RWTH

Statistical Approaches to Image Object Recognition

- ATHENS Intensive Course on Computer Vision - March 16th, 2000.

Dipl.-Inform. Jörg Dahmen
Chair of Computer Science VI
RWTH Aachen - University of Technology
Germany
dahmen@informatik.rwth-aachen.de
http://www.informatik.rwth-aachen.de/I6/i6.html

Lecture Overview

- Introduction to Pattern Recognition
- Databases & 'State-of-the-Art'
- Statistical 'Standard-Approach'
  - Gaussian Mixture Densities in Bayesian Context
- Feature Reduction (KLT, LDA)
- Extension: Invariant Object Recognition
  - Global invariances
  - Local invariances
- Results
- Conclusion

What is pattern recognition?

Given:

- a set of pairs (x_n, k_n)
- usually splitted into training and testing set

Task:

- given observation x_n, find corresponding class k_n

General formulation of a classification problem:

$$x \in \mathbb{R}^D, \; r : \mathbb{R}^D \longrightarrow \{1, ..., K\}$$
$$x \longmapsto r(x)$$

with 'classification function' r.

- $r(x_n) \neq k_n$ is called a misclassification
- Evaluation of classifiers via error rate

Crucial: choice of r.

Well known examples for classifiers:

- Nearest Neighbour Classifier
- Support Vector Machines
- Artificial Neural Nets
- Decision Trees
- Polynomial Classifiers
- ...

This lecture:

- Gaussian Mixture Densities in Bayesian Context.
- Comparison of different classification methods via standard corpora.

Databases 1: USPS

US Postal Service Database (USPS):

- well known standard corpus for classifier evaluation
- isolated, handwritten digits (from US Zip-Codes)
- graylevel images of size 16×16 Pixel
- 7291 training samples, 2007 test samples
- known to be a 'hard' task, human error rate 2.5%
- best result published: 3.0% (Schoelkopf'98, SVM)

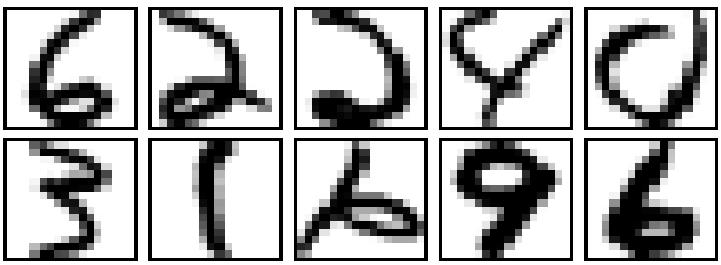

Figure 1: USPS example images

Databases 2: IRMA

- Real-world radiographs from the RWTH Aachen IRMA-Project (Image Retrieval in Medical Applications)
- classes: (1) abdomen, (2) limbs, (3) breast, (4) skull, (5) chest, (6) spine
- 1617 images available (up to 2500×2500 pixels)
- no separation between training and testing data
- no results available from competing groups

Figure 2: IRMA Example images, top-left to bottom-right: abdomen, limbs, breast, skull, chest, spine.

State-of-the-Art: USPS

Method	Error Rate [%]
Human Performance (Simard'93)	2.5
Decision Tree C4.5 (Cortes'95)	16.2
Two-Layer Neural Net (Schoelkopf'95)	5.9
5-Layer Neural Net (LeNet1, LeCun'90)	5.1
Best 5-Layer NN (Simard98)	4.2
Support Vectors (Schoelkopf'97)	4.0
Invariant Support Vectors (Schoelkopf'98)	3.0
Tangent Distance* (Simard'93)	2.6

* 2.500 machine printed digits were added to the training set

Coming up: the statistical standard approach

Bayes Rule

We model $r(x)$ via Bayesian decision rule, i.e.

$$x \longmapsto r(x) = \underset{k}{\text{argmax}} \{p(k)p(x|k)\}$$

with

- prior probabilities $p(k)$
- class-conditional probabilities $p(x|k)$

Note: Bayes rule optimal, if $p(k), p(x|k)$ are known.

Reality: distributions are unknown, i.e.

