

Machine learning – Nonparametric techniques

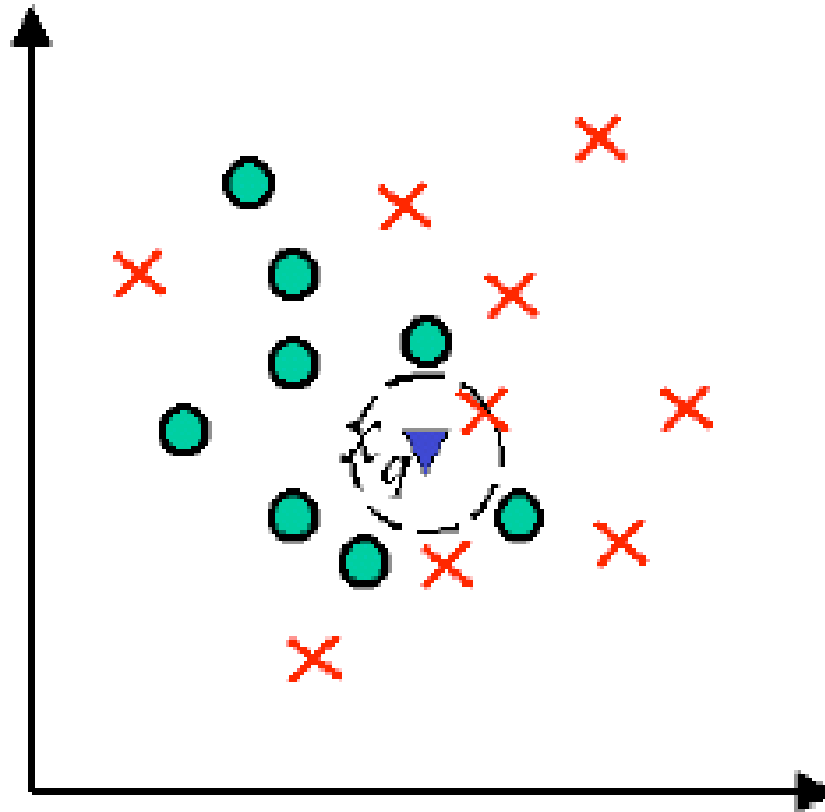
Referent: Peter Hallama

Tutor: Martin Wagner

Algorithm Overview

- K-Nearest Neighbour
- Locally Weighted Regression
- Radial Basis Functions
- Case Based Reasoning
- Short Overview Learning Vector Quantization

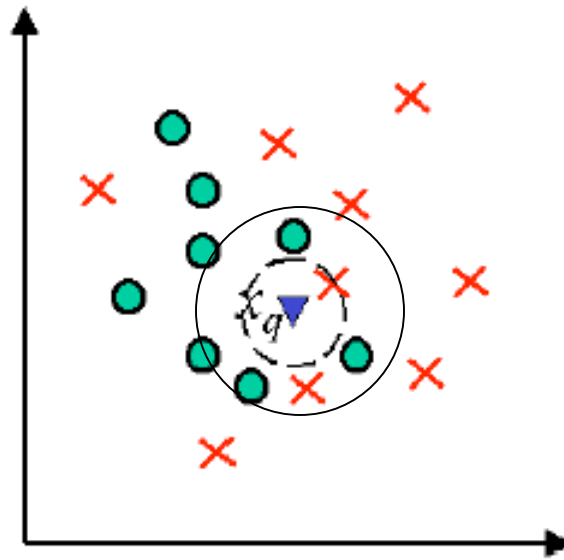
Problem



K-Nearest Neighbour (k-NN)

- Instances correspond to points in the n-dimensional space \mathbb{R}^n
- Instance $\mathbf{x} = (a_1(\mathbf{x}), a_2(\mathbf{x}), \dots, a_n(\mathbf{x}))$
 $a_r(\mathbf{x})$ = value of the rth Attribute
- Distance is defined
$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{r=1}^n (a_r(\mathbf{x}_i) - a_r(\mathbf{x}_j))^2}$$
- Returns most common Value

Example 1-NN



1-NN will classify x_q as cross
5-NN as circle

Distance-Weighted NN

- Giving greater weight to closer neighbours
- May weight at the inverse square of the distance
- $w_i = 1/d(x_q, x_i)^2$

Remarks on NN

- Distance calculated based on all attributes
 - > giving the attributes weight by change their Axes by Factor
- Efficient memory indexing (i.e. kd-tree)
 - > More efficiently at some additional cost in Memory

