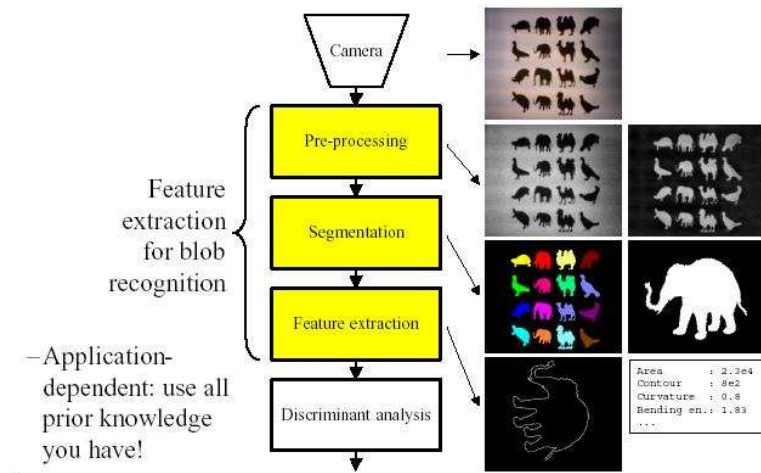


COMBINING CLASSIFIERS: BAGGING and BOOSTING

Danica Kragic

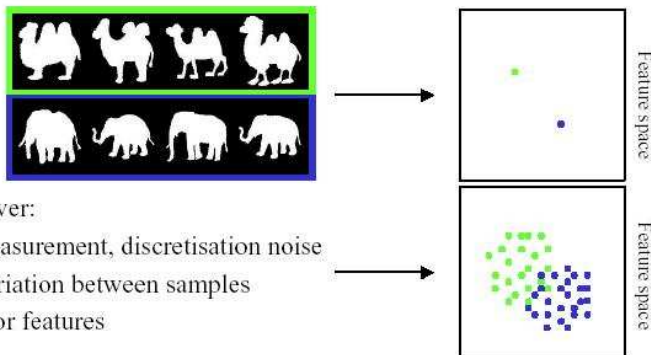
Pattern Recognition



Danica Kragic, 2004

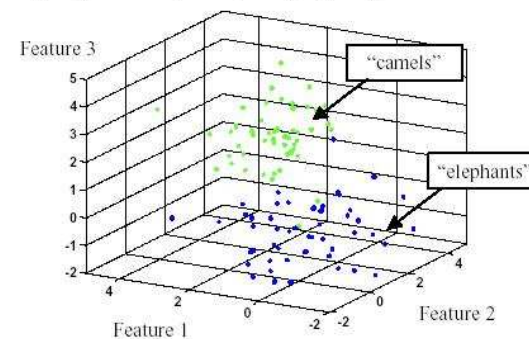
Danica Kragic, 2004

- Sensors give *measurements*, which should be converted to *features* (can be pure measurements, e.g. pixels!)
- Ideally, a feature value is identical for all *samples* in one *class*



- However:
 - Measurement, discretisation noise
 - Variation between samples
 - Poor features

- End result: a k -dimensional space,
 - in which each dimension is a **feature**
 - containing N (labelled) **samples** (objects)



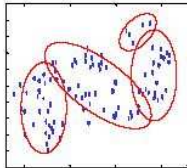
Danica Kragic, 2004

Danica Kragic, 2004

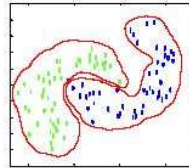
-Pattern recognition

What is our basic problem?

