  - Let $B \in \{r, b\}$ be a random variable for the box we pick.
  - Let $F \in \{a, o\}$ be a random variable for the type of fruit we get.
  - Suppose we pick the red box 40% of the time. We write this as

$$p(B = r) = 0.4 \qquad p(B = b) = 0.6$$

  - The probability of picking an apple *given* a choice for the box is

$$p(F = a \mid B = r) = 0.25 \qquad p(F = a \mid B = b) = 0.75$$

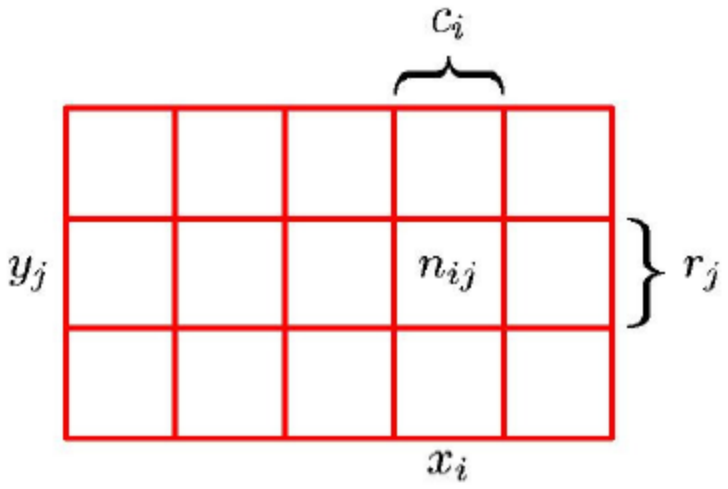
  - What is the probability of picking an apple?

$$p(F = a) = ?$$

Joint, Marginal, Conditional Probability

- More general case
  - Consider two random variables $X \in \{x_i\}$ and $Y \in \{y_j\}$
  - Consider N trials and let

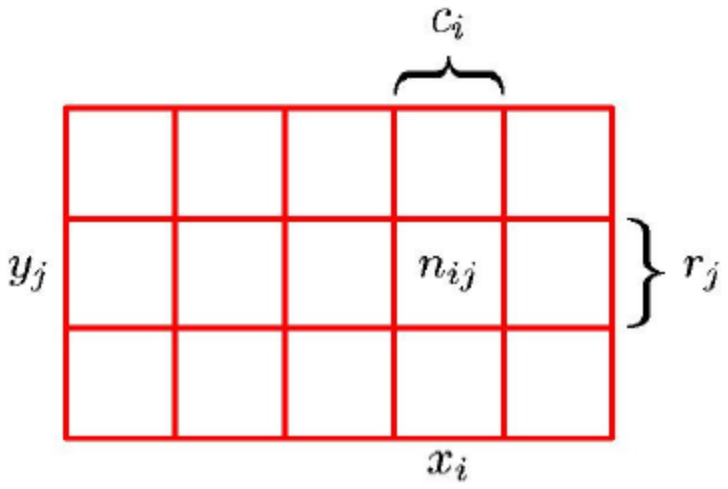
$$n_{ij} = \#\{X = x_i \wedge Y = y_j\}$$
$$c_i = \#\{X = x_i\}$$
$$r_j = \#\{Y = y_j\}$$



Joint, Marginal, Conditional Probability

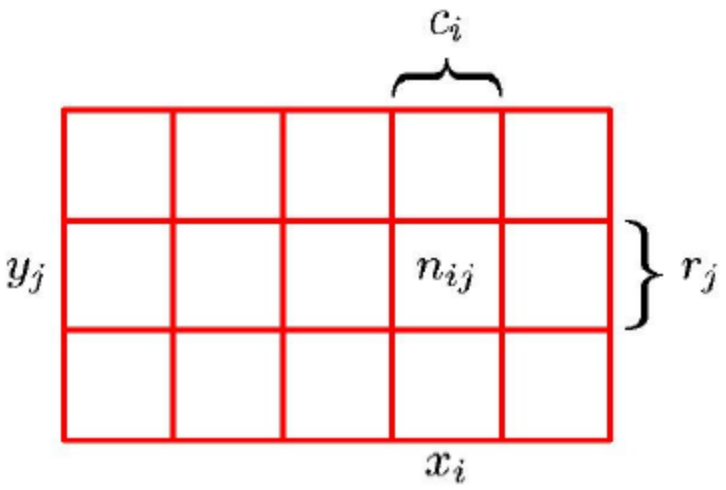
- More general case
  - Consider two random variables $X \in \{x_i\}$ and $Y \in \{y_j\}$
  - Consider N trials and let

$$n_{ij} = \#\{X = x_i \wedge Y = y_j\}$$
$$c_i = \#\{X = x_i\}$$
$$r_j = \#\{Y = y_j\}$$



- Then we can derive
  - Joint probability $$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$
  - Marginal probability $$p(X = x_i) = \frac{c_i}{N}.$$
  - Conditional probability $$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

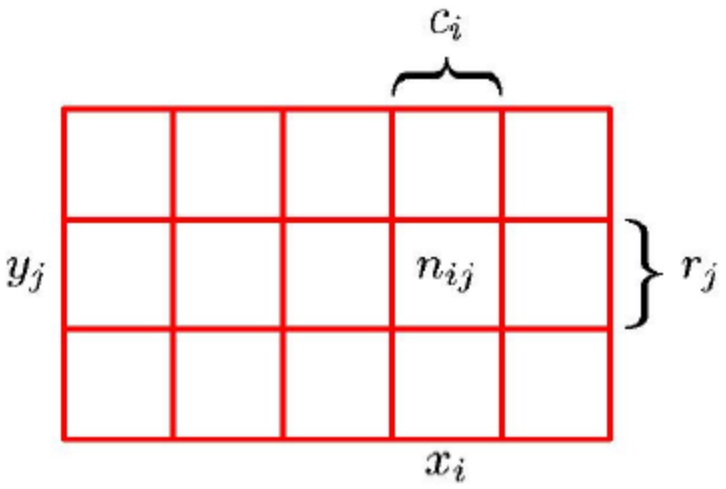
Sum & Product rule



- Rules of probability
  - Sum rule

$$p(X = x_i) = \frac{c_i}{N} = \frac{1}{N} \sum_{j=1}^{L} n_{ij} = \sum_{j=1}^{L} p(X = x_i, Y = y_j)$$

Sum & Product rule



- Rules of probability
  - Sum rule

$$p(X = x_i) = \frac{c_i}{N} = \frac{1}{N}\sum_{j=1}^{L} n_{ij} = \sum_{j=1}^{L} p(X = x_i, Y = y_j)$$

  - Product rule

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N}$$
$$= p(Y = y_j | X = x_i) p(X = x_i)$$

Sum and product rule

- Thus we have

Sum Rule	$p(X) = \sum_{Y} p(X, Y)$
Product Rule	$p(X, Y) = p(Y \mid X) p(X)$

Bayes' theorem

$$P(A = a, B = b) = P(A = a, B = b)$$

Product rule:

$$P(A = a|B = b)P(B = b) = P(B = b|A = a)P(A = a)$$

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

Sum rule:

$$P(A = a|B = b) = \frac{P(B=b|A=a)P(A=a)}{\sum_a P(B=b, A=a)}$$

Product rule:

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

Bayes' theorem

- Thus we have

Sum Rule	$p(X) = \sum_Y p(X, Y)$
Product Rule	$p(X, Y) = p(Y \mid X) p(X)$

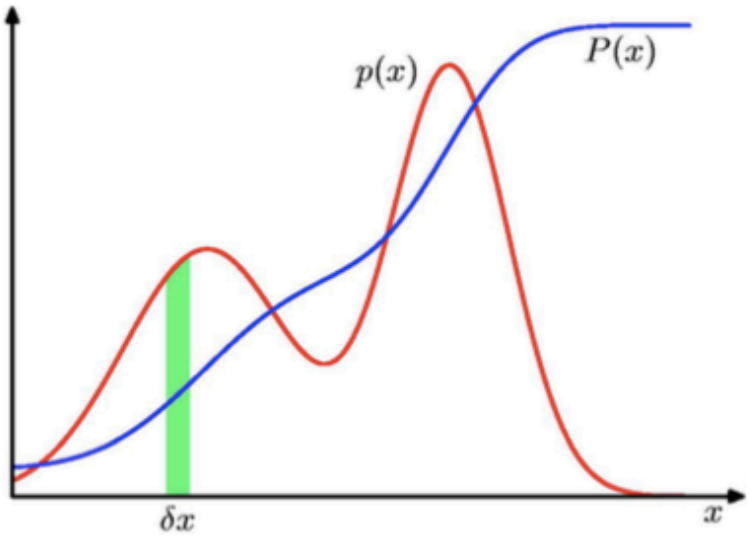
- From those, we can derive

Bayes' Theorem	$p(Y \mid X) = \frac{p(X \mid Y) p(Y)}{p(X)}$
where	$p(X) = \sum_Y p(X \mid Y) p(Y)$

Continuous variables

- **Probabilities over continuous variables are defined over their probability density function (pdf)** $p(x)$**.**

