Sensor Signal Processing

Prof. Dr.-Ing. Andreas König
Lehrstuhl Integrierte Sensorsysteme

FB Elektrotechnik und Informationstechnik
Technische Universität Kaiserslautern

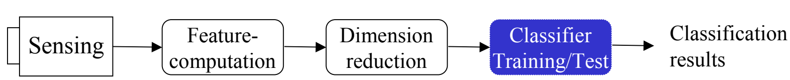
Fall Semester 2005

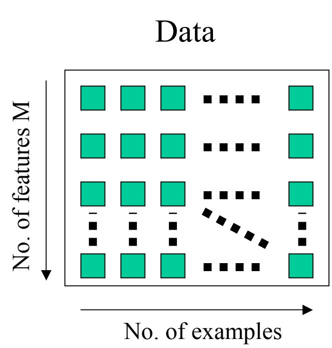
Course Contents

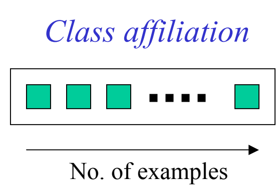
Chapter Contents

Motivation

- After signal processing, feature computation, and condensing or compression of the data the data is subject to classification
- In classification unknown data is affiliated to classes according to previous construction of a classifier



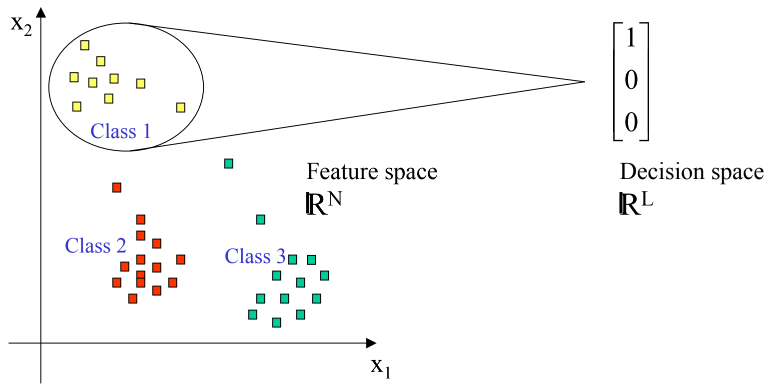




- For known class affiliation (ground truth), classifier performance can be assessed !

Motivation

- Classification can be understood as a special case of general function approximation
- Function values are limited to discrete numbers (class numbers)



(7.1)

- Commonly, integer values of class labels are represented as 1-of-L-coding to avoid spurious neighborhood relations in processing

Sensor Signal Processing
Classification Techniques

Motivation

- Class labels formally are denoted as

$$\begin{array}{l}\Omega = \{\omega_1, \omega_2, \ldots \omega_L,\} \\ \qquad \Updownarrow \quad \Updownarrow \qquad \Updownarrow \\ l \in \{1, \quad 2, \ldots, L \quad \}\end{array} \tag{7.2}$$

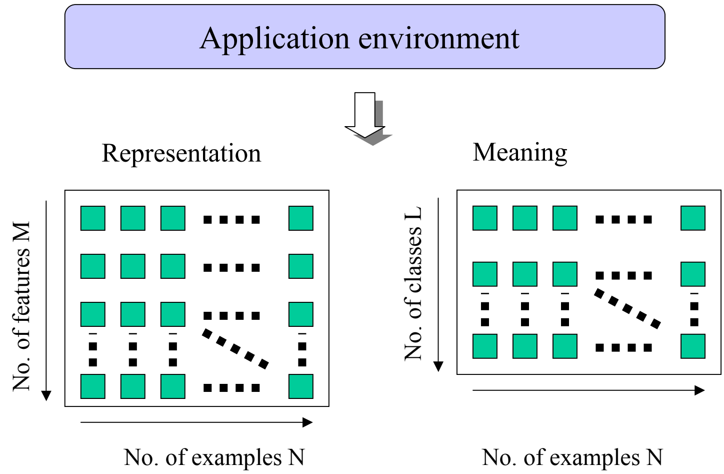
- Alternatively, the target vectors **y**, based on 1-of-L-coding, are employed

$$\begin{array}{l}\Omega = \{\omega_1, \omega_2, \ldots \omega_L,\} \\ \qquad \Updownarrow \quad \Updownarrow \qquad \Updownarrow \\ l \in \left\{ \begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix} \begin{pmatrix}0\\1\\\vdots\\0\end{pmatrix} \cdots \begin{pmatrix}0\\0\\\vdots\\1\end{pmatrix} \right\}\end{array} \tag{7.3}$$

- This representation is favorable for unbiased cost function computation

Motivation

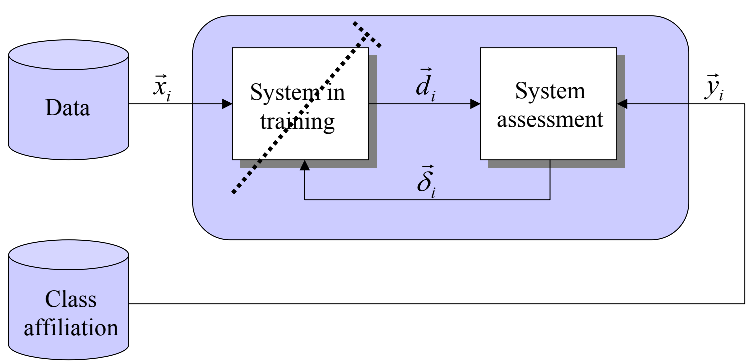
- The construction of a classifier will be regarded in particular for the case of the common machine learning paradigm **learning from examples**