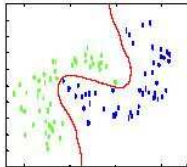
Clustering:
find natural groups of samples in unlabelled data



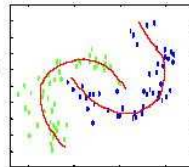
Density estimation:
make a statistical model of the data



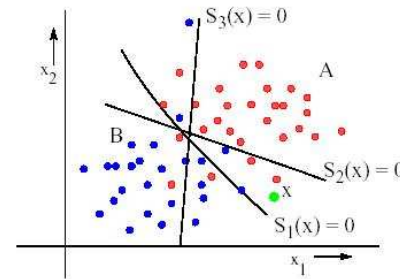
Classification:
find functions separating the classes



Regression:
fit lines or other functions to data (not in this course)



Danica Kragic, 2004



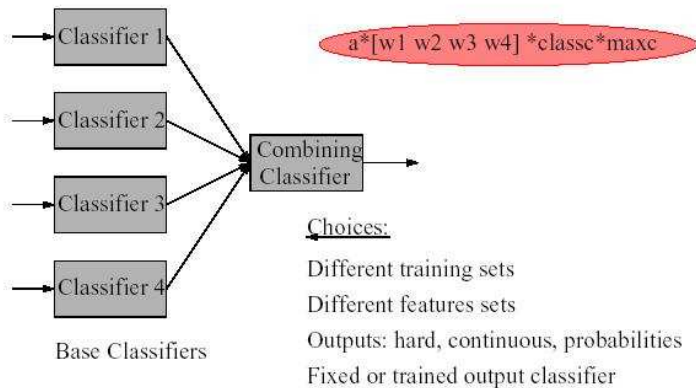
Confidences (Posterior Probabilities):

- Prob ($x \in A$ | $S_1(x)$) = ??
- Prob ($x \in A$ | $S_2(x)$) = ??
- Prob ($x \in A$ | $S_3(x)$) = ??

How to combine $S_1(x)$, $S_2(x)$ and $S_3(x)$??

Danica Kragic, 2004

Pattern Recognition



Danica Kragic, 2004

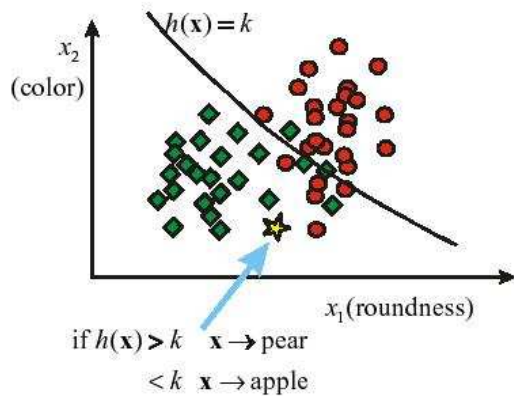
Based on:

- Class-conditional pdfs:
 - Assume a model for pdf and estimate parameters
 - No model assumption: histogram methods, k-nearest neighbours, kernel methods, Bayesian networks
- Discriminant functions: Linear Discriminant Functions (LDA), Support Vector Machines (SVM)
- Similarity measures with stored samples

Danica Kragic, 2004

Discriminant functions

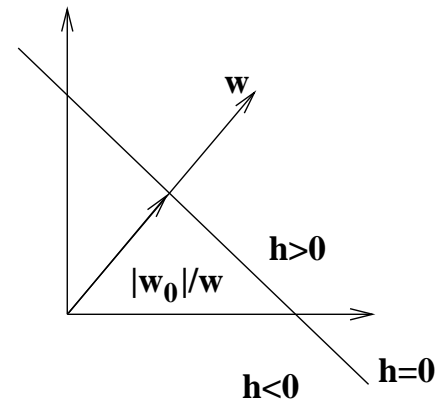
1. Choose class of decision functions in feature space.
2. Estimate the function parameters from the training set.
3. Classify a new pattern on the basis of this decision rule.



Danica Kragic, 2004

Linear Discriminant Functions

$$\mathbf{h}(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 = \sum_i w_i x_i + w_0$$



Danica Kragic, 2004

Linear Discriminant Functions

$$h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$

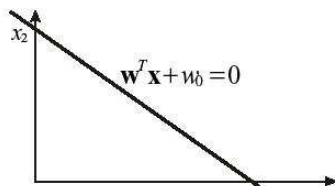
is a linear function of the features \mathbf{x} ,
where \mathbf{w} determines the slope of the
plane and w_0 determines the offset.