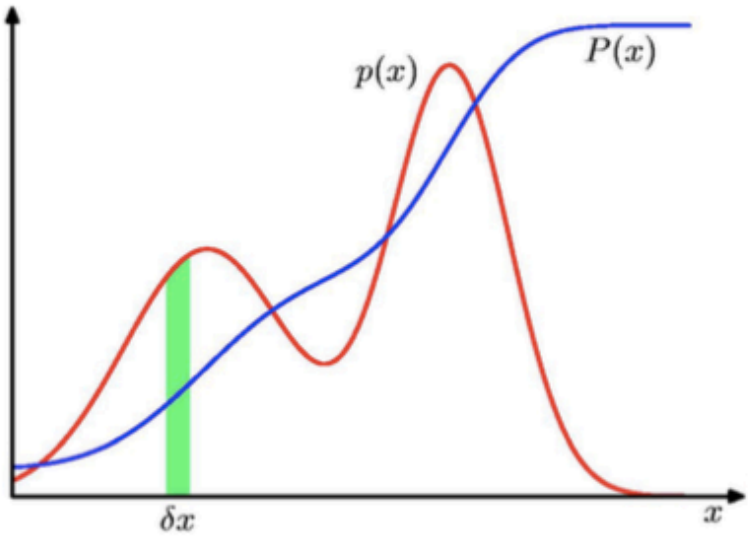
$$p(x \in (a, b)) = \int_a^b p(x)\,\mathrm{d}x$$



Continuous variables

- Probabilities over continuous variables are defined over their probability density function (pdf) $p(x)$.

$$p(x \in (a, b)) = \int_a^b p(x)\,\mathrm{d}x$$



- The probability that x lies in the interval $(-\infty, z)$ is given by the cumulative distribution function

$$P(z) = \int_{-\infty}^{z} p(x)\,\mathrm{d}x$$

All those probabilities..

Probability density function: continuous variables

Probability mass function: discrete variables

Probability distribution: probability mass/density function

Cumulative distribution function: integral of probability density

Expectation

- The average value of some function $f(x)$ under a probability distribution $p(x)$ is called its expectation

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

discrete case

$$\mathbb{E}[f] = \int p(x) f(x) \, \mathrm{d}x$$

continuous case

Expectation

- The average value of some function $f(x)$ under a probability distribution $p(x)$ is called its expectation

$$\mathbb{E}[f] = \sum_x p(x) f(x) \qquad \text{discrete case}$$

$$\mathbb{E}[f] = \int p(x) f(x)\, \mathrm{d}x \qquad \text{continuous case}$$

- If we have a finite number N of samples drawn from a pdf, then the expectation can be approximated by

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^{N} f(x_n)$$

Expectation

- The average value of some function $f(x)$ under a probability distribution $p(x)$ is called its expectation

$$\mathbb{E}[f] = \sum_x p(x) f(x) \qquad \text{discrete case}$$

$$\mathbb{E}[f] = \int p(x) f(x)\, \mathrm{d}x \qquad \text{continuous case}$$

- If we have a finite number N of samples drawn from a pdf, then the expectation can be approximated by

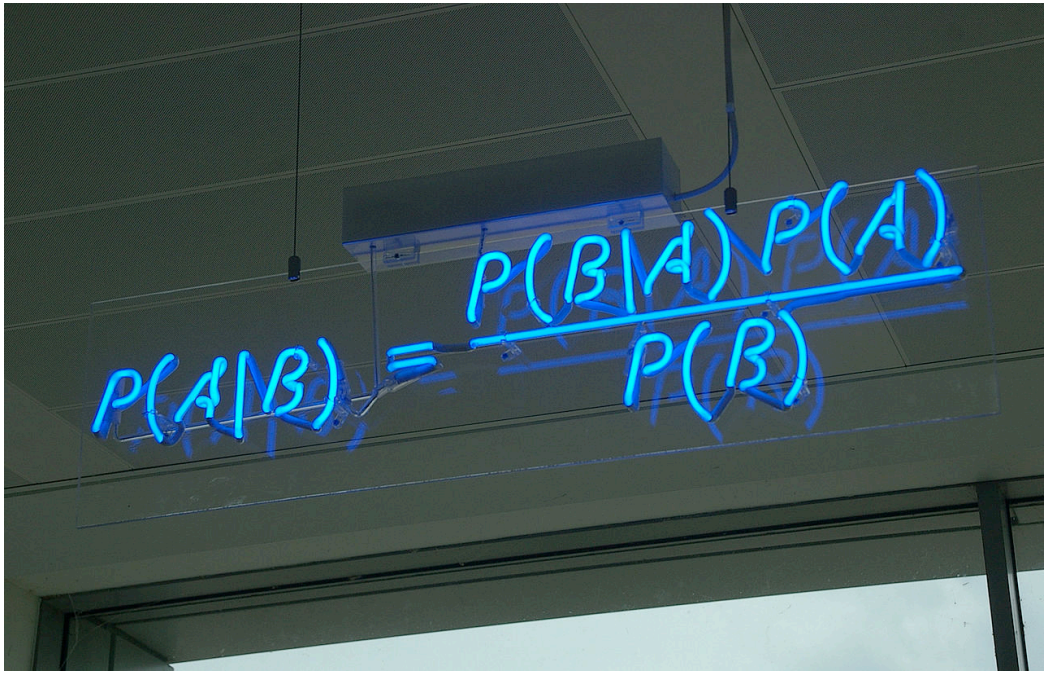
$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^{N} f(x_n)$$

- We can also consider a conditional expectation

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

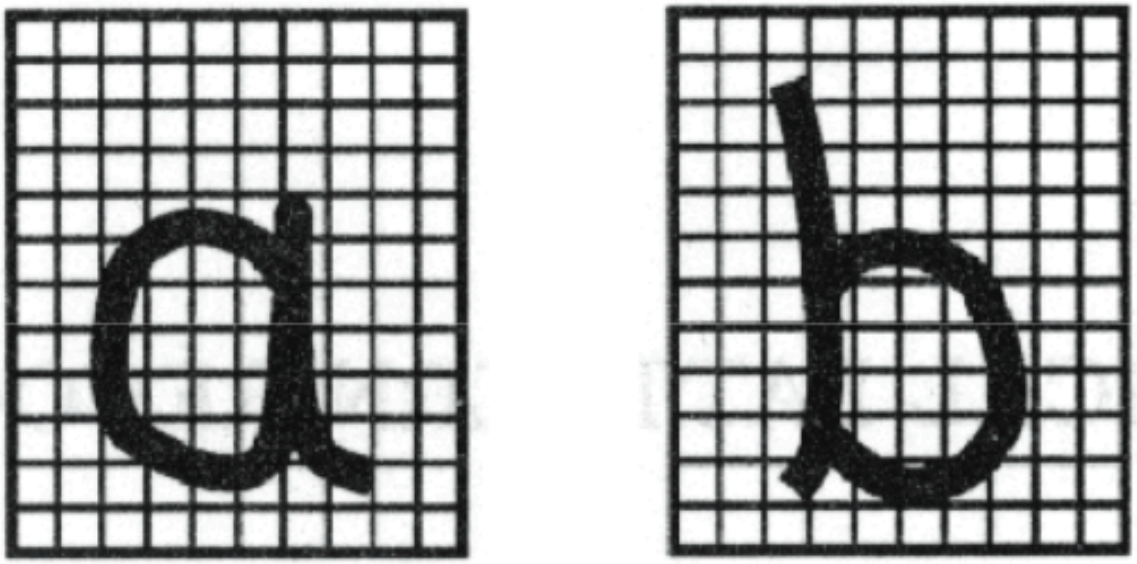
1st Lecture Layout

- Introduction to Pattern Recognition
- Introduction to Classification
  - Probability theory review
  - Bayes Decision Theory
  - Discriminative and Generative modelling



Binary classification problem

- Example: handwritten character recognition



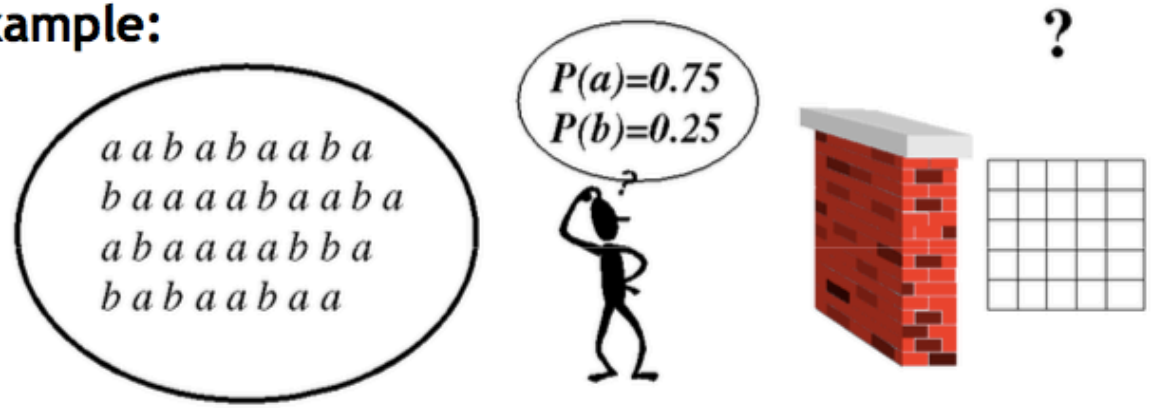
- Goal:
  - Classify a new letter such that the probability of misclassification is minimized.