- choose models for the unknown distributions
- estimate their parameters (training data)

Here:

- $p(k)$: relative frequencies, i.e. $p(k) = \frac{N_k}{N}$
- $p(x|k)$: Gaussian Mixture Densities

Gaussian Mixture Densities

Modelling $p(x|k)$ via Gaussian Mixtures yields

$$p(x|k) = \sum_{i=1}^{I_k} c_{ki} \cdot \mathcal{N}(x|\mu_{ki}, \Sigma_{ki})$$

where

- $\mathcal{N}(x|\mu_{ki}, \Sigma_{ki}) = \frac{1}{\sqrt{(2\pi)^D det(\Sigma_{ki})}} \exp\left[-\frac{1}{2}(x-\mu_{ki})^T \Sigma_{ki}^{-1}(x-\mu_{ki})\right]$
- class index k, density index i
- weighted sum of normal distributions with parameter set $\lambda_{ki} = \{c_{ki}, \mu_{ki}, \Sigma_{ki}\}$

Note:
Gaussian mixture densities can approximate *any* density function with arbitrary precision, but

- parameter estimation?
- number of model parameters?

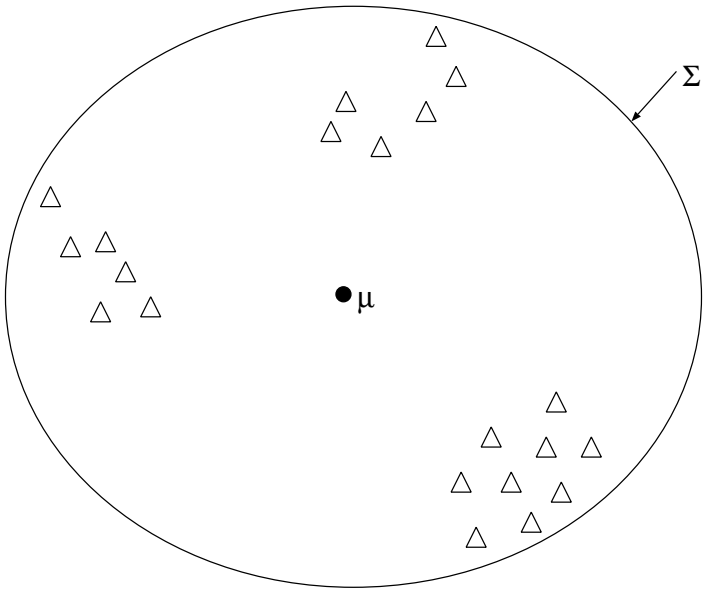

Figure 3: Single Density Model

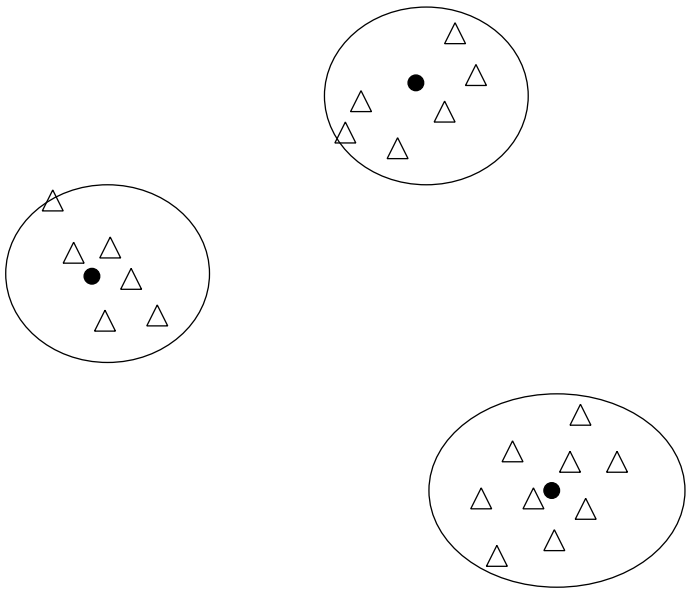

Figure 4: Mixture Density Model

Parameter Estimation I

Problem:

- **Density estimation given a '*hidden variable*'**
- **to which density i does observation x_n belong?**

Solution:
EM-algorithm (Dempster et al., 1977) yields the following reestimation formulae:

$$
\begin{aligned}
p(i|x_n,k,\lambda_{ki}) &= \frac{c_{ki}\cdot\mathcal{N}(x_n|\mu_{ki},\Sigma_{ki})}{\sum_{i'}c_{ki'}\cdot\mathcal{N}(x_n|\mu_{ki'},\Sigma_{ki'})} \\
\gamma_{ki}(n) &= \frac{p(i|x_n,k,\lambda_{ki})}{\sum_{n'}p(i|x_{n'},k,\lambda_{ki})} \\
\bar{c}_{ki} &= \frac{1}{N_k}\sum_{n=1}^{N_k}p(i|x_n,k,\lambda_{ki}) \\
\bar{\mu}_{ki} &= \sum_{n=1}^{N_k}\gamma_{ki}(n)\cdot x_n \\
\bar{\Sigma}_i &= \sum_{n=1}^{N_k}\gamma_{ki}(n)\cdot[x_n-\bar{\mu}_{ki}][x_n-\bar{\mu}_{ki}]^T
\end{aligned}
$$

Parameter Estimation II

Two problems still remain:

a) initial values for $\lambda_{ki} = \{c_{ki}, \mu_{ki}, \Sigma_{ki}\}$?

b) number of parameters?

