Representation Related Problems in Pattern Recognition

London, 6 October 2006

Robert P.W. Duin

Delft University of Technology

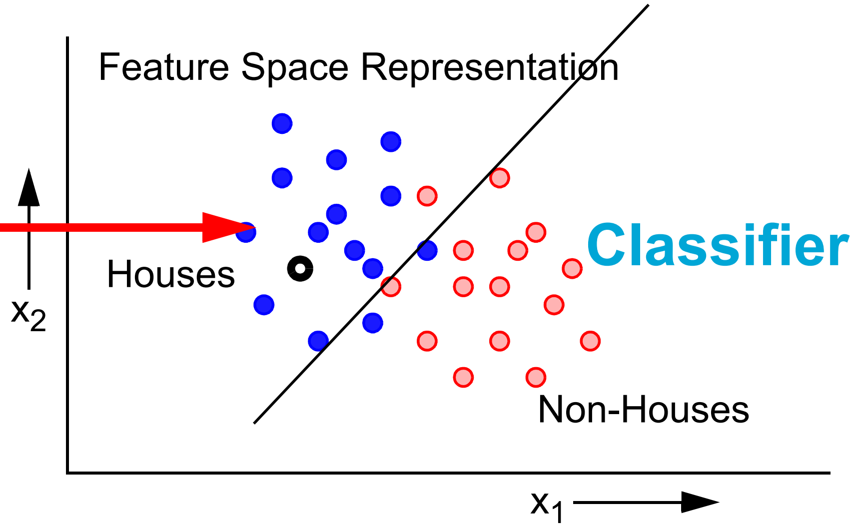
The Netherlands

TU Delft

Model Driven ⟷ Data Driven

Model of a house

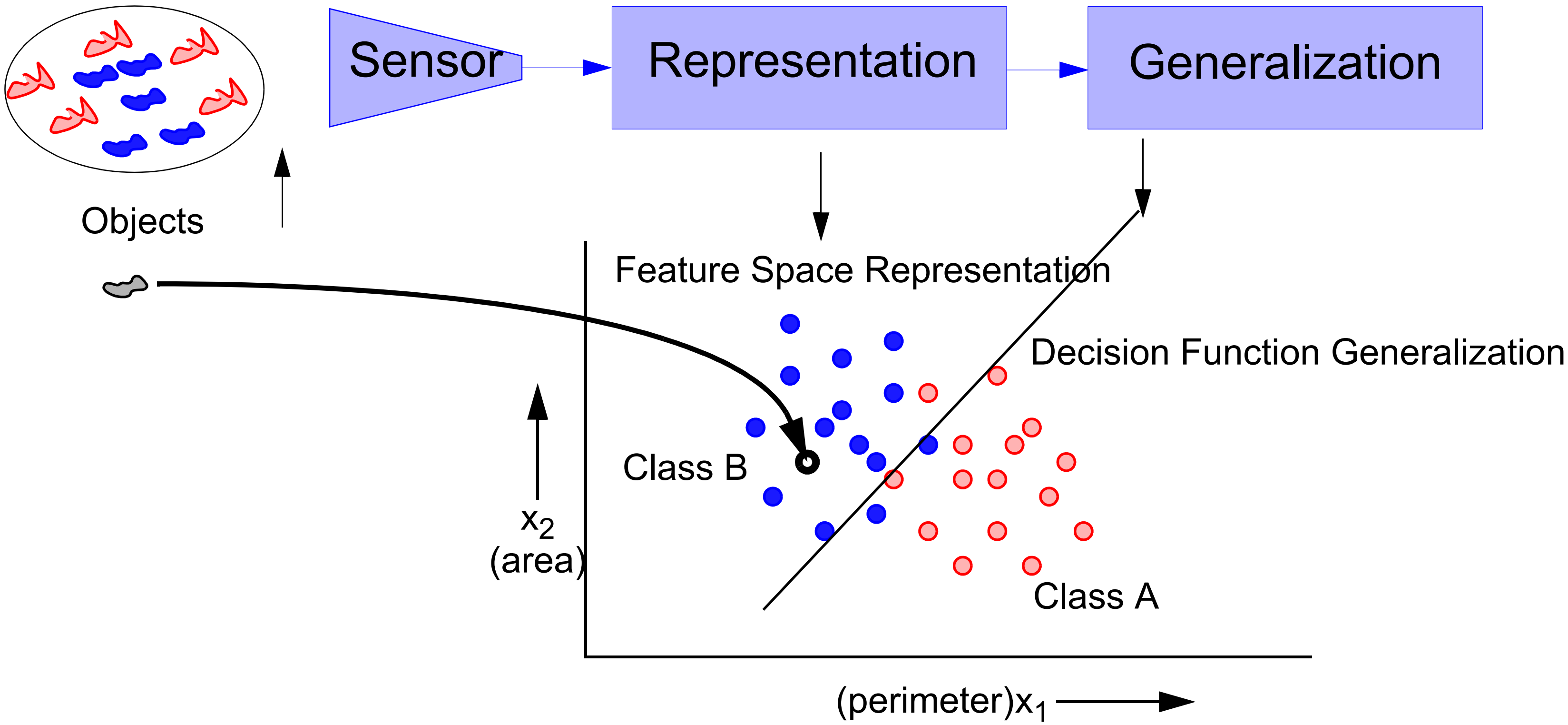
Are these houses?



Observations of houses

TU Delft

The Pattern Recognition System



Learning from examples
Finding concepts (classes) from observations

Representation

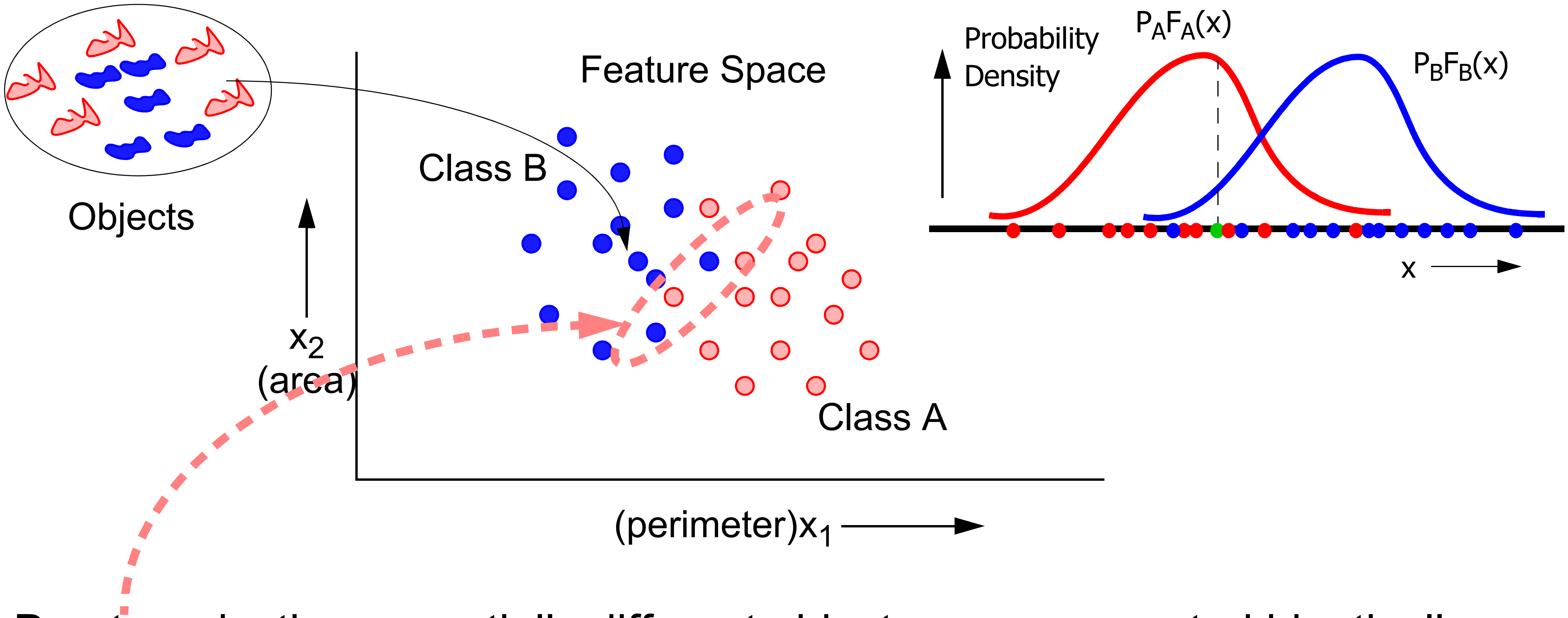
Object representation

- Vector representations
  - Features
  - Samples (Pixels)
  - Dissimilarities
  - Dimensionality problems
- Non-vectorial representations

Class representation

- Sampling: aselective - selective
- Supervised - Unsupervised
- Number of objects

Feature Representation



Due to reduction essentially different objects are represented identically

The feature representation needs a statistical (probabilistic) generalization

TU Delft

Feature Space Assumptions, The Ideal World

- A (small) set of informative features
- Euclidean analysis is possible (after feasible corrections)

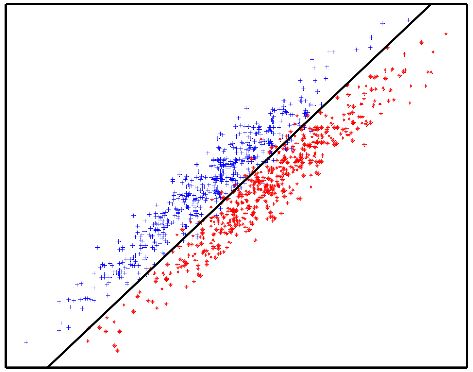
- Classes have known, not very different priors
- Natural classes (e.g. correspond to a unsupervised clustering result)
- Training set is representative for the problem
  e.g. aselectively drawn from the same universe as the test set
  sufficiently large for the given feature size
  classes do not drift
  labels are correct

In this ideal world we can nicely study generalization procedures
Applicable?

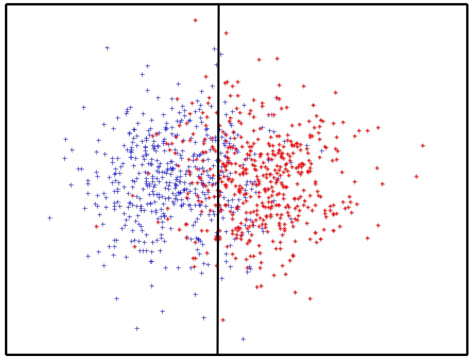
TU Delft

Classifier typology

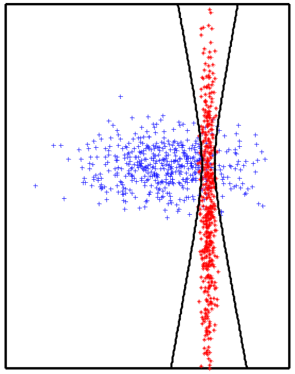
Each classifier has a problem for which it is the best classifier



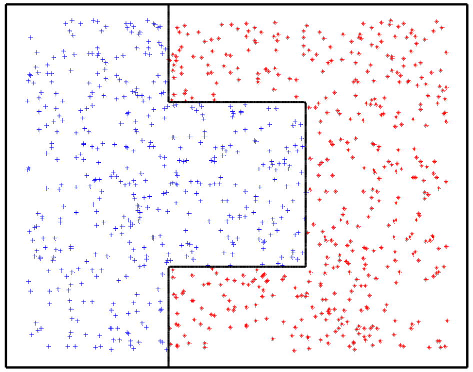
Fisher



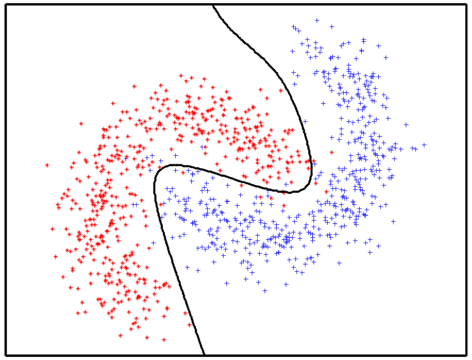
Nearest Mean



Bayes Normal



Decision Tree



Neural Network

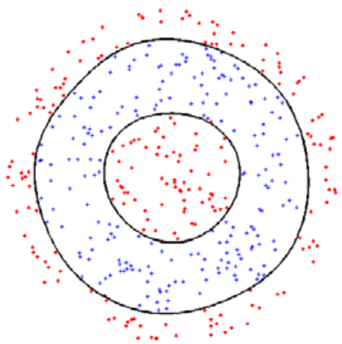
Can we create a

library of problems corresponding to the library of classifiers?

Classifier Problem Archtypes

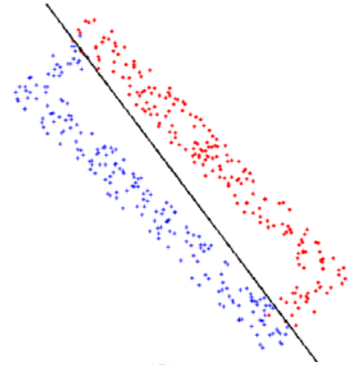
RB-SVM (10)

80 RB-SVM*
131 RB-Neural-Net
146 1-NN
147 Parzen
153 K-NN
270 Dec-Tree
303 LM-Neural-Net
305 Naive-Bayes
467 UDA
473 QDA
499 Logistic
499 LDA
499 Nearest-Mean
500 SVM-1



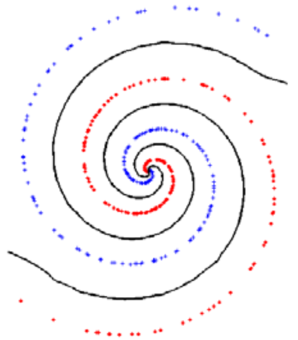
SVM-1 (463)

16 SVM-1*
16 Logistic*
18 RB-SVM*
18 LM-Neural-Net
19 RB-Neural-Net
19 1-NN
21 K-NN
28 LDA
29 QDA
43 Parzen
67 Dec-Tree
132 Nearest-Mean
172 UDA
302 Naive-Bayes



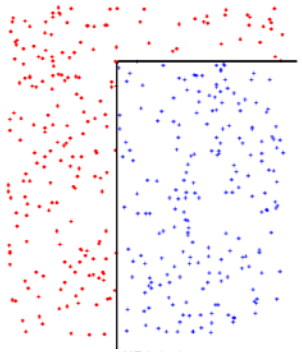
1-NN

13 1-NN*
14 K-NN*
48 Parzen
134 Dec-Tree
180 RB-SVM
205 RB-Neural-Net
273 Naive-Bayes
333 LM-Neural-Net
454 SVM-1
455 LDA
455 Logistic
459 Nearest-Mean
460 UDA
460 QDA



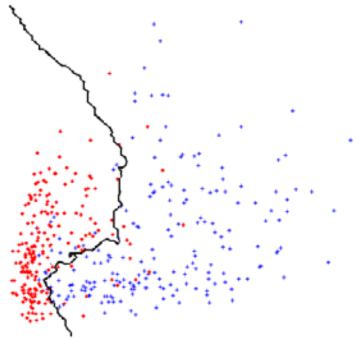
Dec-Tree (44)

