Learning Vector Quantization and Relevance Learning

Michael Biehl
Intelligent Systems Group
University of Groningen

Outline

Learning Vector Quantization

- introduction to prototype learning and LVQ
- distance based classification
- basic training prescription: LVQ1

Example: Intron/Exon classification

- based on tiling microarray data
- application of standard LVQ1 (fixed metric)

Adaptive metrics and relevance learning

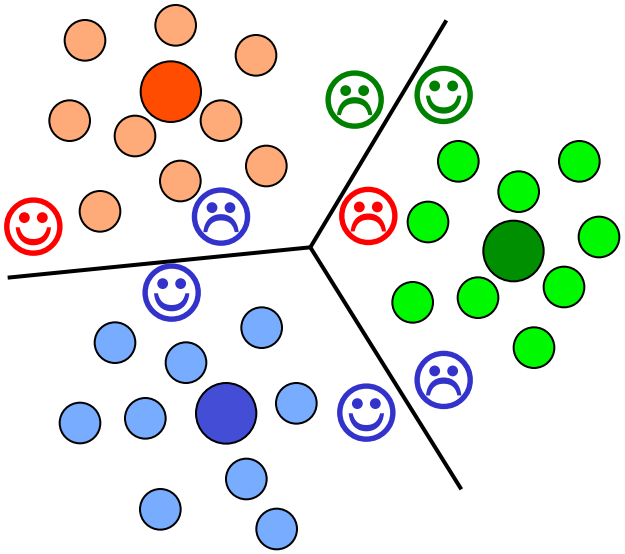
- weighted Euclidean distance, relevance learning
- feature weighting, feature selection

Learning Vector Quantization (LVQ)

- identification of **prototype vectors** from labelled example data
- distance based **classification** (e.g. Euclidean distance)

often: **heuristically** motivated variations of **competitive learning**

example: basic LVQ scheme **"LVQ1"** [Kohonen]



classification:

assignment of a vector ξ
to the class of the closest
prototype **w**

aim: **generalization ability**
classification of novel data
after learning from examples

Learning Vector Quantization (LVQ)

- identification of **prototype vectors** from labelled example data
- parameterization of distance based **classification** (e.g. Euclidean)

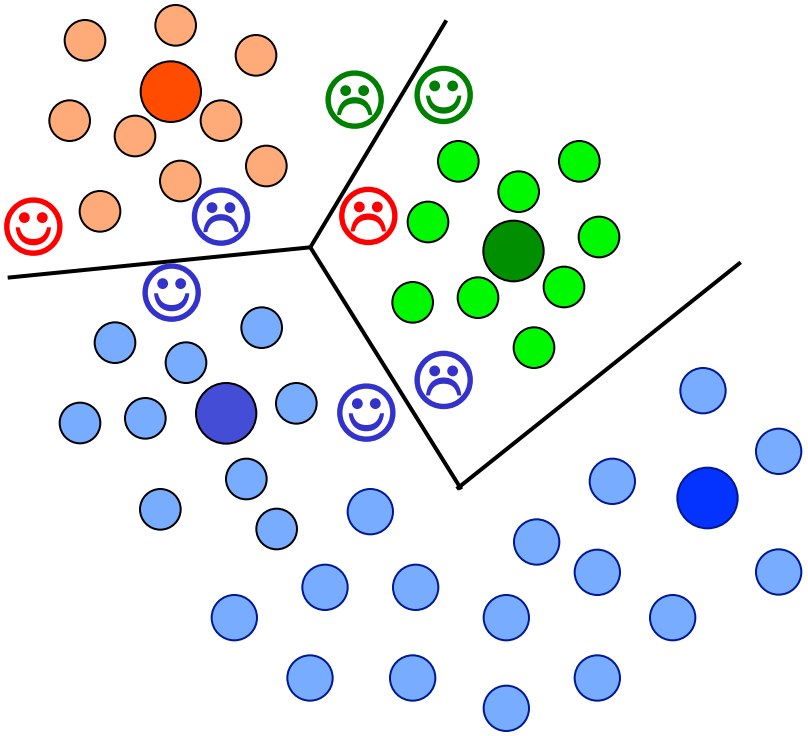
often: **heuristically** motivated variations of **competitive learning**

example: basic LVQ scheme **"LVQ1"** [Kohonen]