Binary classification problem

- Concept 1: Priors (a priori probabilities) $p(C_k)$
  - What we can tell about the probability *before seeing the data*.

Binary classification problem

- **Concept 1: Priors (a priori probabilities)** $p(C_k)$
  - What we can tell about the probability *before seeing the data*.
  - Example:

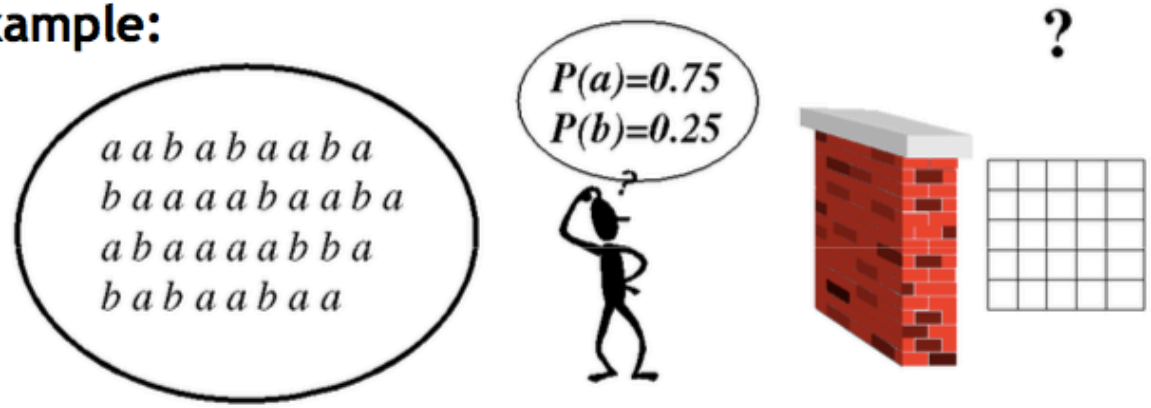


$$C_1 = a \qquad p(C_1) = 0.75$$

$$C_2 = b \qquad p(C_2) = 0.25$$

Binary classification problem

- **Concept 1: Priors (a priori probabilities)** $p(C_k)$
  - What we can tell about the probability *before seeing the data*.
  - Example:



$$C_1 = a \qquad p(C_1) = 0.75$$

$$C_2 = b \qquad p(C_2) = 0.25$$

- **In general:** $\sum_k p(C_k) = 1$

Binary classification problem

Concept 2: Conditional probabilities $p(x \mid C_k)$

Binary classification problem

Concept 2: Conditional probabilities

$$p(x \mid C_k)$$

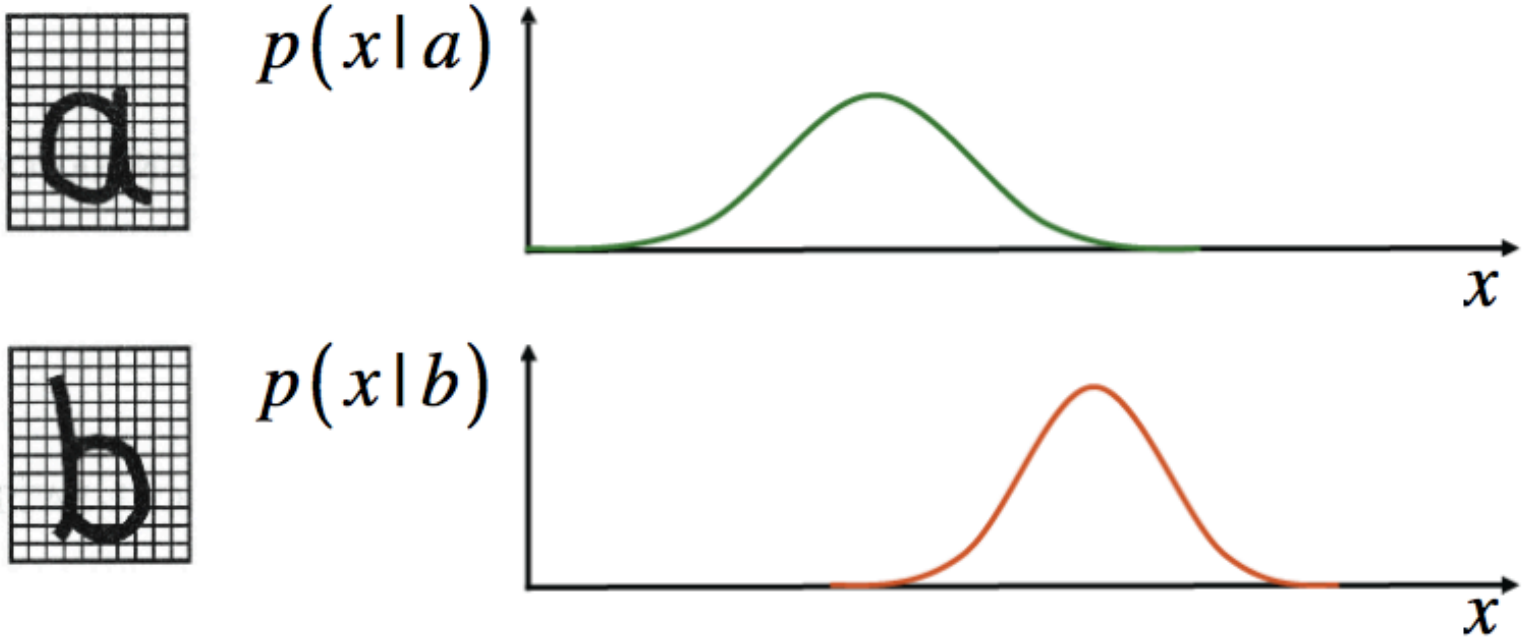
- Let x be a feature vector.
- x measures/describes certain properties of the input.
  - E.g. number of black pixels, aspect ratio, ...
- $p(x|C_k)$ describes its likelihood for class C_k.

Binary classification problem

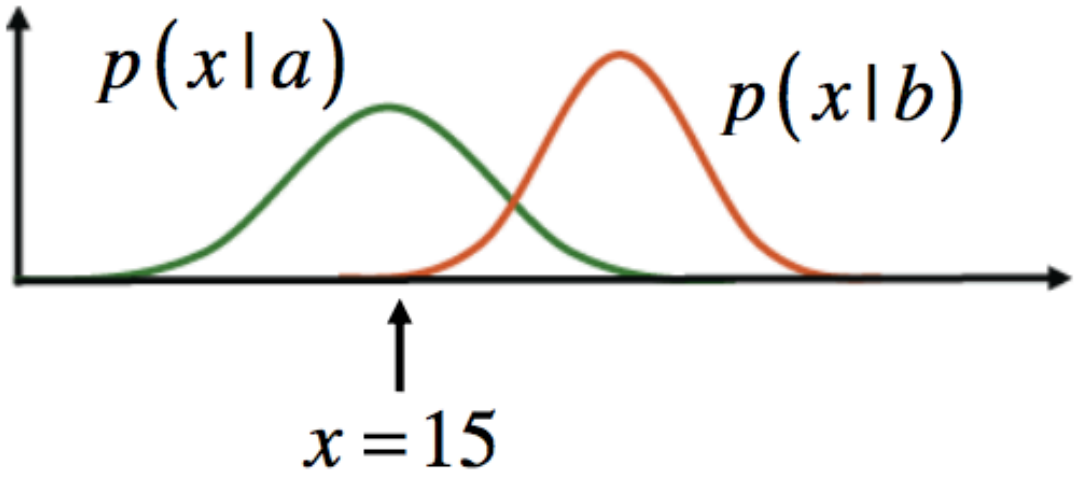
Concept 2: Conditional probabilities

$p(x \mid C_k)$

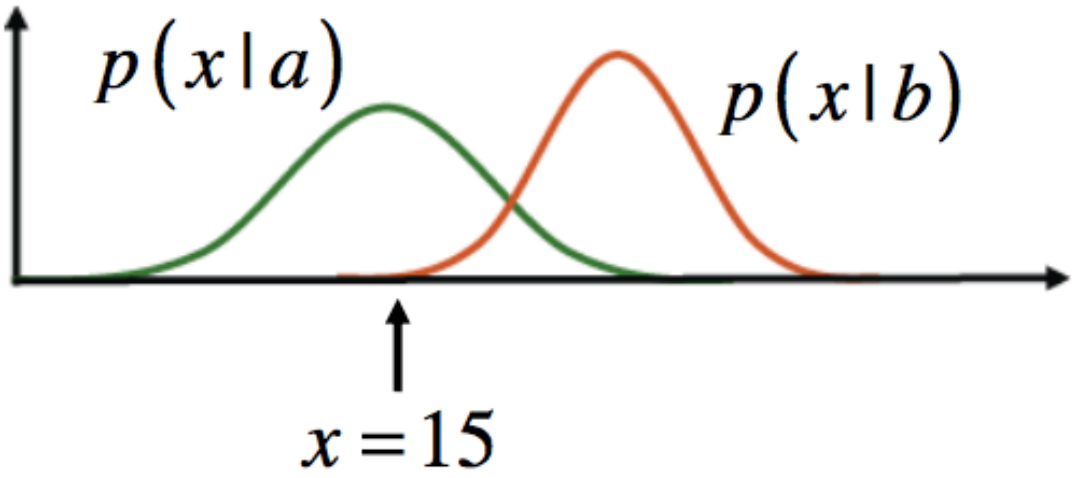
- Let x be a feature vector.
- x measures/describes certain properties of the input.
  - E.g. number of black pixels, aspect ratio, ...
- $p(x|C_k)$ describes its likelihood for class C_k.