If we define

$$\mathbf{z} = (1, x_1, \dots, x_p)^T, \quad \mathbf{v} = (w_0, w_1, \dots, w_p)^T$$

then the function can be written as

$$h(\mathbf{x}) = \mathbf{v}^T \mathbf{z}.$$



Danica Kragic, 2004

Linear Discriminant Functions

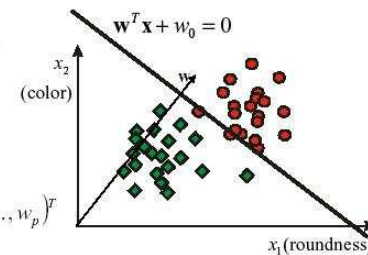
$$h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \mathbf{x} \in \begin{cases} \omega_1 \\ \omega_2 \end{cases}$$

or, shorter,

$$\mathbf{v}^T \mathbf{z} \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \mathbf{x} \in \begin{cases} \omega_1 \\ \omega_2 \end{cases}$$

with

$$\mathbf{z} = (1, x_1, \dots, x_p)^T, \quad \mathbf{v} = (w_0, w_1, \dots, w_p)^T$$



Webb writes this as follows. Suppose

$\mathbf{y}_i = \mathbf{z}_i$ for $\mathbf{x}_i \in \omega_1$ and $\mathbf{y}_i = -\mathbf{z}_i$ for $\mathbf{x}_i \in \omega_2$

then we seek for a value of \mathbf{v} for which

$\mathbf{v}^T \mathbf{y}_i > 0$ for all \mathbf{y}_i corresponding to the \mathbf{x}_i in the train set.

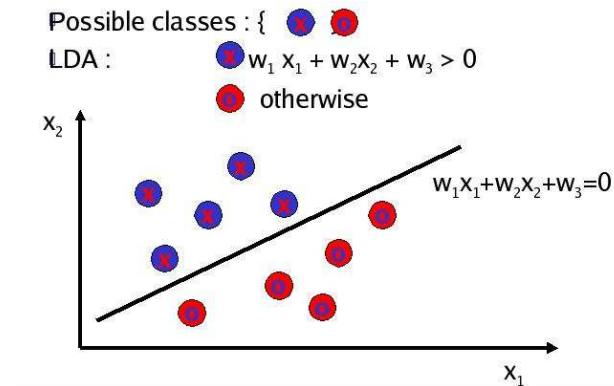
Danica Kragic, 2004

Classification problem

- Assume a set S of N instances $x_i \in X$ each belonging to one of M classes $\{c^1, \dots, c^M\}$
- The training set consists of pairs $\langle x_i, c_i \rangle$
- A classifier C assigns a classification $C(x) \in \{c^1, \dots, c^M\}$ to an instance x
- The classifier learned in trial t is denoted C^t while C^* is the composite bagged or boosted classifier

Danica Kragic, 2004

Linear Discriminant Analysis



Danica Kragic, 2004

Bagging and Boosting

- **Bagging and Boosting aggregate multiple hypotheses generated by the same learning algorithm invoked over different distributions of training data** [Breiman 1996, Freund and Schapire 1996]
- **Bagging and Boosting generate a classifier with a smaller error on the training data as it combines multiple hypotheses which individually have a large error**

Danica Kragic, 2004

Bagging and Boosting

- Bagging replicates training sets by sampling with replacement from the training instances.
- Boosting uses all instances but weights them and therefore produces different classifiers .
- Classifiers are then combined by voting to create a composite classifier .
- Bagging: classifiers have same votes;
- Boosting: vote dependant on the classifiers' accuracy

Danica Kragic, 2004

Bagging

- For each trial, generate *new* training set of size N with replacements (multiple occurrence of instances)
- A classifier C^t is generated for each training set
- Final classifier C^* is formed by aggregating the T classifiers
- An instance x is classified by counting votes for which