27 Dec-Tree*
30 LM-Neural-Net*
36 RB-SVM
47 RB-Neural-Net
51 Parzen
51 Naive-Bayes
52 1-NN
54 K-NN
103 QDA
123 UDA
139 LDA
142 Logistic
143 SVM-1
156 Nearest-Mean



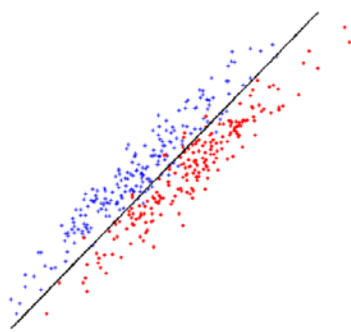
K-NN (6)

157 K-NN*
160 Parzen*
165 RB-Neural-Net*
165 LM-Neural-Net*
184 RB-SVM
187 SVM-1
189 Logistic
190 1-NN
203 QDA
206 LDA
216 Naive-Bayes
219 Nearest-Mean
223 UDA
233 Dec-Tree



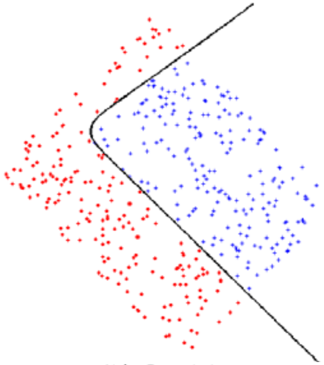
LDA (32)

59 LDA*
60 Logistic*
60 SVM-1*
61 QDA*
81 LM-Neural-Net
83 RB-SVM
89 RB-Neural-Net
95 Parzen
109 K-NN
116 1-NN
151 Dec-Tree
290 Nearest-Mean
312 UDA
365 Naive-Bayes



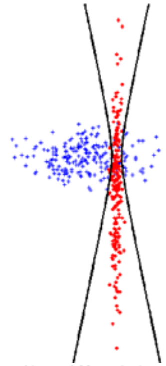
LM-Neural-Net (10)

32 LM-Neural-Net*
36 RB-SVM*
44 RB-Neural-Net
50 Parzen
52 1-NN
54 K-NN
89 Dec-Tree
103 QDA
111 UDA
136 Naive-Bayes
139 LDA
142 Logistic
143 SVM-1
156 Nearest-Mean



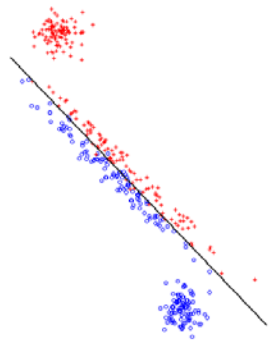
UDA (12)

53 UDA*
54 QDA
81 RB-Neural-Net
89 RB-SVM
92 Naive-Bayes
93 Parzen
93 Dec-Tree
99 LM-Neural-Net
102 1-NN
104 K-NN
180 Logistic
193 SVM-1
202 LDA
258 Nearest-Mean



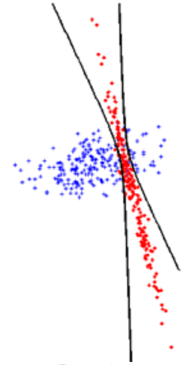
Logistic (566)

51 Logistic*
51 SVM-1*
72 RB-SVM
74 RB-Neural-Net
75 LDA
81 QDA
98 LM-Neural-Net
112 1-NN
118 K-NN
124 Parzen
134 Dec-Tree
233 Nearest-Mean
246 UDA
255 Naive-Bayes



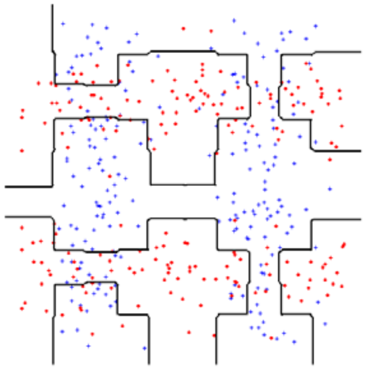
QDA (10)

54 QDA*
86 LM-Neural-Net
89 RB-SVM
90 RB-Neural-Net
92 Parzen
102 1-NN
104 K-NN
124 Dec-Tree
162 UDA
180 Logistic
187 Naive-Bayes
193 SVM-1
202 LDA
258 Nearest-Mean



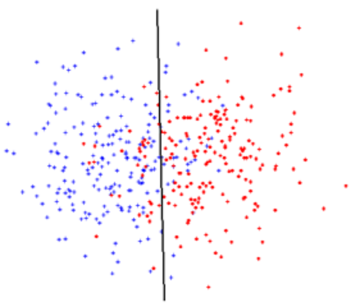
Naive-Bayes (11)

317 Naive-Bayes*
328 RB-Neural-Net
333 Parzen
342 RB-SVM
352 K-NN
353 1-NN
377 LM-Neural-Net
377 Dec-Tree
493 UDA
500 SVM-1
500 Nearest-Mean
501 LDA
501 Logistic
502 QDA



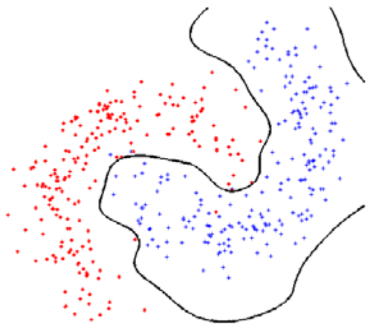
Nearest-Mean (39)

148 Nearest-Mean*
150 LDA*
150 Logistic*
150 SVM-1*
151 UDA
153 QDA
165 Parzen
165 K-NN
179 Naive-Bayes
193 LM-Neural-Net
196 RB-Neural-Net
225 1-NN
226 RB-SVM
228 Dec-Tree



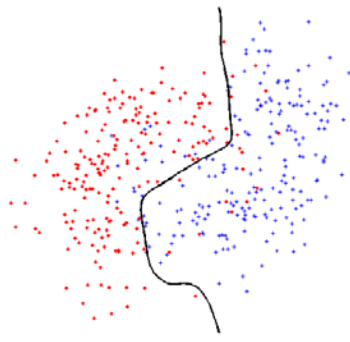
RB-Neural-Net (152)

29 RB-Neural-Net*
30 Parzen*
33 K-NN
34 RB-SVM
36 1-NN
56 LM-Neural-Net
69 Dec-Tree
124 Naive-Bayes
159 Logistic
159 LDA
160 SVM-1
161 QDA
203 UDA
205 Nearest-Mean

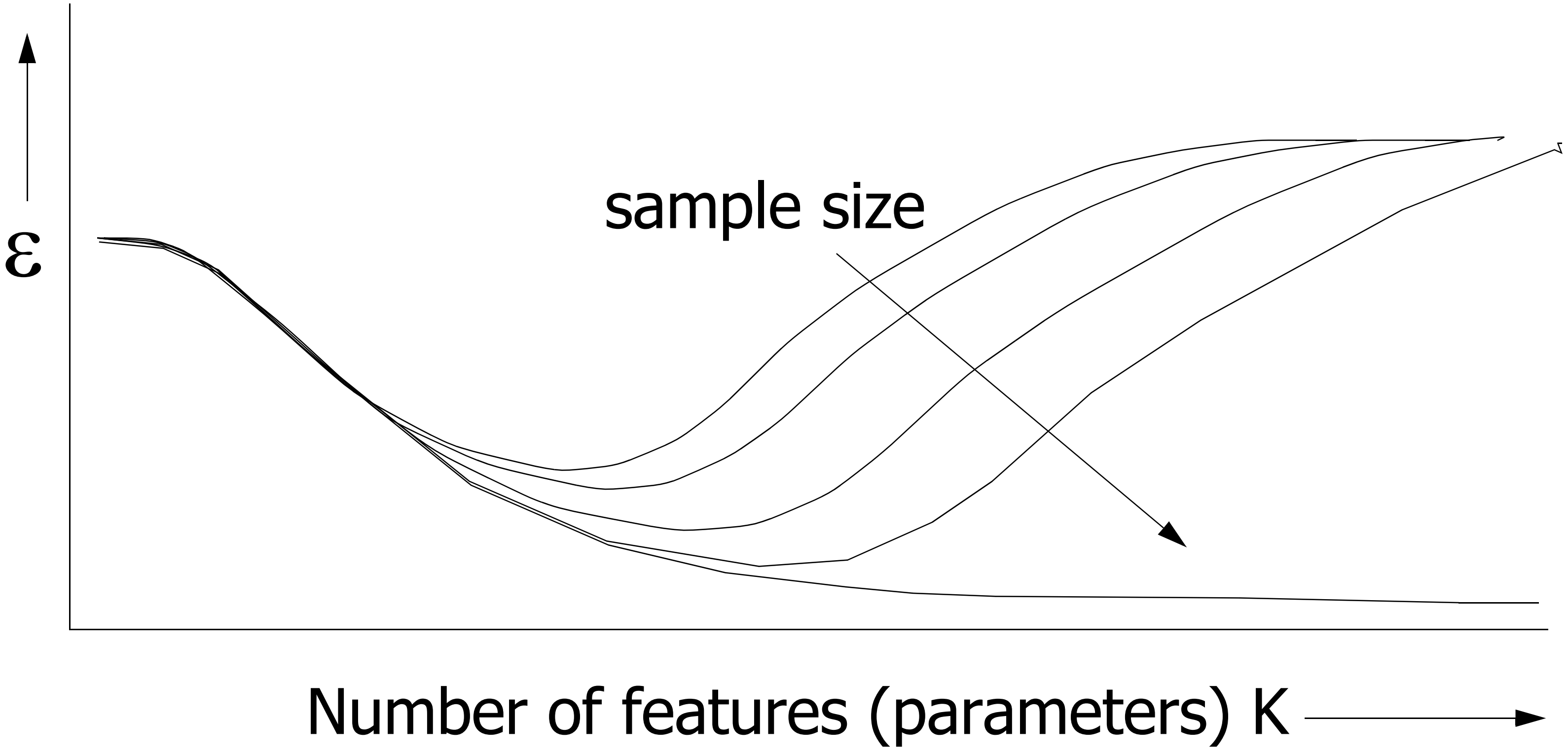


Parzen (14)

146 Parzen*
150 K-NN*
158 RB-Neural-Net
171 LM-Neural-Net
184 Logistic
184 LDA
184 SVM-1
184 RB-SVM
185 QDA
191 1-NN
199 Naive-Bayes
208 Nearest-Mean
208 UDA
232 Dec-Tree



Bad Features → More Features → Complexity Problem



The real world, first problem: peaking

TUDelft

No Feature Reduction

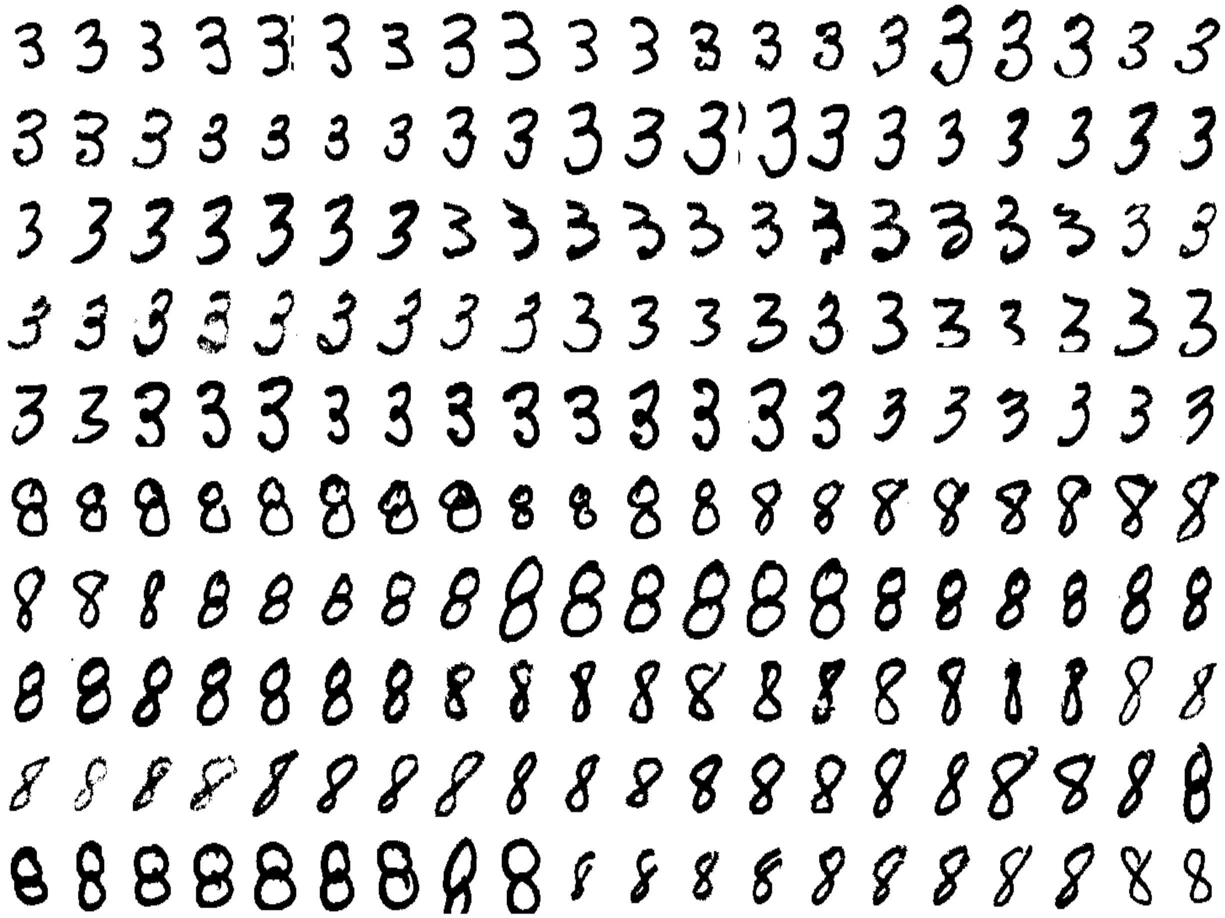
The feature representation enforces class overlap.
To be solved by a probabilistic approach.

However:

Are densities needed in high dimensional spaces?
Are classes to be represented by densities?