Example:



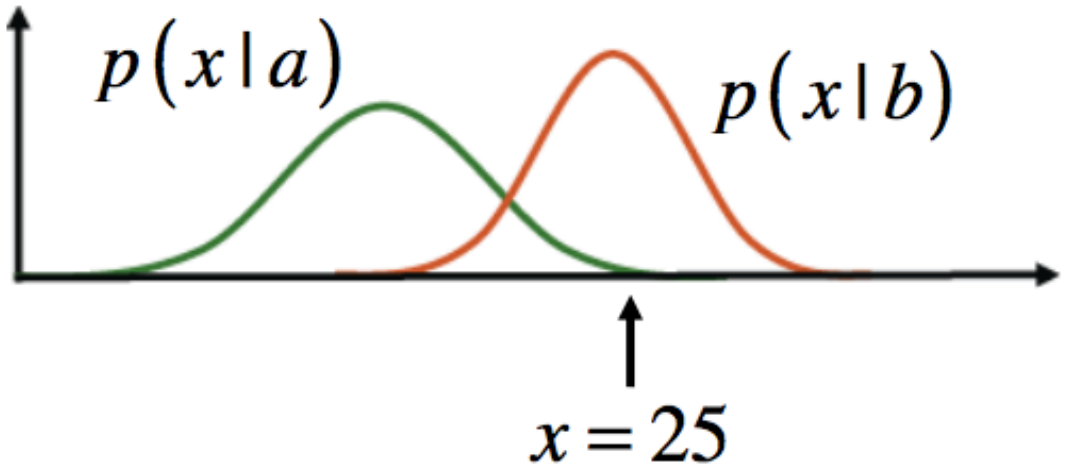
Example:



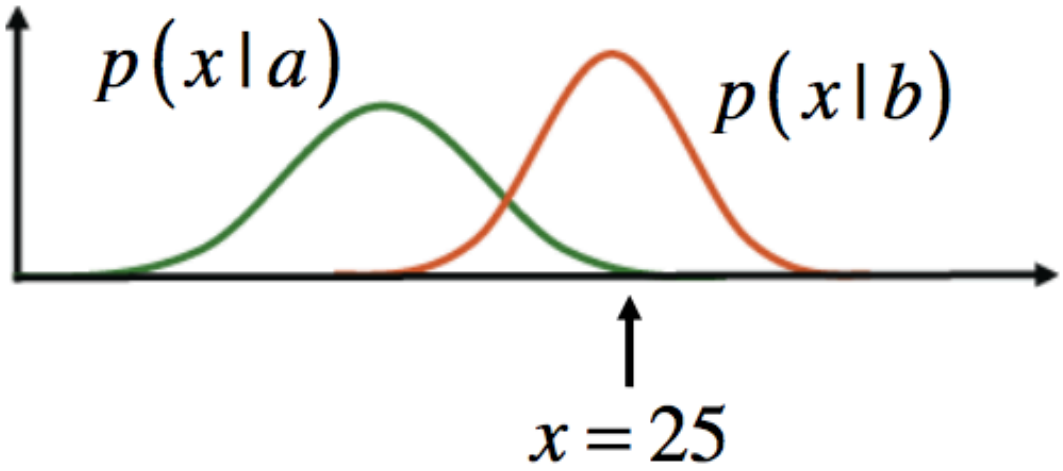
Question:

- Which class?
- The decision should be ‘a’ here.

Example:



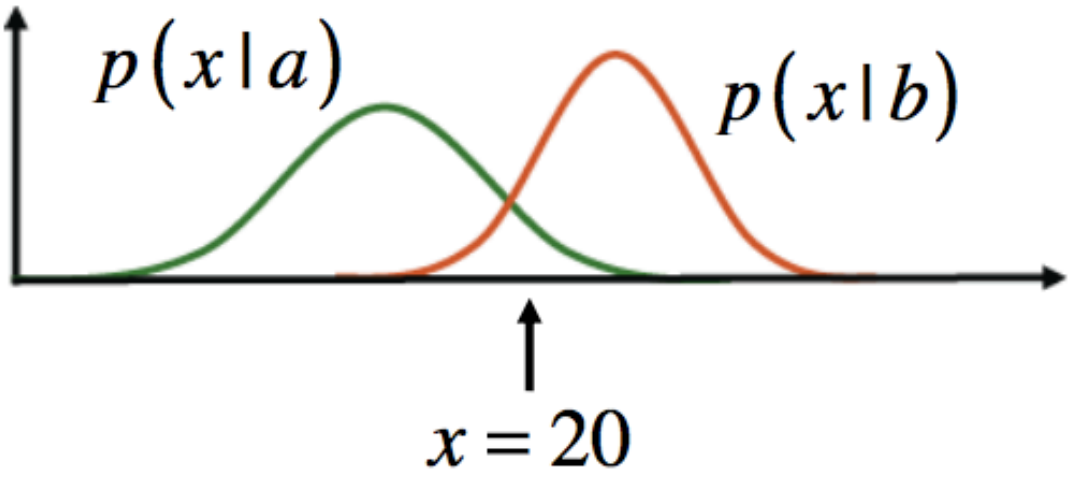
Example:



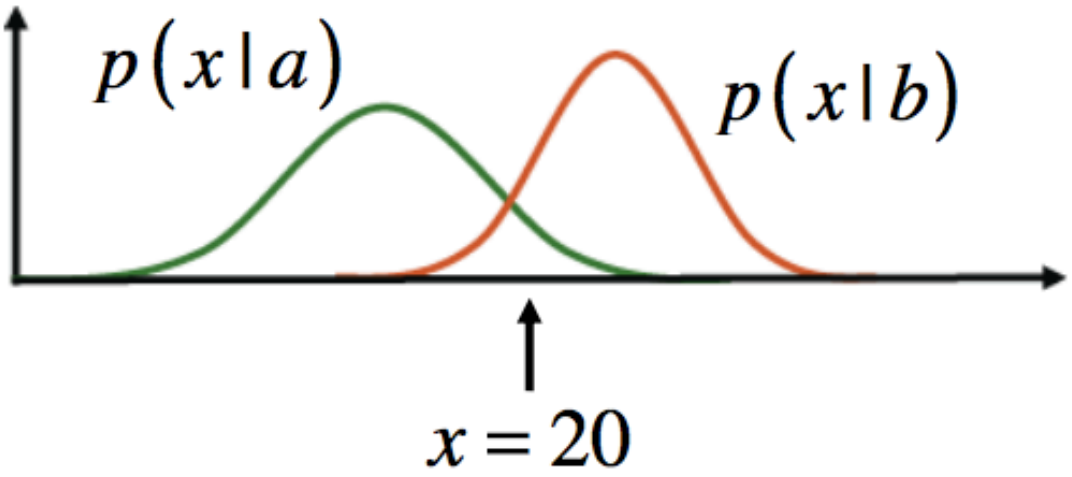
Question:

- Which class?
- Since $p(x \mid a)$ is much smaller than $p(x \mid b)$, the decision should be 'b' here.

Example:



Example:



Question:

- Which class?
- Remember that $p(a) = 0.75$ and $p(b) = 0.25$...
- I.e., the decision should be again ‘a’.

$\Rightarrow$ How can we formalize this?

Concept 3: Posterior probabilities

$p(C_k \mid x)$

- We are typically interested in the *a posteriori* probability, i.e. the probability of class C_k given the measurement vector x.

Bayes' Theorem:

$$p(C_k \mid x) = \frac{p(x \mid C_k)\, p(C_k)}{p(x)} = \frac{p(x \mid C_k)\, p(C_k)}{\sum_i p(x \mid C_i)\, p(C_i)}$$

Concept 3: Posterior probabilities

$$p(C_k \mid x)$$

- We are typically interested in the *a posteriori* probability, i.e. the probability of class C_k given the measurement vector x.