Pro/Contra k-NN

- + Training is very fast
- + Does not lose any Information
- lazy method / slow at query time
- Irrelevant attributes considered (weight the attributes)

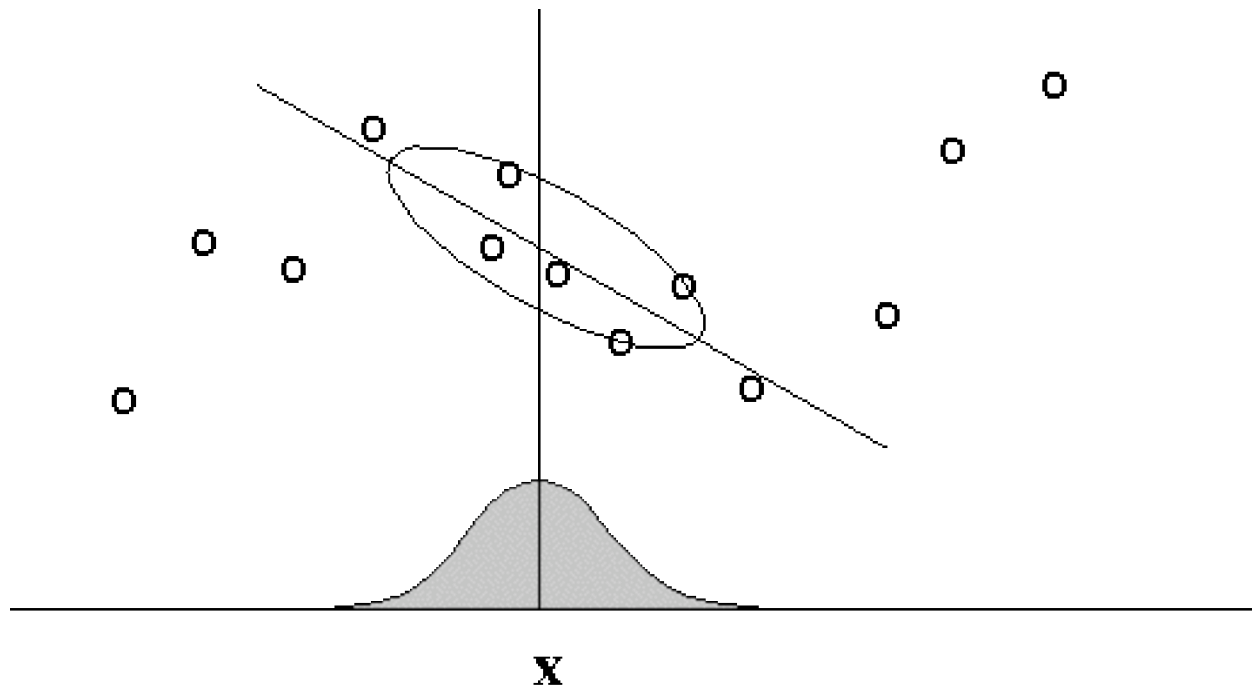
Locally Weighted Regression

- Trying to approximate the target function in the Neighbourhood of x_q using linear/quadratic functions or multilayer neural networks
- Minimizing the squared error over k nearest neighbours

Locally Weighted Linear Regression

- adjust the values of slope and intercept to find the line that best predicts Y from X .
- minimize the sum of the squares of the vertical distances of the points from the line.