Sensor Signal Processing
Classification Techniques

Motivation

- System to be trained by presentation of examples, i.e. pairs of measurement data and affiliated meaning



- The system is adapted until it gives the correct answer (in tolerable limits)

Motivation

- Statistical motivation according to Bayes-theory:

$$P(x,\omega) = P(x \mid \omega) \cdot P(\omega) = P(\omega \mid x) \cdot P(x) \qquad (7.4)$$

Marginal distribution | Class specific probability | A priori probability | A posteriori probability | Pattern probability

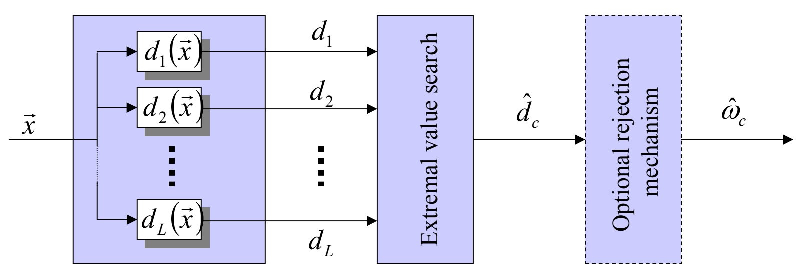
- Class specific, a priori, and pattern probabilities are assumed to be known
- Class affiliation of new data requires the computation (estimation) of the a posteriori probability for each class and determine the maximum:

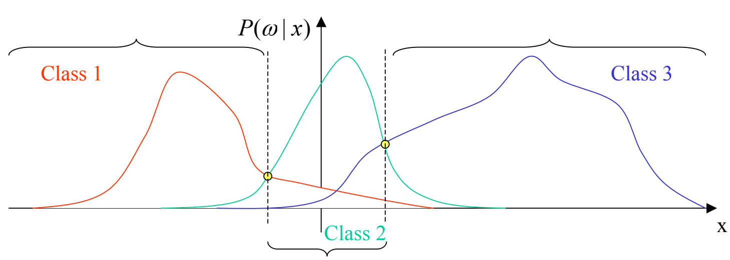
$$P(\omega_c \mid x) = \max_{j=0}^{N} \left(P(\omega_j \mid x) = \frac{P(x \mid \omega_j) \cdot P(\omega_j)}{P(x)} \right) \qquad (7.5)$$

- Index c implies the class index of the class with the highest a posteriori value, i.e. the most probable affiliation of the pattern

Motivation

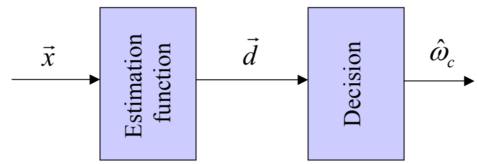
- A classification system or a classifier can be defined by the following structure:

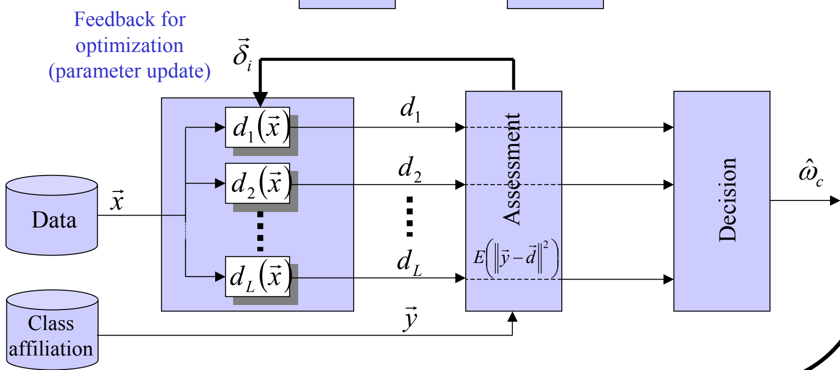




Motivation

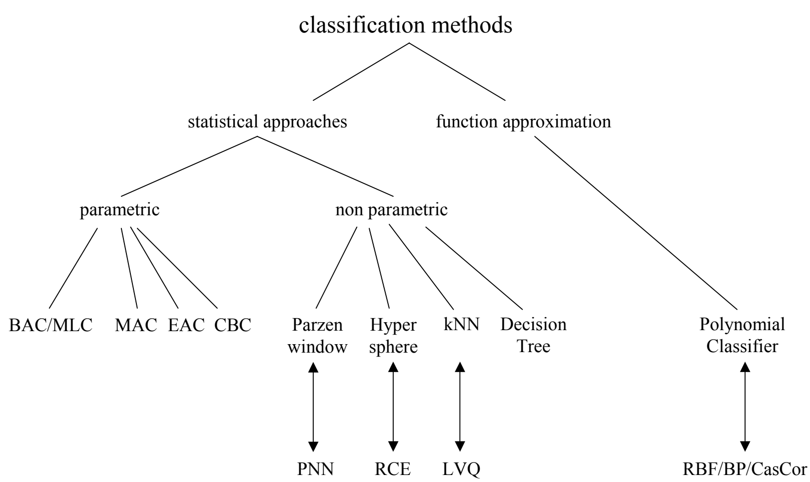
- Unfortunately, the class specific, a priori, and pattern probabilities are commonly unknown and have to be estimated from examples for classification