Solutions:

a) perform '*successive splitting*':

- train a single density $\mathcal{N}(x|\mu_k, \Sigma_k)$ (easy, as there is no hidden variable)
- split μ_k by adding distortion vector ϵ, yielding a mixture with 2 densities $\mathcal{N}(x|\mu_{k1}, \Sigma_k)$ and $\mathcal{N}(x|\mu_{k2}, \Sigma_k)$
- perform parameter reestimation
- repeat splitting until 'model is OK'

Kernel Densities

other ways to model $p(x|k)$:

- kernel densities
  - each reference defines a density, i.e. $p_{KD}(x|k) = \frac{1}{N_k} \sum_{n=1}^{N_k} \mathcal{N}(x|x_{nk}, \Sigma_{x_{nk}})$ with
    - mean vector x_{nk}
    - estimated covariance matrix $\Sigma_{x_{nk}}$
    - $c_{x_{nk}} = \frac{1}{N_k}$
- nearest neighbour
  assign x to the class that the 'nearest' references image belongs to.
- nearest neighbour can be interpreted as a 'maximum-approximation' of kernel densities.

Parameter Reduction

Problem:

- 16×16 pixels reference images
  $\rightarrow$ covariance matrices $\Sigma_{ki} \in \mathbb{R}^{65.536}$(!)
- number of training samples is limited, thus, parameter reduction is essential

Possible approaches include:

- class-specific variance pooling: estimate only a Σ_k for each class k, i.e. $\Sigma_{ki} = \Sigma_k \; \forall i = 1, ..., I_k$
- global variance pooling: estimate only a single Σ, i.e. $\Sigma_{ki} = \Sigma \; \forall i = 1, ..., I_k$ and $\forall k = 1, ..., K$
- estimate only a diagonal Σ, i.e. a variance vector
- reduce feature space dimensionality

Important:
suitable feature reduction

Feature Reduction I

Important: Goal of feature reduction?

a) reduce number of model parameters

b) minimize 'reconstruction error'
(data compression, KLT)

c) maximize class separability
(pattern recognition, LDA)

General approach:

- given: observations $x_n \in \mathbb{R}^D$
- find suitable function $\varphi : \mathbb{R}^D \longmapsto \mathbb{R}^d$
  - feature reduction:
    $x_n \in \mathbb{R}^D \longmapsto \varphi(x_n) = y_n \in \mathbb{R}^d$
  - reconstruction:
    $y_n \in \mathbb{R}^d \longmapsto \varphi^{-1}(y_n) = \hat{x}_n \in \mathbb{R}^D$
- reconstruction error: $||x_n - \hat{x}_n||^2$

How do we find φ ?

Karhunen-Loève-Transformation
(‘*principal components analysis*’)

Method:

- compute $\Sigma = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu) \cdot (x_n - \mu)^T$
- compute eigenvectors Φ_i , -values δ_i of Σ, sort in decreasing order (with respect to δ_i)
- construct an $d \times D$ transformation matrix φ_{KLT} by interpreting the first d eigenvectors as a matrix
- feature reduction: $y_n = x_n^T * \varphi_{KLT}$
- reconstruction: $\hat{x}_n = \sum_{i=1}^{d} y_{n,i} \cdot \Phi_i$

Properties/ Notes:

- minimizes reconstruction error
- decorrelates features (diagonal Σ)
- divide eigenvector components by $\sqrt{\delta_i}$
  $\rightarrow$ Whitening, $\Sigma = I$
- often used for pattern recognition, although no class information is considered

Feature Reduction III - LDA

Linear-Discriminant-Analysis (LDA)
optimal linear projection (class separability)

Method:

- **compute Within-/ Between-Class-Scatter**
  - $\mathbf{\Sigma_w} = \frac{1}{N} \sum_{k=1}^{K} \sum_{n=1}^{N_k} (x_{nk} - \mu_k) \cdot (x_{nk} - \mu_k)^T$
  - $\mathbf{\Sigma_b} = \frac{1}{N} \sum_{k=1}^{K} N_k \cdot (\mu_k - \mu) \cdot (\mu_k - \mu)^T$
- **compute eigenvectors Φ_i , -values δ_i of $\mathbf{\Sigma_w^{-1}\Sigma_b}$, sort in decreasing order (with respect to δ_i)**
- **construct an $d \times D$ transformation matrix φ_{LDA} by interpreting the first d eigenvectors as a matrix**
- **feature reduction: $y_{nk} = x_{nk}^T * \varphi_{LDA}$**