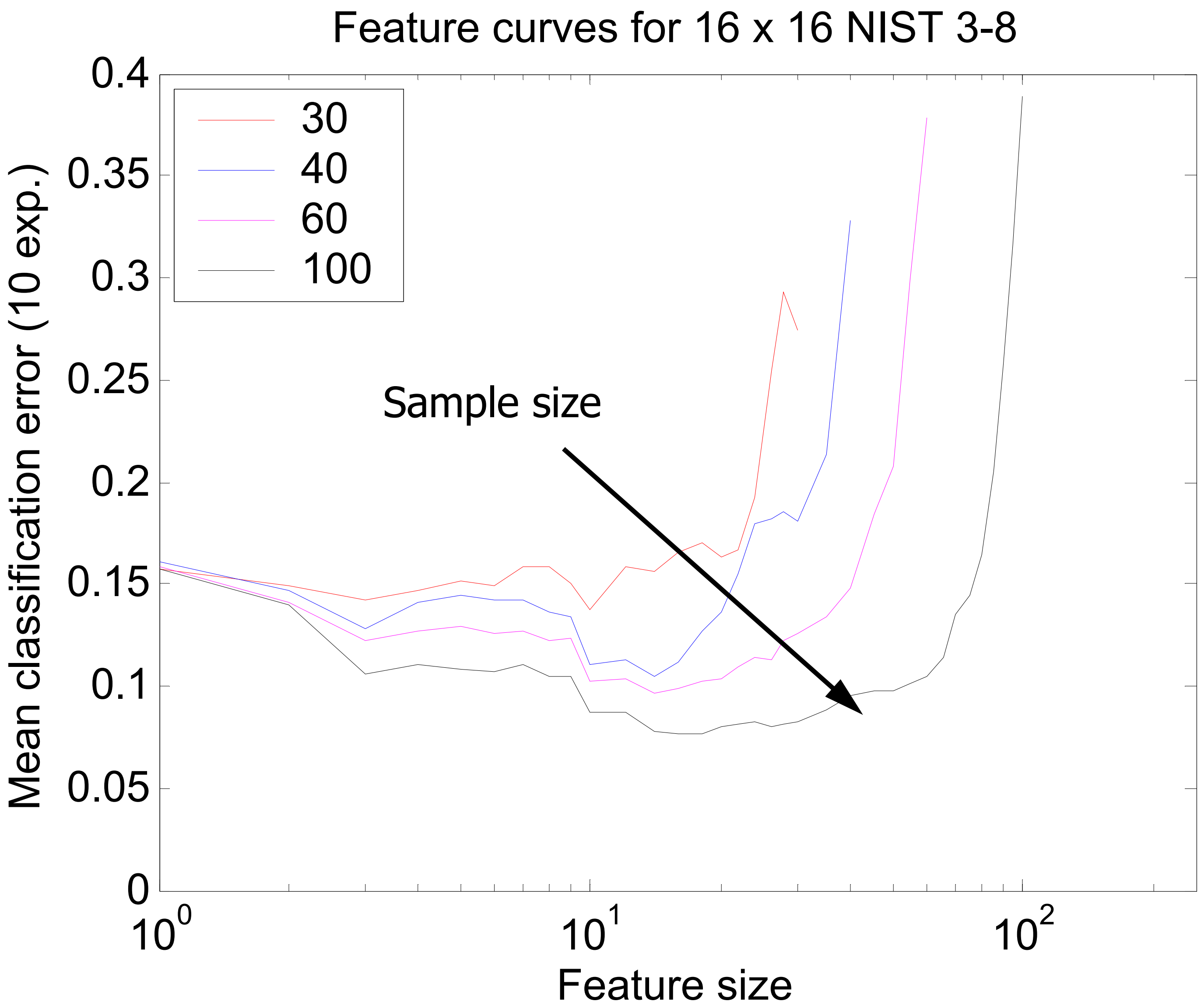
Can we construct domain based classifiers?

Example Dissimilarity Space: NIST Digits 3 and 8



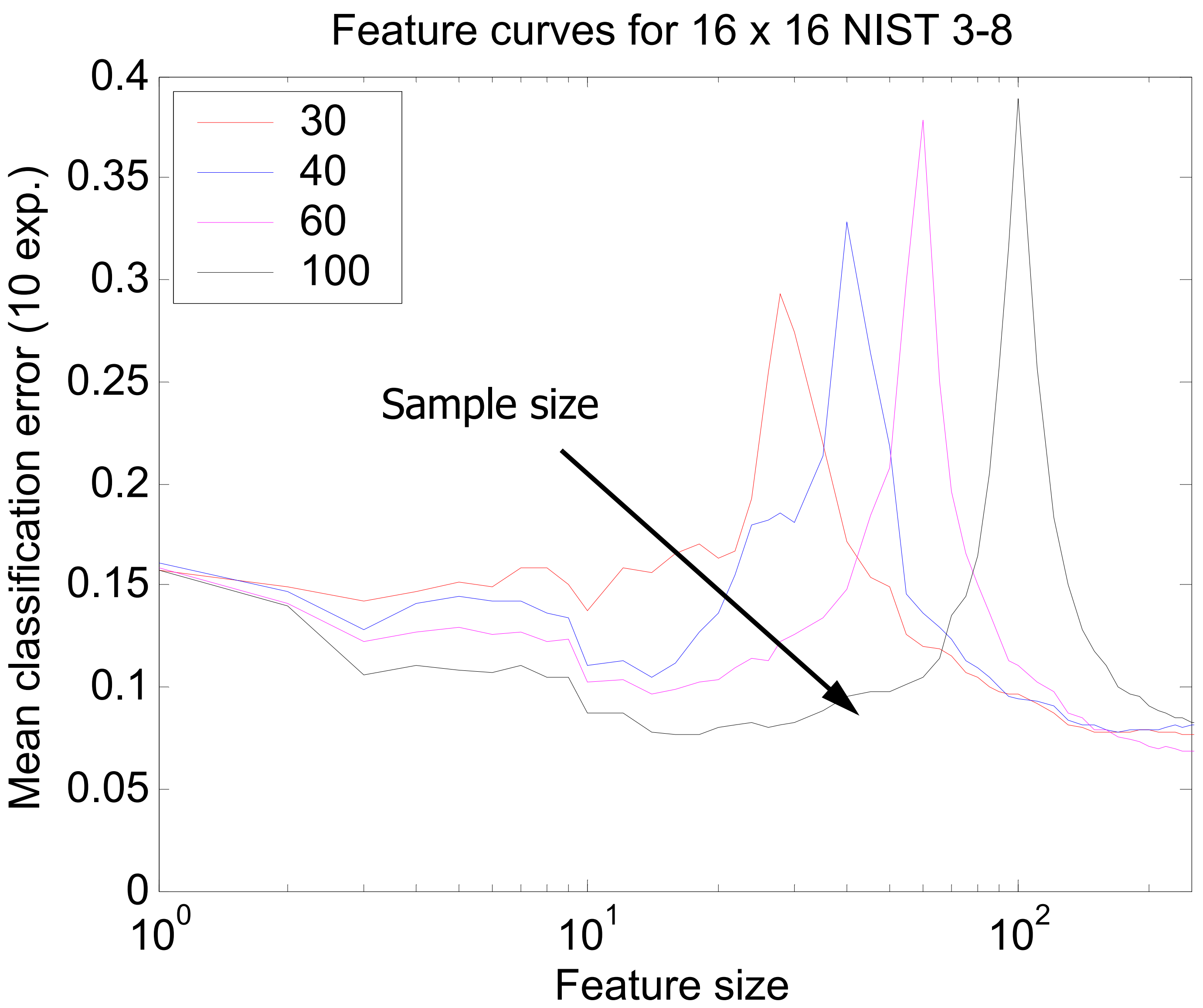
Examples of the raw data

Peaking



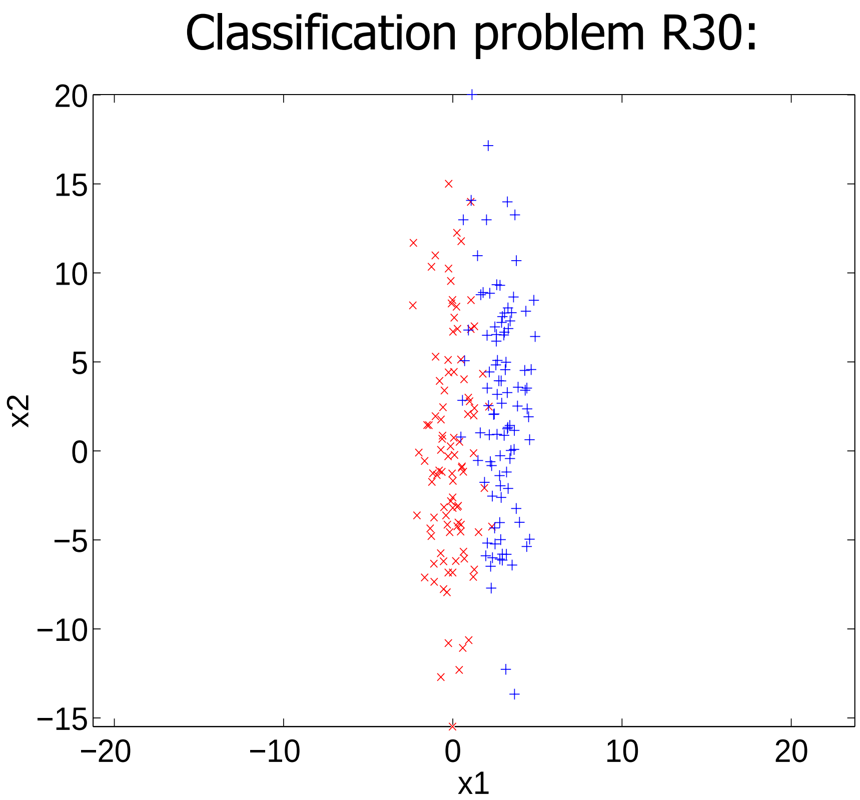
Feature curves for 16 x 16 NIST 3-8
30
40
60
100
Sample size
Mean classification error (10 exp.)
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
10^0
10^1
10^2
Feature size

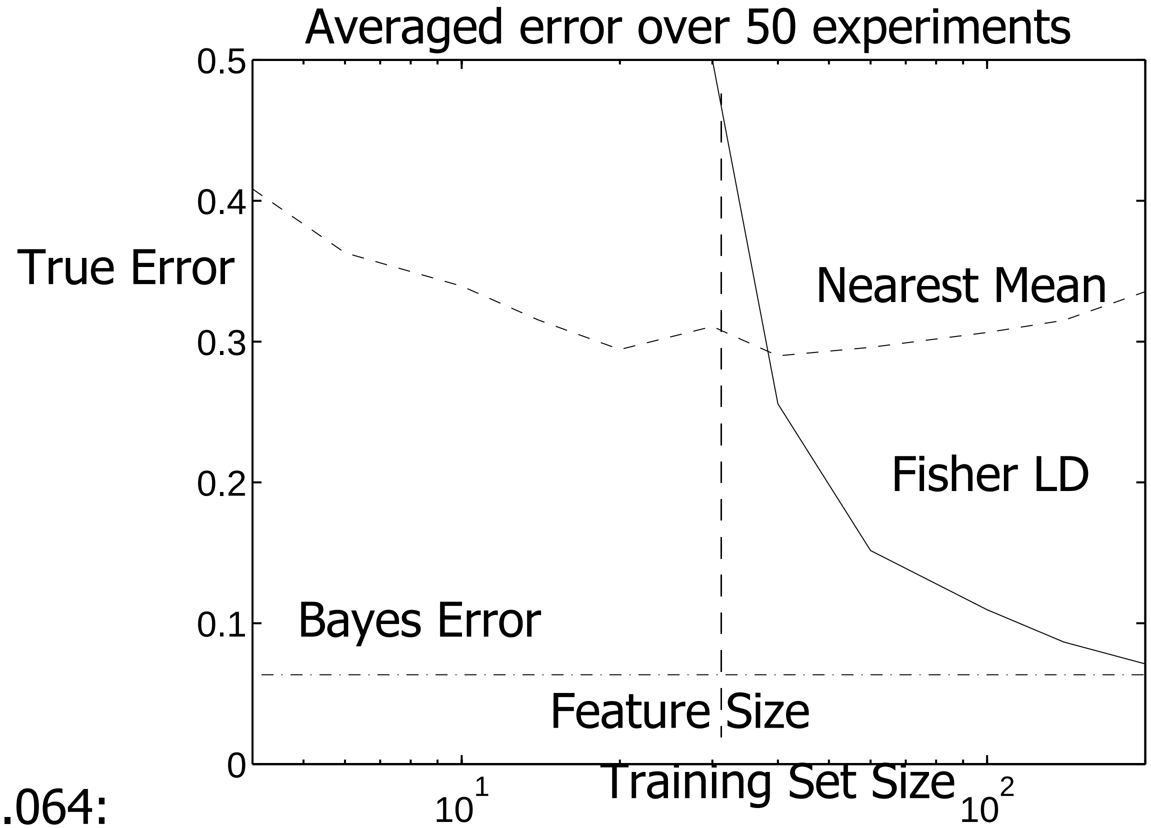
Overpeaking



TU Delft

Small Sample Size





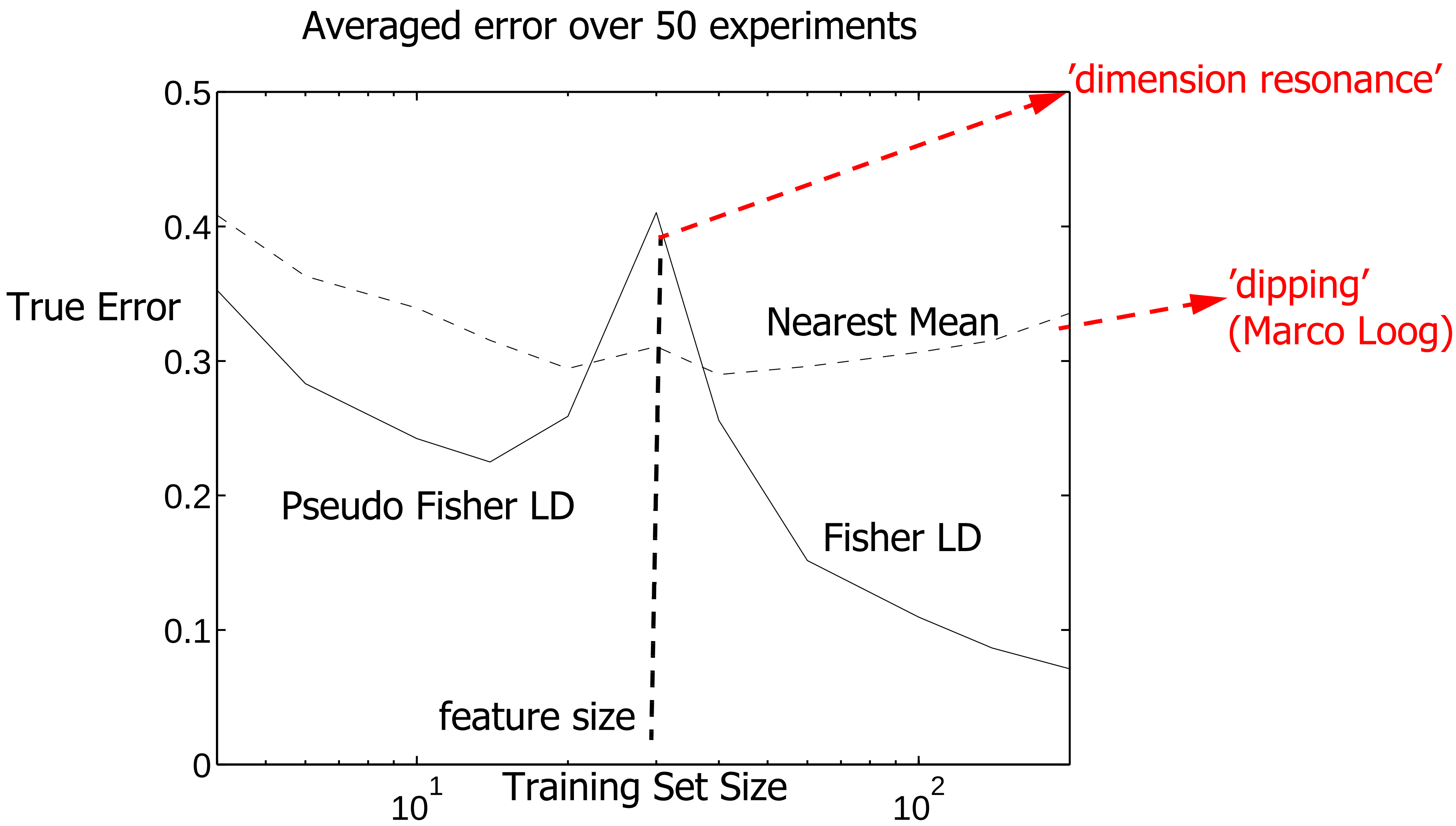
Two normal distributions, overlap: $\varepsilon^* = 0.064$:

