Statistical Approaches
to
Image Object Recognition

- ATHENS Intensive Course on Computer Vision -
March 22nd, 2001

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

Lecture Overview

- Introduction to pattern recognition
- Databases & ‘State-of-the-Art’
- Statistical ‘standard-approach’
  - Gaussian mixture densities
  - Bayesian decision rule
  - Feature reduction (KLT, LDA)
- Extension: Invariant image object recognition
  - virtual data creation
  - invariant distance measures
- Theoretical considerations
- Results
- Conclusion

What is pattern recognition?

Given:

- set of pairs (x_n, k_n) ('supervised learning')
  alternative: $x_{nk},\ n = 1,..,N,\ k = 1,..,K$
- usually split into training and testing set

Task: Given a 'new' x, find corresponding k

General formulation of a classification problem:

$$x \in \mathrm{I\!R}^D,\ r : \mathrm{I\!R}^D \longrightarrow \{1, ..., K\}$$
$$x \longmapsto r(x)$$

with 'decision function' r.

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

$$e[\%] = \frac{\#\text{misclassifications}}{\#\text{trials performed}} \cdot 100$$

Crucial: choice of r.

Well known examples for classifiers:

- Nearest Neighbour Classifier (NN)
- Support Vector Machine (SVM)
- Relevance Vector Machine (RVM)
- Artificial Neural Nets (ANN)
- Decision Trees
- Polynomial Classifiers
- ...

Applications:

- Robot vision
- Handwriting recognition
- Quality control
- Image retrieval
- ...

This lecture:

- Gaussian mixture densities in Bayesian context
- Comparison of different classification methods via standard corpora
- Data: handwritten digits, X-ray images

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 pixels
- $7,291$ training samples, $2,007$ test samples
- known to be a 'hard' task, human error rate 2.5%
- best result published so far: 3.0%
(Schölkopf'98, invariant support vector machine)

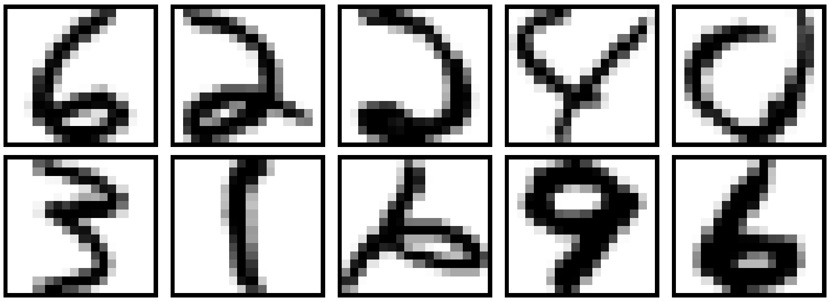


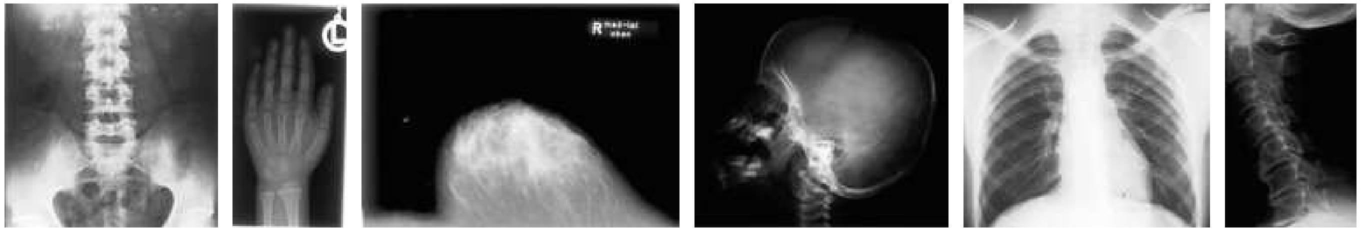
Figure 1: USPS example images

Feature analysis:
appearance based (pixels are features)