Example



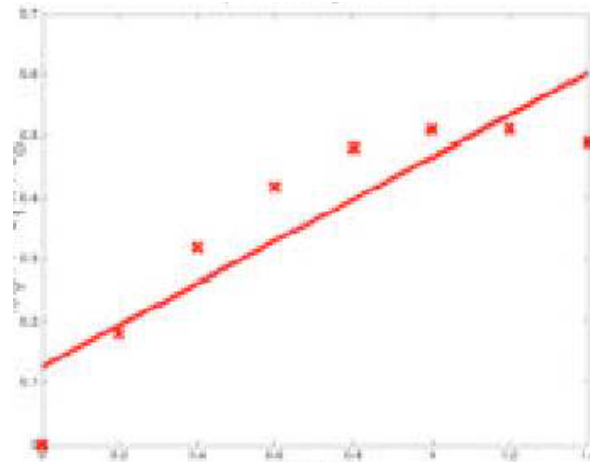
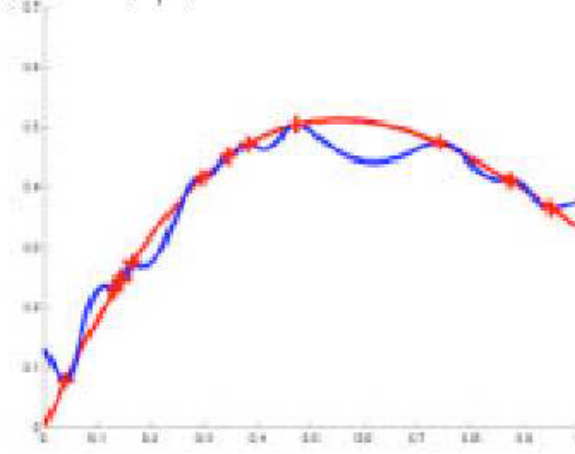
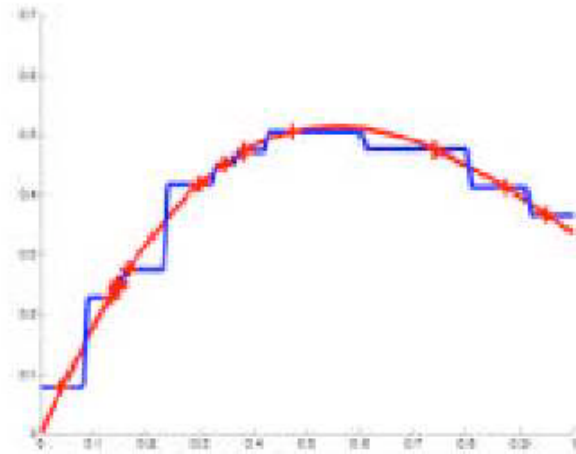
Pro/Contra LWR

- + approximates the real target function
- + Does not lose any information
- lazy method / slow at query time
- Irrelevant attributes may be considered
- very slow for complex functions

NN / WNN / LWLR

1-nearest neighbor

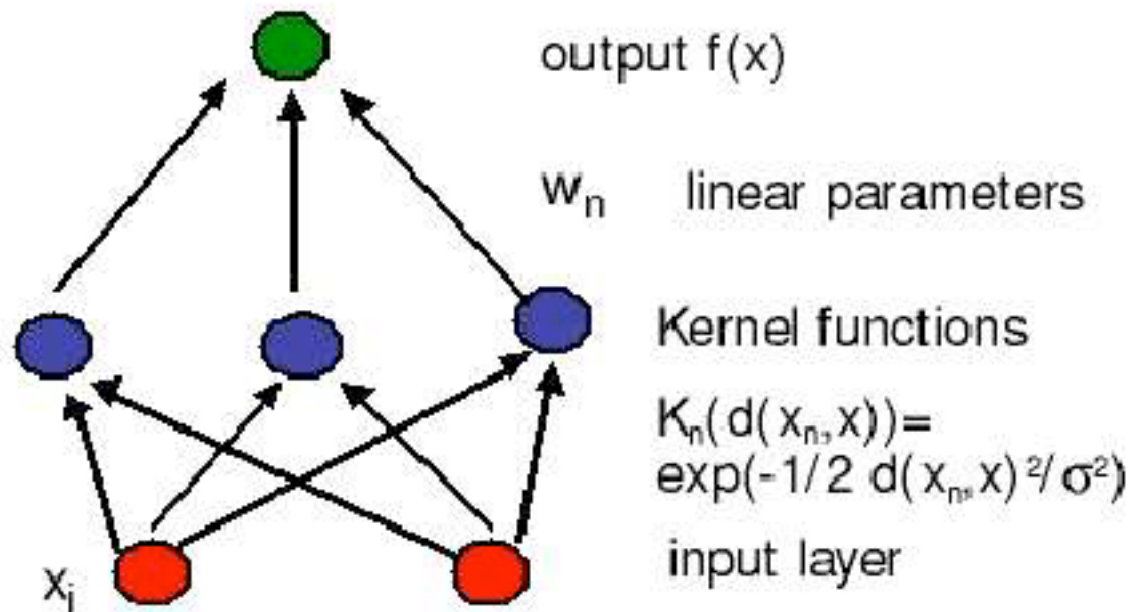
$$K(d(x_q, x_i)) = 1 / d(x_q, x_i)^2$$



Radial Basis Functions

- $\hat{f}(\mathbf{x}) = \omega_0 + \sum_{u=1}^k \omega_u K_u(d(\mathbf{x}_u, \mathbf{x}))$
- \mathbf{x}_u is a instance from X ; K_u decreases as the distance $d(\mathbf{x}_u, \mathbf{x})$ increases
- K = number of kernel functions included

Radial Basis Function



Pro/Contra RBF

- + linear combination of local approximations
leads to global approximation to the target f
- + Closely related to Distance Weighted
Regression but eager not lazy
- how to choose the kernel functions?
- how to train the weights?