Properties:

- **maximizes class separability**
- **might be numerically instable**
- **robust LDA-variant: '*Subspace-Method*'**

The Subspace Method (SSM)

Idea:
if $S_W = I$, the LDA can be computed as follows

- compute prototypes $\mu_k - \mu,\ k = 1, ..., K$
- compute orthonormal basis for the subspace spanned (max. K-1 dimensions)
- compute projection of feature vectors into that subspace

Now: 'force' $S_W = I$ ($\rightarrow$ whitening transform)

Notes:

- Subspace Method exactly computes an LDA
- computation numerically more stable (avoids matrix inversion)
- yields better results in the presence of numerical problems (cp. results)

PCA vs. LDA/ Subspace

adidas, aufkleben von Schukat S. 116

Feature Reduction IV

Note:
feature reduction *always* **means a** *loss of information*.

Problem:

- **what if K is small?**
- **USPS: $K = 10$ implies an LDA reduction from $x \in \mathbb{R}^{256} \rightarrow y \in \mathbb{R}^{9}$.**

Solution:

- **create 'pseudoclasses':**
  - **train a mixture density for each class k**
  - **interpret each density as a pseudoclass**
- **USPS: 4 pseudoclasses/ class, yielding $y \in \mathbb{R}^{39}$**

First Summary

What has happened until now?

(1) use Bayes Rule for classification

(2) use mixture densities to model $p(x|k)$

(3) use EM-algorithm for parameter estimation

(4) reduction of model parameters, in the following:

- **perform feature reduction (subspace method)**
- **estimate only a diagonal, globally pooled Σ**

Drawback of statistical classifiers:
despite (4), a large amount of training data is required, which is not always available

Solution?

Virtual Data Creation

Idea: Create Virtual Training Data

Method:
chose a 'suitable' transformation T (with parameter α) and create virtual data by applying T to the training data.

Effects:

- **we gain additional training data, leading to a more robust parameter estimation**
- **local invariance with respect to T**

US Postal Service Task:

- **chose ± 1 pixel shifts**
- **yields $7.291 \cdot 9 = 65.619$ training samples**

Extension:
Apply this approach to the testing data, too.
$\rightarrow$ Virtual-Test-Sample Method (VTS)

Use T to create virtual test samples
$x_{\alpha_j} = T(x, \alpha_j),\ j = 1, ..., J$

Assuming that the transformations considered are mutually exclusive yields:

$$\begin{aligned} p(x|k) &= \sum_{j=1}^{J} p(\alpha_j) \cdot p(x|\alpha_j, k) \\ &= \frac{1}{J} \sum_{j=1}^{J} p(x_{\alpha_j}|k) \end{aligned}$$

Notes:

- we assume that the prior probabilities for all transformations are equal
- the term $\frac{1}{J}$ is independent of k and may be neglected
- the result looks like Kittler's 'sum rule' for classifier combination, but ...

RWTH Classifier Combination: Sum Rule

Kittler's approach:

- train J classifiers
- let each of them compute $p(x|k,j),\ j = 1, ..., J$
- compute $p(x|k) = \sum_{j=1}^{J} p(x|k,j)$

Comments:

- decision looks like VTS
- to justify the sum-rule, Kittler has to assume $p(k) \sim p(k|x)$ ($\rightarrow$ useless features!)
- VTS: we only assume that the transformations considered are mutually exclusive
- VTS: no need to train different classifiers (computational complexity!)

First results

Table 1: USPS-Results without/ with Feature reduction (39 Subspace Features)

Method:	Error without SSM [%]				Error with SSM [%]			
	1-1	1-9	9-1	9-9	1-1	1-9	9-1	9-9
SD	19.5	18.3	22.3	21.5	12.8	12.4	13.1	11.7
MD	7.2	6.6	6.4	6.0	6.7	5.9	4.5	3.4
KD	6.5	5.5	5.9	5.1	6.3	5.3	4.2	3.4
NN	6.8	5.9	6.2	5.3	7.0	5.9	4.9	3.6

Note:

- 4 pseudoclasses per USPS class yielded 39 subspace features.
- global variance pooling, diagonal Σ

Comments to result overview

- feature reduction is effectice: 5.1% to 3.4%
- virtual training data on MD: 6.7% to 4.5%
- virtual test sample method: 4.5% to 3.4%