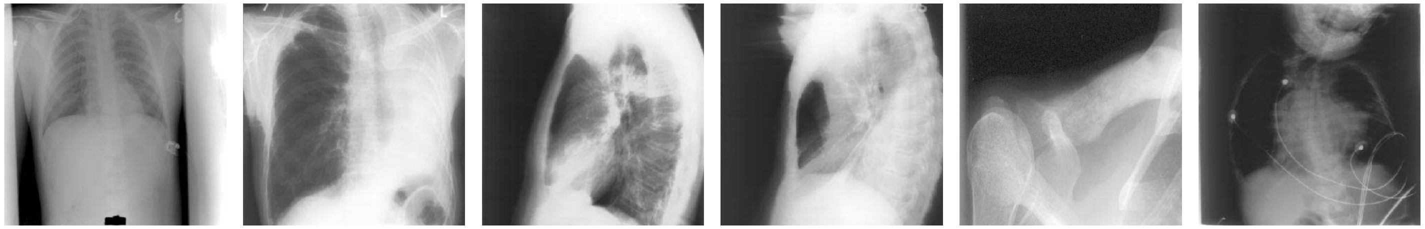
Databases 2: IRMA

IRMA (Image Retrieval in Medical Applications, RWTH Aachen project)

- real-world radiographs from clinical routine, few results
- 1,617 images of varying size (200×200 to 2,000×2,000 → 32×...)
- leaving-one-out approach (L-1-O)
- six classes (abdomen, limbs, breast, skull, chest, spine)



Example images from the IRMA database



Variation within class 'chest'

State-of-the-Art: USPS

Method	Group	Error Rate [%]
Human Performance	Simard'93, AT&T	2.5
Decision Tree C4.5	Cortes'95, AT&T	16.2
Relevance Vector Machine	Tipping'00, Microsoft	5.1
Artificial Neural Net	LeCun'90, AT&T	4.2
Support Vector Machine	Schölkopf'97-98, GMD	4.0
+ invariance		3.0
Tangent Distance*	Simard'93, AT&T	2.5

* 2.418 machine printed digits were added to the training set

Coming up: the statistical standard approach

The Bayesian Decision Rule

Classification via Bayesian decision rule, i.e.

$$x \longmapsto r(x) = \operatorname*{argmax}_{k} \{p(k) \cdot p(x|k)\}$$

with

- observation $x \in \mathbb{R}^D$
- prior probabilities $p(k)$
- class-conditional probabilities $p(x|k)$

Bayes rule is optimal w.r.t. expected number of misclassifications, if $p(k), p(x|k)$ are known.

Reality: distributions are unknown, i.e.

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

Here:

- $p(k)$: relative frequencies (i.e. $p(k) = \frac{N_k}{N}$) or uniform distribution (digits)
- $p(x|k)$: Gaussian mixture densities (GMD)

Gaussian Mixture Densities

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

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

where

- $\mathcal{N}(x|\mu_{ki}, \Sigma_{ki}) =$
  $\frac{1}{\sqrt{\det(2\pi \cdot \Sigma_{ki})}} \exp\left[-\frac{1}{2}(x - \mu_{ki})^T \Sigma_{ki}^{-1} (x - \mu_{ki})\right]$
- 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?