feature 1 $N_A(0, 1)$, $N_B(3, 1)$

feature 2 $N_A(0, 40)$, $N_B(3, 40)$

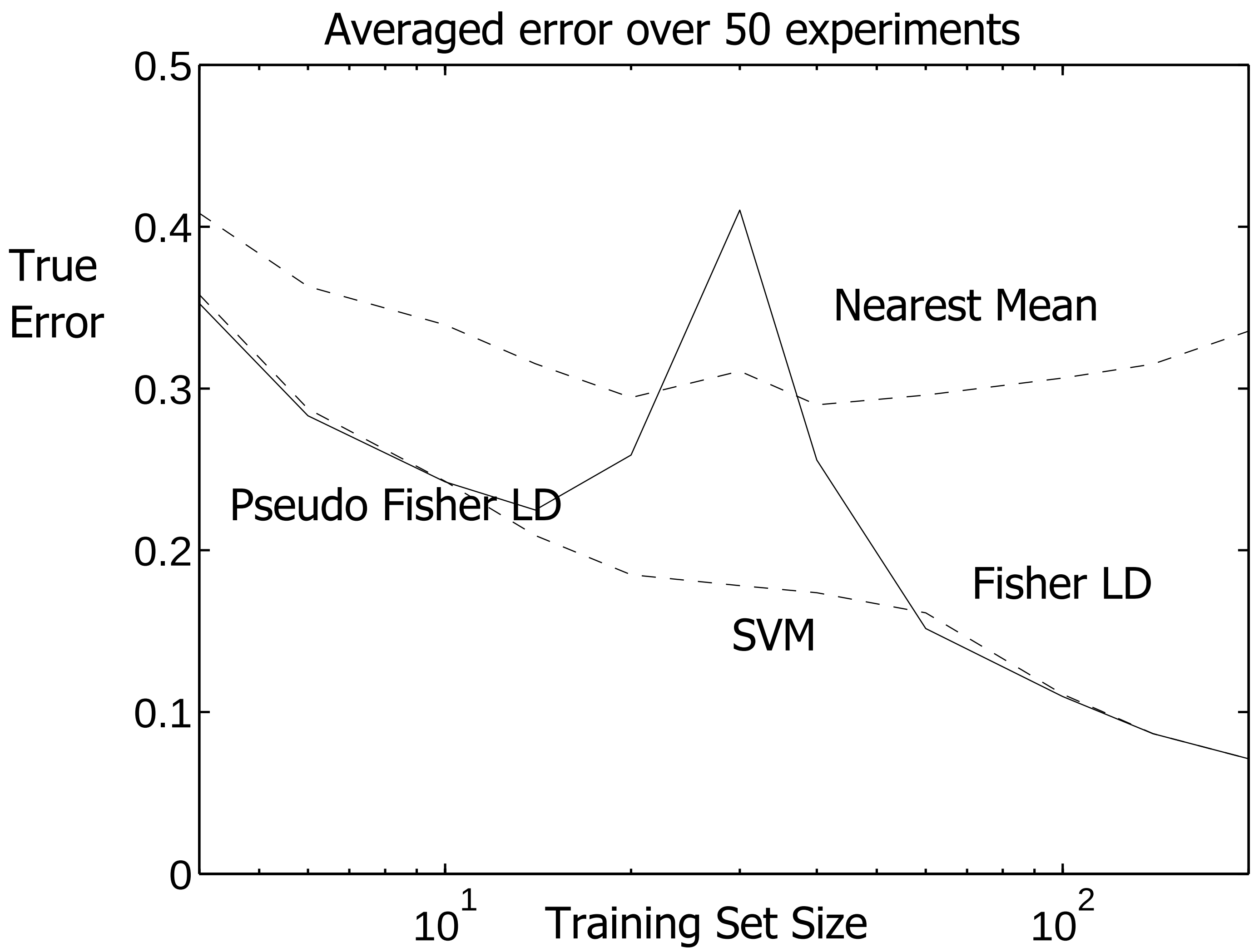
feature 3-30 $N_A(0, 1)$, $N_B(0, 1)$

Dimension Resonance and Dipping



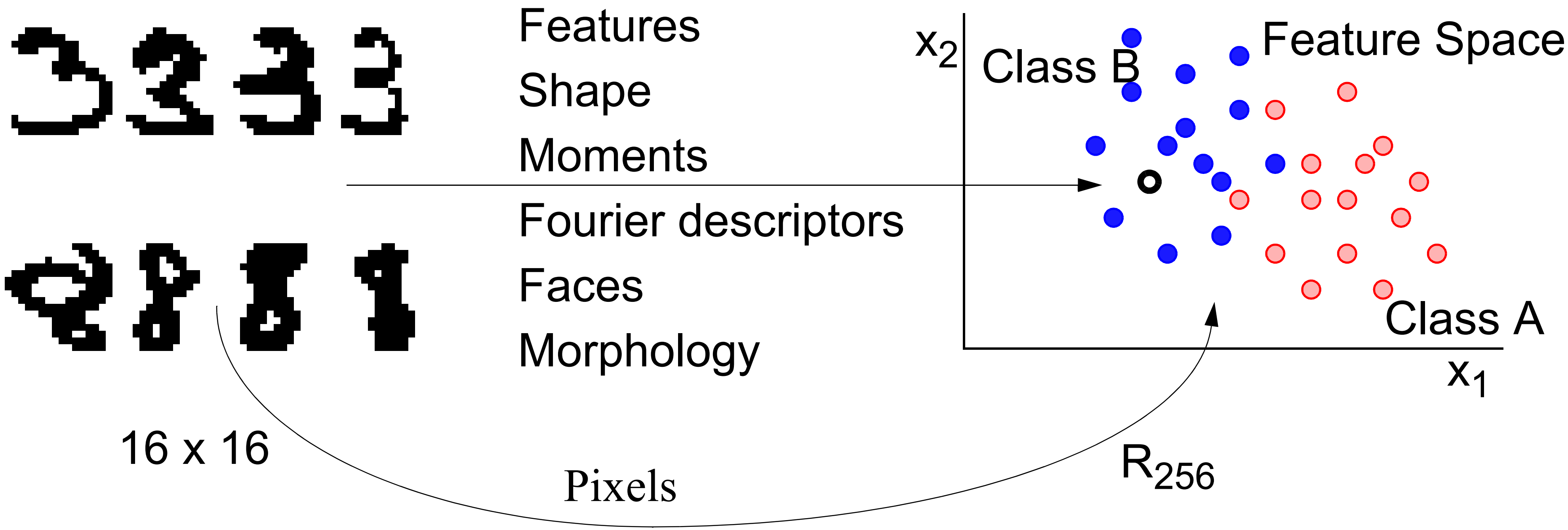
TUDelft

Support Vector Machine for Small Sample Sizes



TU Delft

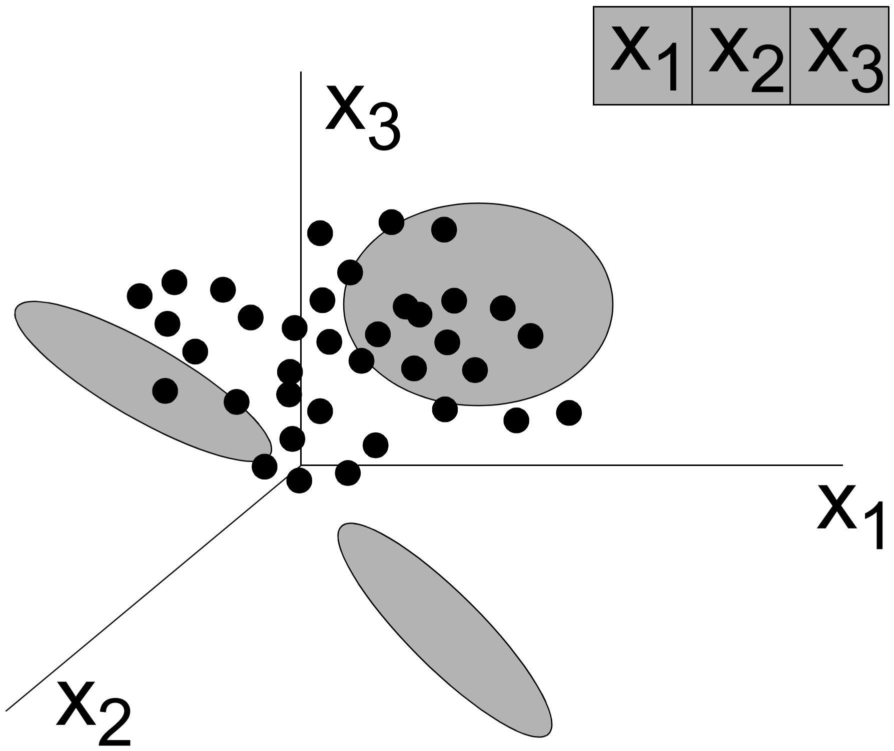
Pixel Representation: Samples Instead of Features



Pixels are more general, initially complete representation
Large datasets available → good results for OCR

The Connectivity Problem in the Pixel Representation

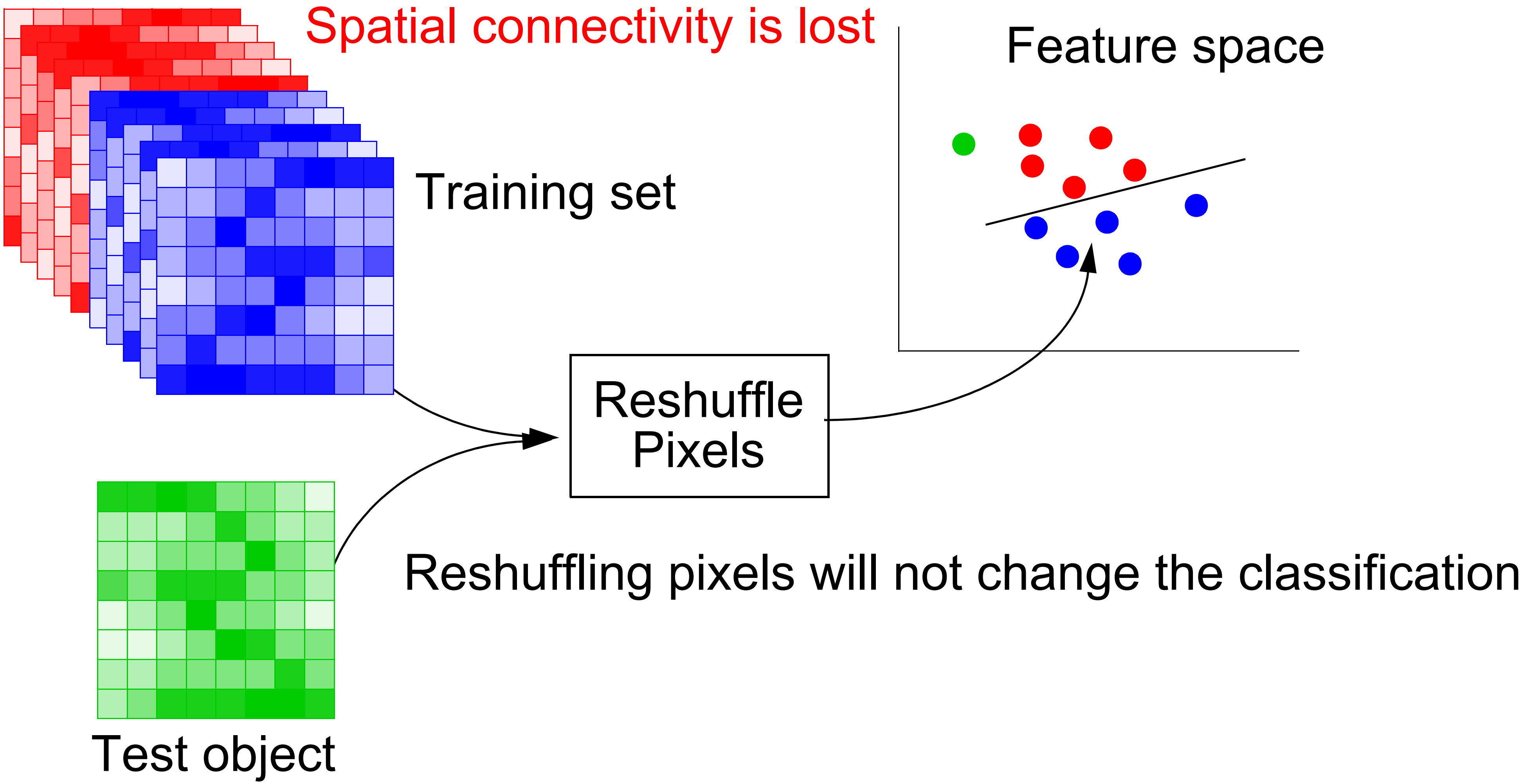
Spatial connectivity is lost



Dependent (connected) measurements are represented independently,
The dependency has to be rediscovered from the data.

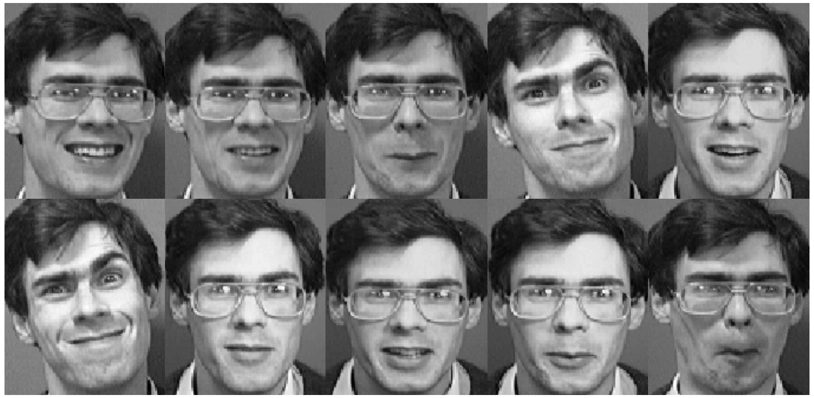
TU Delft

The Connectivity Problem in the Pixel Representation



Can connectivity be taken into account in the representation?