Motivation

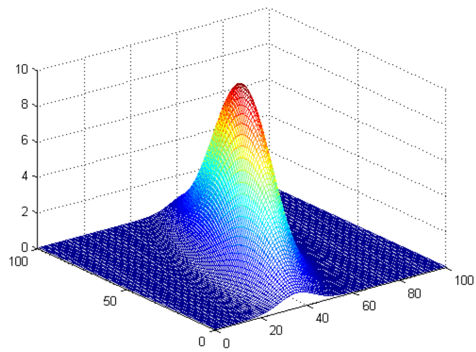
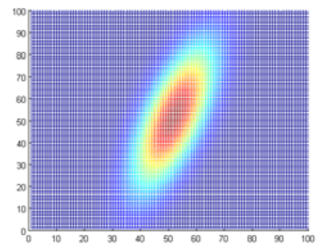
- Numerous options to define estimation functions
- Taxonomy of important classification methods:



Parametric Methods

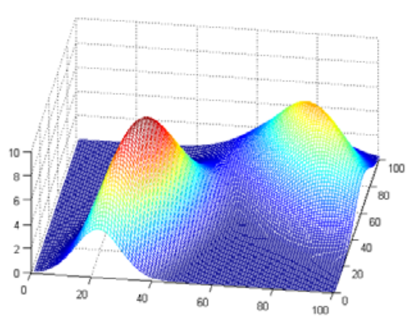
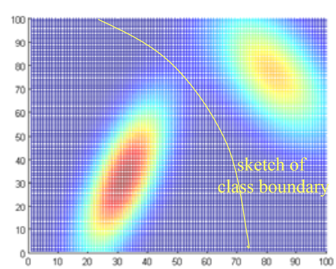
- The Bayes-Normal distribution Classifier (BAC) assumes Gaussian distributions for the classes
- The decision function are determined as

$$\hat{d}_i(\vec{x}) = \frac{P(\omega_i)}{\sqrt{(2\pi)^N \det K_i}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T K_i^{-1}(\vec{x}-\vec{\mu}_i)} \tag{7.6}$$

Parametric Methods

- The intersections of two class region probability distribution functions (pdf) define the class borders as lines of equiprobability

- Parabolic class boundaries result in the two-dimensional example
- In the case of more than two classes, class regions are defined by intersections of resulting parabolic functions

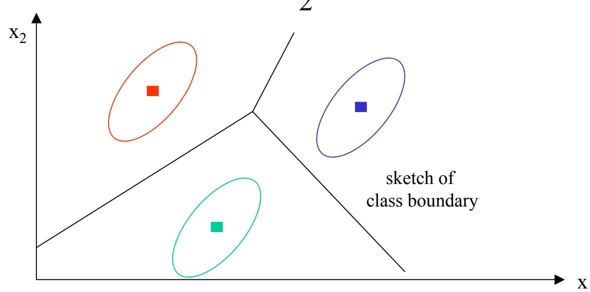
Parametric Methods

- Assuming equal a priori values for all classes return the Maximum-Likelihood-Classifier (MLC):

$$\hat{d}_i(\vec{x}) = -\frac{1}{2}\ln(\det K_i) - \frac{1}{2}(\vec{x}-\vec{\mu}_i)^T K_i^{-1}(\vec{x}-\vec{\mu}_i) \qquad (7.7)$$

- Assuming further equal covariance for all classes return the Mahalanobis-Classifier (MAC):

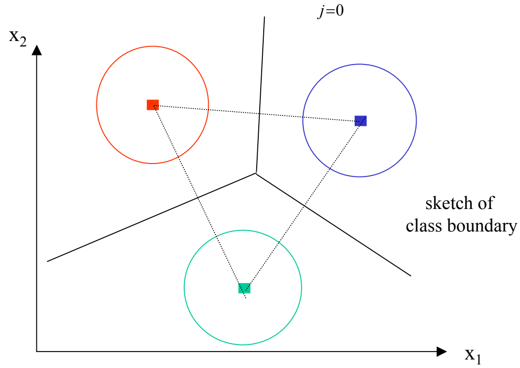
$$\hat{d}_i(\vec{x}) = -\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T K^{-1}(\vec{x}-\vec{\mu}_i) \qquad (7.8)$$



Parametric Methods

- Assuming further the covariance matrix to be the identity matrix return the Centroid or Euclidean Distance Classifier (EAC):

$$\hat{d}_i(\vec{x}) = \|\vec{x}-\vec{\mu}_i\|^2 = \sum_{j=0}^{N-1}(x_j-\mu_j)^2 \qquad (7.9)$$



- Simplifying the metric returns the City-Block-Classifier (CBC):

$$\hat{d}_i(\vec{x}) = \sum_{j=0}^{N-1}|x_j-\mu_j| \qquad (7.10)$$

Nonparametric Methods

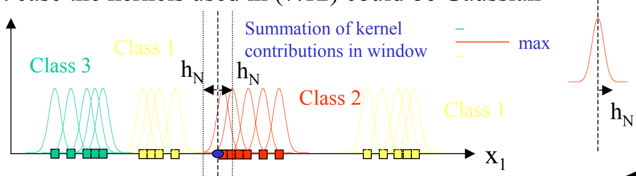
- Parametric methods condense all the sample set information into few model parameters and perform a global pdf estimation
- In contract, nonparametric methods perform a local pdf estimation:

$$p(\vec{x}) = \frac{k(\vec{x})}{N \cdot \nu} \qquad (7.11)$$

- Assumption of either fixed volume ν or fixed number of patterns k
- The Parzen-window classifier bases on the first alternative

$$p_{Parzen}(\vec{x}) = \frac{1}{N \cdot h_N^M} \sum_{i=1}^{N} \kappa\left(\frac{\vec{x} - \vec{x}_i}{h_N}\right) \qquad (7.12)$$