Single vs. Mixture Densities

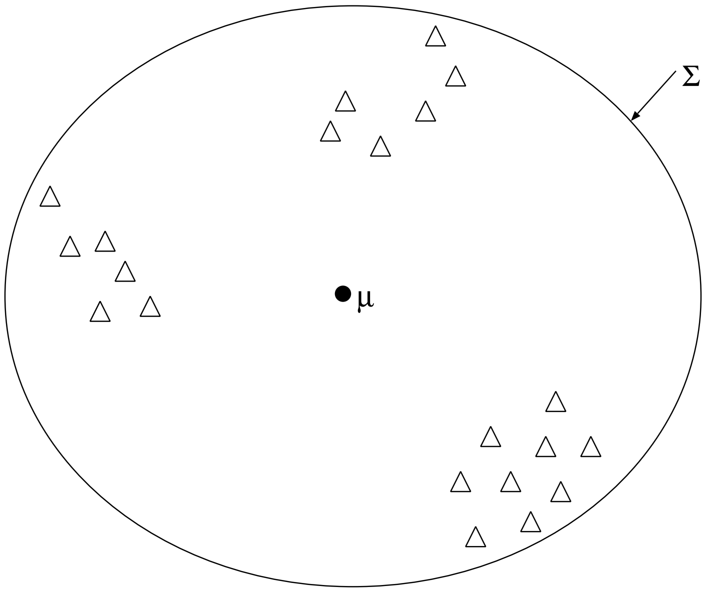


Figure 2: Single Density Model

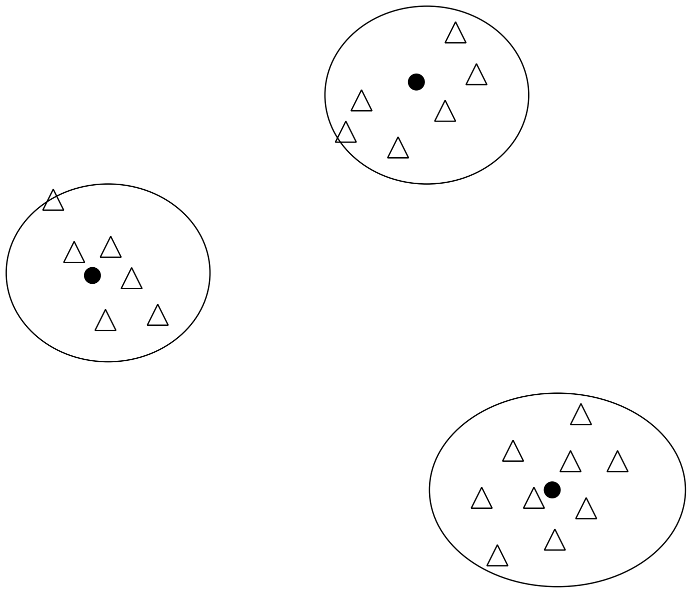


Figure 3: Mixture Density Model

Kernel Densities

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

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

Parameter Estimation I

Problem:

- **mixture 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:

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

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/ subtracting a 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' (ideally: evaluation on development test set)**

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

b) maximize class separability
   (pattern recognition, LDA)

General approach:

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

How do we find φ ?

Feature Reduction II - KLT

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** λ_i **of** Σ**, sort in decreasing order (with respect to** λ_i**)**
- **construct an** $d \times D$ **transformation matrix** φ_{KLT} **by interpreting the first** d **eigenvectors as a matrix**
- **feature reduction:** $y_n = \varphi_{KLT}^T * x_n$
- **reconstruction:** $\hat{x}_n = \varphi_{KLT} * y_n$

Properties/ Notes:

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

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

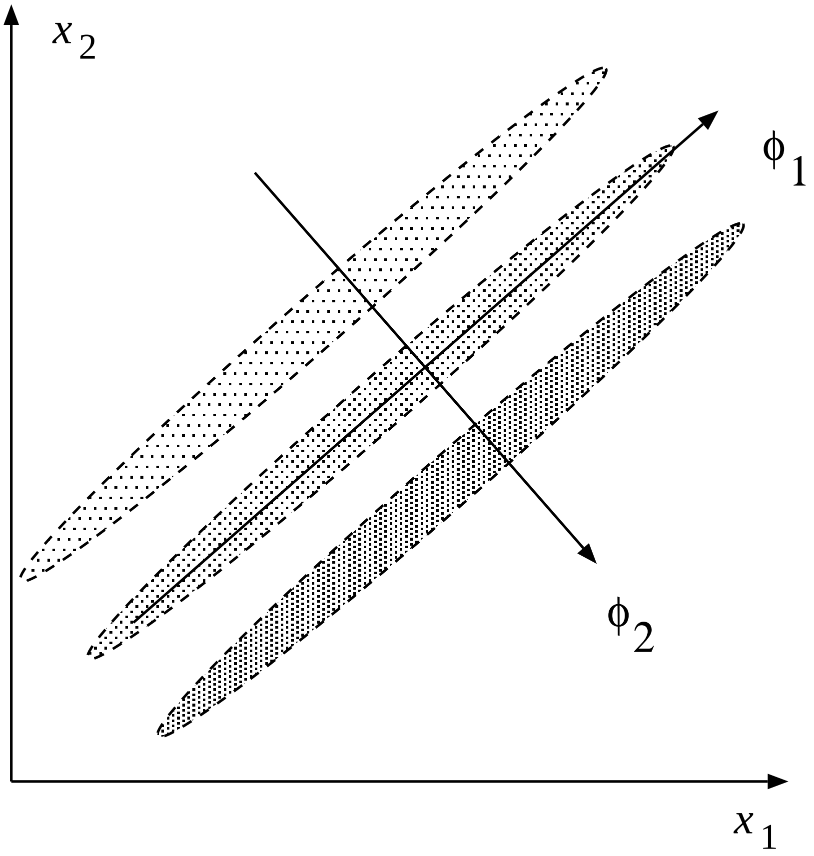
Method:

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

Properties:

- maximizes class separability
- yields a maximum of $(K-1)$ features
- might be numerically instable in case of few training examples

PCA vs. LDA/ Subspace



Feature Reduction IV

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

In other words, it can be shown that

$$\text{“}\boldsymbol{Error}(x \in \mathbb{R}^{D+1}) \leq \boldsymbol{Error}(x \in \mathbb{R}^{D})\text{”}$$

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:
   - optionally: perform feature reduction (via LDA)
   - 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

Typical drawback of statistical classifiers:

- insufficient amount of training data
- idea: create virtual training data

Method:
choose 'suitable' transformation t with parameter α, create virtual data by applying t to the training data

Effects:

- we gain additional training data, leading to more reliable parameter estimation
- local invariance with respect to t

US Postal Service Task:

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

Extension:
Apply this approach to the testing data, too.
(inspired by classifier combination schemes (CC))
$\rightarrow$ Virtual-Test-Sample Method (VTS)

Visualization: CC vs. VTS

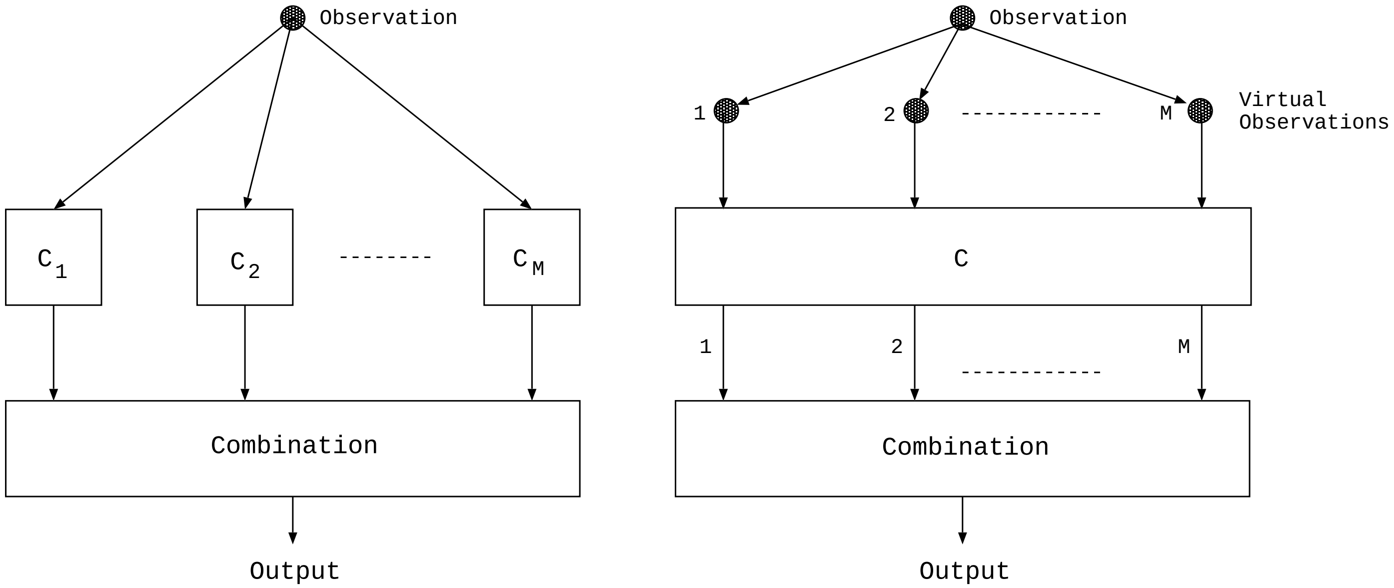


Figure 4: Classifier combination (left) vs. the virtual test sample method (right)

The VTS Method

Virtual Test Sample Method:

- create virtual test data $x(\alpha)$ from original observation x using $t(x, \alpha),\ \alpha \in \mathcal{M}$
- combined decision:

$$
\begin{aligned}
x \longmapsto r(x) \quad &= \quad \operatorname*{argmax}_k \{p(k|x)\} \\
&= \quad \operatorname*{argmax}_k \left\{ \sum_{\alpha \in \mathcal{M}} p(k, \alpha | x) \right\} \\
&= \quad \operatorname*{argmax}_k \left\{ \sum_{\alpha \in \mathcal{M}} p(\alpha|x) \cdot p(k|x, \alpha) \right\} \\
&\overset{model}{=} \quad \operatorname*{argmax}_k \left\{ \sum_{\alpha \in \mathcal{M}} p(\alpha) \cdot p(k|x(\alpha)) \right\}
\end{aligned}
$$