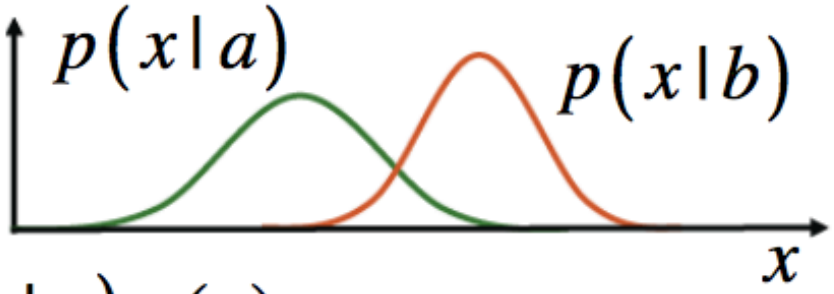
Bayes' Theorem:

$$p(C_k \mid x) = \frac{p(x \mid C_k)\, p(C_k)}{p(x)} = \frac{p(x \mid C_k)\, p(C_k)}{\sum_i p(x \mid C_i)\, p(C_i)}$$

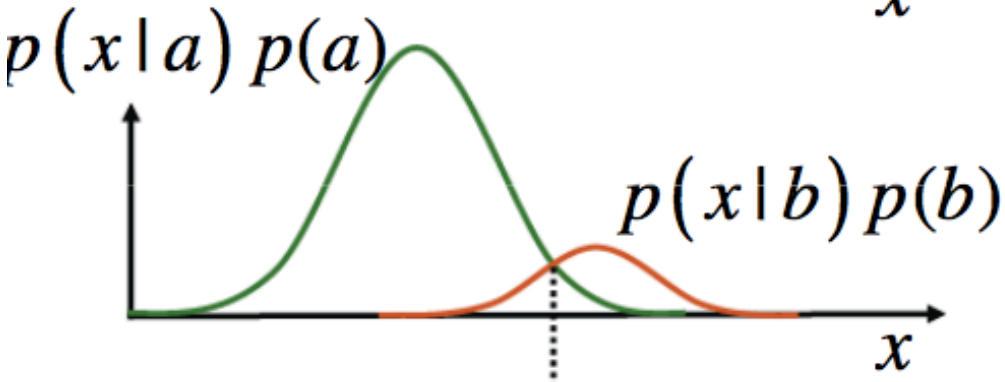
Interpretation

$$Posterior = \frac{Likelihood \times Prior}{Normalization\ Factor}$$

Binary classification problem & posteriors

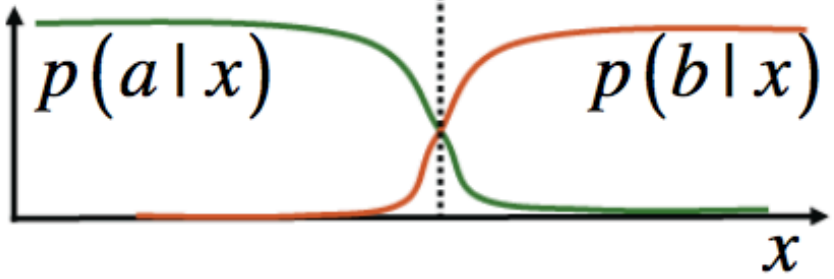


Likelihood



Likelihood $\times$ *Prior*

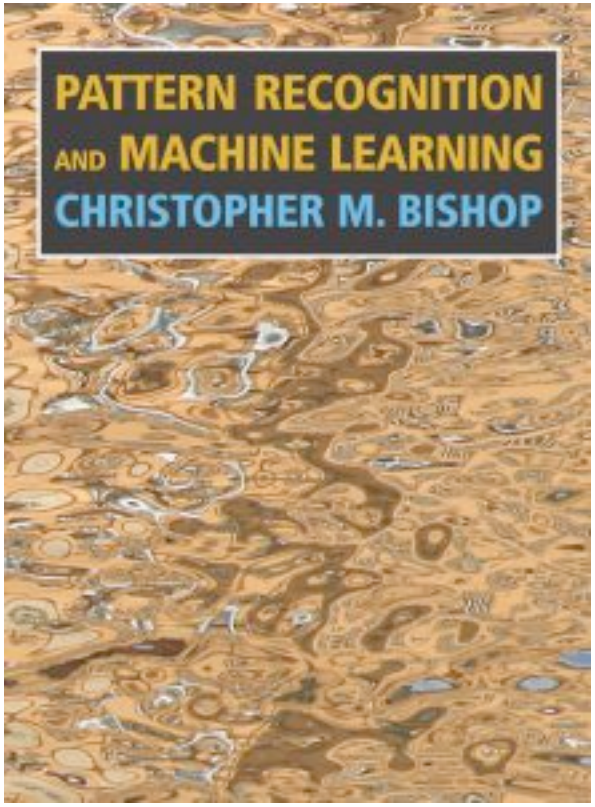
Decision boundary



$$Posterior = \frac{Likelihood \times Prior}{NormalizationFactor}$$

Reference for the next few slides:

C. Bishop, Pattern Recognition and Machine learning



Decision Theory (Section 1.5.1 of C. Bishop's book)

Goal: Minimize the probability of a misclassification

$$p(\text{correct}) = \sum_{k=1}^{K} p(\mathbf{x} \in \mathcal{R}_k, \mathcal{C}_k)$$

$$= \sum_{k=1}^{K} \int_{\mathcal{R}_k} p(\mathbf{x}, \mathcal{C}_k)\, \mathrm{d}\mathbf{x}$$

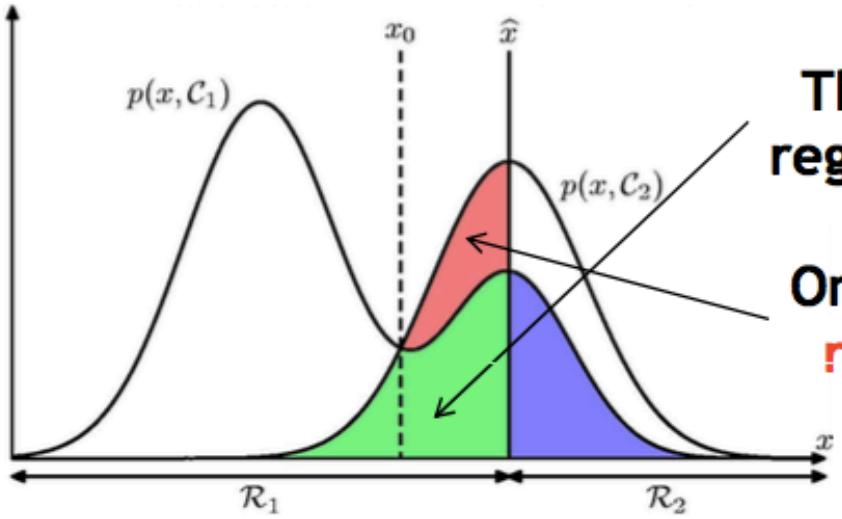
Probability of mistake, 2 classes

$$p(\text{mistake}) = p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1)$$

$$= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\, \mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\, \mathrm{d}\mathbf{x}.$$

Decision Theory

Goal: Minimize the probability of a misclassification

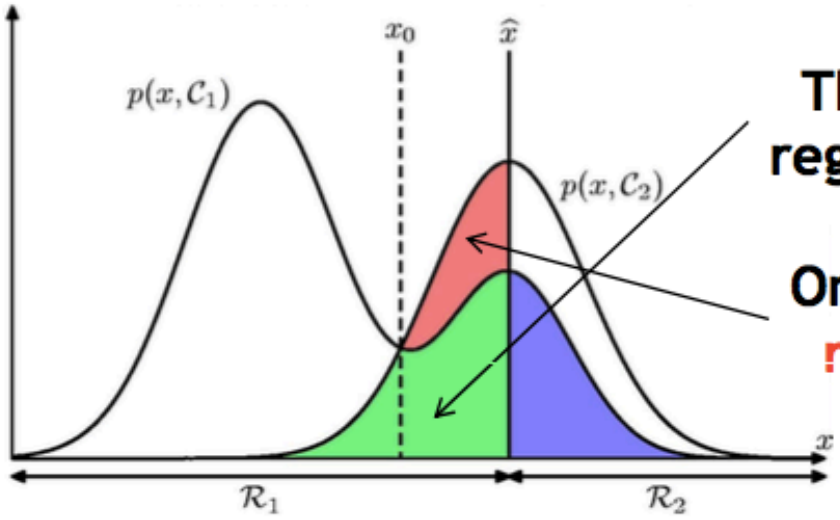


The green and blue regions stay constant

Only the size of the red region varies!

Decision Theory

Goal: Minimize the probability of a misclassification



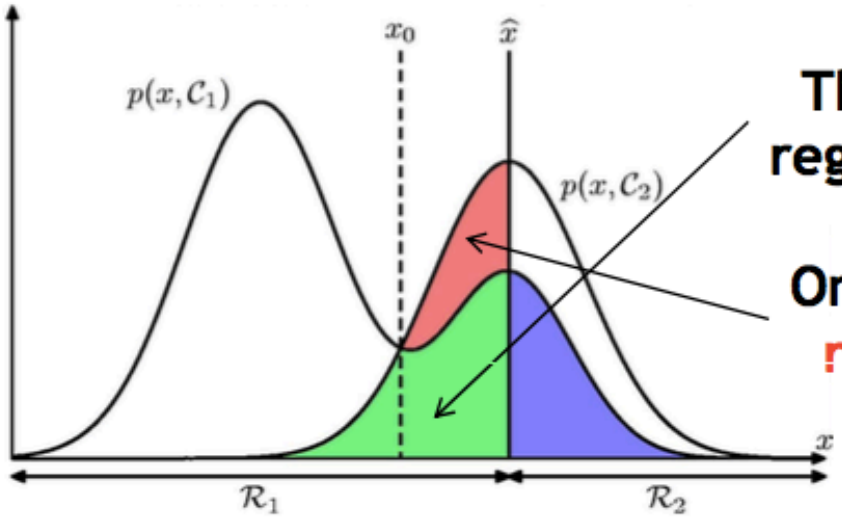
The green and blue regions stay constant

Only the size of the red region varies!