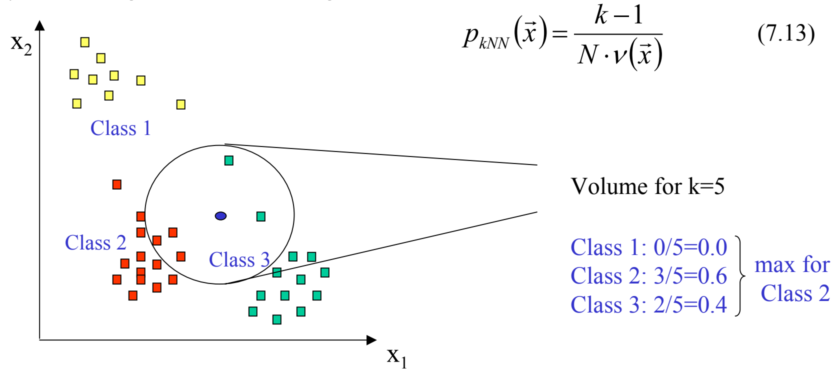
- In the simplest case the kernels used in (7.12) could be Gaussian functions:



Nonparametric Methods

- The kernel width must be adapted to data density
- k-nearest-neighbor classifiers (kNN) under control of the parameter k estimate the densitiy and determine the class affiliation of a new pattern by evaluating the k-nearest-neigbors

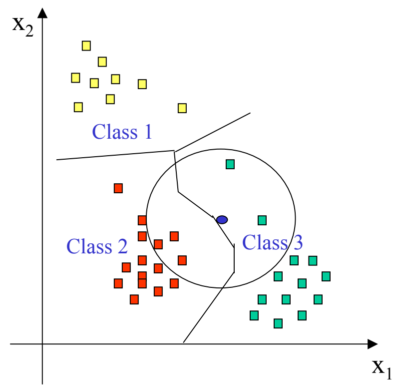
$$p_{kNN}(\vec{x}) = \frac{k-1}{N \cdot \nu(\vec{x})} \qquad (7.13)$$



- kNN training means storage of all patterns !
- Two basic variations: voting and volumetric kNN

Nonparametric Methods

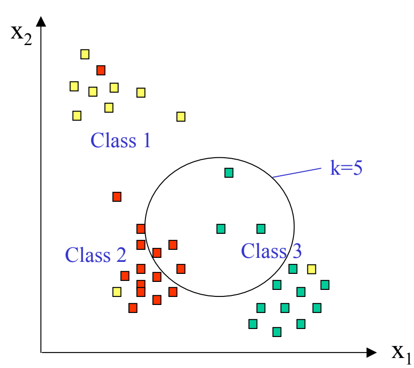
- Sketch of class boundary for k=1 1NN-classification



- Class-specific Voronoi tesselation defines the class borders in this case

Nonparametric Methods

- Edited-Nearest-Neighbor-classification (ENN, Devijver and Kittler, 1980)
- For k=1 resubstitution is guaranteed, however generalization can be affected by this over-specialization in following situations:



- Automatic determination of the outliers in the data set:

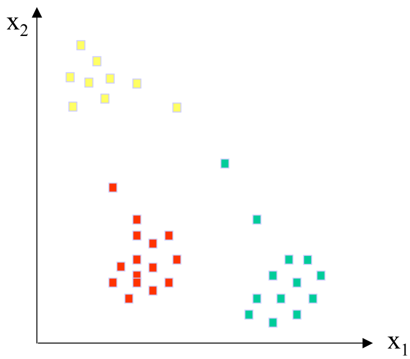
$$G_{ENN} = \frac{kNN_{\omega_j = \omega_j}}{kNN_{\omega_j \neq \omega_j}} \tag{7.14}$$

- k neighbors determined and investigated for class affiliation
- Elimination if same to different class affiliations are below threshold:

$$G_{ENN} < \Theta \tag{7.15}$$

- Edited sample set is reduced, resubstitution no longer assured, generalization generally improved by outlier elimination (rf. q_o)

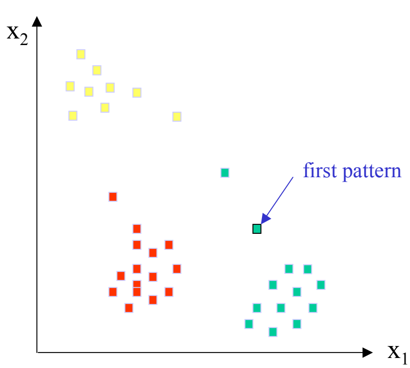
- Condensed-Nearest-Neighbor-classification (CoNN, Hart, 1968)
- The complete storage in kNN requires large memory & long computation
- Reduction by limiting storage to vectors defining class boundary



Pseudo-Code of CoNN algorithm:
1. Initially empty classifier
2. Get first (next) pattern x_i
3. 1-NN-classify sample set pattern
   If correct_classification goto 4
   Else Insert pattern x_i; goto 2
4. If all_patterns_corr_class break;
   Else goto 2

- Algorithm reduces effort but depends on sample set presentation order and leaves redundancy in the CoNN; recall by 1-NN mandatory !

