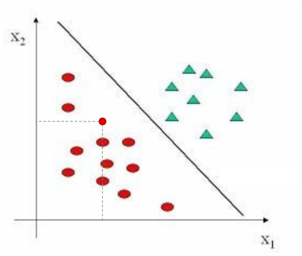
INF 5300 – Introduction II/II

- Feature-based classification basics (reminder)

Background material: The INF4300 lecture notes and supporting material on classification.

All there is to it I / II



All there is to it II / II

- Task: Classify object $x = \{x_1,\ldots,x_n\}$ to one of K classes $\omega_1,\ldots\omega_K$
- *Decision rule* f(**x**)=ω_i divides the feature space into K disjoint subsets R_i, i=1,...K.
- The decision rule is often defined by K scalar *discriminant functions* $g_1(\mathbf{x}),\ldots.g_K(\mathbf{x})$
- The pattern **x** will be classified to the class whose discriminant function gives a maximum:
  $f(\mathbf{x})=\omega_i \Leftrightarrow g_i(\mathbf{x}) = \max_{j=1,\ldots K} g_j(\mathbf{x})$
- Discriminant hypersurfaces are thus defined by
  $g_i(\mathbf{x})-g_j(\mathbf{x})=0$
- Training data vs test/unseen data

Key concept!

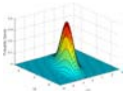
Density-based classifiers

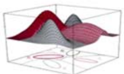
Model/estimate $p(\mathbf{x}|c)$

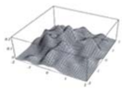
(Density-based classifiers interlude)

$p(\mathbf{x}|c)$

- Gaussian normal
  - Spherical (nearest mean)
  - Uncorrelated (naive Bayes)
  - Model-restrained
  - Equal-shaped classes
  - Full
- Mixture models (mixture of Gaussians)
- Kernel-based (e.g. Parzen-windows)







More complex

* Illustrations from Duda et al

Density-based classifiers cont.

Model/estimate $p(\mathbf{x}|c)$

Prior probability for class c

$$P(c|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|c)P(c)}{p(\mathbf{x}_i)}$$

Here this is our discriminant function

Bayes' rule

Our decision rule

$$f(\mathbf{x_i}) = \underset{c}{\operatorname{argmax}}\ p(c|\mathbf{x_i})$$

A "quite" popular density-based classifier

Classification with Gaussian distributions

- **x** normally distributed / Gaussian pdf:

$$p(x \mid \omega_s) = \frac{1}{(2\pi)^{d/2} |\Sigma_s|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_s)^t \Sigma_s^{-1} (x-\mu_s) \right]$$

- μ_s and Σ_s are generally not known, often using sample mean and sample covariance:

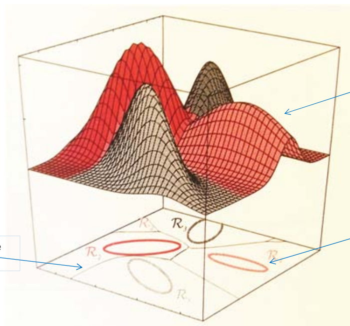
$$\hat{\mu}_s = \frac{1}{M_s} \sum_{m=1}^{M_s} x_m,$$

$$\hat{\Sigma}_s = \frac{1}{M_s} \sum_{m=1}^{M_s} (x_m - \hat{\mu}_s)(x_m - \hat{\mu}_s)^t$$

where the sum is over all training samples belonging to class s

Often easier to work with log $p(x|\omega_s)$

Gaussian distributions | example



Discriminant function for class 4

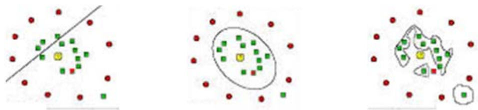
Equidensity contour, class 4

Discriminant surface (here, in 2D; curve)

(Duda et.al 2000, fig 2.16)

Model complexity

- We seek a suitable model complexity to reveal «true» underlying structure
- However, one should not model «noise» or be prone to arbitrary sampling
- Goal: The location (in feature space) of a new, unseen sample should have correct label



«underfit» Cf. slide 5; increasing complexity -> overfit

Classifier performance

- Do not test your classifier using your training data!
- We are interested in how our classifier does on new, unobserved samples
- Separate training and test set
  - However, often a limited number of labeled data
- Can switch dataset roles for training and testing, cf. cross-validation

Summary

- Decision rule and division of feature space
- Discriminant functions
- Probability-density-based classifiers
  - Often maximizing aposteriori probability, p(c|x)
- Model complexity, overfitting and the importance of a separate training and test set