High dimensional data often does not overlap

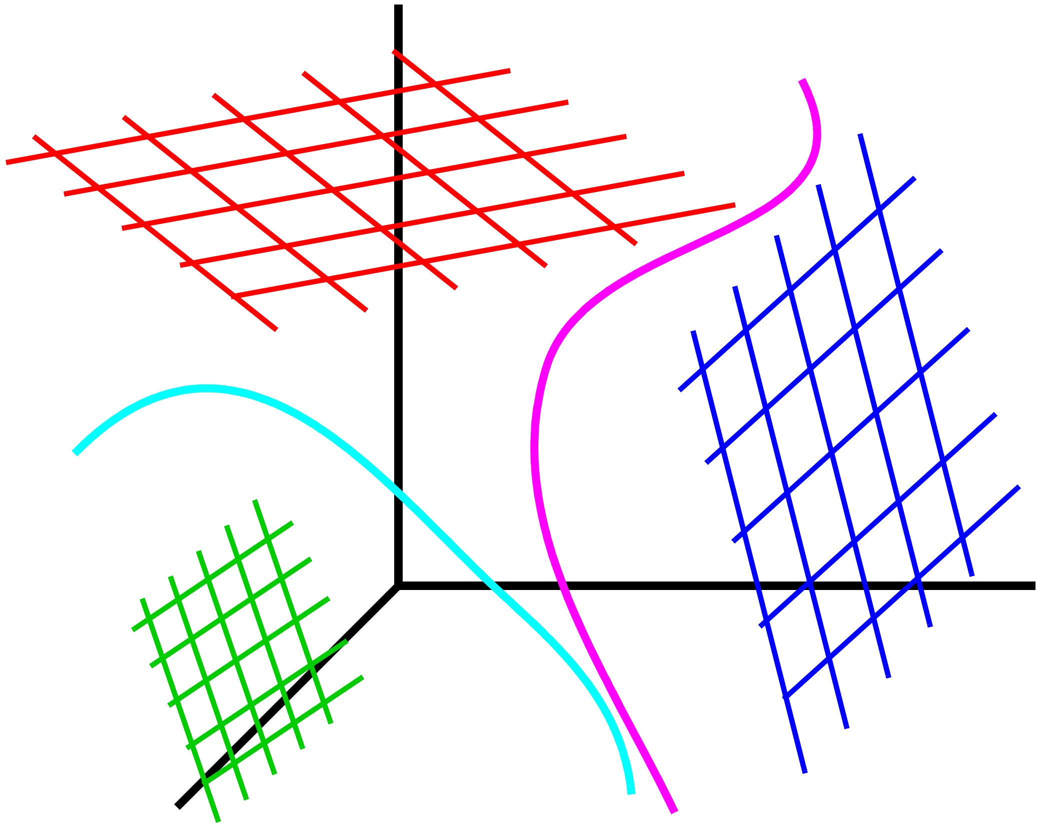


Complete feature representations, which enable the reconstruction of human recognizable, may yield separable classes.

There is no picture that could be member of different classes.

In some representations classes are separable

Domains instead of Densities

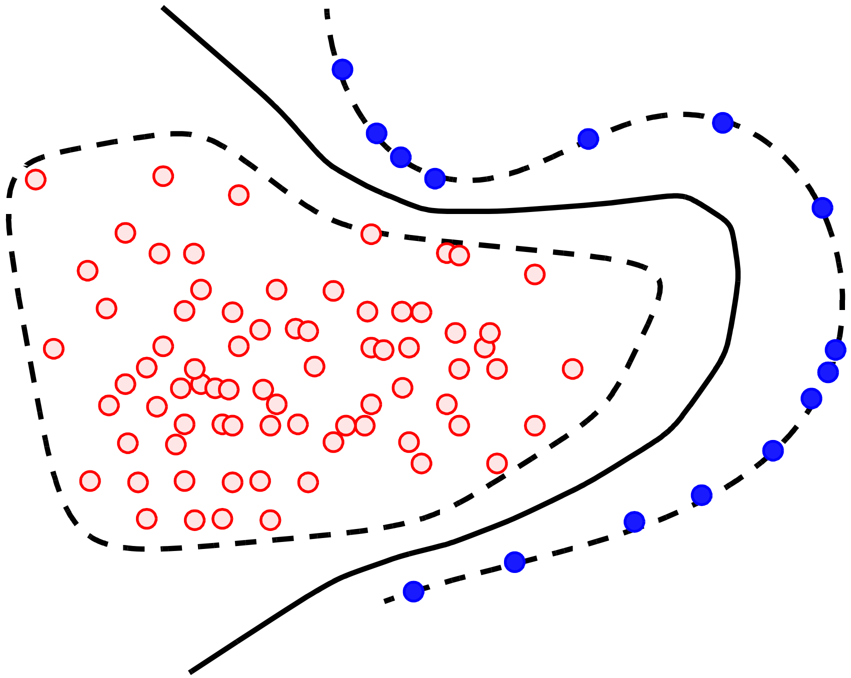


No well sampled training sets are needed.

Classifiers still to be developed.

Class structure $\longleftrightarrow$ Object invariants

Domain based classification



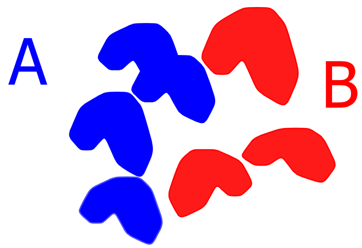
Don't trust class densities
Estimate for each class a domain
Assign new objects to nearest domain

Outlier dependent
Distances instead of densities

How to construct domain based classifiers?

No Features: Dissimilarities

Define dissimilarity measure d_{ij} between raw data of objects i and j



Given labeled training set T

Unlabeled object x to be classified

$$D_T = \begin{pmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} & d_{17} \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & d_{26} & d_{27} \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} & d_{36} & d_{37} \\ d_{41} & d_{42} & d_{43} & d_{44} & d_{45} & d_{46} & d_{47} \\ d_{51} & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} & d_{57} \\ d_{61} & d_{62} & d_{63} & d_{64} & d_{65} & d_{66} & d_{67} \\ d_{71} & d_{72} & d_{73} & d_{74} & d_{75} & d_{76} & d_{77} \end{pmatrix}$$

not used by NN rule

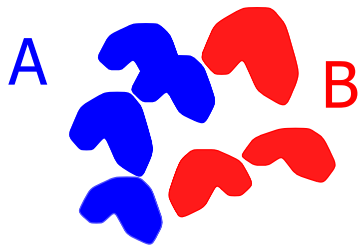
$$\mathbf{d}_x = (d_1 \;\; d_2 \;\; d_3 \;\; d_4 \;\; d_5 \;\; d_6 \;\; d_7)$$

The traditional Nearest Neighbor rule (template matching) just finds:
label(argmin$_{trainset}$(d_i)),
without using D_T. Can we do any better?

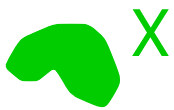
TU Delft

No Features: Dissimilarities

Define dissimilarity measure d_{ij} between raw data of objects i and j



Given labeled training set T



Unlabeled object x to be classified

$$D_T = \begin{pmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} & d_{17} \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & d_{26} & d_{27} \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} & d_{36} & d_{37} \\ d_{41} & d_{42} & d_{43} & d_{44} & d_{45} & d_{46} & d_{47} \\ d_{51} & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} & d_{57} \\ d_{61} & d_{62} & d_{63} & d_{64} & d_{65} & d_{66} & d_{67} \\ d_{71} & d_{72} & d_{73} & d_{74} & d_{75} & d_{76} & d_{77} \end{pmatrix}$$

not used by NN rule

$$\mathbf{d}_x = (d_1 \;\; d_2 \;\; d_3 \;\; d_4 \;\; d_5 \;\; d_6 \;\; d_7)$$

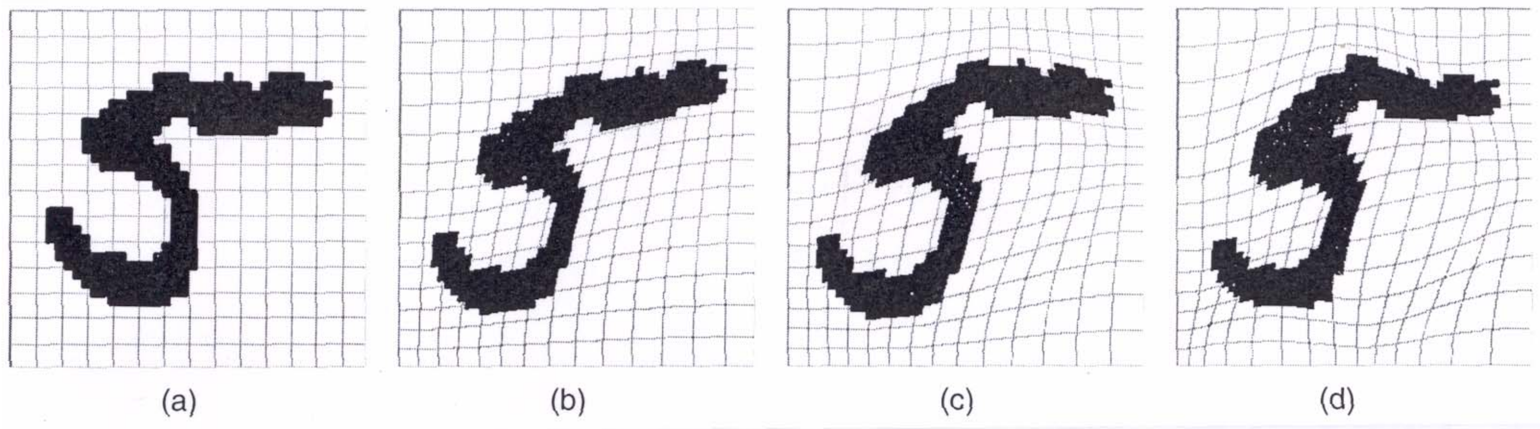
The traditional Nearest Neighbor rule (template matching) just finds:
label($\text{argmin}_{\text{trainset}}(d_i)$),
without using D_T. Can we do any better?

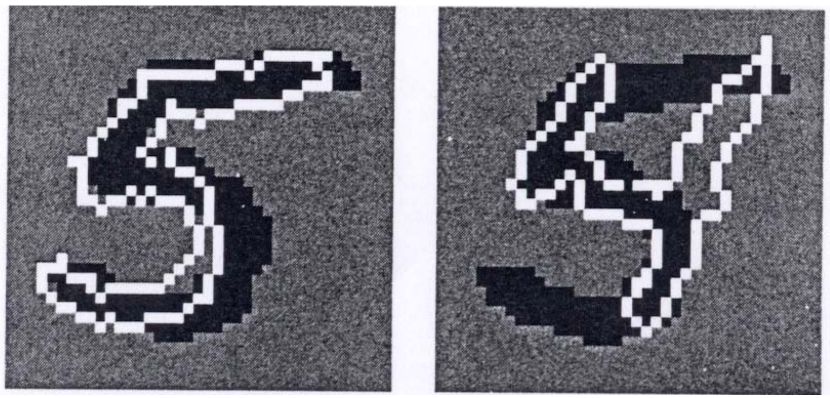
Pekalska, The Dissimilarity Representation for PR, World Scientific, 2005

TU Delft

Example: Deformable Templates

Examples of deformed templates





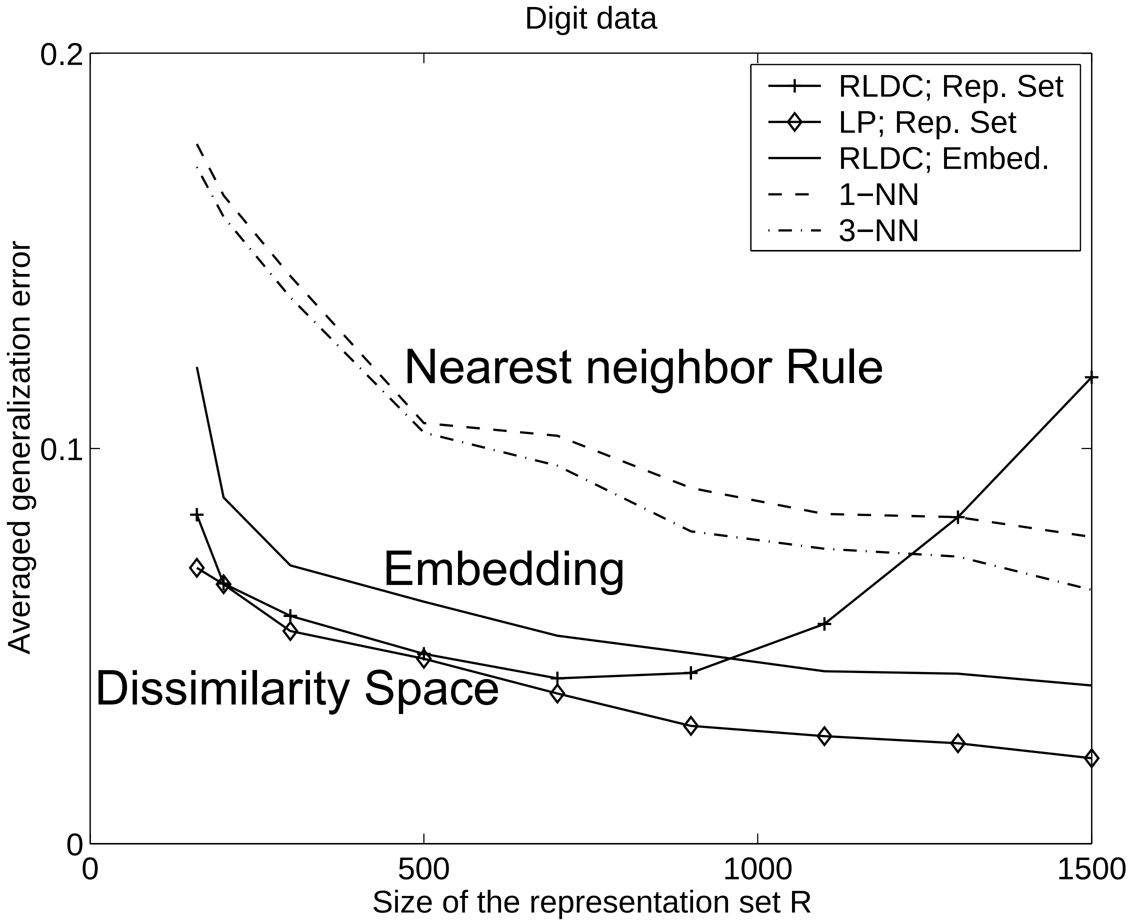
Matching new objects x to various templates y

class(x) = class(argmin$_y$(D(x, y)))

Dissimilarity measure appears to be non-metric

A.K. Jain, D. Zongker, Representation and recognition of handwritten digit using deformable templates, IEEE-PAMI, vol. 19, no. 12, 1997, 1386-1391.