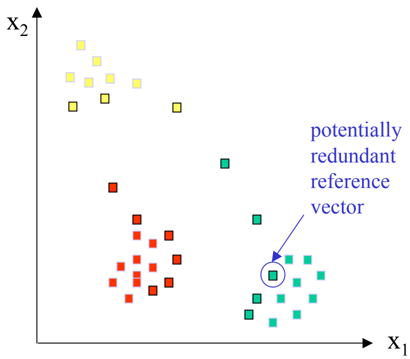
- Condensed-Nearest-Neighbor-classification (CoNN, Hart, 1968)
- Step-wise demonstration:



Pseudo-Code of CoNN algorithm:
1. Initially empty classifier
2. Get first (next) pattern x_i
3. 1-NN-classify sample set pattern
   If correct_classification goto 4
   Else Insert pattern x_i; goto 2
4. If all_patterns_corr_class break;
   Else goto 2

- All class 3 patterns in the following will be classified correctly

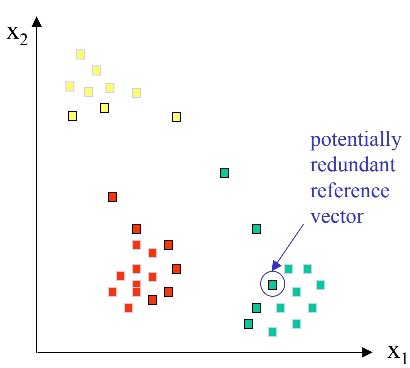
- Condensed-Nearest-Neighbor-classification (CoNN, Hart, 1968)
- Sketch of potential final solution:



Pseudo-Code of CoNN algorithm:
1. Initially empty classifier
2. Get first (next) pattern x_i
3. 1-NN-classify sample set pattern
   If correct_classification goto 4
   Else Insert pattern x_i; goto 2
4. If all_patterns_corr_class break;
   Else goto 2

- Removal of existing redundancy by following step

- Reduced-Nearest-Neighbor-classification (RNN, Gates, 1972)
- Addition of a removal step to clean reference vector set from redundant instances, i.e., vectors not required for perfect resubstitution

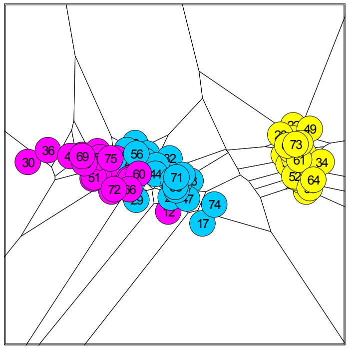


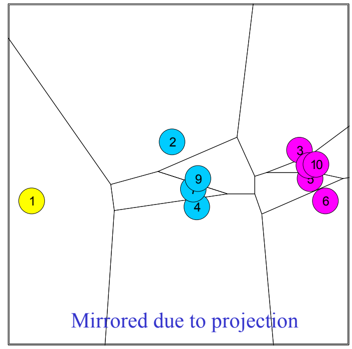
Pseudo-Code of RNN algorithm:
1. Run CoNN algorithm
2. Tentatively remove first (next) reference vector r_i
3. 1-NN-classify all sample set patterns
   If correct_classification permanently remove r_i; goto 4
   Else restore r_i; goto 4
4. If last_ref_vec break;
   Else goto 2

- CoNN redundancy eliminated, trong dependance on presentation order
- Alternatively, CoNN and RNN steps can be interleaved in a modified, potentially faster algorithm, however, limit cycles can occur (rf. q_s)

Nonparametric Methods

- QuickCog example for Iris data:





- With the given settings, 10 reference vectors are chosen (q_s=0.906666)

Nonparametric Methods

- A traditional method in OCR used for nonparametric classification, based on function approximation approach, is the polynomial classifier

$$\hat{d}_i(\vec{x}) = a_{0,i} + a_{1,i}\cdot x_1 + a_{2,i}\cdot x_2 + \ldots + a_{N,i}\cdot x_N + a_{N+1,i}\cdot x_1^2 + a_{N+2,i}\cdot x_1\cdot x_2 + \ldots \quad (7.16)$$

- Expressing (7.14) by new coordinates results in

$$\vec{v} = (v_1, v_2, \ldots, v_p) = (1, x_1, x_2, \ldots, x_N, x_1^2, x_1\cdot x_2, \ldots) \quad (7.17)$$

$$\vec{\hat{d}} = A^T \cdot \vec{v}(\vec{x}) \quad (7.18)$$

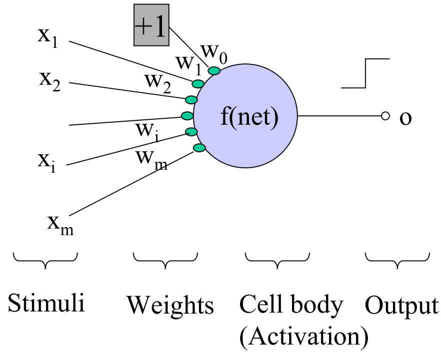
- LMS-optimization to determine polynomial coefficients of A
- Similarity to dot product neuron with nonlinear synapses !
- Excessive growth of variables in (7.15) for increasing order G of polynomial:

$$p = \binom{N+G}{G} = \frac{(N+G)!}{N!\cdot G!} \quad (7.19)$$

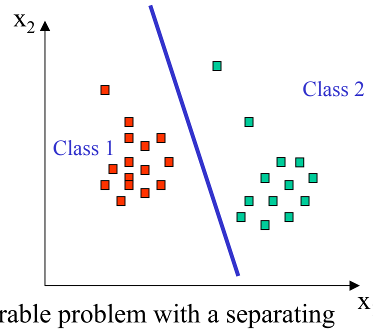
- Data compression/term selection, G 1-3 (Schürmann, Kressel)

Neural Networks

- Simple dot product neurons can serve as linear classifiers



$$o = f\left(\sum_{i=1}^{m} x_i w_i\right)$$

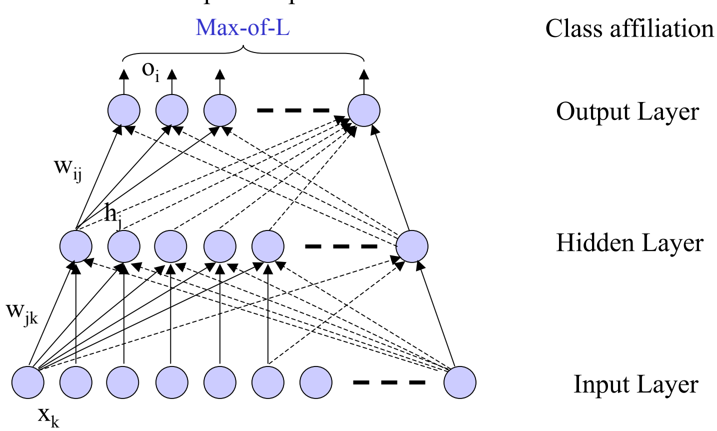


- A single neuron can separate a linear separable problem with a separating line (plane, hyperplane)
- Logical combinations of a layer allow tackling non-linear problems !