Table 2: LDA vs. Subspace method on USPS

Method:	Error [%]	
	1-1	9-1
Mixture Densities, Subspace	6.7	4.5
Mixture Densities, LDA	10.9	4.5

Notes:

- numerical problems on 1-1 data $\rightarrow$ bad LDA result
- virtual training data eliminates numerical problems, LDA and subspace yield identical results

Comparison of USPS-Results

Method	Error Rate [%]
Human Performance (Simard'93)	2.5
Decision Tree C4.5 (Cortes'95)	16.2
Two-Layer Neural Net (Schoelkopf'95)	5.9
5-Layer Neural Net (LeNet1, LeCun'90)	5.1
Support Vectors (Schoelkopf'97)	4.0
Invariant Support Vectors (Schoelkopf'98)	3.0
Tangent Distance* (Simard'93)	2.6
i6: MD, VTS	3.4

3.4% is not bad, but... can we do better?

Possible solution:

focus on *invariant* object recognition!

Approaches to invariant IOR

Approaches to invariant image object recognition:

a) extract invariant features

- amplitude spectrum of Fourier-Transform (shift-invariant)
- Fourier-Mellin Transformation (RST-invariant)
- invariant moments (RST-invariant)
- histogram based features (R(S)T-invariant)
- ...

b) adaptive classification

- globally invariant distance measures (Simard's tangent distance)
- locally invariant distance measures (image distortion model)

We will concentrate on b).

Tangent Distance I

Starting point:

- reasonably small transformations of an object do not affect class-membership
- given a transformation T_1 with parameter α_1 (e.g. the scaling factor), the set of all transformed images $M_x = \{T_1(x, \alpha_1) : \alpha_1 \in \mathbb{R}\} \subset \mathbb{R}^{I \times J}$ spans a one-dimensional curve in input space
- if there are L such transformations, M_x is a manifold of at most dimension L

$\rightarrow$ defining $d(x, \mu)$ to be the distance between the according manifolds yields invariance with respect to $T_1, ..., T_L$

Problem:
computation of this distance is a very hard problem, as the manifolds needed have no analytic expression in general

Idea: Tangent approximation of the manifolds!

Tangent Distance II

Small transformations of the object can be approximated via tangent subspace $\hat{M}_x$:

$\hat{M}_x = \{x + \sum_{l=1}^{L} \alpha_l \cdot T_l(x) \ : \alpha \in \mathbb{R}^L\} \subset \mathbb{R}^{I \times J}$

- with tangent vectors $T_l(x)$
- transformed versions of x are obtained by adding a linear combination of these to x

Setting $\bar{x} = x + \sum_{l=1}^{L} \alpha_l \cdot T_l(x)$ we now define

- $D_{ss}(x, \mu) = \min_{\alpha}\{\|\bar{x} - \mu\|^2\}$ (single-sided TD)
- $D_{ds}(x, \mu) = \min_{\alpha,\beta}\{\|\bar{x} - \bar{\mu}\|^2\}$ (double-sided TD)

Remaining questions:

- computation of tangent vectors?
- efficient computation of tangent distance?

Tangent Distance III

Given the tangents, D_{ss} is computed as follows:

1) compute tangent vectors for observation x
2) compute orthonormal basis for tangent subspace (SVD)
3) compute projection $\mu(x)$ of reference image into tangent subspace of x
4) compute $D_{ss}(x, \mu) = \|\mu(x) - \mu\|^2$

Steps 1) and 2) can be computed in advance
(for the references)
Steps 3) and 4) can be done efficiently by computing

$$D_{ss}(x, \mu) = \|x - \mu\|^2 - \sum_{l=1}^{L} \frac{[(x-\mu)^t \cdot T_l(x)]^2}{\|T_l(x)\|^2}$$

Note:

- step 2) can't be done in advance for D_{ds}, as the subspace is spanned by tangent vectors of x and μ
- computation of D_{ds} is *very* expensive

Tangent vector computation (finite differences):

1) **horizontal shift:**
$$T_1(x(i,j)) = x(i+1,j) - x(i,j)$$

2) **vertical shift:**
$$T_2(x(i,j)) = x(i,j+1) - x(i,j)$$

3) **rotation:**
$$T_3(x(i,j)) = j \cdot T_1(x(i,j)) - i \cdot T_2(x(i,j))$$

4) **scale:**
$$T_4(x(i,j)) = i \cdot T_1(x(i,j)) + j \cdot T_2(x(i,j))$$

5) **axis deformation:**
$$T_5(x(i,j)) = j \cdot T_1(x(i,j)) + i \cdot T_2(x(i,j))$$