Only assumption:
transformations considered are mutually exclusive

Advantages of VTS:

- incorporate prior knowledge via $p(\alpha)$
  here: $p(\alpha) = const$
- easy to implement
- adds invariance properties
- experiments: ± 1 pixel shifts

Classifier Combination: Sum Rule

Idea of sum rule: classifier combination

- train M classifiers $C_1, ..., C_M$
- let each of them compute $p_m(k|x),\ m = 1, ..., M$
- compute $p(k|x) = \sum_{m=1}^{M} p_m(k|x)$

Assuming $p(k) \sim p_m(k|x)$, Kittler managed to justify the sum rule in 1998.
This implies: extracted feature are useless!

Sum Rule decision looks like VTS, but...

- VTS is straightforward to derive
- VTS only assumes that the transformations considered are mutually exclusive
- no need to train different classifiers for VTS (computational complexity!)
- but: combination of CC and VTS is possible (and it even works!)

First results

Table 1: USPS-Results without/ with feature reduction (39 LDA features)

Method:	Error without LDA [%]				Error with LDA [%]			
	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	8.0	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 LDA features
- global variance pooling, diagonal Σ

RWTH

Comparison of USPS Results

Method	Group	Error Rate [%]
1-Nearest Neighbour	"Baseline"	5.6
Decision Tree C4.5	AT&T: Cortes'95	16.2
Relevance Vector Machine	Microsoft: Tipping'00	5.1
Artificial Neural Net	AT&T: LeCun'90	4.2
Invariant Support Vectors	GMD: Schölkopf'98	3.0
Tangent Distance*, 1-NN	AT&T: Simard'93	*2.5
MD, LDA, (VTS)	RWTH-i6	4.5 (3.4)

* 2,418 machine printed digits were added to the training set

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

Possible solution: focus on *invariant* object recognition!

Approaches to invariant image object recognition:

a) image normalization (RST-invariant)
b) 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)
   - ...
c) adaptive classification
   - globally invariant distance measures (tangent distance)
   - locally invariant distance measures (image distortion model)

We will concentrate on c) in the following.

Tangent Distance I (Simard '93)

<u>Idea:</u> An image x is not a point in feature space!
Instead, transformations of x span a manifold M_x.

$$D(x, y) = \min_{\bar{x} \in M_x, \bar{y} \in M_y} \{d(\bar{x}, \bar{y}\}$$

with Euclidean distance $d(x, y)$.

Solution of this is a hard problem, therefore

- approximate M_x using a linear subspace, spanned by so-called 'tangent vectors' x_l
- x_l can be easily computed
  (i.e. affine transformations via image gradients)
- local transformations (small α_l):
  $x(\alpha) = x + \sum_{l=1}^{L} \alpha_l \cdot x_l$
- results in linear optimization problem,
  which is easy to solve

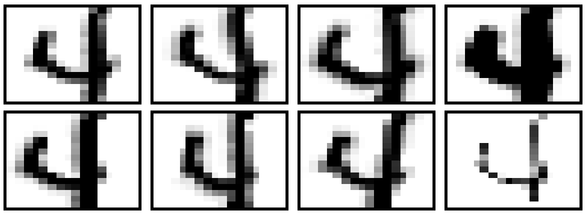


Figure 5: Examples of local tangent approximations.

Tangent Distance II

We define

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

Remaining questions:

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

Tangent Distance III

Conceptionally, D_{ss} is computed as follows:

1) compute tangent vectors for observation x
2) compute orthonormal basis for tangent subspace ($\rightarrow$ singular value decomposition)
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

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

Note:

- step 2) can't be done in advance for D_{ds}, as the subspace depends on μ AND x.
- thus, computation of D_{ds} is *very* expensive.
- strictly speaking, TD is not a metric: e.g. it violates the triangle inequation.