Sensor Signal Processing
Classification Techniques

Neural Networks

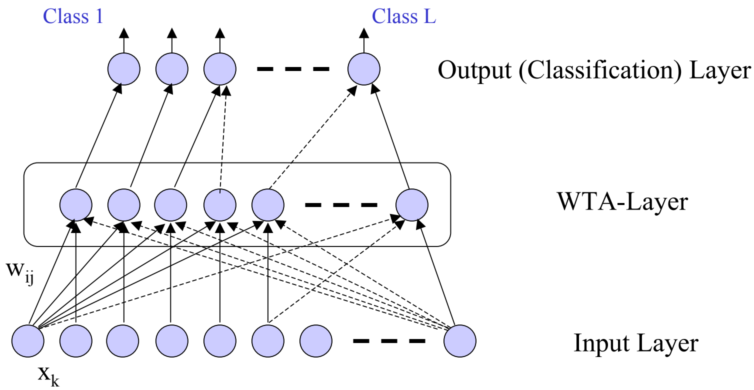
- A multilayer perceptron with backpropagation algorithm serves as a classifier for non-linear separable problems:



- Training bases on the approach introduced in section 5
- Choice of hidden layer size and learning parameters can be difficult
- Resubstitution not guaranteed, generalization can be surprizing

Neural Networks

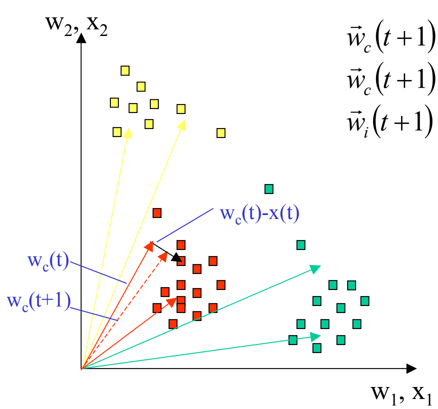
- A learning vector quantization network (LVQ) employs winner-take-all mechanism that leaves but the strongest response active:



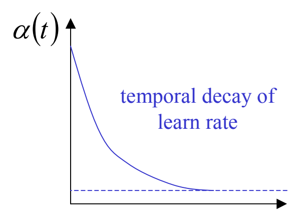
- Adjusting of a fixed number, commonly randomly or by SOM training initialized vectors
- Recall via 1-NN classification

Neural Networks

- Basic LVQ-1 learning method [Kohonen 1989]
- Iterative presentation of training data and WTA-computation finding $w_c(t)$



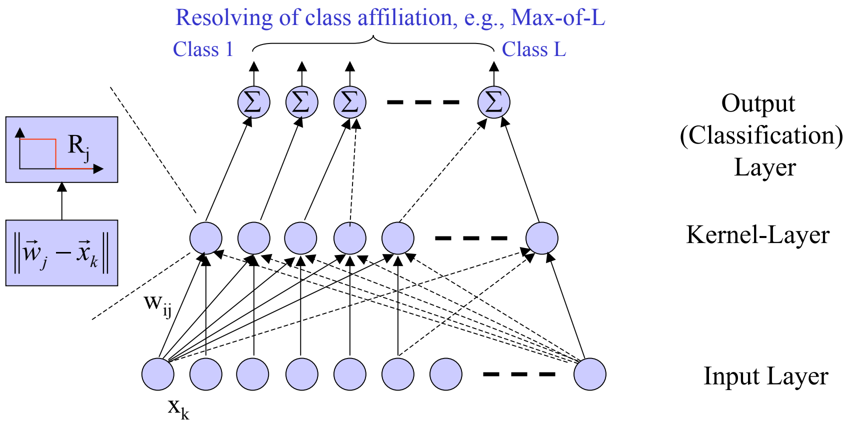
$$\vec{w}_c(t+1)=\vec{w}_c(t)+\alpha(t)\left[\vec{w}_c(t)-\vec{x}(t)\right] \quad if \quad \omega_c=\omega_j$$
$$\vec{w}_c(t+1)=\vec{w}_c(t)-\alpha(t)\left[\vec{w}_c(t)-\vec{x}(t)\right] \quad if \quad \omega_c\neq\omega_j$$
$$\vec{w}_i(t+1)=\vec{w}_i(t) \quad \forall \quad i\neq c \tag{7.20}$$



- Dead reference vectors w_i can occur, sufficient no. per class not assured
- Different initialization, e.g., RNN, could be suitable
- Basic methods extended by improved versions LVQ2/2.1 & LVQ3

Neural Networks

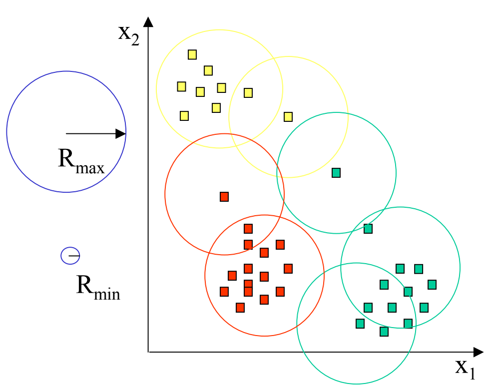
- A special class of ANN consists of a kernel layer and an output layer