6) **diagonal deformation:**
$$T_6(x(i,j)) = i \cdot T_1(x(i,j)) - j \cdot T_2(x(i,j))$$

7) **line thickness:**
$$T_7(x(i,j)) = T_1(x(i,j))^2 + T_2(x(i,j))^2$$

Tangent Distance IV

Notes:

- using the tangent vectors described above, invariance to affine transformation (1-6) and to line-thickness is achieved

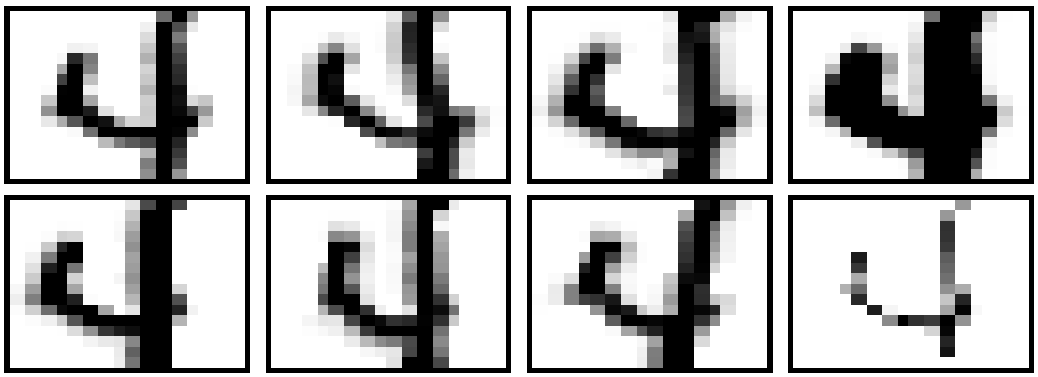

Figure 5: Example images generated via tangent approximation. Original image is at top-left.

Application of TD

Usage of tangent distance within our statistical classifier?

1) **replace Euclidean distance with tangent distance (is straightforward)**

2) **use tangent approximation for robust parameter estimation**

Robust parameter estimation:

- **we implicitly create a infinite amount of training samples via tangent approximation**
- **influence to parameters μ, Σ ?**

We will start by investigating on the means.

used symbols:

- N: number of training samples
- x_n: n-th training sample
- α: vector notation of the α_l
- $p(\alpha)$: distribution of the α_l
  note that by definition $\int p(\alpha)\ d\alpha = 1$
- T_n: matrix notation of the tangent vectors belonging to x_n
- Σ_α: covariance matrix of the α_l

$$
\begin{aligned}
\mu_T &= \frac{1}{N} \int p(\alpha) \cdot \sum_{n=1}^{N} x_n + T_n \cdot \alpha \; d\alpha \\
&= \frac{1}{N} \int p(\alpha) \sum_{n=1}^{N} x_n \; d\alpha + \frac{1}{N} \int p(\alpha) \sum_{n=1}^{N} T_n \cdot \alpha \; d\alpha \\
&= \frac{1}{N} \sum_{n=1}^{N} x_n \cdot \int p(\alpha) \; d\alpha + \frac{1}{N} \int p(\alpha) \sum_{n=1}^{N} T_n \cdot \alpha \; d\alpha \\
&= \mu + \frac{1}{N} \int p(\alpha) \sum_{n=1}^{N} T_n \cdot \alpha \; d\alpha
\end{aligned}
$$

Last term is of the the form:

- $\int p(\alpha) f(\alpha) \; d\alpha = E(f(\alpha))$
- as $E(\alpha) = 0$ and f linear, it follows $E(f(\alpha)) = 0$.
- Thus: $\mu_T = \mu$

$\rightarrow$ **Mean vectors do not change!**

Parameter Estimation with TV III

Estimation of Σ under consideration of tangent approximation:

$$
\begin{aligned}
\boldsymbol{\Sigma_T} &= \frac{1}{N} \int p(\alpha) \cdot \sum_{n=1}^{N} (x_n + T_n \cdot \alpha - \mu)(x_n + T_n \cdot \alpha - \mu)^T \; d\alpha \\
&= \frac{1}{N} \int p(\alpha) \cdot \sum_{n=1}^{N} \left[(x_n - \mu)(x_n - \mu)^T + (T_n \alpha)(T_n \alpha)^T + ... \right] d\alpha \\
&= \frac{1}{N} \left[N \cdot \Sigma + \int p(\alpha) \sum_{n=1}^{N} (T_n \alpha \alpha^T T_n^T) \; d\alpha \right] \\
&= \frac{1}{N} \left[N \cdot \Sigma + \sum_{n=1}^{N} T_n \left(\int p(\alpha) \alpha \alpha^T \; d\alpha \right) T_n^T \right] \\
\boldsymbol{\Sigma_T} &= \Sigma + \frac{1}{N} \sum_{n=1}^{N} T_n \Sigma_\alpha T_n^T
\end{aligned}
$$