Computation of Tangent Vectors

Tangent vector computation (finite differences):

1) **horizontal shift:**
   $$x_1(x(i,j)) = x(i+1,j) - x(i,j)$$

2) **vertical shift:**
   $$x_2(x(i,j)) = x(i,j+1) - x(i,j)$$

3) **rotation:**
   $$x_3(x(i,j)) = j \cdot x_1(x(i,j)) - i \cdot x_2(x(i,j))$$

4) **scale:**
   $$x_4(x(i,j)) = i \cdot x_1(x(i,j)) + j \cdot x_2(x(i,j))$$

5) **axis deformation:**
   $$x_5(x(i,j)) = j \cdot x_1(x(i,j)) + i \cdot x_2(x(i,j))$$

6) **diagonal deformation:**
   $$x_6(x(i,j)) = i \cdot x_1(x(i,j)) - j \cdot x_2(x(i,j))$$

7) **line thickness:**
   $$x_7(x(i,j)) = x_1(x(i,j))^2 + x_2(x(i,j))^2$$

Note:
“Good” calculation of x_1, x_2 is important:
e.g preprocessing via Sobel or similar operator

Visualization: Tangent Distance

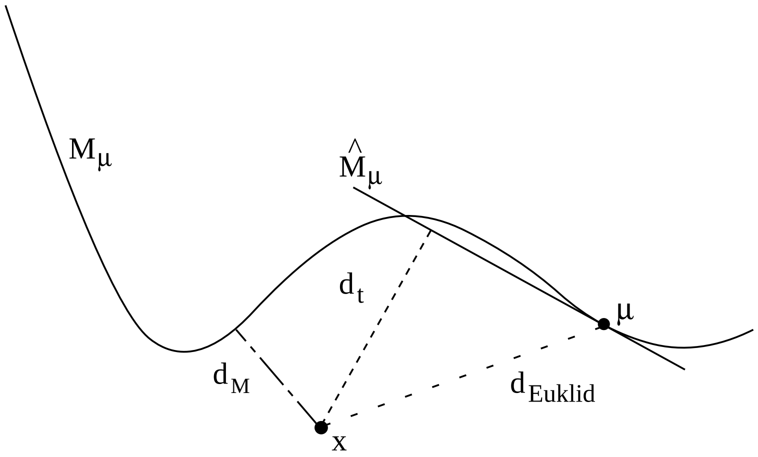


Figure 6: Single-sided tangent distance (TD).

Applications:

- **Simard '93: 1-NN classifier (cp. results)**
- **RWTH-i6:**
  a) use TD to 'replace' Mahalanobis distance in Gaussians,

  b) reliable parameter estimation using tangent approximations.

Interpretation of TD I

What happens if we use TD in a Gaussian?

Approach: integrate over the unknown $\alpha \in \mathbb{R}^L$.

$$\begin{aligned} p(x|\mu,\Sigma) &= \int p(x,\alpha|\mu,\Sigma)\, d\alpha \\ &= \int p(\alpha|\mu,\Sigma)\cdot p(x|\mu,\alpha,\Sigma)\, d\alpha \\ &\overset{model}{=} \int p(\alpha)\cdot p(x|\mu,\alpha,\Sigma)\, d\alpha \end{aligned}$$

where

$$\begin{aligned} p(\alpha) &= \mathcal{N}(\alpha|0,\gamma^2 I) \\ p(x|\alpha,\mu,\Sigma) &= \mathcal{N}(x|\mu + \sum_{l=1}^{L} \alpha_l \mu_l, \Sigma) \end{aligned}$$

After a lengthy calculation:

$$p(x|\mu,\Sigma) = \mathcal{N}(x|\mu,\Sigma') \text{ with } \Sigma' = \Sigma + \sum_{l=1}^{L} \gamma^2 \mu_l \mu_l^T$$

Interestingly:
Computing SS-TD on the side of the references only modifies the covariance matrix!

Interpretation of TD II

What happens on the side of the observations?

Using similar considerations as above, we obtain:

$$p(x|\mu,\Sigma) = \left(\prod_{l=1}^{L}(1+\gamma^2 x_l^T\Sigma^{-1}x_l)^{-\frac{1}{2}}\right)\cdot \det(2\pi\Sigma)^{-\frac{1}{2}}\cdot$$
$$\exp\left[-\frac{1}{2}\left((\mu-x)^T\underbrace{(\Sigma^{-1}-\sum_{l=1}^{L}\frac{(x_l^T\Sigma^{-1})^T(x_l^T\Sigma^{-1})}{\frac{1}{\gamma^2}+x_l^T\Sigma^{-1}x_l})}_{(*)}(\mu-x)\right)\right]$$

Note:

- resulting distribution is not Gaussian, as matrix $(*)$ depends on x
- but: we can still compute its mean vector μ_T and the covariance matrix Σ_T by applying the moment method for parameter estimation.
  Here: equivalent to using the $x_n(\alpha)$ for parameter estimation

Interpretation of TD III

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

T_n is the matrix notation of the tangent vectors of reference image x_n.