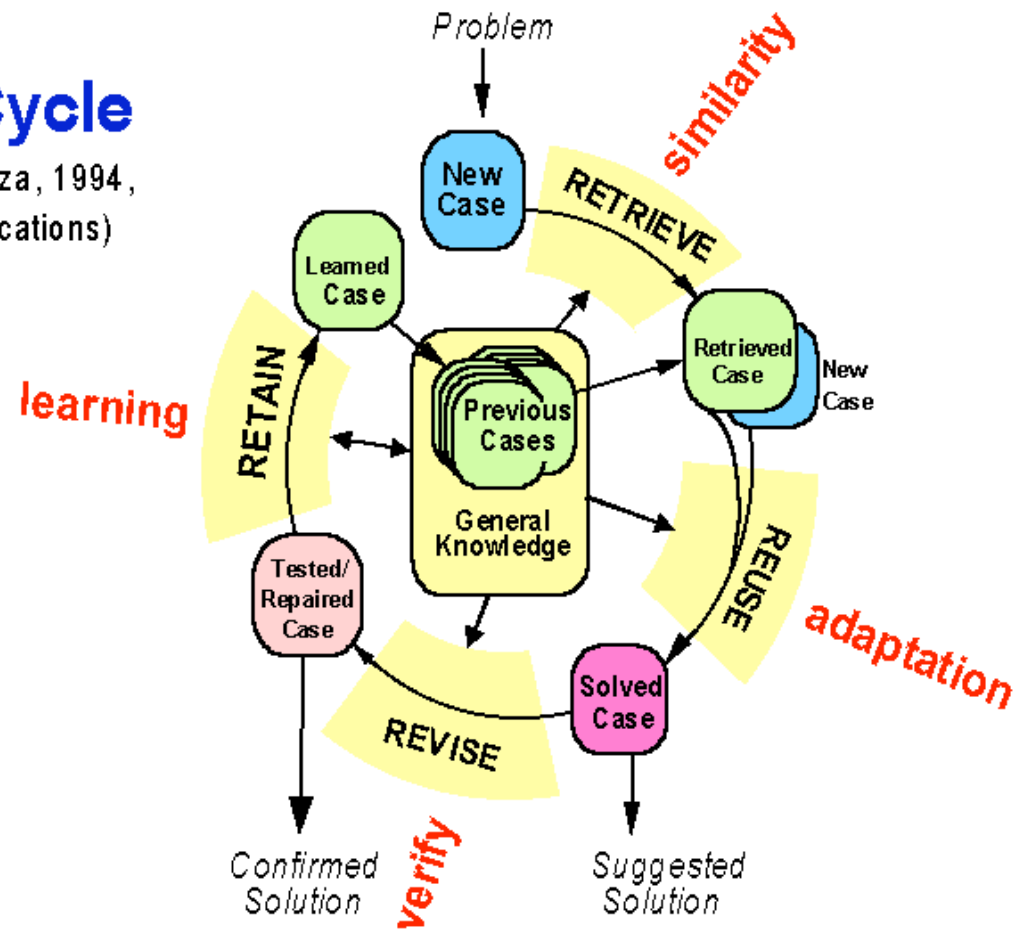
Case-Based Reasoning

- Not representing in N-dimensional Space
- Solving Problems by reusing and combining portions of previous solutions
- Example the CADET system
- Assist in the conceptual design of simple mechanical devices such as water faucets
- Library containing about 75 previous designs and design fragments
- Instance is represented by describing structure and its qualitative function