classification:

assignment of a vector ξ
to the class of the closest
prototype **w**

aim: **generalization ability**
classification of novel data
after learning from examples



piecewise linear decision boundaries

formally:

set of **prototypes** $w^1, w^2, \ldots, w^K \qquad w^k \in \mathbb{R}^N$

representing class $S^1, S^2, \ldots, S^K \qquad S^k \in \{1, 2, \ldots, C\}$

nearest prototype classifier

based on similarity/distance measure $d[w, \xi] \geq 0$

given feature vector ξ, determine the *winner* $w^{i^*} = \underset{j}{\operatorname{argmin}} \{ d[w^j, \xi] \}$

→ assign ξ to class S^{i^*}

examples: squared **Euclidean distance** $d[w, \xi] = \sum_{j=1}^{N} (w_j - \xi_j)^2$

Manhattan distance $d[w, \xi] = \sum_{j=1}^{N} |w_j - \xi_j|$

LVQ1 training:

randomized initial w^k, e.g. close to the class-conditional means

sequential presentation of labelled examples $\{\xi^t, \sigma^t\} \quad t = 1,2,\ldots,P,1,2,\ldots$

… the winner takes it all: $w^{i^*} = \underset{j}{\operatorname{argmin}} \{ d[w^j, \xi^t] \}$

$$\mathbf{w}^{i^*} \rightarrow \mathbf{w}^{i^*} + \eta_w \, \psi(S^{i^*}, \sigma^t) \left(\xi^t - \mathbf{w}^{i^*} \right) \qquad \psi(S,\sigma) = \begin{cases} +1 & \text{if } S = \sigma \\ -1 & \text{else} \end{cases}$$

η_w : learning rate, step size of update (repulsion/attraction)

many variants/modifications:

- learning rate schedule $\eta_w(t)$
- update of more than one prototype
- more general update functions $\Psi(S,\sigma,w^{i^*},\ldots)$

LVQ algorithms ...

- intuitive
- fast, easy to implement
- natural tool for multi-class problems

- frequently applied in a variety of practical problems

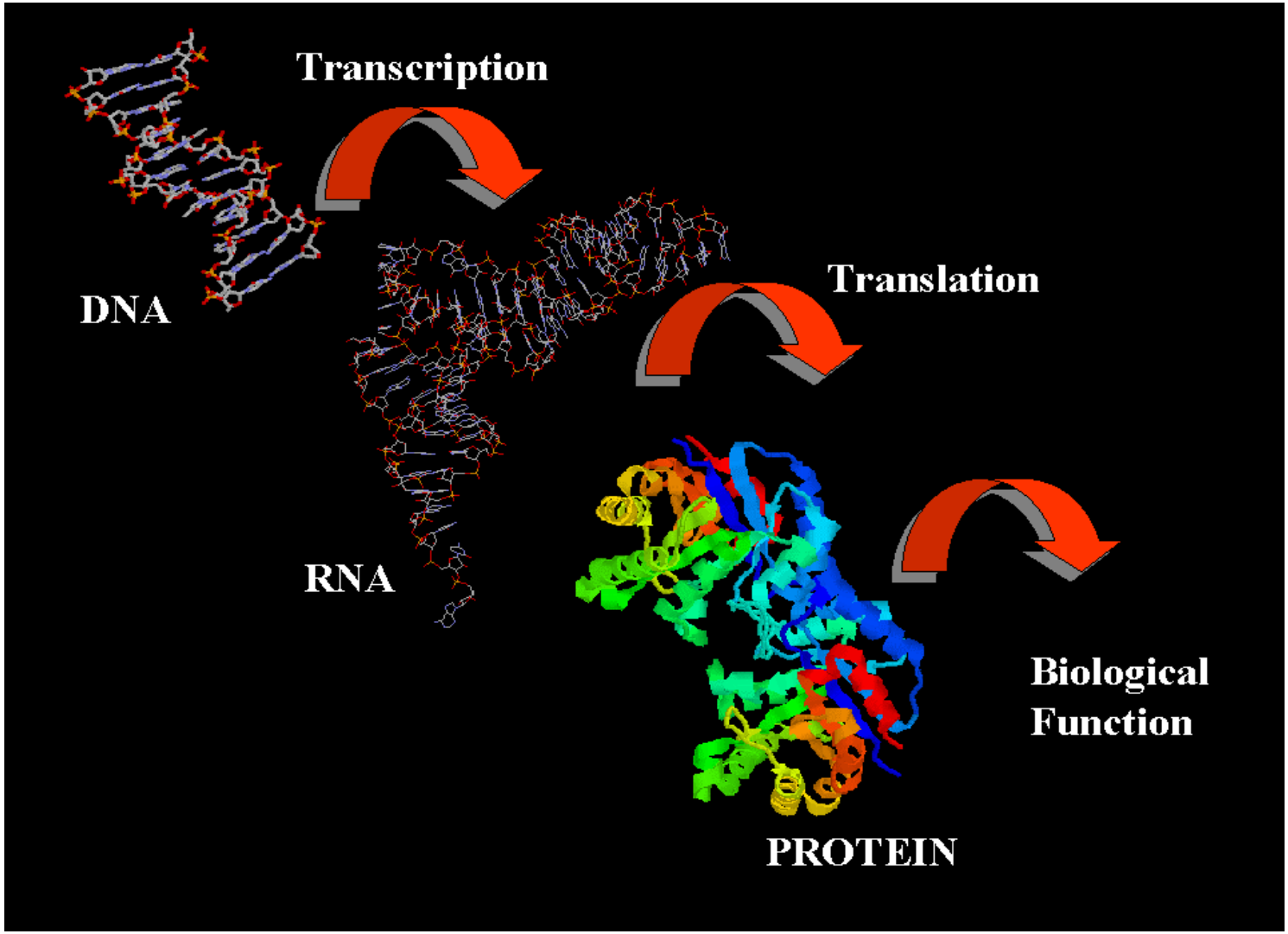
- often based on purely heuristic arguments or cost functions with unclear relation to classification error