Σ_T is called tangent covariance matrix.

Experiments with TD

In our experiments:

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

Table 2: USPS-Results, no feature reduction, with and without tangent-variance (TV)

Method:	Error rate [%]			
	1-1	1-9	9-1	9-9
Baseline MD	8.0	6.6	6.4	6.0
MD, TV	6.4	4.8	4.5	4.3

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

use D_{ss}: 4.3% to 2.9%.
use D_{ds}: 4.3% to 2.7%.

invariant kernel densities (+ classifier combination): 2.4% (2.2%) error

Final USPS Overview

Method	Group	Error Rate [%]
1-Nearest Neighbour	“Baseline”	5.6
Decision Tree C4.5	AT & T: Cortes’95	16.2
Relevance Vector Machine	Microsoft: Tipping’00	5.1
Artificial Neural Net	AT & T: LeCun’90	4.2
Invariant Support Vectors	GMD: Schölkopf’98	3.0
Tangent Distance*, 1-NN	AT & T: Simard’93	*2.5
MD, LDA, (VTS)	RWTH-i6	4.5 (3.4)
MD, tangents, VTS		2.7
KD, tangents, VTS, (CC)		2.4 (2.2)

* 2,418 machine printed digits were added to the training set

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 transformations

Simple "Image Distortion Model" (IDM):

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

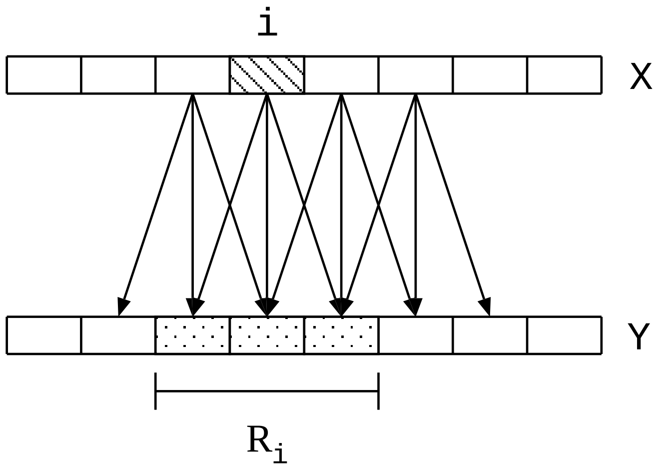


Figure 7: 1D example for the IDM

Image Distortion Model II

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

Choosing $c(i', j')$:

- learn $c(i', j')$ from training data
- here: Euclidean distance between (i, j), (i', j')

'Distorted Tangent Distance':

1) registration via optimal tangent approximation
2) compute IDM-distance

Motivation: Effects of

- tangent distance (global transformations) and
- image distortion model (local transformations)

should be "additive".

Experiments:
Use thresholding, i.e. local pixel distance restricted by threshold S:

$$d(x_{ij}, \mu_{ij}) := \min \left\{ \|x_{ij} - \mu_{ij}\|^2, S \right\}$$

Notes: IRMA Recognition

2 Problems:

a) Only few images available:
Leaving-One-Out (L-1-O) approach: one image is classified, the remaining are used as references.

b) Different sized images

- **look for best horizontal shift**
- **pixels with no counterpart are treated as background (grayvalue zero)**

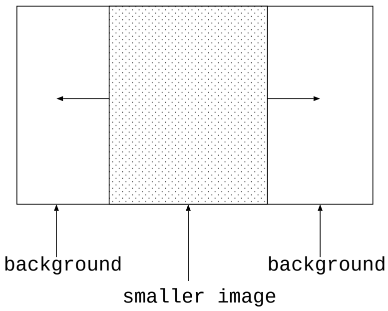


Figure 8: The IRMA matching step

Note: This is just a simple description of a complex model. More detail: *"An Automatic Approach to Invariant Radiograph Classification", Dahmen et al., BVM2001*

State of the Art & Results (IRMA)