Beispiel

CBR Cycle

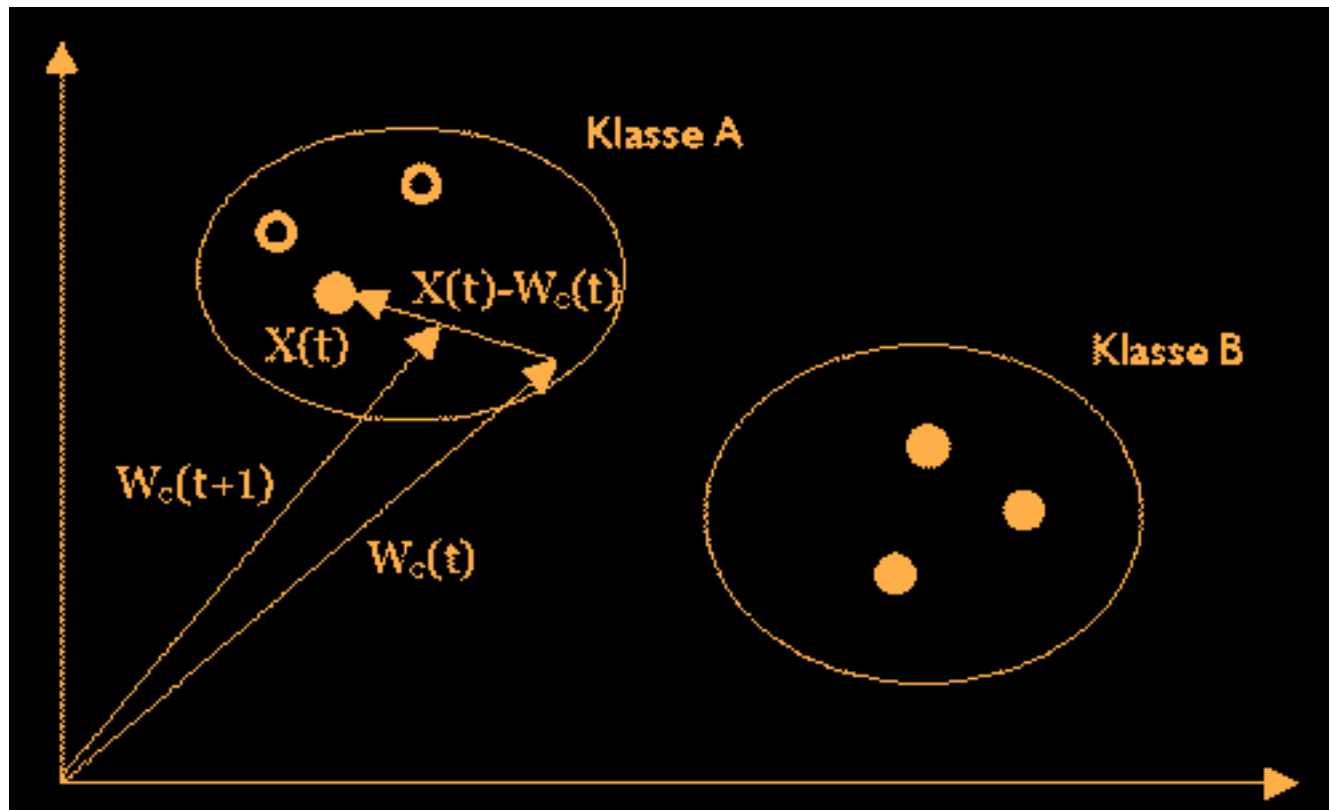
(Aamodt & Plaza, 1994,
AI Communications)



Short Overview Learning Vector Quantization

- Classes Represented by Prototyp Vectors
- Training points attract prototypes of the correct class (LVQ1)
- They may also disattract prototypes of the wrong classes (LVQ2)
- LVQ3 trys to fix some LVQ2 problems.

LVQ Beispiel



Benchmark on Spam mail

- Benchmark setup:
- Mixture of spam messages and messages received via a mailinglist ratio 481: 2412 = 16,6 % spam
- Algorithms: Naive Bayesian / TiMBL

TiMBL

- Variant of k-NN
- Doesn't consider only the k-nearest but all instances at the k-closest distance
- This could lead to a high number of instances

TCR definition

- $N_{L \rightarrow L}$ Number of legal mail classified as legal
 $N_{L \rightarrow S}$ = legal mail classified as spam

$$Acc = \frac{N_{L \rightarrow L} + N_{S \rightarrow S}}{N_L + N_S} \quad Err = \frac{N_{L \rightarrow S} + N_{S \rightarrow L}}{N_L + N_S}$$

- Weighting the legal mail λ -times more than spam messages

$$WAcc = \frac{\lambda \cdot N_{L \rightarrow L} + N_{S \rightarrow S}}{\lambda \cdot N_L + N_S} \quad WErr = \frac{\lambda \cdot N_{L \rightarrow S} + N_{S \rightarrow L}}{\lambda \cdot N_L + N_S}$$