- The Restricted-Coulomb-Energy network (RCE) employs step functions in the kernel layer and or-gate or summation of activated kernel neurons
- The network is generated from scratch by (patented) dynamic training

Neural Networks

- RCE-training is part of the Nestor-Learning-System (NLS)
- Dynamic placement and scaling of hyperspheres:



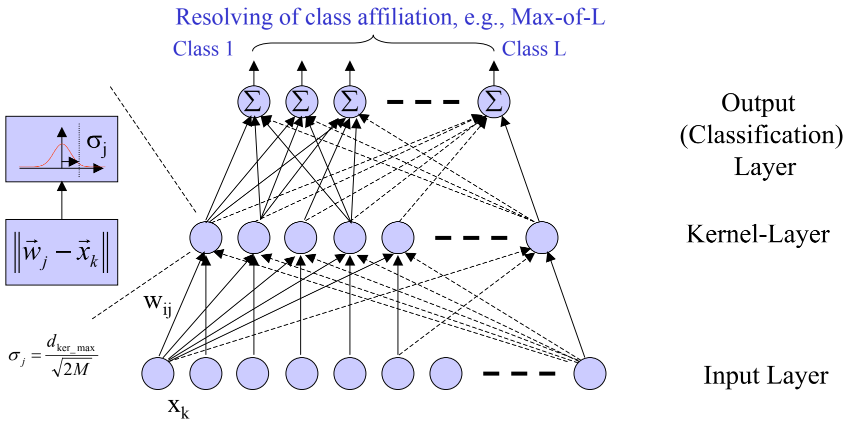
Pseudo-Code of RCE algorithm:

1. Initially empty classifier
2. Get first (next) pattern $\mathbf{x}_i$
3. RCE-classify pattern
   If *correct_classification* goto 4
   Else If *unknown* Insert $\mathbf{x}_i$; goto 2
   Else If *ambiguous* reduce
   radii of $\mathbf{r}_i$ with $\omega_i \neq \omega_j$;goto 2
   Else (*misclassification*)
   reduce radii of $\mathbf{r}_i$ with
   $\omega_i \neq \omega_j$; Insert $\mathbf{x}_i$; goto 2
4. If all_patterns_corr_class break;
   Else goto 2

- Result strongly depends on presentation order (Pro-RCE extension)
- Insertion only does not remove redundant neurons
- Classification resolving by voting or kernel based pdf estimation (PRCE)
- Additional attributes unknown and uncertain/ambiguous

Neural Networks

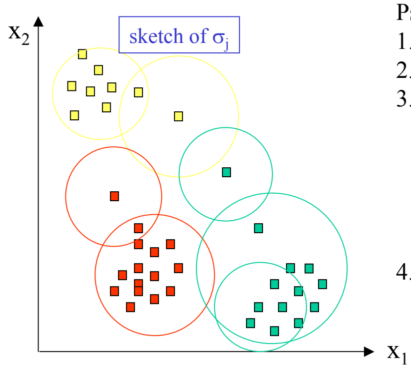
- The Radial-Basis-Function network (RBF) employs (commonly Gaussian) kernel functions in the kernel layer and dot product output layer neurons



- RBF-networks are also universal function approximators !
- Number of kernels fixed by choice and random initialization or SOM

Neural Networks

- RBF-training on a fixed hidden layer might not be efficient
- Dynamic training algorithms for function approximation and classification
- The first one is Platt´s Resource-Allocation-Network RAN



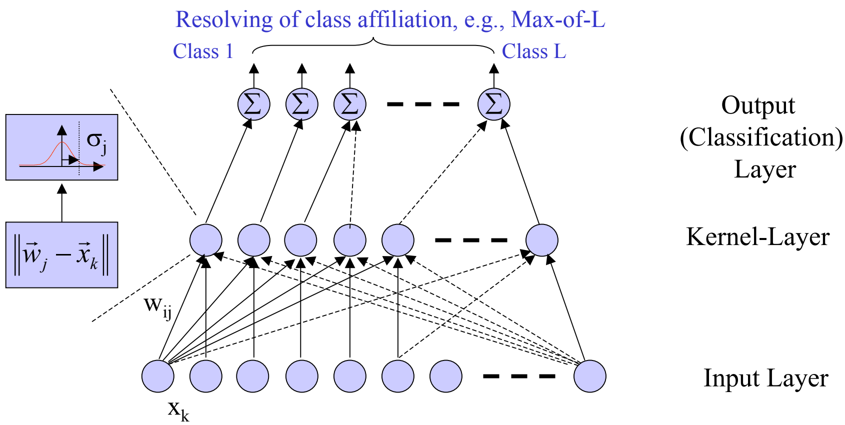
Pseudo-Code of principle RAN algorithm:

1. Initially empty kernel layer
2. Get first (next) pattern $\mathbf{x}_i$
3. Compute output value
   If *Error* $> \varepsilon$ & $d > \delta$
   Insert $\mathbf{x}_i$ as new kernel;
   $\sigma_j = \min(\kappa\bullet\delta, \kappa\bullet d)$
   Adapt output layer weights ... ;
4. If *sum_err* $< \varepsilon 2$ break;
   Else $\delta = \delta \bullet e^{-1/\tau}$; goto 2

- Fast (evolving) training, insertion only does not remove redundant neurons
- Classification by determination of maximum pdf; background applicable !
- Training for all RBF-parameters can be achieved by gradient descent !
- Smooth and well-generalizing behavior of RBF-networks