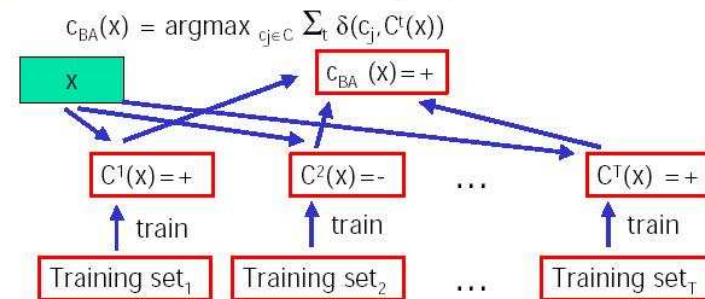
$$C^t(x) = k$$

and $C^*(x)$ represents the class with most votes

Danica Kragic, 2004

Bagging

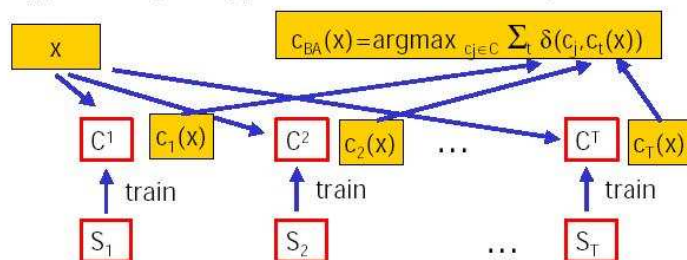
- For each trial $t=1,2,\dots,T$ create a bootstrap sample S^t .
- Obtain an hypothesis C^t on the bootstrap sample S^t .
- The final classification is the majority class



Danica Kragic, 2004

Bagging

- From the overall training S set randomly sample (with replacement) T different training sets S_1, \dots, S_T of size N .
- For each sample set S_t obtain a hypothesis C^t .
- To an unseen instance x assign the majority classification $c_{BA}(x)$ among the hypotheses C^t classifications $c_t(x)$.



Danica Kragic, 2004

Bagging

..... requires "instable" classifiers such as decision trees or concept learners

Danica Kragic, 2004

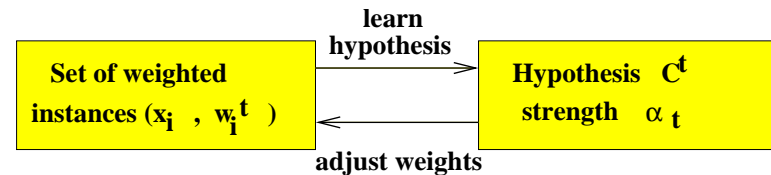
Boosting

- Boosting maintains a weight w_i for each instance $\langle x_i, \dots, c_i \rangle$ in the training set
- The higher the weight w_i , the more the instance x_i influences the next hypotheses learned
- At each trial, the weights are adjusted to reflect the performance of the previously learned hypothesis, with the result that the weight of correctly classified instances is decreased and the weight of incorrectly classified instances is increased

Danica Kragic, 2004

Boosting

- Construct a hypothesis C^t from the current distribution of instances described by w^t
- Adjust the weights according to the classification error ε^t of classifier C^t
- The strength α_t of a hypothesis depends on its training error $\alpha_t = \frac{1}{2} \ln((1 - \varepsilon_t)/\varepsilon_t)$



Danica Kragic, 2004

Boosting

- The final hypothesis c_{BO} aggregates the individual hypotheses C^t by weighted voting

$$c_{BO}(x) = \operatorname{argmax}_{c_j \in \mathcal{C}} \sum_{t=1}^T \alpha_t \delta(c_j, c_t(x))$$

- Each hypothesis vote is a function of its accuracy

Danica Kragic, 2004

Boosting

- Let w_i^t denote the weight of an instance x_i at trial t , for every x_i , $w_i^1 = 1/N$. The weight w_i^t reflects the importance (e.g. probability of occurrence) of the instance x_i in the sample set S^t
- At each trial $t = 1, \dots, T$, an hypothesis C^t is constructed from the given instances under the distribution w^t . This requires that the learning algorithm can deal with fractional examples.