TCR definition

- The weighted baseline is the cost without any filter

$$WAcc^b = \frac{\lambda \cdot N_L}{\lambda \cdot N_L + N_S} \quad WErr^b = \frac{N_S}{\lambda \cdot N_L + N_S}$$

- This leads to the Total Cost Ratio (TCR)

$$TCR = \frac{WErr^b}{WErr} = \frac{N_S}{\lambda \cdot N_{L \rightarrow S} + N_{S \rightarrow L}}$$

Results

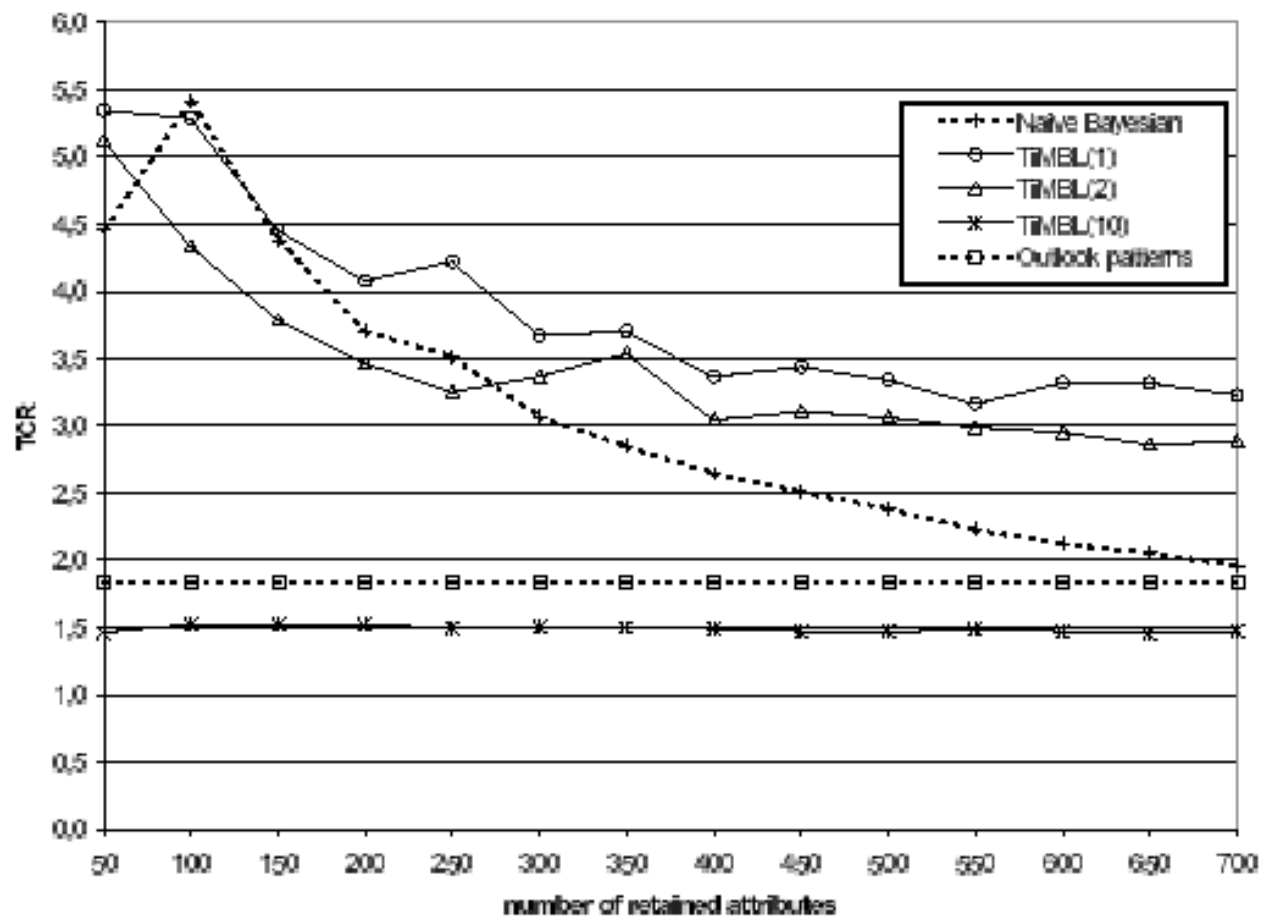