Figure 1.24 Schematic illustration of the joint probabilities $p(x, \mathcal{C}_k)$ for each of two classes plotted against x, together with the decision boundary $x = \widehat{x}$. Values of $x \geqslant \widehat{x}$ are classified as class $\mathcal{C}_2$ and hence belong to decision region $\mathcal{R}_2$, whereas points $x < \widehat{x}$ are classified as $\mathcal{C}_1$ and belong to $\mathcal{R}_1$. Errors arise from the blue, green, and red regions, so that for $x < \widehat{x}$ the errors are due to points from class $\mathcal{C}_2$ being misclassified as $\mathcal{C}_1$ (represented by the sum of the red and green regions), and conversely for points in the region $x \geqslant \widehat{x}$ the errors are due to points from class $\mathcal{C}_1$ being misclassified as $\mathcal{C}_2$ (represented by the blue region). As we vary the location $\widehat{x}$ of the decision boundary, the combined areas of the blue and green regions remains constant, whereas the size of the red region varies. The optimal choice for $\widehat{x}$ is where the curves for $p(x, \mathcal{C}_1)$ and $p(x, \mathcal{C}_2)$ cross, corresponding to $\widehat{x} = x_0$, because in this case the red region disappears. This is equivalent to the minimum misclassification rate decision rule, which assigns each value of x to the class having the higher posterior probability $p(\mathcal{C}_k|x)$.

Decision Theory

Goal: Minimize the probability of a misclassification



The green and blue regions stay constant

Only the size of the red region varies!

$$
\begin{aligned}
p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\
&= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\, \mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\, \mathrm{d}\mathbf{x}. \\
&= \int_{\mathcal{R}_1} p(\mathcal{C}_2|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} + \int_{\mathcal{R}_2} p(\mathcal{C}_1|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}
\end{aligned}
$$

Bayes-optimal classification rule

Optimal decision rule

- Decide for C_1 if

$$p(\mathcal{C}_1|x) > p(\mathcal{C}_2|x)$$

Bayes-optimal classification rule

Optimal decision rule

- Decide for C_1 if

$$p(\mathcal{C}_1|x) > p(\mathcal{C}_2|x)$$

- This is equivalent to

$$p(x|\mathcal{C}_1)p(\mathcal{C}_1) > p(x|\mathcal{C}_2)p(\mathcal{C}_2)$$

Bayes-optimal classification rule

Optimal decision rule

- Decide for C_1 if

$$p(\mathcal{C}_1|x) > p(\mathcal{C}_2|x)$$

- This is equivalent to

$$p(x|\mathcal{C}_1)p(\mathcal{C}_1) > p(x|\mathcal{C}_2)p(\mathcal{C}_2)$$

- Which is again equivalent to (Likelihood-Ratio test)

$$\frac{p(x|\mathcal{C}_1)}{p(x|\mathcal{C}_2)} > \underbrace{\frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)}}_{\text{Decision threshold } \theta}$$

Extension to more than 2 classes

Decide for class k whenever it has the greatest posterior probability of all classes:

$$p(\mathcal{C}_k|x) > p(\mathcal{C}_j|x) \quad \forall j \neq k$$

Extension to more than 2 classes

Decide for class k whenever it has the greatest posterior probability of all classes:

$$p(\mathcal{C}_k|x) > p(\mathcal{C}_j|x) \quad \forall j \neq k$$

$$p(x|\mathcal{C}_k)p(\mathcal{C}_k) > p(x|\mathcal{C}_j)p(\mathcal{C}_j) \quad \forall j \neq k$$

Extension to more than 2 classes

Decide for class k whenever it has the greatest posterior probability of all classes:

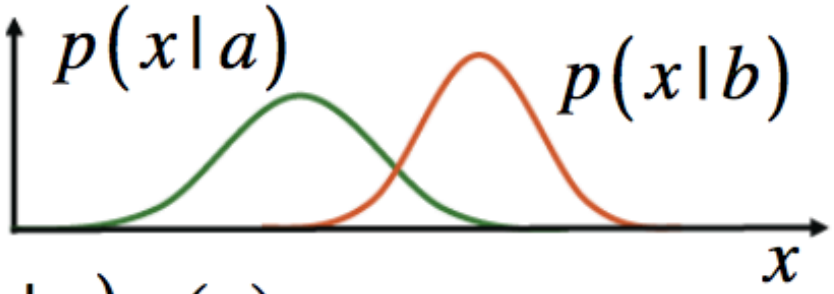
$$p(\mathcal{C}_k|x) > p(\mathcal{C}_j|x) \quad \forall j \neq k$$

$$p(x|\mathcal{C}_k)p(\mathcal{C}_k) > p(x|\mathcal{C}_j)p(\mathcal{C}_j) \quad \forall j \neq k$$

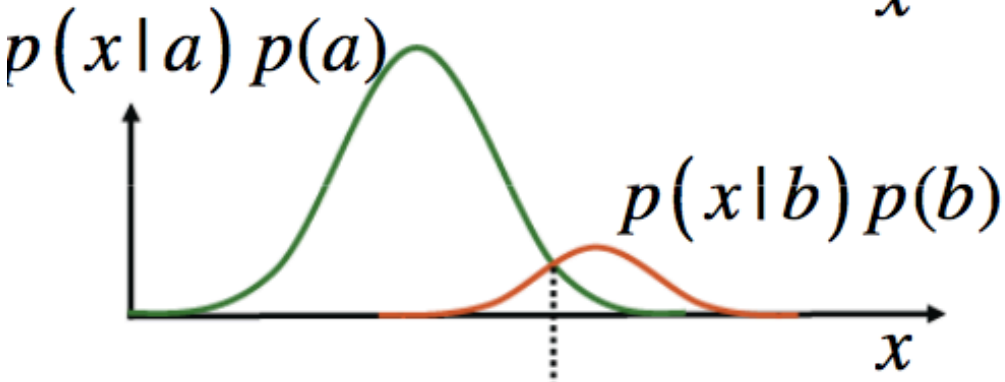
Likelihood-ratio test

$$\frac{p(x|\mathcal{C}_k)}{p(x|\mathcal{C}_j)} > \frac{p(\mathcal{C}_j)}{p(\mathcal{C}_k)} \quad \forall j \neq k$$

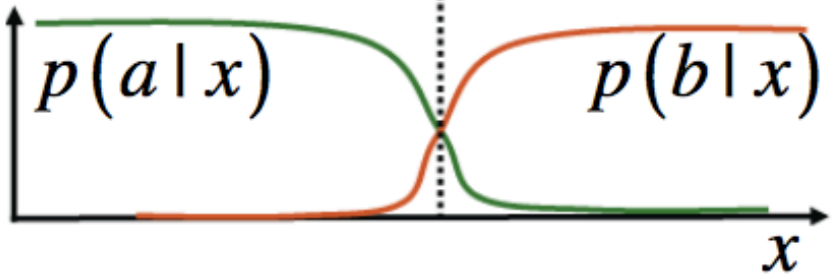
Binary classification problem & posteriors, revisited