Danica Kragic, 2004

Boosting

- The error of the hypothesis C^t is measured with respect to the weights

$$\epsilon_t = \sum_{\forall i \text{ such that } C^t(x_i) \neq c_i} w_i^t / \sum_i w_i^t$$

$$\alpha_t = \frac{1}{2} \ln((1 - \epsilon_t) / \epsilon_t)$$

- Update the weights w_i^t of *correctly* and *incorrectly* classified instances by

$$w_i^{t+1} = w_i^t e^{-\alpha_t} \text{ if } C^t(x_i) = c_i$$

$$w_i^{t+1} = w_i^t e^{\alpha_t} \text{ if } C^t(x_i) \neq c_i$$

- Afterwards normalize the w_i^{t+1} such that they form a proper distribution $\sum_i w_i^{t+1} = 1$

Danica Kragic, 2004

Boosting

Given $\{(x_1, c_1), \dots, (x_m, c_m)\}$, initialize $w_i^1 = 1/m$

For $t = 1, \dots, T$

train weak learner using distribution w_i^t

get weak hypothesis $C^t : X \rightarrow C$ with error

$$\epsilon_t = \sum_{\forall i \text{ such that } C^t(x_i) \neq c_i} w_i^t / \sum_i w_i^t$$

choose $\alpha_t = \frac{1}{2} \ln((1 - \epsilon_t) / \epsilon_t)$

update

$$w_i^{t+1} = w_i^t e^{-\alpha_t} \text{ if } C^t(x_i) = c_i$$

$$w_i^{t+1} = w_i^t e^{\alpha_t} \text{ if } C^t(x_i) \neq c_i$$

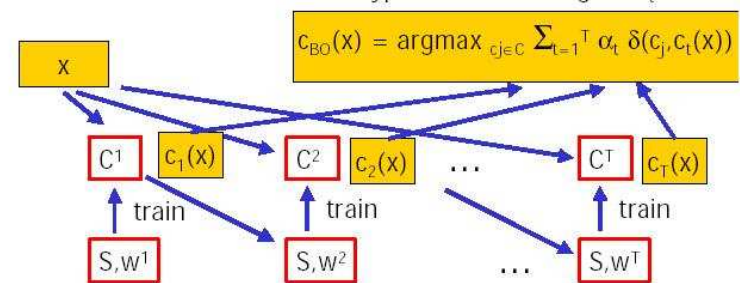
Output the final hypothesis

$$c_{BO}(x) = \operatorname{argmax}_{c_j \in C} \sum_{t=1}^T \alpha_t \delta(c_j, c_t(x))$$

Danica Kragic, 2004

Boosting

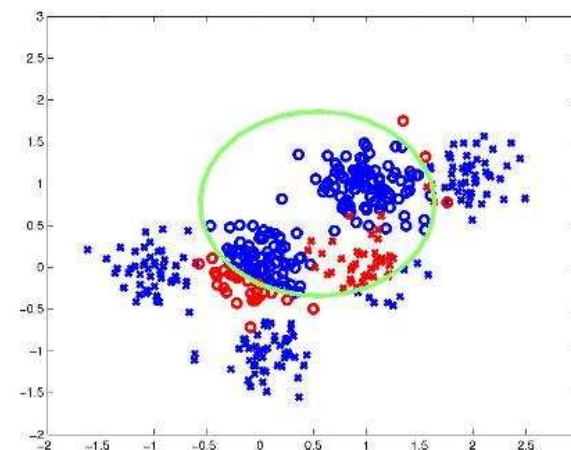
- The classification $c_{BO}(x)$ of the boosted hypothesis is obtained by summing the votes of the hypotheses C^1, C^2, \dots, C^T where the vote of each hypothesis C^t is weight α_t