Figure 1. TCR scores for $\lambda=1$

Results

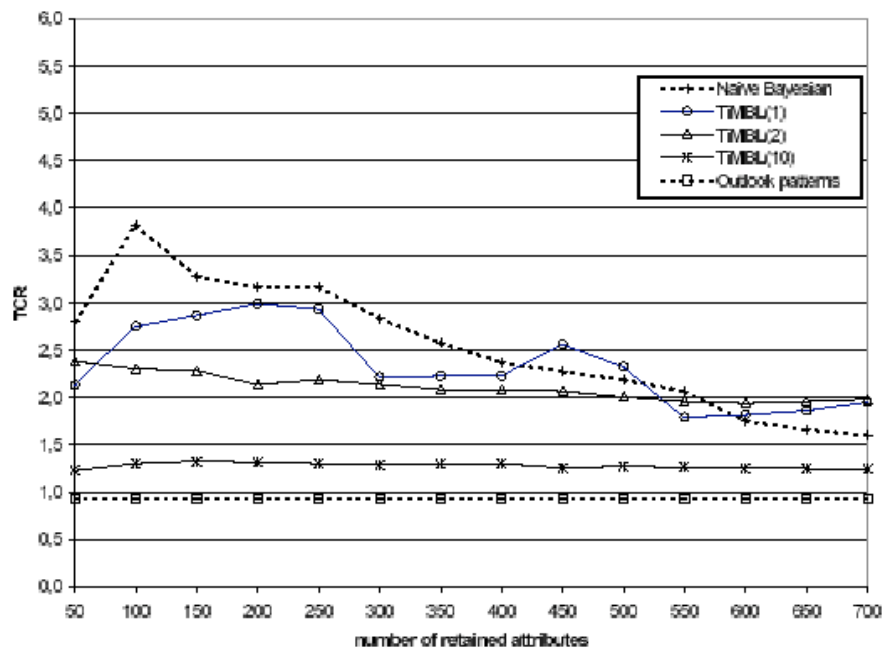


Figure 2. TCR scores for $\lambda=9$

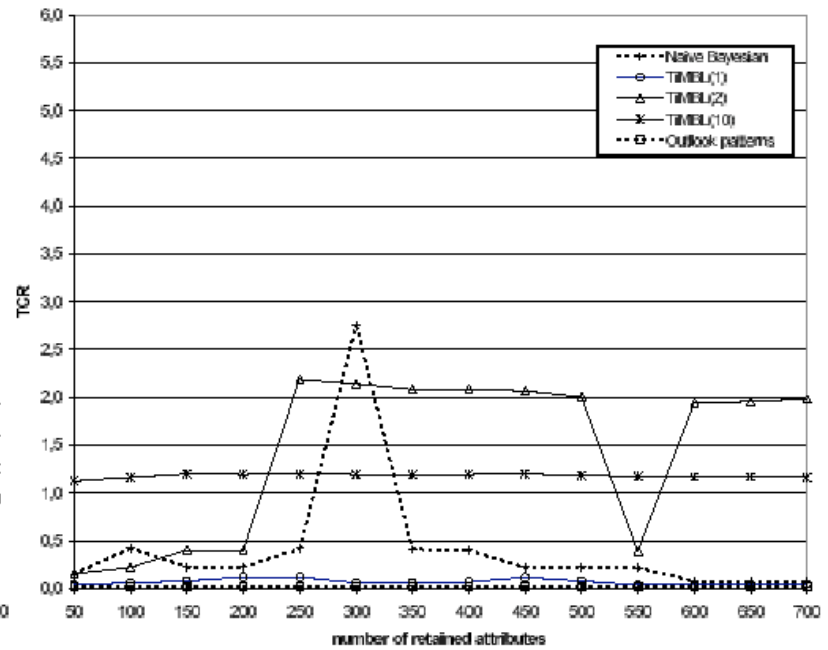


Figure 3. TCR scores for $\lambda=999$

More Results

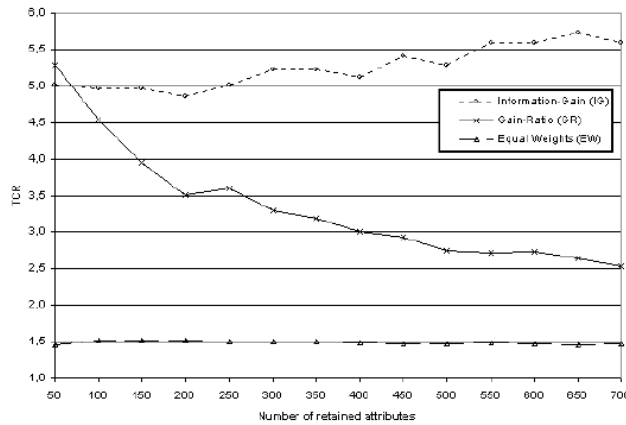


Figure 1: TCR of 10-NN for $\lambda = 1$ and three attribute-weighting functions.