Table 3: Obtained L-1-O results in comparison to reported results

Method		ER [%]
Cooccurrence Matrices (Theiner 2000)		29.0
Active Contours (Bredno 2000)		51.1
RWTH-i6:	1-NN & “Matching”	15.8
	+ Kernel Densities	14.0
	+ Thresholding (S=5000)	11.2
	+ Image Distortion Model	9.0
	+ Tangent Distance	8.2
	(+ Disregard background)	7.6

New test set of 332 images: 9.0% ER

Conclusions

What we did:

- Classification approach:
  - Gaussian Mixtures in Bayesian context
  - parameter estimation via EM
  - feature reduction (KLT, LDA)
- dealing with invariances
  - virtual data (training, VTS)
  - tangent distance
  - image distortion model
- parameter estimation with respect to tangents
- theoretical interpretation of tangent distance
- experimental results:
  - excellent on USPS
  - very good on medical images
    (few competing results available)

More information: see literature-slides

Outlook: Examples

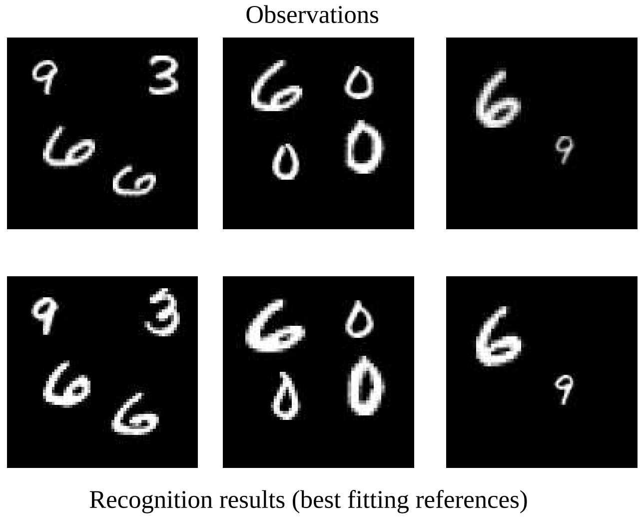


Figure 9: Example: multi object recognition.

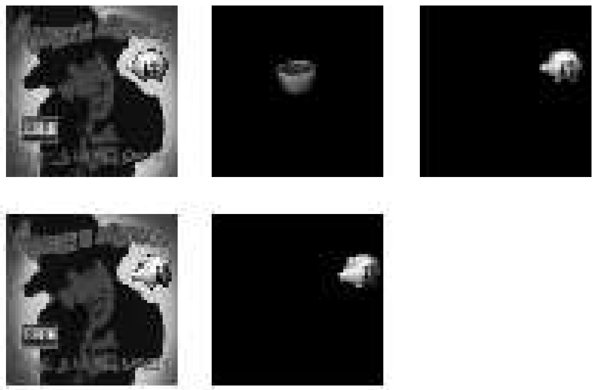


Figure 10: Object detection in a complex scene.

Related Literature

ALL RWTH-i6 PAPERS ARE AVAILABLE AT
http://www-i6.informatik.rwth-aachen.de/Colleagues/dahmen/joerg.html

- **B. Schölkopf,** *Support Vector Learning* **, Oldenbourg Verlag, Munich, 1997.**
- **B. Schölkopf, P. Simard, A. Smola, V. Vapnik,** *Prior Knowledge in Support Vector Kernels*, **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*, **Advances in Neural Information Processing Systems 5, Morgan Kaufmann, San Mateo CA, pp. 50-58, 1993.**
- **J. Dahmen, D. Keysers, H. Ney, M. Gueld,** *Statistical Image Object Recognition using Mixture Densities*, **accepted for publication in Journal of Mathematical Imaging and Vision, Kluwer Academic Publishers**
- **J. Dahmen, D. Keysers, M. Güld, H. Ney,** *Invariant Image Object Recognition using Mixture Densities*, **15th Int. Conf. on Pattern Recognition, Barcelona, Spain, Vol. 2, pp. 614-617, September 2000.**
- **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 Int. RIAO Conference on Content-Based Multimedia Information Access, Paris, France, pp. 551-566, April 2000.**
- **J. Dahmen, K. Beulen, M. Güld, H. Ney,** *A Mixture Density Based Approach to Object Recognition for Image Retrieval*, **6th Int. RIAO Conference on Content-Based Multimedia Information Access, Paris, France, pp. 1632-1647, April 2000.**