Danica Kragic, 2004

Bayes MAP Hypothesis

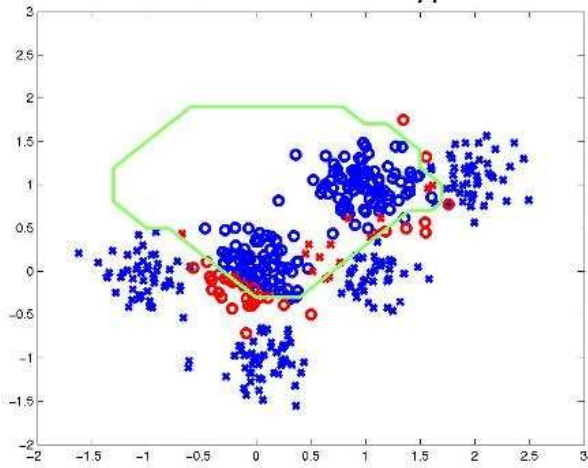
Bayes MAP hypothesis for two classes x and o
red: incorrect classified instances



Danica Kragic, 2004

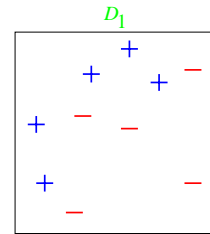
Boosted Bayes MAP Hypothesis

- Boosted Bayes MAP hypothesis has more complex decision surface than individual hypothesis alone



Danica Kragic, 2004

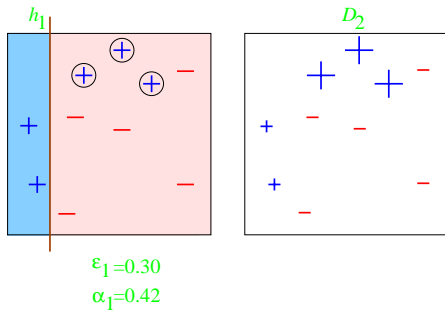
Toy Example



weak classifiers = vertical or horizontal half-planes

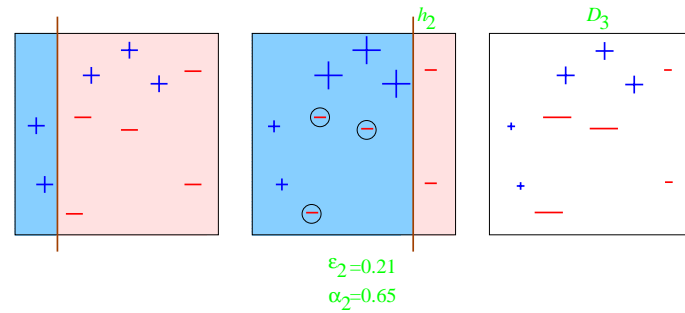
Danica Kragic, 2004

Round 1



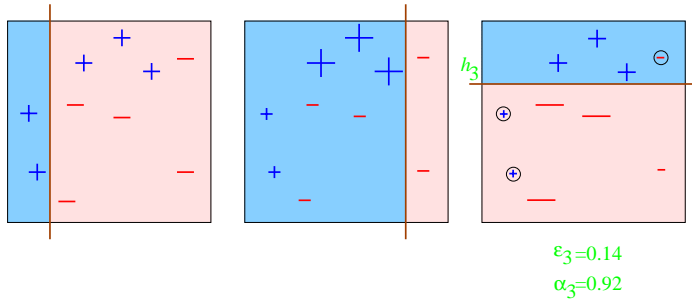
Danica Kragic, 2004

Round 2



Danica Kragic, 2004

Round 3



Danica Kragic, 2004

Final Classifier

$$H_{\text{final}} = \text{sign} \left(0.42 \begin{array}{|c|} \hline \text{blue} \\ \hline \text{pink} \\ \hline \end{array} + 0.65 \begin{array}{|c|} \hline \text{blue} \\ \hline \text{pink} \\ \hline \end{array} + 0.92 \begin{array}{|c|} \hline \text{blue} \\ \hline \text{pink} \\ \hline \end{array} \right)$$

$$= \begin{array}{|c|} \hline \text{blue} \\ \hline \text{pink} \\ \hline \end{array}$$

Danica Kragic, 2004