- limited theoretical understanding of convergence etc. important issue: which is the 'right' distance measure ?

Relevance Learning: adaptation of the metrics / distance measure during training

here: applied in a (non-standard) classification problem from bioinformatics

Gene expression

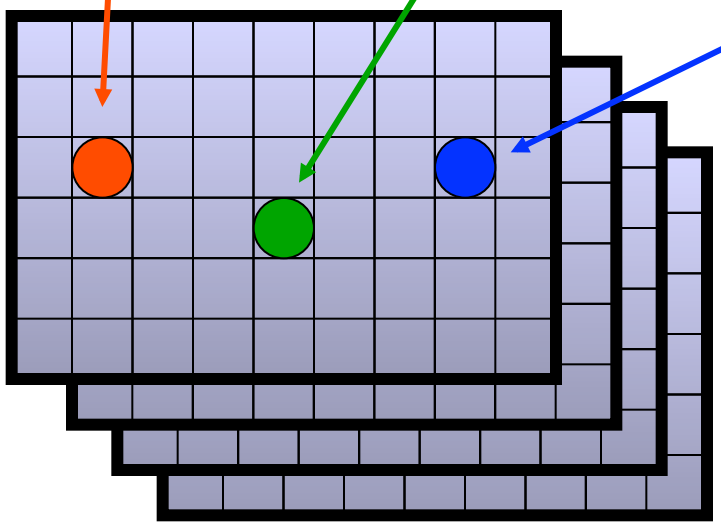


c/o R. Breitling

Genomic tiling array data

sequence covered by 'path' or 'tiling'