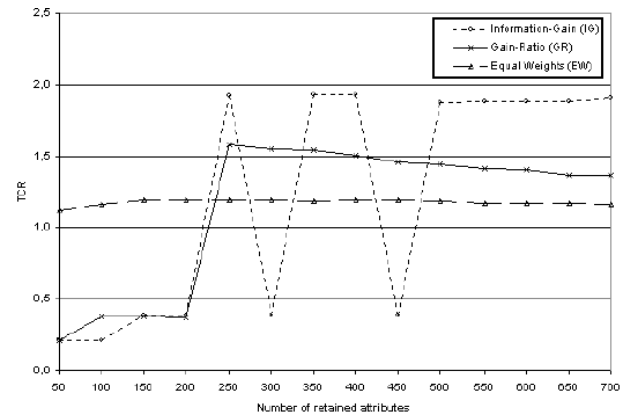


Figure 2: TCR of 10-NN for $\lambda = 999$ and three attribute-weighting functions.

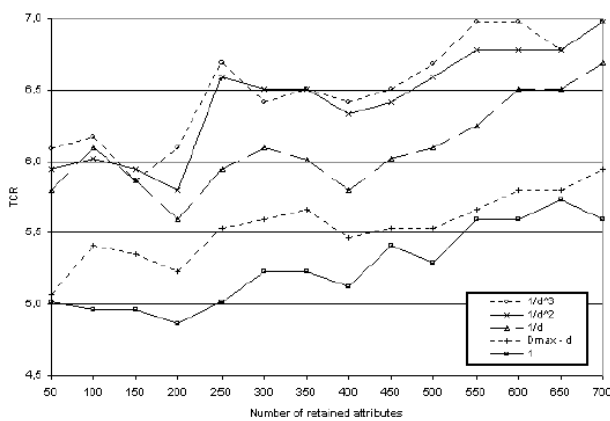


Figure 3: TCR of 10-NN for $\lambda = 1$ and different distance weighting functions (using IG for attribute weighting).

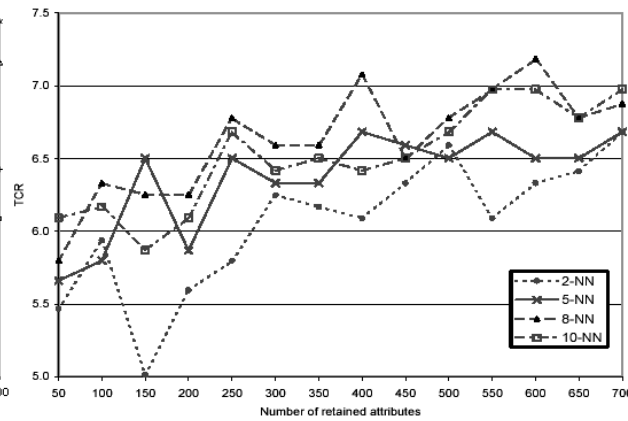


Figure 4: TCR of k -NN for $\lambda = 1$ and different k values (using IG for attribute weighting and $1/d^3$ for distance weighting).

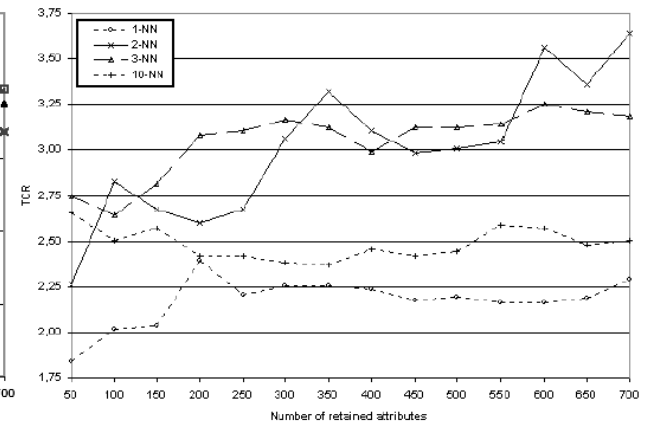


Figure 5: TCR of k -NN for $\lambda = 9$ and different k values (using IG for attribute weighting and $1/d^3$ for distance weighting).

Other Examples

- Text recognition
- Image Scene Classification