Likelihood



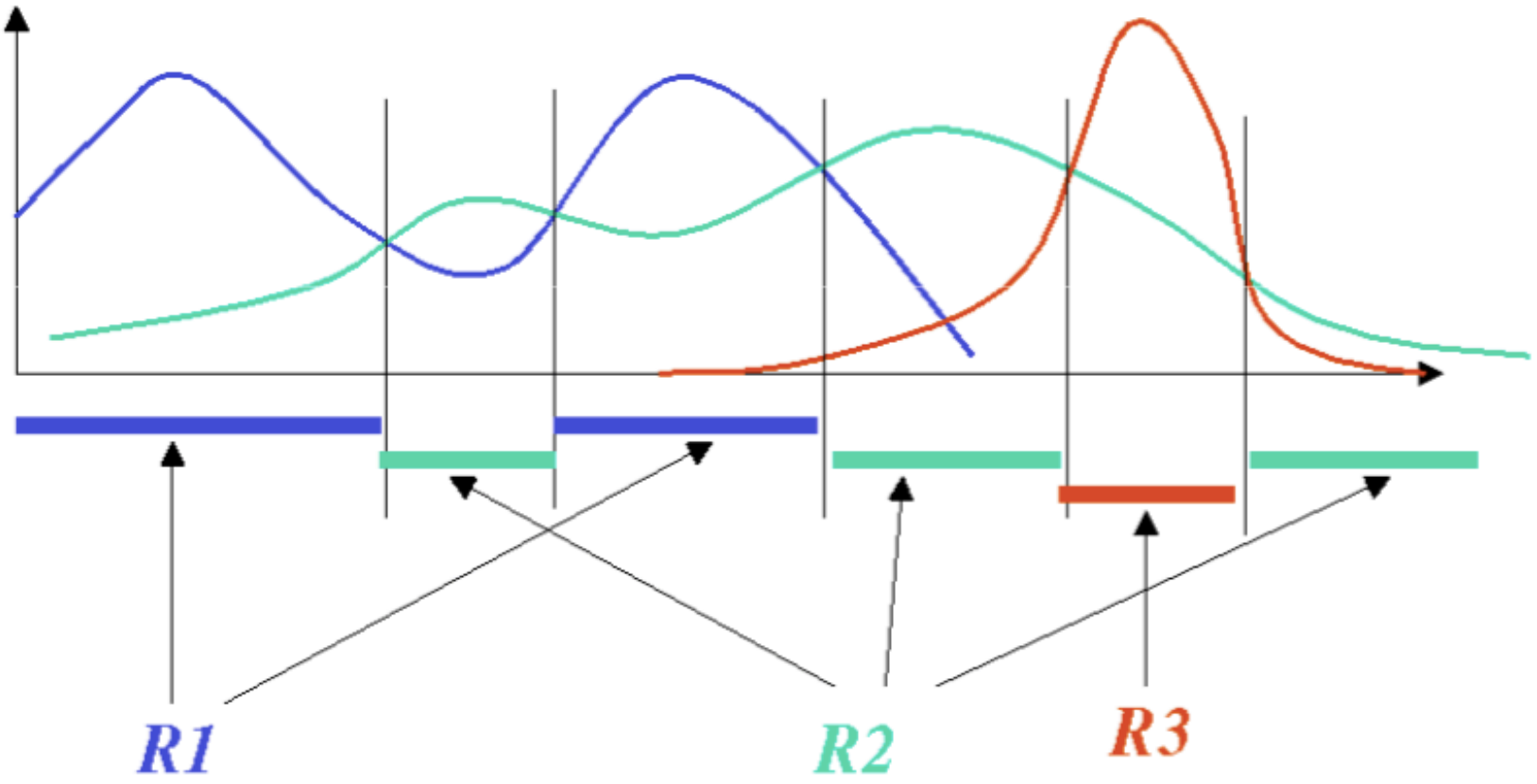
Likelihood $\times$ *Prior*



$$Posterior = \frac{Likelihood \times Prior}{NormalizationFactor}$$

Extension to more than 2 classes

Decision regions: $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$, ...



Classifying with loss functions

Generalization to decisions with a loss function

- Differentiate between the possible decisions and the possible true classes.
- Example: medical diagnosis
  - Decisions: sick or healthy (or: further examination necessary)
  - Classes: patient is sick or healthy

Classifying with loss functions

Generalization to decisions with a loss function

- Differentiate between the possible decisions and the possible true classes.
- Example: medical diagnosis
  - Decisions: sick or healthy (or: further examination necessary)
  - Classes: patient is sick or healthy
- The cost may be asymmetric:

$$loss(decision = healthy | patient = sick) >> loss(decision = sick | patient = healthy)$$

Classifying with loss functions

In general, we can formalize this by introducing a loss matrix L_{kj}

$$L_{kj} = loss\ for\ decision\ C_j\ if\ truth\ is\ C_k.$$

Classifying with loss functions

In general, we can formalize this by introducing a loss matrix L_{kj}

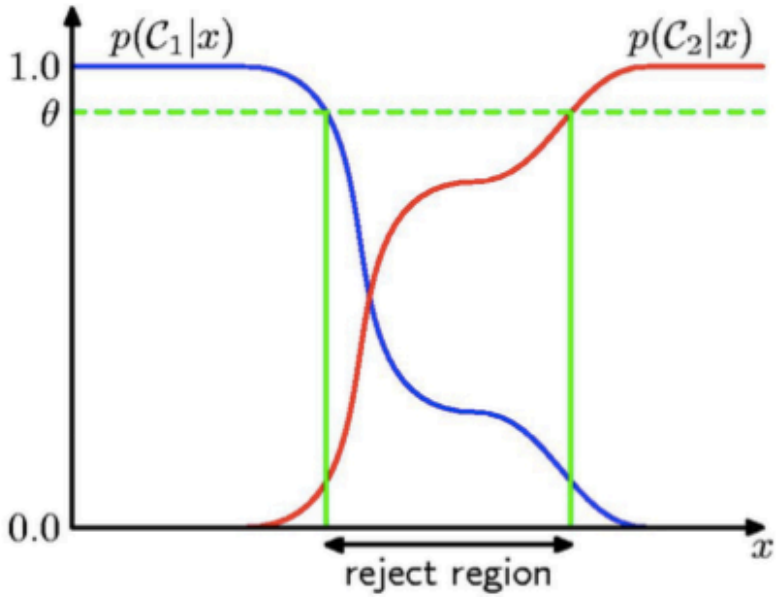
$$L_{kj} = loss\ for\ decision\ C_j\ if\ truth\ is\ C_k.$$

Example: cancer diagnosis

$$L_{cancer\ \ diagnosis} = \begin{array}{c} \\ \text{Truth} \\ \end{array} \begin{array}{c} \text{Decision} \\ \begin{array}{cc} \text{cancer} & \text{normal} \\ \end{array} \\ \begin{array}{l} \text{cancer} \\ \text{normal} \end{array} \begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix} \end{array}$$

Reject option

- 2 classes, 3 actions



Classification errors arise from regions where the largest posterior probability $p(\mathcal{C}_k|\mathbf{x})$ is significantly less than 1.

- **These are the regions where we are relatively uncertain about class membership.**
- **For some applications, it may be better to reject the automatic decision entirely in such a case and e.g. consult a human expert.**

Minimizing the expected loss

Optimal solution is the one that minimizes the loss.

- **But: loss function depends on the true class, which is unknown**

Minimizing the expected loss

Optimal solution is the one that minimizes the loss.

- But: loss function depends on the true class, which is unknown

Solution: Minimize the expected loss

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) \, \mathrm{d}\mathbf{x}$$

Minimizing the expected loss

Optimal solution is the one that minimizes the loss.

- But: loss function depends on the true class, which is unknown

Solution: Minimize the expected loss

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) \, \mathrm{d}\mathbf{x}$$

This can be done by choosing the regions $\mathcal{R}_j$ such that

$$\mathbb{E}[L] = \sum_k L_{kj} p(\mathcal{C}_k|\mathbf{x})$$

which is easy to do once we know the posterior class probabilities $p(\mathcal{C}_k|\mathbf{x})$.

Minimizing the expected loss

Example:

- 2 Classes: C_1, C_2
- 2 Decision: α_1, α_2
- Loss function: $L(\alpha_j|C_k) = L_{kj}$

Minimizing the expected loss

Example:

- 2 Classes: C_1, C_2
- 2 Decision: α_1, α_2
- Loss function: $L(\alpha_j|\mathcal{C}_k) = L_{kj}$
- Expected loss (= risk R) for the two decisions:

$$\mathbb{E}_{\alpha_1}[L] = R(\alpha_1|\mathbf{x}) = L_{11}p(\mathcal{C}_1|\mathbf{x}) + L_{21}p(\mathcal{C}_2|\mathbf{x})$$
$$\mathbb{E}_{\alpha_2}[L] = R(\alpha_2|\mathbf{x}) = L_{12}p(\mathcal{C}_1|\mathbf{x}) + L_{22}p(\mathcal{C}_2|\mathbf{x})$$

Minimizing the expected loss

Example:

- 2 Classes: C_1, C_2
- 2 Decision: α_1, α_2
- Loss function: $L(\alpha_j|\mathcal{C}_k) = L_{kj}$
- Expected loss (= risk R) for the two decisions:

$$\mathbb{E}_{\alpha_1}[L] = R(\alpha_1|\mathbf{x}) = L_{11}p(\mathcal{C}_1|\mathbf{x}) + L_{21}p(\mathcal{C}_2|\mathbf{x})$$
$$\mathbb{E}_{\alpha_2}[L] = R(\alpha_2|\mathbf{x}) = L_{12}p(\mathcal{C}_1|\mathbf{x}) + L_{22}p(\mathcal{C}_2|\mathbf{x})$$

Goal: Decide such that expected loss is minimized

- I.e. decide α_1 if $R(\alpha_2|\mathbf{x}) > R(\alpha_1|\mathbf{x})$

Minimizing the expected loss

$$R(\alpha_2|\mathbf{x}) \;>\; R(\alpha_1|\mathbf{x})$$

Minimizing the expected loss

$$
\begin{aligned}
R(\alpha_2|\mathbf{x}) &> R(\alpha_1|\mathbf{x}) \\
L_{12}p(\mathcal{C}_1|\mathbf{x}) + L_{22}p(\mathcal{C}_2|\mathbf{x}) &> L_{11}p(\mathcal{C}_1|\mathbf{x}) + L_{21}p(\mathcal{C}_2|\mathbf{x})
\end{aligned}
$$