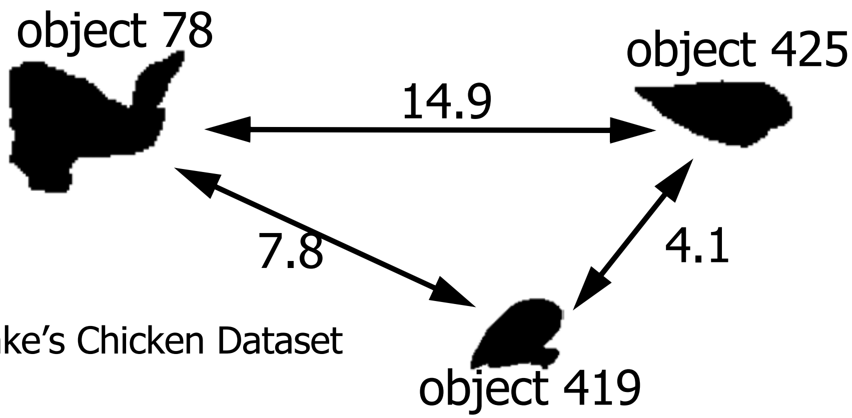
Three Approaches Compared for the Zongker Data



Dissimilarity Space better than Embedding better than Nearest Neighbor Rule

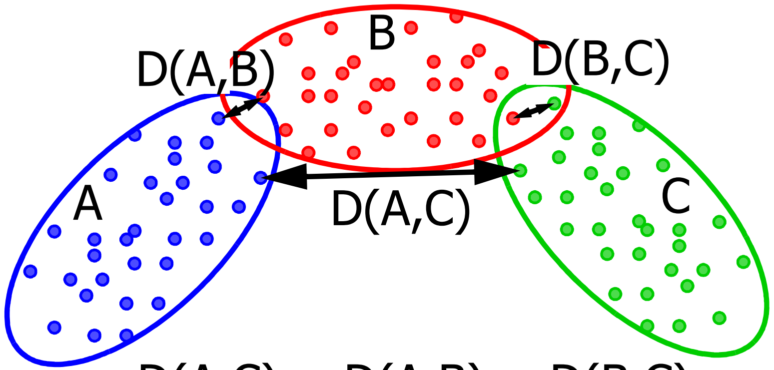
The Non-Euclidean World of Pattern Recognition

Weighted edit-distance for strings



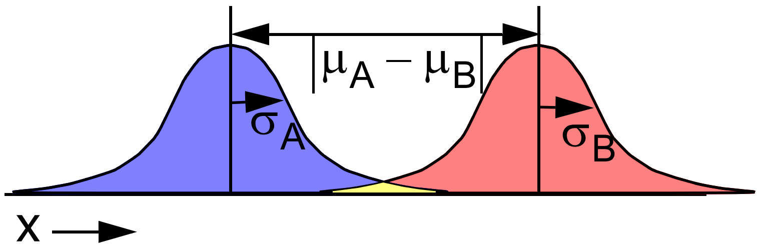
Bunke's Chicken Dataset

Single Linkage Clustering

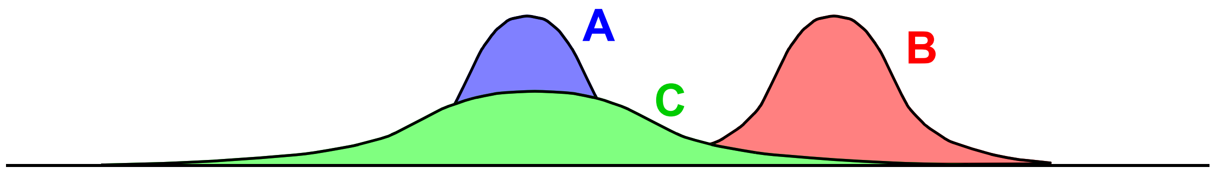


$$D(A,C) > D(A,B) + D(B,C)$$

The Fisher Criterion



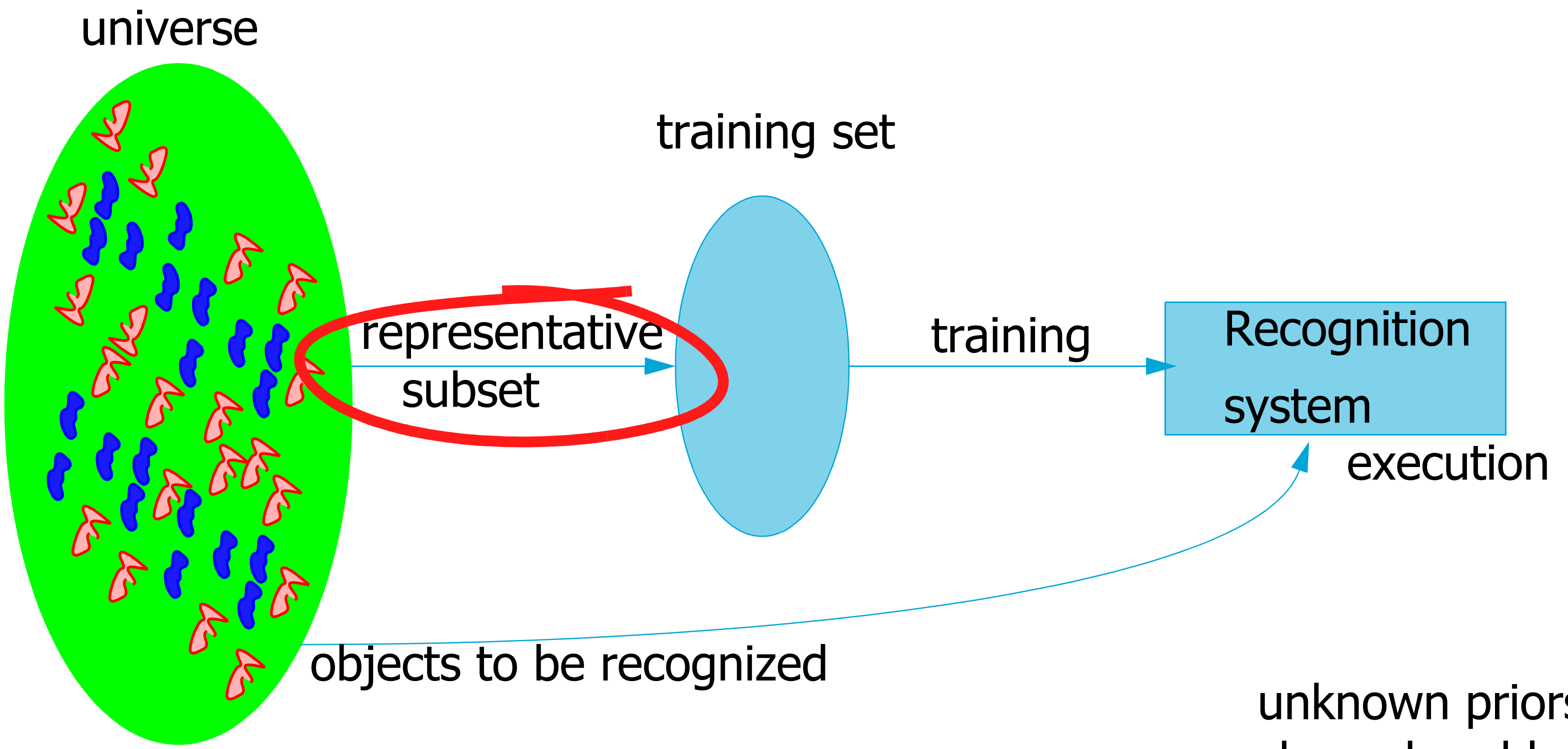
$$J(A, B) = \frac{|\mu_A - \mu_B|^2}{\sigma_A^2 + \sigma_B^2}$$



$J(A,C) = 0$; $J(A,B) =$ large; $J(C,B) =$ small $\neq J(A,B)$

TU Delft

Class Representation Problems

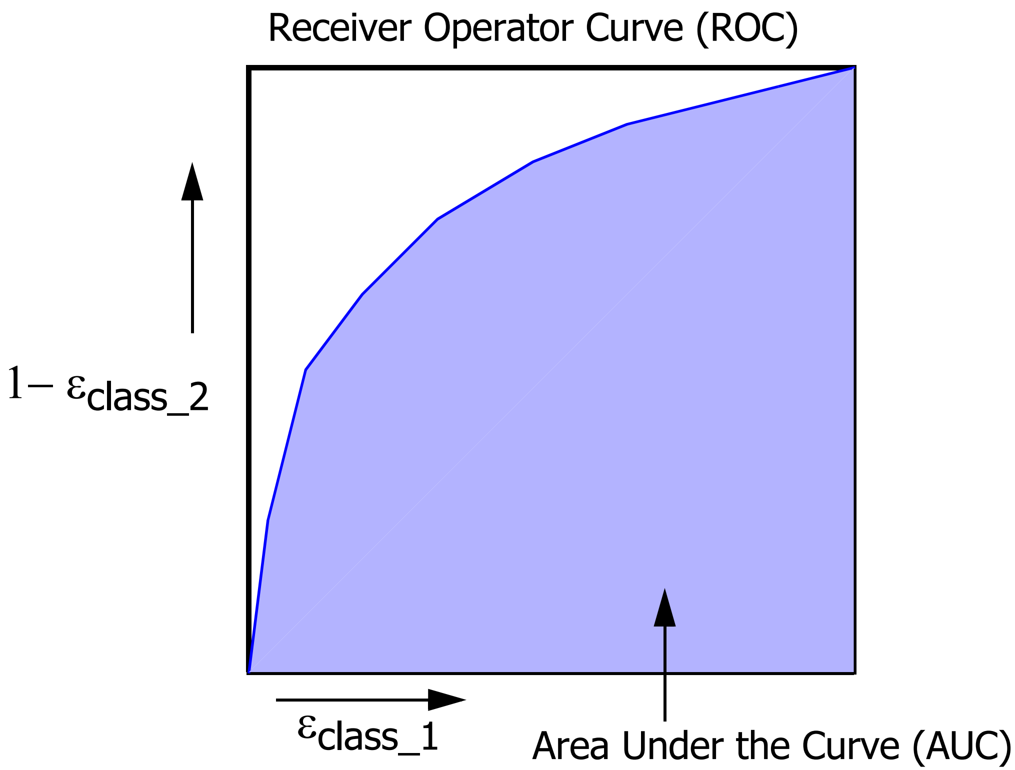


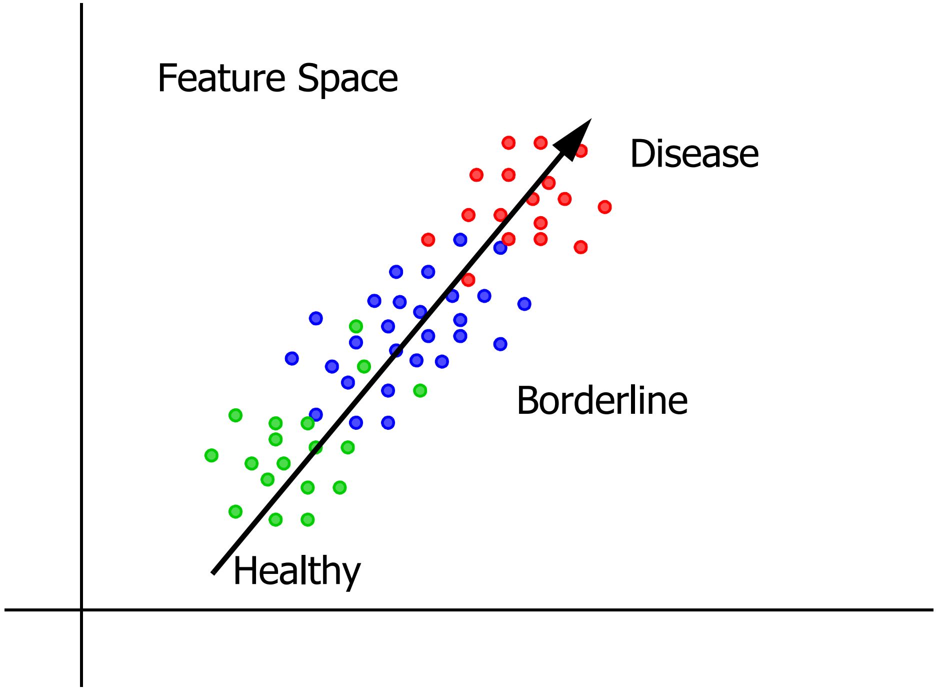
unknown priors
skewed problems
ill sampled problems
label uncertainty
population drift

What to do if no good definition of the universe can be found?

TU Delft

ROC, AUC

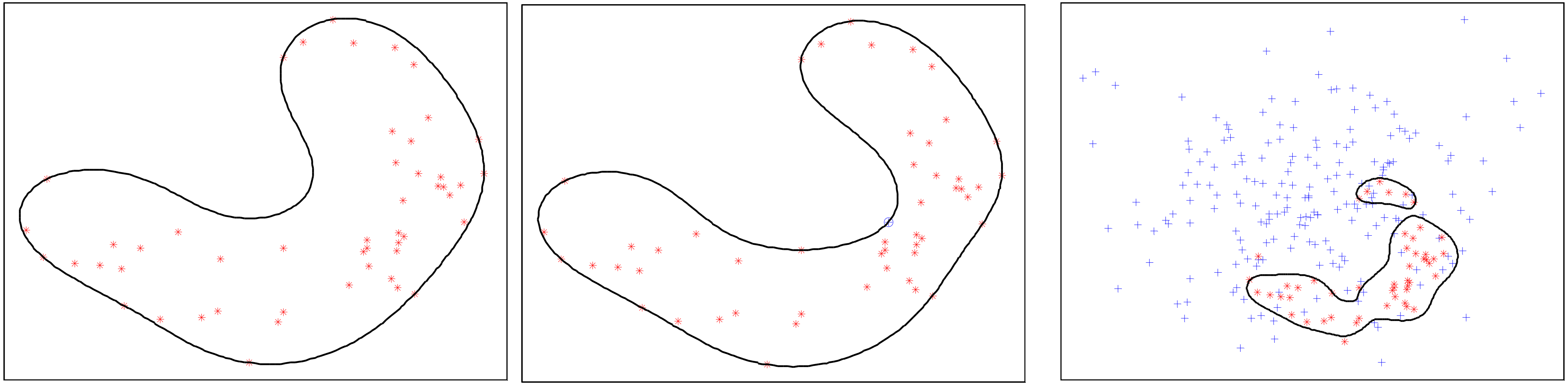




1 AUC: Robust performance measure
(unknown priors/costs, unbalanced sampling)

2 AUC optimizing classifiers may find 'good' directions in case of higly overlapping, ill defined classes

One-class problems



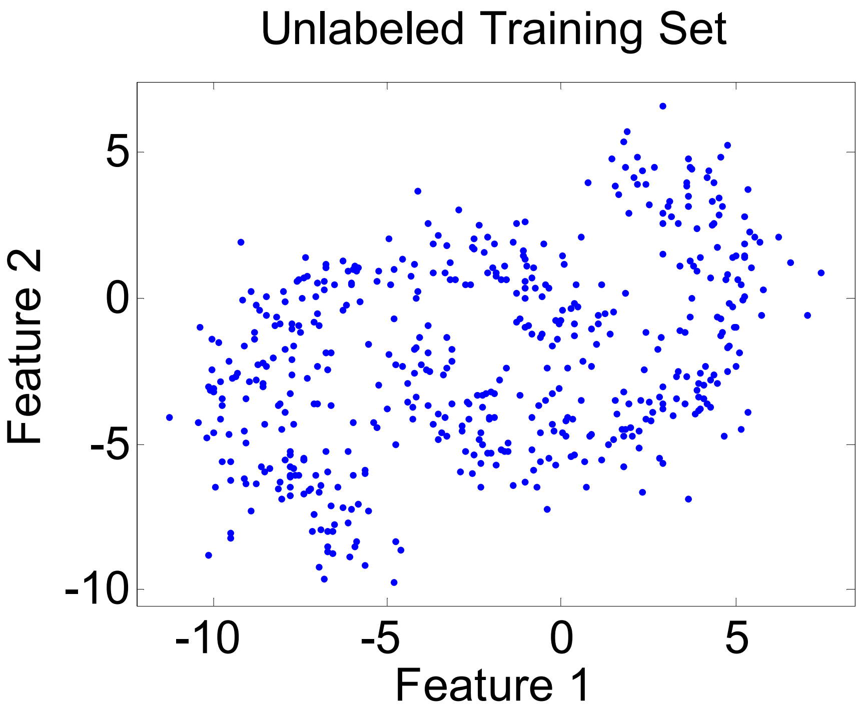
training set of a single class only + an outlier in a sea of outliers