In our experiments:

- $\Sigma_\alpha = I$
- no feature reduction

Table 3: USPS-Results, no feature reduction, with and without tangent-variance

Method:	Error rate [%]			
	1-1	1-9	9-1	9-9
Baseline SD	19.5	18.3	22.3	21.5
Baseline MD	8.0	6.6	6.4	6.0
SD, TV	19.8	17.4	21.5	-
MD, TV	6.4	4.8	4.5	4.3

Tangent variance: MD error from 6.0% to 4.3%.

replacing Euclidean distance by D_{ss}: 4.3% to 2.9%.
replacing Euclidean distance by D_{ds}: 4.3% to 2.7%.

Best result: invariant kernel densities: 2.4% error

Final USPS Overview

Method	Error Rate [%]
Human Performance (Simard'93)	2.5
Decision Tree C4.5 (Cortes'95)	16.2
Two-Layer Neural Net (Schoelkopf'95)	5.9
5-Layer Neural Net (LeNet1, LeCun'90)	5.1
Support Vectors (Schoelkopf'97)	4.0
Invariant Support Vectors (Schoelkopf'98)	3.0
Tangent Distance* (Simard'93)	2.6
i6: Mixture Densities	4.5
Mixture Densities, VTS	3.4
invariant Mixture Densities, VTS	2.7
invariant Kernel Densities, VTS	2.4

Image Distortion Model I

TD compensates small *global* transformations, but ...

Medical IRMA images:

- **images with and without pathologies**
- **varying scribor position**

$\rightarrow$ need for local distortions

Simple Image Distortion Model:

$$D_{dist}(x,y) = \sum_{i=1}^{M} \sum_{j=1}^{N} \min_{i',j' \in R_{i,j}} \left(x_{i,j} - y_{i',j'}\right)^2$$

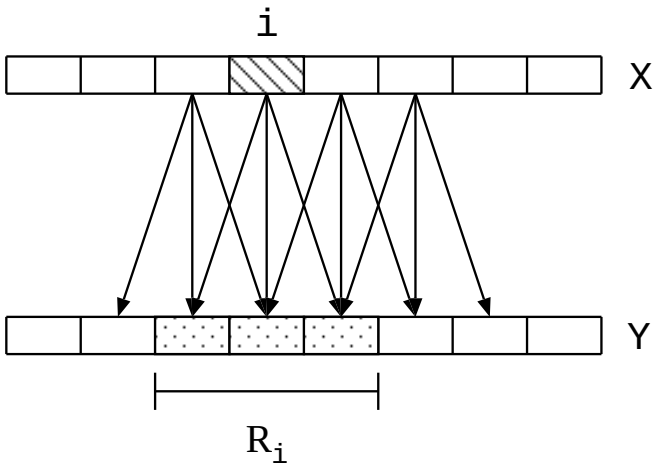

Figure 6: 1D example for the image distortion model

Image Distortion Model II

Restrict possible transformations by incorporating a cost function $c(i', j')$:

$$D_{dist}(x, y) = \sum_{i=1}^{M} \sum_{j=1}^{N} \min_{i',j' \in R_{i,j}} \left\{ (x_{i,j} - y_{i',j'})^2 + c(i', j') \right\}$$

Chosing $c(i, j)$:

- empirically (e.g. Euclidean distance of (i, j), (i', j'))
- learn $c(i, j)$ from training data

Experiments:

- images scaled down to 32×32 pixels
- empirical $c(i, j)$
- introduce tresholding, i.e. local pixel distance restricted by treshold S

IRMA Results

Table 4: L-1-O error rates without tresholding

Distance Measure	Error Rate [%]	
	NN	KD
Euclidean Distance	18.1	16.4
Tangent Distance	15.3	14.8
Distortion Model	16.5	14.7
Distorted Tangent Distance	14.2	12.5

Table 5: L-1-O error rates, tresholding (S=3500)

Distance Measure	Error Rate [%]	
	NN	KD
Euclidean Distance	14.8	14.2
Tangent Distance	13.7	12.9
Distortion Model	14.5	13.2
Distorted Tangent Distance	10.6	10.3

incorporate aspect-ratio: 10.3% to 8.2%

Analysis of results

Notes:

- IDM better than TD without tresholding!
- best region size: $R = (2r+1) \times (2r+1), r = 0.8$
- IDM significantly reduces TD error rate
- not surprisingly: choice of r is crucial