Neural Networks

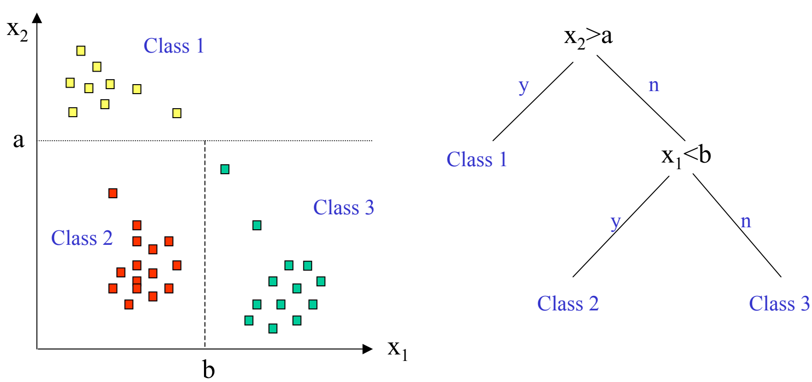
- Probabilistic-Neural-Networks of Specht resemble Parzen-Window:



- Each training data vector is stored as Gaussian kernel with fixed global σ
- According to class labels, kernel are wired to pdf summation nodes
- Explicit cost or a priori weighting can be employed before pdf max-of-L

Rule-based Approaches

- Inherently, rules can be extracted/employed in classification problems
- Decision trees are one common example for crisp techniques:

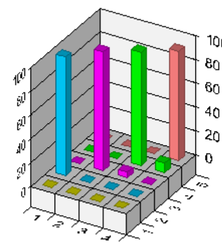


- Decision trees construction by dedicated machine learning algorithm (Idx.y)
- Knowledge application/extraction; hybrid combinations feasible
- Fuzzy and neuro-fuzzy approaches extend the classification approach

Evaluation Techniques

- The performance of a classification system as well as of a classifier is assessed by the classification rate
- Based on the available *ground truth* of a test data set, the number of correctly classified patterns vs. the total pattern number is computed and given in %
- More detail about confusion between classes is given by the confusion matrix (Mechatronic example):

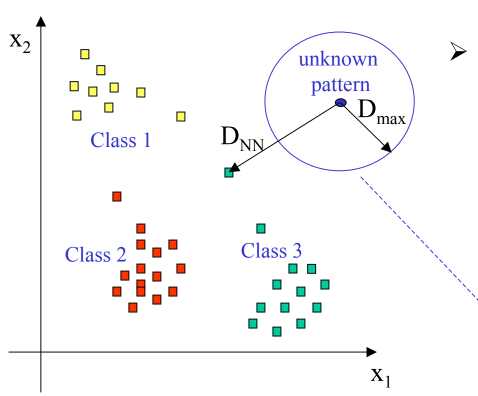
```
Confusion matrix
Class1 (50) :  0  (R)  100  0      0       0
Class2 (100):  0  (R)  0    100    0       0
Class3 (150):  0  (R)  0    4.660  95.330  0
Class4 (25) :  0  (R)  0    0      8       92
Classification rate: 97.231 %
```



- The confusion matrix indicates the additional option of rejection

Evaluation Techniques

- Several situations can occur, where the decision of the classifier could be doubtable (pattern close to the decision boundary or far from training data)



- Rejection criterion:

MAX: $\hat{d}_{\max} < \Theta$ (7.11)

DIF: $\left(\hat{d}_{\max} - \hat{d}_{2nd\,\max}\right) < \Theta$ (7.12)

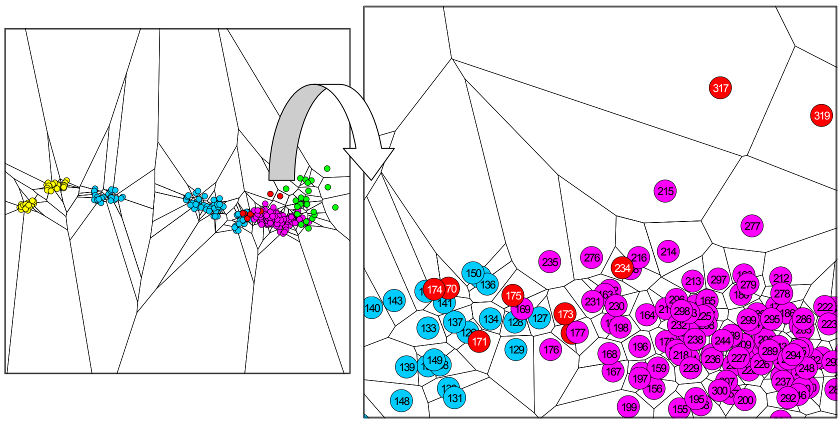
RAD: $D_{NN} > D_{\max}$ (7.13)

$R_{NN} > R_{\max}$

- Guessing can be avoided by rejection (Decision can always be enforced)
- Contradictions can be resolved on higher level (fusion aspects !)

Evaluation Techniques

- The location and identification of misclassifications in feature space and in the example database, e.g., corresponding raw image, is salient
- Feature space visualization techniques allow just this:



- Misclassified patterns can rapidly be investigated for cause elimination

Summary

- The chapter gave a survey of the concepts and common conventional as well as neural classification methods
- Recently, powerful techniques, such as the support vector machines have extended the available method pool
- In practical application, the choice of approach should be governed by the need and complexity of the underlying problem and data
- For a system well optimized in the first stages the simplest choice will provide robustness and good performance
- For small data sets, appropriate partitioning in training and test data is not feasible
- For statistical meaningful evalution leave-one-out techniques (LOO) are commonly applied
- In addition to rejection threshold, also the notion of cost can be introduced into the design of the discussed classifier models