C. elegans



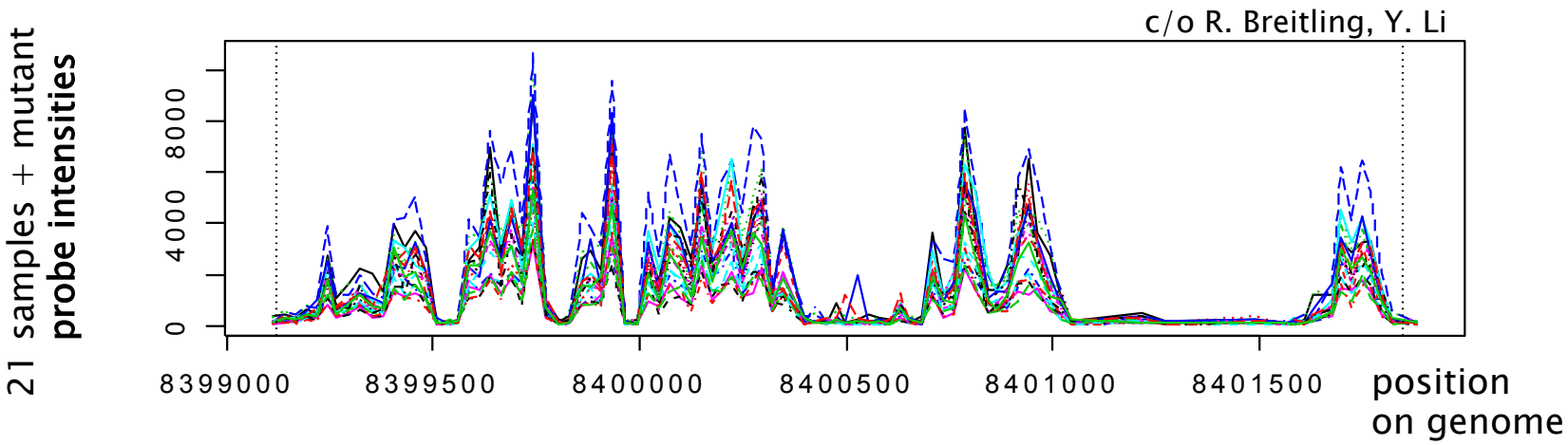
microarray: **transcription intensity vs. genomic position**

repeated for many **samples**:

- different developmental stages
- varying external conditions
- different strains (variants)
- mutants …

perfect match intensities G A G T C T A G G (PM)

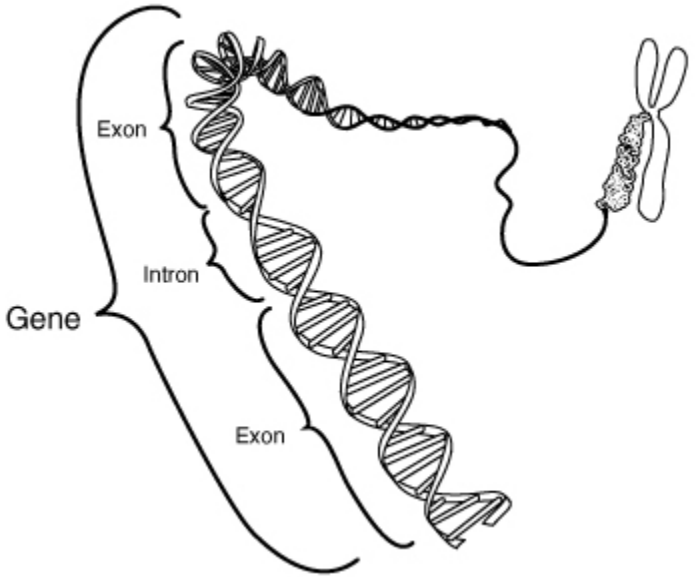
mismatch intensities G A G T G T A G G (MM)



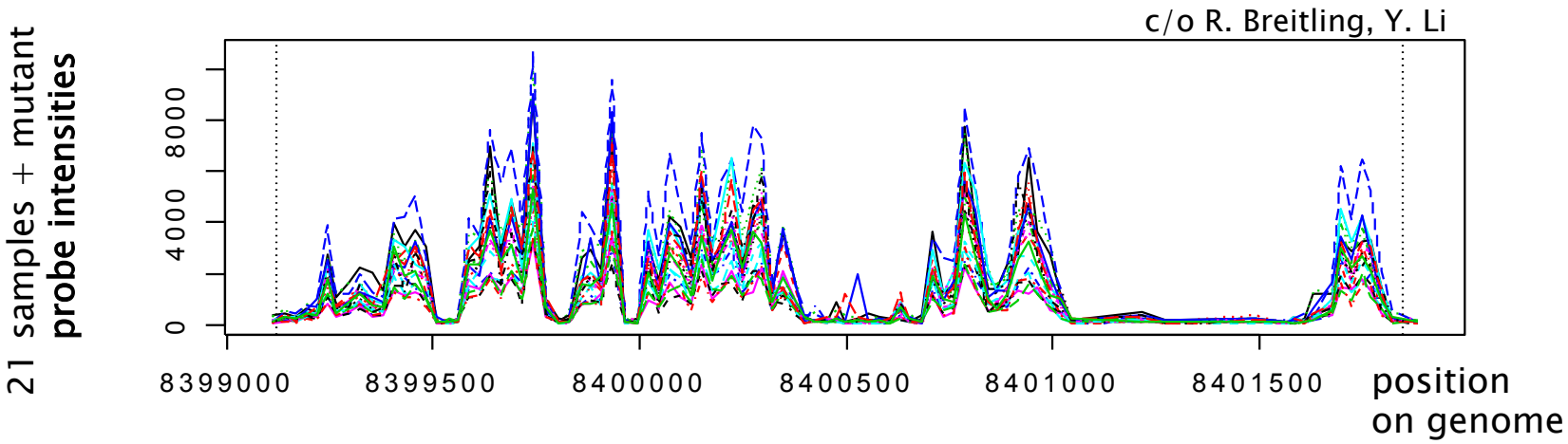
(4120) genomic positions, classified as intronic / exonic