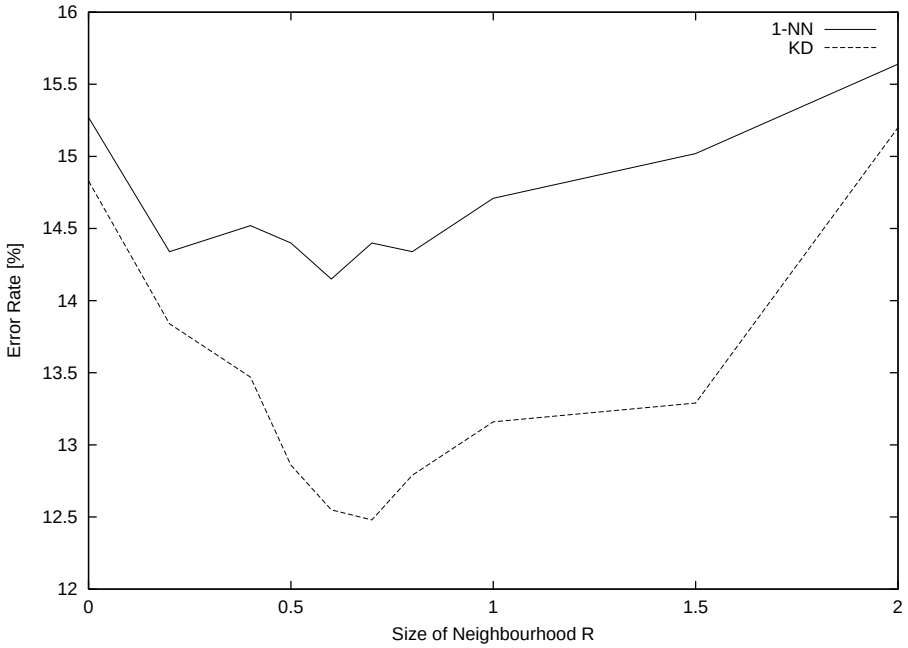

Figure 7: Error rates for distorted tangent distance with respect to size of neighbourhood R, without local tresholding

Conclusions

What we did:

- Classification approach:
  - Gaussian Mixtures in Bayes context
  - parameter estimation via EM
  - feature reduction (KLT, LDA)
- dealing with invariances
  - tangent distance
  - image distortion model
- parameter estimation with respect to tangents
- replacing Euclidean distance with invariant distance measures
  - excellent results on USPS
  - very good results on medical images (no competing results available)

Any questions left?

Related Literature I

More infomation:

- J. Dahmen, T. Theiner, D. Keysers, H. Ney, T. Lehmann, B. Wein, "Classification of Radiographs in the 'Image Retrieval in Medical Applications' System (IRMA)", 6th International RIAO Conference on Content-Based Multimedia Information Access, Paris, France, April 2000, in press.
- J. Dahmen, K. Beulen, M. Güld, H. Ney, "A Mixture Density Based Approach to Object Recognition for Image Retrieval", Proceedings of the 6th International RIAO Conference on Content-Based Multimedia Information Access, Paris, France, April 2000, in press.
- J. Dahmen, J. Hektor, R. Perrey, "Automatic Classification of Red Blood Cells using Gaussian Mixture Densities", Bildverarbeitung fr die Medizin 2000, München, 2000 (in press).

Related Literature II

- J. Dahmen, R. Schlüter, H. Ney, "Discriminative Training of Gaussian Mixtures for Image Object Recognition", in W. Förstner, J. Buhmann, A. Faber, P. Faber (eds.): Procs. 21. Symposium German Association for Pattern Recognition, pp. 205-212, Bonn, September 1999.
- B. Schölkopf, *Support Vector Learning* , Oldenbourg Verlag, Munich, 1997.
- B. Schölkopf, P. Simard, A. Smola, V. Vapnik, "Prior Knowledge in Support Vector Kernels", M. Jordan, M. Kearns, S. Solla (eds.): *Advances in Neural Information Processing Systems 10*, MIT Press, pp. 640-646, 1998.
- J. Kittler, "On Combining Classifiers", IEEE Trans. Pattern Analysis & Machine Intelligence, Vol. 20, No. 3, pp. 226-239, March 1998.
- P. Simard, Y. Le Cun, J. Denker, "Efficient Pattern Recognition Using a New Transformation Distance," S.J. Hanson, J.D. Cowan, C.L. Giles (eds.): *Advances in Neural Information Processing Systems 5*, Morgan Kaufmann, San Mateo CA, pp. 50-58, 1993.