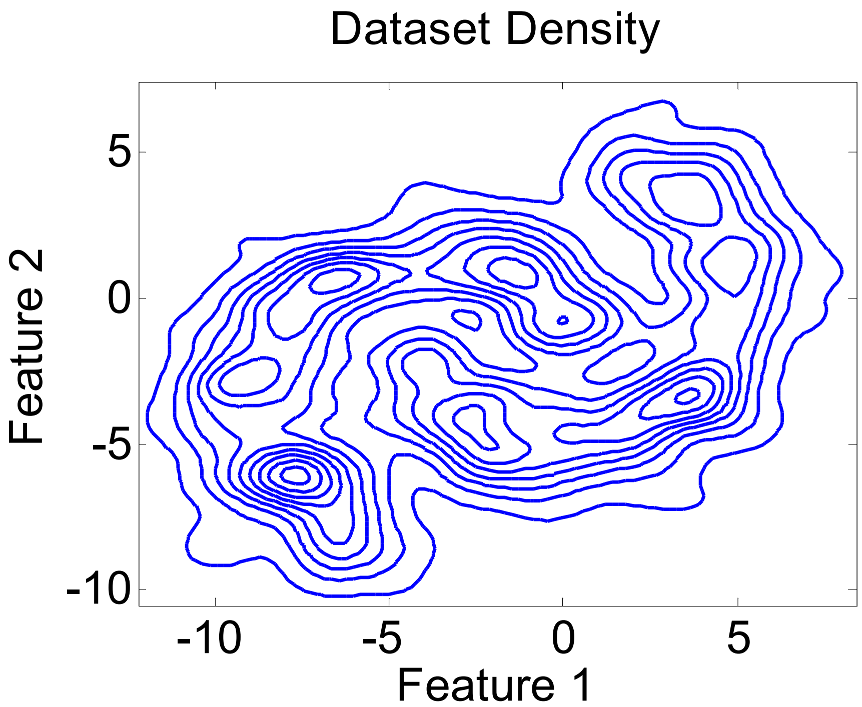
How to generalize well: no empty areas included
stay outside boundary objects

What is a proper one-class classifier?

TU Delft

An Opportunity: Large Unlabeled Training Set





- Given: A large, but finite, unlabeled training set X_u, or a density function.
- Ask labels for a small set of objects (of given size), X_l.
- Task: design a classifier, or label X_u.

TU Delft

Approaches

Selective Sampling:

- Determine a small set of objects from X_u that represents the dataset well
- Ask for the labels: X_l
- Train a classifier

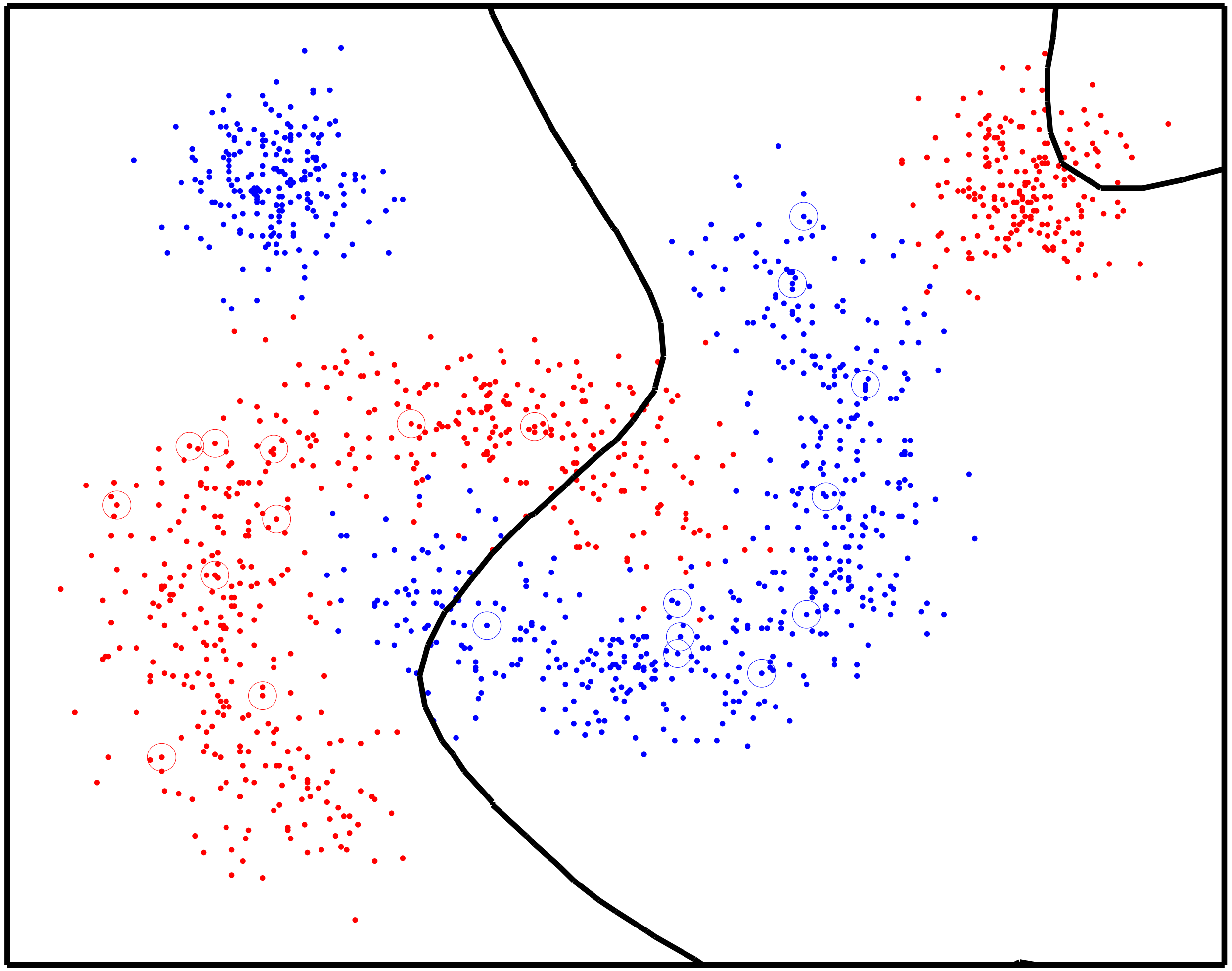
Active learning

- Select (at random?) an initially small training set. Ask for the labels, X_l
- Compute a classifier
- Select, given the classifier and X_u, more objects, ask the labels, extend X_l
- Repeat

Semi-Supervised Learning

- Compute classifiers from X_l combined with X_u

The use of unlabeled objects and active learning



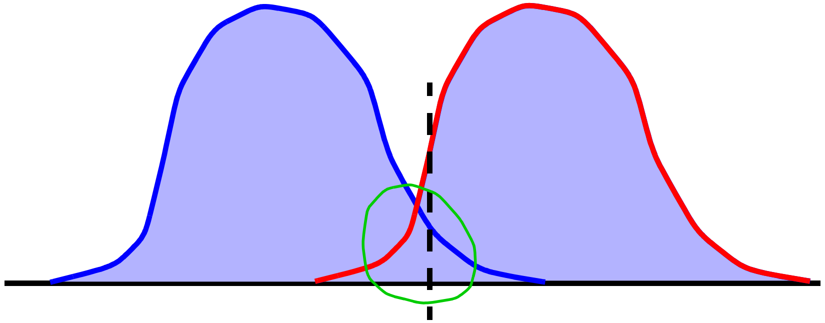
Assume labeling is expensive

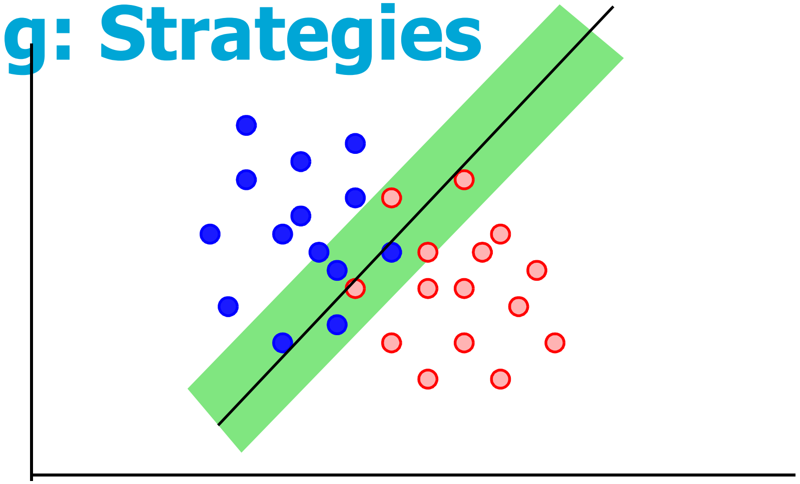
Can we make us of unlabeled objects for better classification?

Can we select a few to improve the classifier?
- close to the decision boundary?
- far away from the dec. boundary?
- at random?

How to make use of unlabeled data to construct classifiers?

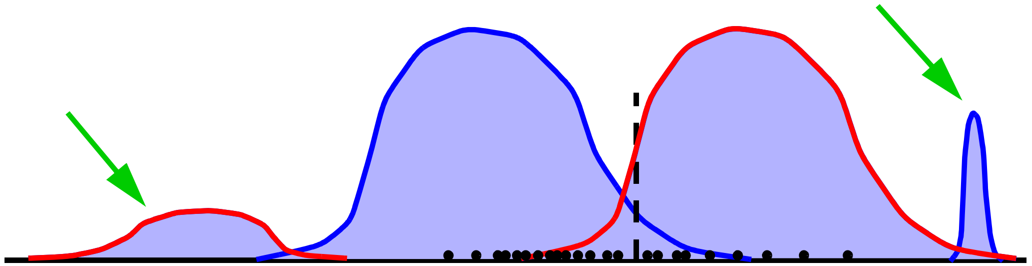
Active Learning: Strategies





Exploitation

Add unlabeled objects close the classifier to the training set.

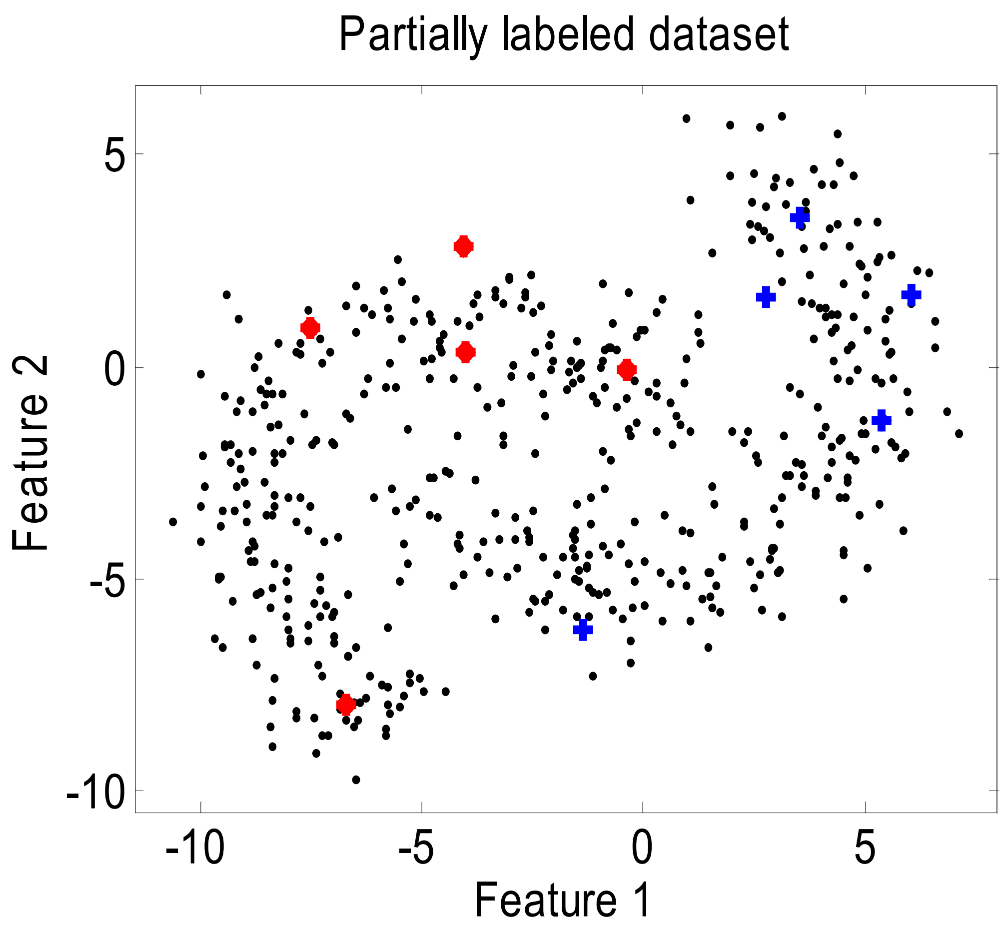


Exploration

Add remote unlabeled objects that represent unvisited clusters.

Is the set of objects representative for the problem?

Semi - Supervised Learning



Can better classifiers be designed by using labeled and unlabeled objects simultaneously?