exons: transcribed → mRNA →
translated → protein

introns: transcribed → (pre-) mRNA
but *spliced out* before
leaving the nucleus,
→ no translation



Wikipedia: *non-coding DNA inside a gene*



(4120) genomic positions, classified as intronic / exonic

Note: class membership labels according to the current genome annotation, true introns / exons are not exactly known!

Aim: identify *false introns* = potential new exons (or even genes) ?

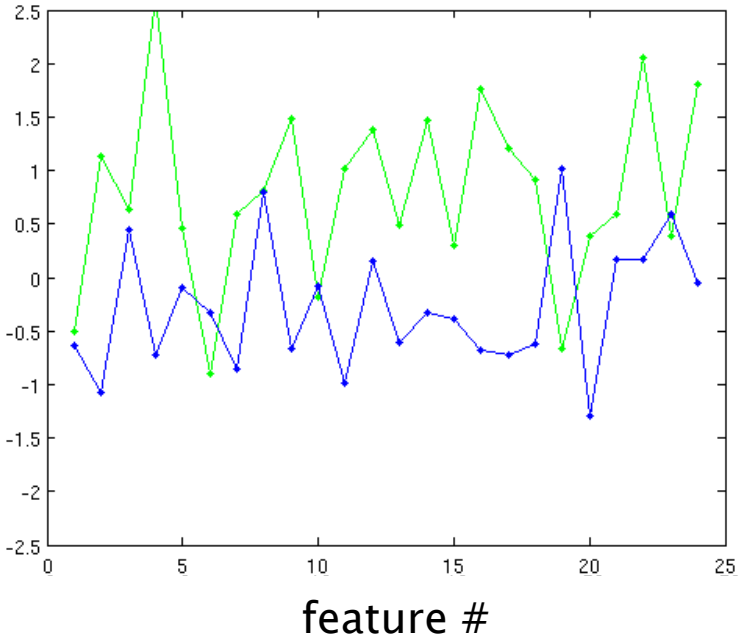
24 features constructed from 'raw data'

including: median PM and MM probe intensities, correlations of neighboring genome positions, melting temperatures, …

4120 **labelled vectors** (2587 from class "0", 1533 from class "1")

24 **features** (real numbers), z-score transformed: $\langle \xi_j \rangle = 0, \quad \langle \xi_j^2 \rangle = 1$

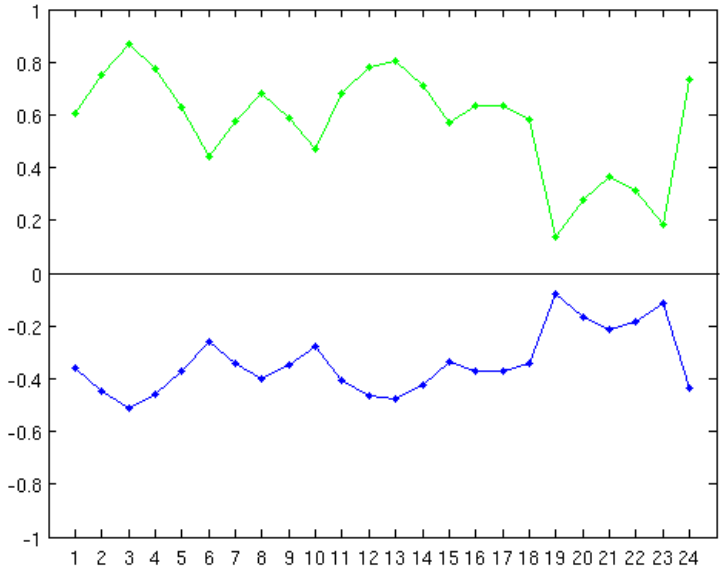
example feature vectors



4120 **labelled vectors** (2587 from class "0", 1533 from class "1")