Minimizing the expected loss

$$
\begin{aligned}
R(\alpha_2|\mathbf{x}) &> R(\alpha_1|\mathbf{x}) \\
L_{12}p(\mathcal{C}_1|\mathbf{x}) + L_{22}p(\mathcal{C}_2|\mathbf{x}) &> L_{11}p(\mathcal{C}_1|\mathbf{x}) + L_{21}p(\mathcal{C}_2|\mathbf{x}) \\
(L_{12} - L_{11})p(\mathcal{C}_1|\mathbf{x}) &> (L_{21} - L_{22})p(\mathcal{C}_2|\mathbf{x})
\end{aligned}
$$

Minimizing the expected loss

$$
\begin{aligned}
R(\alpha_2|\mathbf{x}) &> R(\alpha_1|\mathbf{x}) \\
L_{12}p(\mathcal{C}_1|\mathbf{x}) + L_{22}p(\mathcal{C}_2|\mathbf{x}) &> L_{11}p(\mathcal{C}_1|\mathbf{x}) + L_{21}p(\mathcal{C}_2|\mathbf{x}) \\
(L_{12} - L_{11})p(\mathcal{C}_1|\mathbf{x}) &> (L_{21} - L_{22})p(\mathcal{C}_2|\mathbf{x}) \\
\frac{(L_{12} - L_{11})}{(L_{21} - L_{22})} &> \frac{p(\mathcal{C}_2|\mathbf{x})}{p(\mathcal{C}_1|\mathbf{x})} = \frac{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}
\end{aligned}
$$

Minimizing the expected loss

$$
\begin{aligned}
R(\alpha_2|\mathbf{x}) &> R(\alpha_1|\mathbf{x}) \\
L_{12}p(\mathcal{C}_1|\mathbf{x}) + L_{22}p(\mathcal{C}_2|\mathbf{x}) &> L_{11}p(\mathcal{C}_1|\mathbf{x}) + L_{21}p(\mathcal{C}_2|\mathbf{x}) \\
(L_{12} - L_{11})p(\mathcal{C}_1|\mathbf{x}) &> (L_{21} - L_{22})p(\mathcal{C}_2|\mathbf{x}) \\
\frac{(L_{12} - L_{11})}{(L_{21} - L_{22})} &> \frac{p(\mathcal{C}_2|\mathbf{x})}{p(\mathcal{C}_1|\mathbf{x})} = \frac{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)} \\
\frac{p(\mathbf{x}|\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)} &> \frac{(L_{21} - L_{22})}{(L_{12} - L_{11})}\frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)}
\end{aligned}
$$

⇒ **Adapted decision rule taking into account the loss.**

Discriminant functions

Formulate classification in terms of comparisons

- Discriminant functions

$$y_1(x), \ldots, y_K(x)$$

- Classify x as class C_k if

$$y_k(x) > y_j(x) \quad \forall j \neq k$$

Discriminant functions

Formulate classification in terms of comparisons

- Discriminant functions

$$y_1(x), \ldots, y_K(x)$$

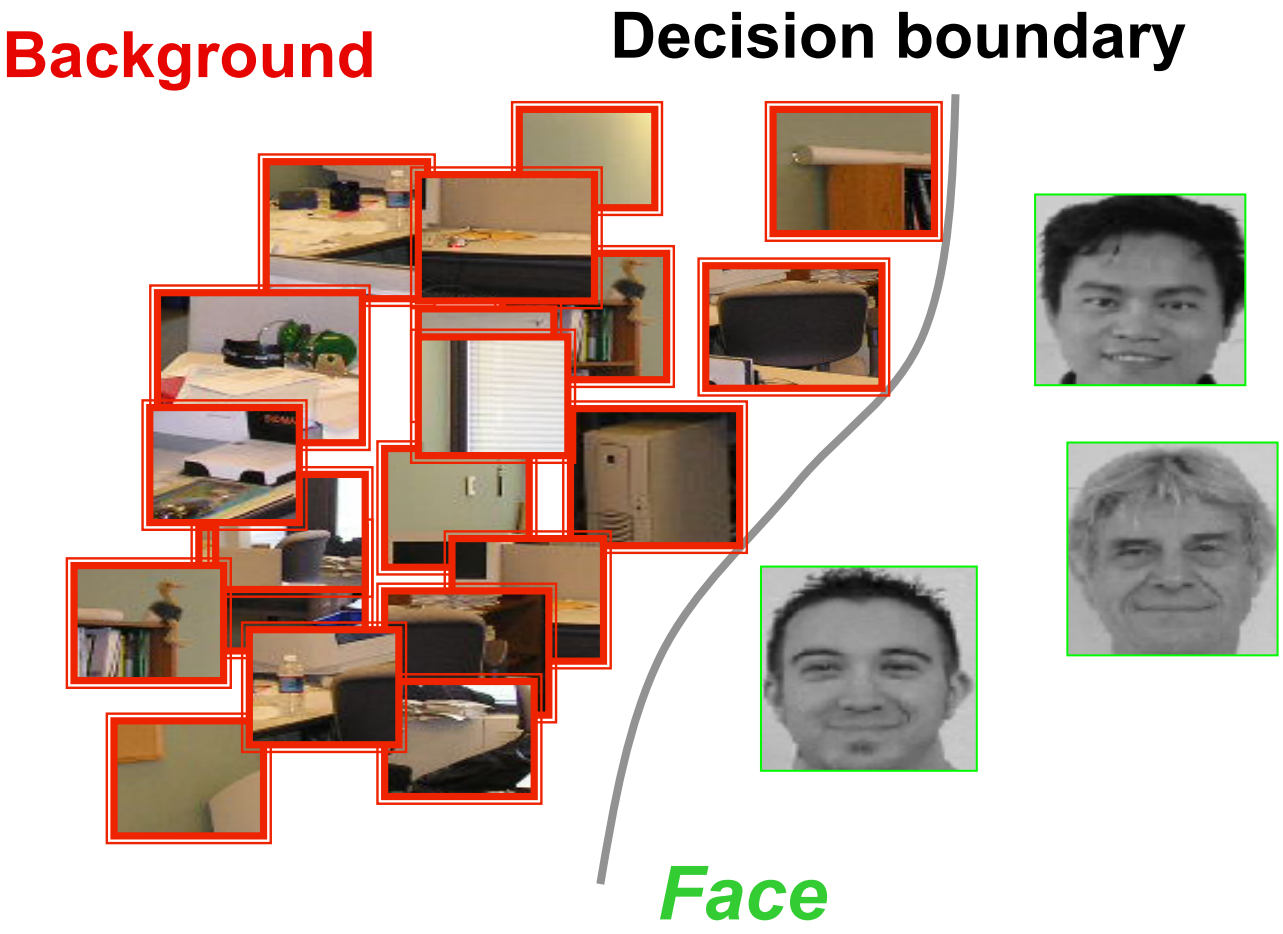
- Classify x as class C_k if

$$y_k(x) > y_j(x) \quad \forall j \neq k$$

Examples (Bayes Decision Theory)

$$\begin{aligned} y_k(x) &= p(\mathcal{C}_k|x) \\ y_k(x) &= p(x|\mathcal{C}_k)p(\mathcal{C}_k) \\ y_k(x) &= \log p(x|\mathcal{C}_k) + \log p(\mathcal{C}_k) \end{aligned}$$

`Faceness': discriminant function



Generative vs. Discriminative classifiers

$$y_k(x) \propto p(x|\mathcal{C}_k)p(\mathcal{C}_k)$$

- First determine the class-conditional densities for each class individually and separately infer the prior class probabilities.
- Then use Bayes' theorem to determine class membership.

$\Rightarrow$ *Generative methods*

Generative vs. Discriminative classifiers

$$y_k(x) \propto p(x|\mathcal{C}_k)p(\mathcal{C}_k)$$

- First determine the class-conditional densities for each class individually and separately infer the prior class probabilities.
- Then use Bayes' theorem to determine class membership.

$\Rightarrow$ *Generative methods*

$$y_k(x) = p(\mathcal{C}_k|x)$$

- First solve the inference problem of determining the posterior class probabilities.
- Then use decision theory to assign each new x to its class.

$\Rightarrow$ *Discriminative methods*

Generative vs. Discriminative classifiers

$$y_k(x) \propto p(x|\mathcal{C}_k)p(\mathcal{C}_k)$$

- First determine the class-conditional densities for each class individually and separately infer the prior class probabilities.
- Then use Bayes' theorem to determine class membership.

$\Rightarrow$ *Generative methods*

$$y_k(x) = p(\mathcal{C}_k|x)$$

- First solve the inference problem of determining the posterior class probabilities.
- Then use decision theory to assign each new x to its class.

$\Rightarrow$ *Discriminative methods*

Alternative

- Directly find a discriminant function $y_k(x)$ which maps each input x directly onto a class label.

Tentative Course Schedule: generative & discriminative

- 1st week: Introduction, fundamental concepts
- 2nd week: Linear Regression
- 3rd week: Logistic Regression
- 4th week: Support Vector Machines
- 5th week: Ensemble Models (Adaboost, Random Forests)
- 6th week: Unsupervised learning (K-means, PCA, Sparse Coding)
- 7th week: Deep Learning (neural networks, backpropagation, SGD)
- 8th week: Probabilistic modelling (hidden variable models, EM)
- 9th week: Intro to Structured Prediction (Random Fields, Graphical Models)
- 10th week: review and applications