Representation Related Problems in Pattern Recognition

London, 6 October 2006

Robert P.W. Duin

Delft University of Technology

The Netherlands
Model Driven $\leftrightarrow$ Data Driven

Model of a house

Are these houses?

Observations of houses

Feature Space Representation

Classifier

Houses

Non-Houses
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.
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?
Classifier typology

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

Fisher Bayes Normal

Nearest Mean

Bayes Normal

Decision Tree

Neural Network

Can we create a library of problems corresponding to the library of classifiers?
Bad Features $\rightarrow$ More Features $\rightarrow$ Complexity Problem

The real world, first problem: peaking
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

Mean classification error (10 exp.)

Sample size

Feature size
Overpeaking

Feature curves for 16 x 16 NIST 3-8

Mean classification error (10 exp.)

Sample size

Feature size
Small Sample Size

Classification problem R30:

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

- feature 1 $\mathcal{N}_A(0, 1), \mathcal{N}_B(3, 1)$
- feature 2 $\mathcal{N}_A(0, 40), \mathcal{N}_B(3, 40)$
- feature 3-30 $\mathcal{N}_A(0, 1), \mathcal{N}_B(0, 1)$

Averaged error over 50 experiments

- Nearest Mean
- Fisher LD

True Error

Bayes Error

Feature Size

Training Set Size
Dimension Resonance and Dipping

Averaged error over 50 experiments

True Error

0.5
0.4
0.3
0.2
0.1
0

Pseudo Fisher LD

Fisher LD

Nearest Mean

feature size

Training Set Size

10^1
10^2

'dimension resonance'

'dipping' (Marco Loog)
Support Vector Machine for Small Sample Sizes

Averaged error over 50 experiments

- True Error
- Nearest Mean
- Pseudo Fisher LD
- SVM
- Fisher LD

Training Set Size

Error

Averaged error over 50 experiments

0

10^1 10^2

5/10/06  R.P.W. Duin
Pixel Representation: Samples Instead of Features

Features
- Shape
- Moments
- Fourier descriptors
- Faces
- Morphology

Class A

Class B

Feature Space

$16 \times 16$ Pixels

$R_{256}$

Pixels are more general, initially complete representation
Large datasets available $\rightarrow$ 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.
The Connectivity Problem in the Pixel Representation

Spatial connectivity is lost

Training set

Reshuffle Pixels

Reshuffling pixels will not change the classification

Feature space

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 $\leftrightarrow$ Object invariants
Domain based classification

How to construct domain based classifiers?

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

- Outlier dependent
- Distances instead of densities
Define dissimilarity measure $d_{ij}$ between raw data of objects $i$ and $j$

Given labeled training set $T$

Unlabeled object $x$ to be classified

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

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

Pekalska, The Dissimilarity Representation for Pattern Recognition, World Scientific, 2005
Example: Deformable Templates

Examples of deformed templates

Matching new objects $x$ to various templates $y$

$$\text{class}(x) = \text{class}(\arg\min_y (D(x, y)))$$

Dissimilarity measure appears to be non-metric

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

object 78

14.9

7.8

4.1

object 425

object 419

Bunke’s Chicken Dataset

\[ D(A,C) > D(A,B) + D(B,C) \]

Single Linkage Clustering

The Fisher Criterion

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

\[ J(A,C) = 0; \quad J(A,B) = \text{large}; \quad J(C,B) = \text{small} \neq J(A,B) \]

5/10/06
R.P.W. Duin
Class Representation Problems

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

unknown priors
skewed problems
ill sampled problems
label uncertainty
population drift
**ROC, AUC**

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

2. AUC optimizing classifiers may find ‘good’ directions in case of highly 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?
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 $X_u$. 

Unlabeled Training Set

Dataset Density
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 use of unlabeled objects for better classification?

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

Partially labeled dataset

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’?

Learning from the Test Set

2 x 2 training samples and x 98 test samples

Soft label propagation

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.
Conclusions

Pattern recognition research is solving representation problems