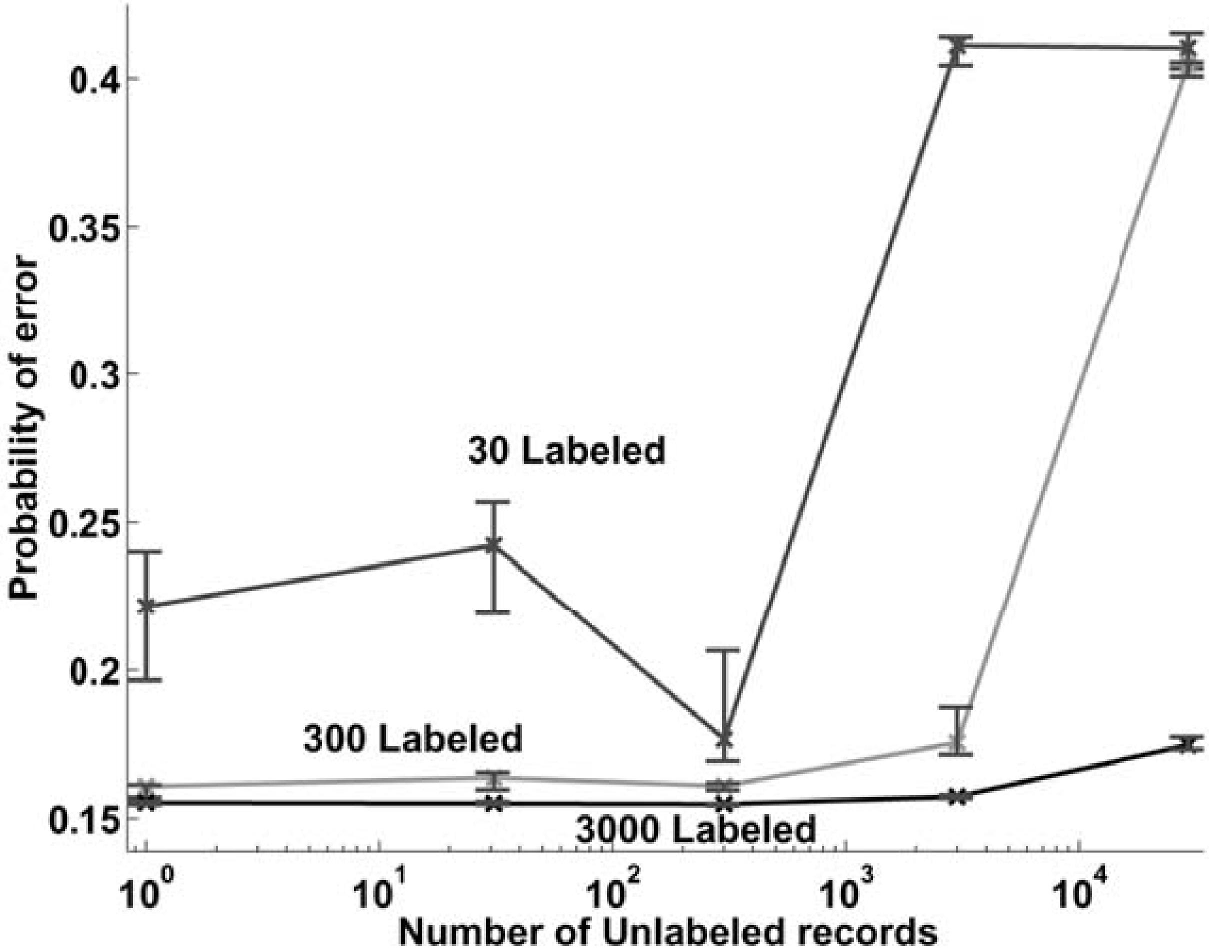
Two possible approaches:

- Combine supervised and unsupervised models
- Label propagation

Application: learn from the test set!

How to build a good semi-supervised classifier?

Semi Supervised Learning: Combining Supervised and Unsupervised Models

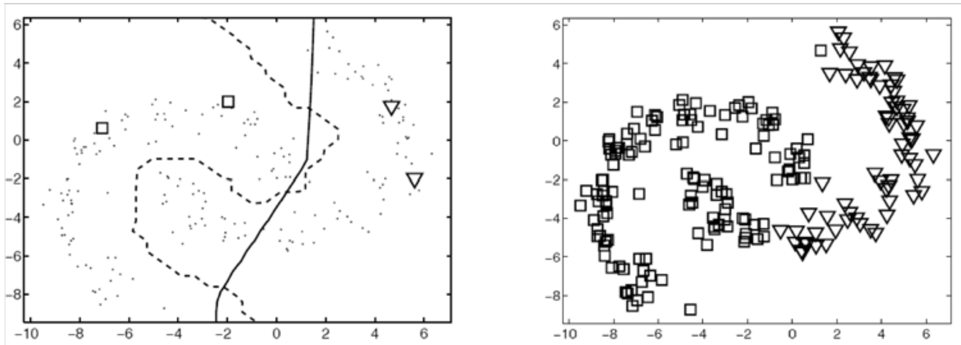


Another example of 'dipping'?

I. Cohen, F.G. Cozman, N. Sebe, M.C. Cirelo, T.S. Huang, Semisupervised learning of classifiers: theory, algorithms, and their application to human-computer interaction, IEEE-PAMI, 26, 1553-1566, 2004.

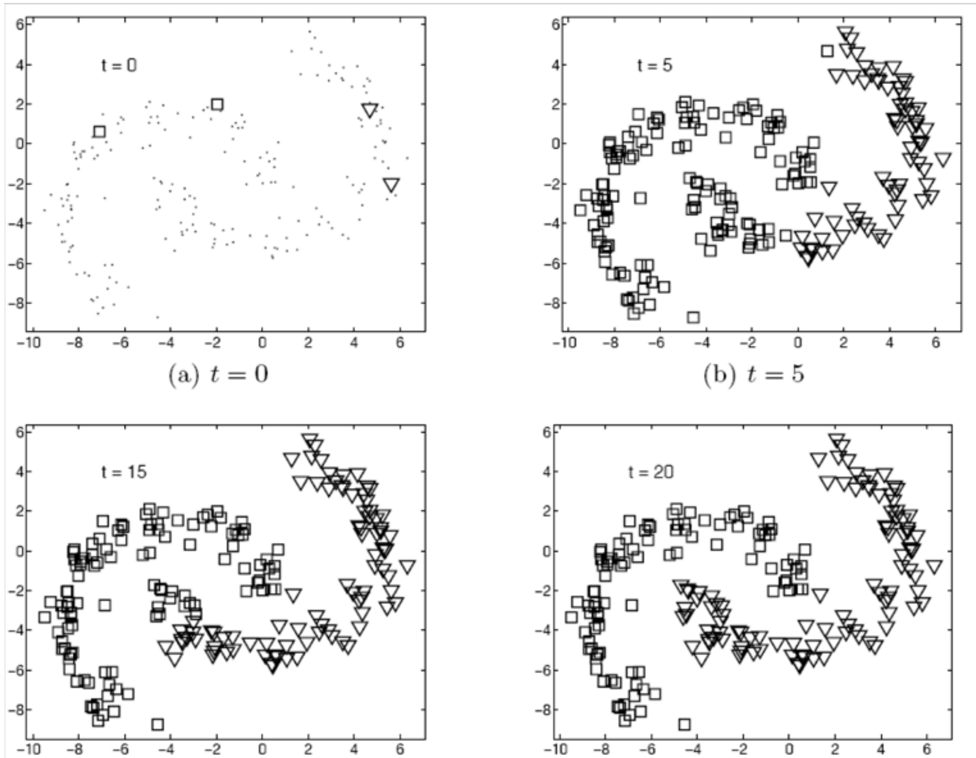
TU Delft

Learning from the Test Set



2 x 2 training samples and x 98 test samples

Soft label propagation

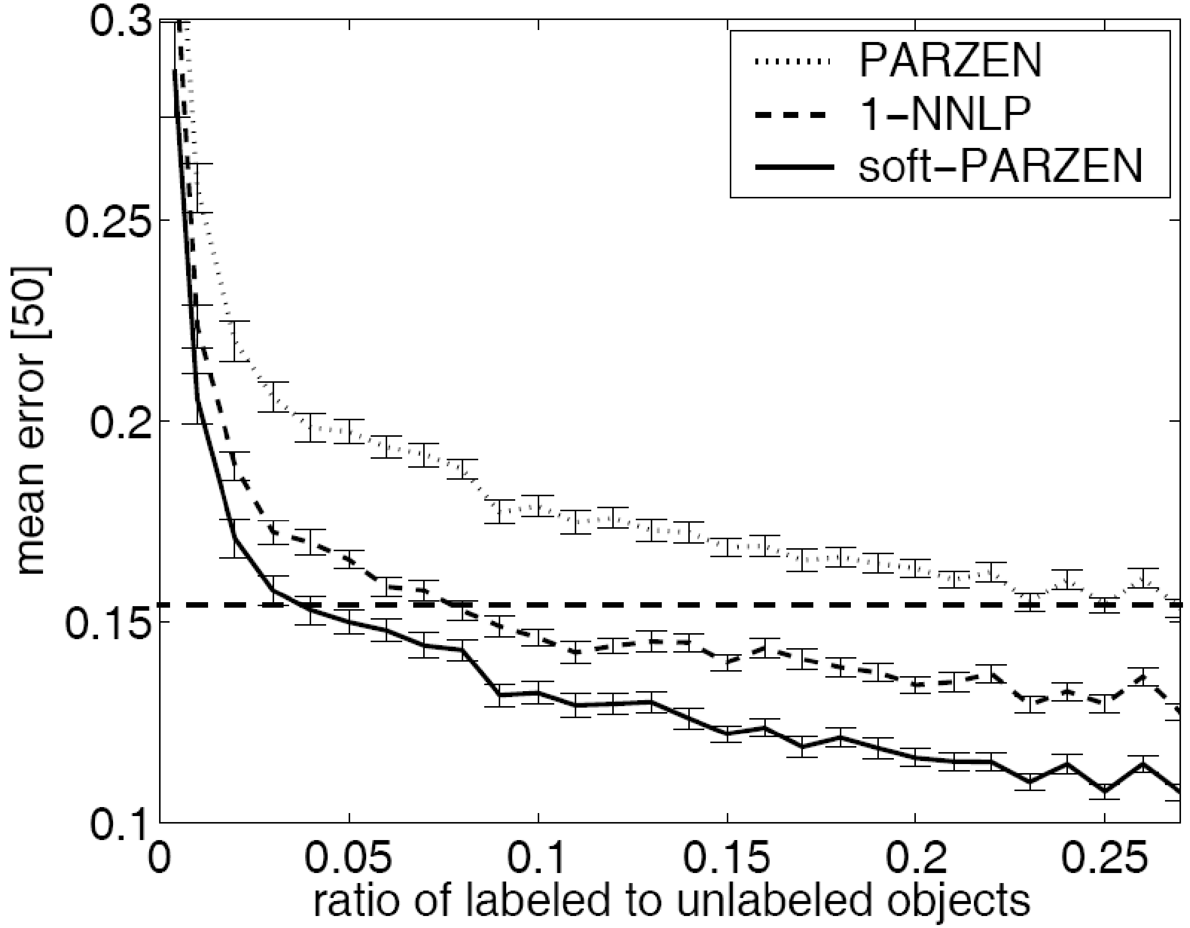


(a) $t = 0$ (b) $t = 5$ (c) $t = 15$ (d) $t = 20$

20 iterations of soft Parzen

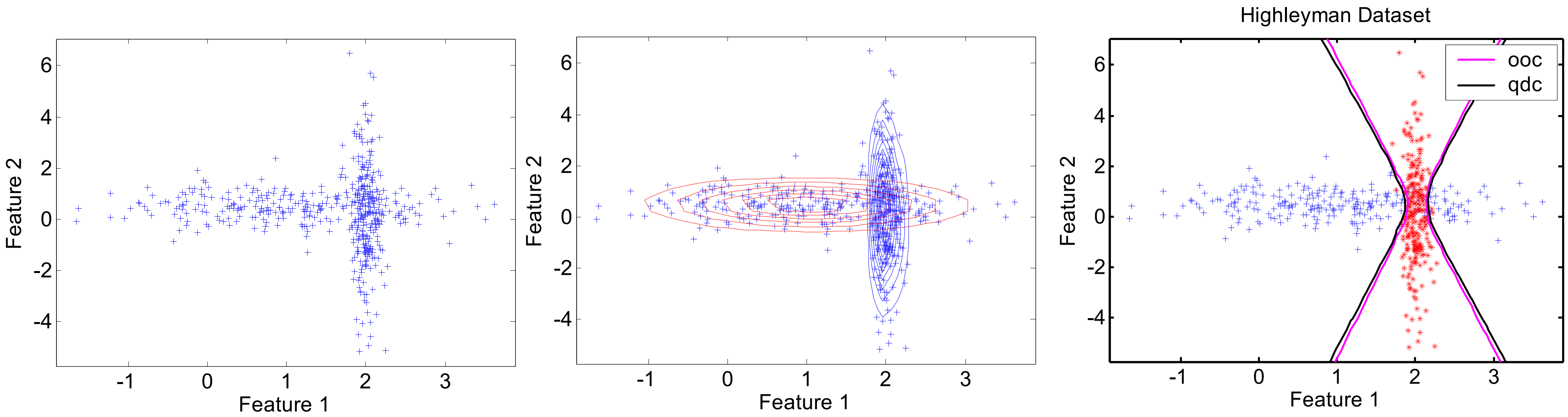
Piotr Juszczak, Learning to recognise, Ph.D. Thesis, Delft Univ. of Technology, 2006
see also Cores 2005.

Semi-Supervised Learning by Soft Parzen

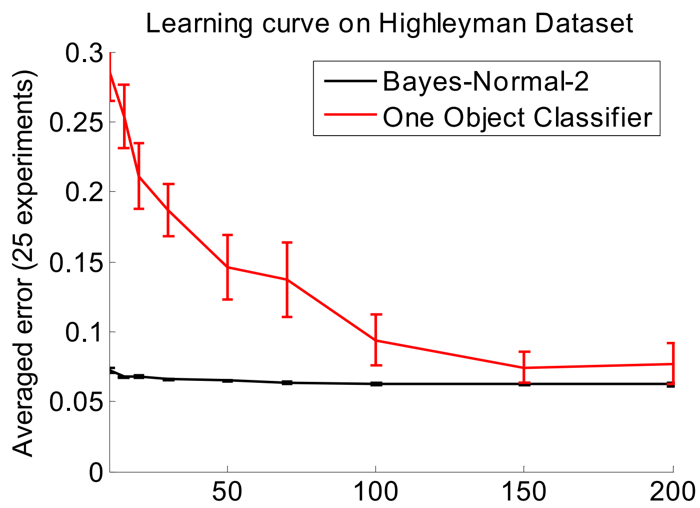


(b) satellite (500, 36, 6)

The One-Object Classifier (OOC)



1. Cluster the dataset into two clusters.
2. Select a most 'typical' object in one of the clusters.
3. Ask for its label.
4. Label the clusters accordingly.
5. Compute the classifier.



TU Delft

Conclusions

Pattern recognition research is solving representation problems