24 **features** (real numbers), z-transformed: $\langle \xi_j \rangle = 0, \quad \langle \xi_j^2 \rangle = 1$

class conditional mean vectors



→ (Manhattan) distance based classifier

evaluation scheme:

training from 3000 examples

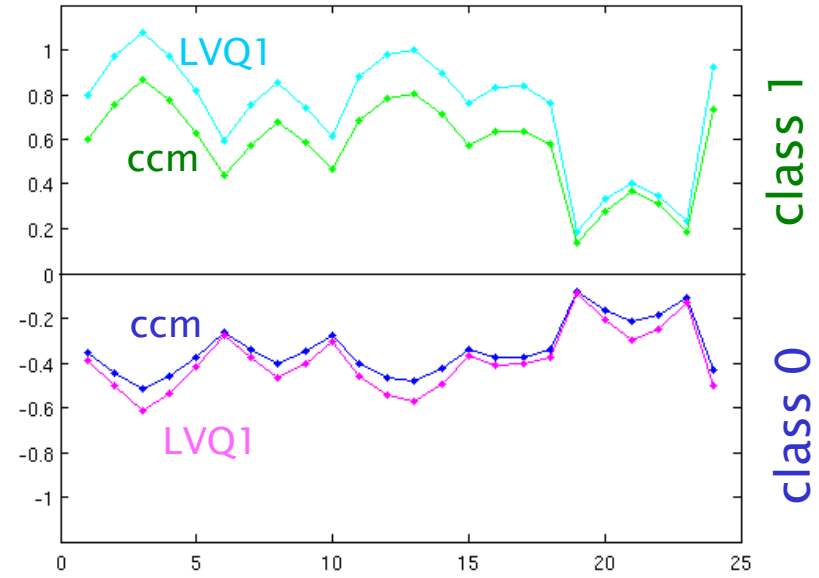
testing on 1000 examples

(avg. over >10 random permutations)

error rates:

	all	class 0	class 1
training	0.125	0.050	0.253
test	0.126	0.052	0.253

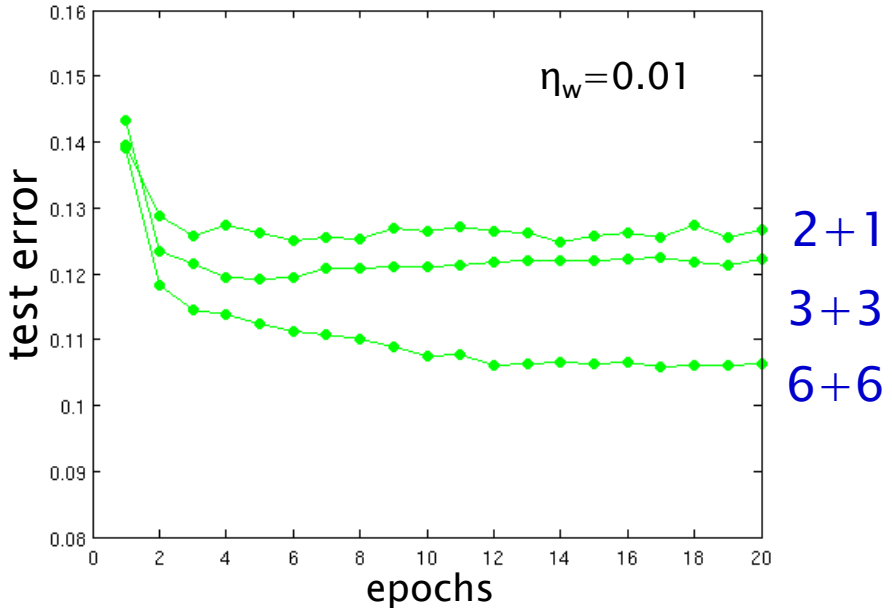
LVQ1 training:



one prototype per class

compared to ccm-prototypes:

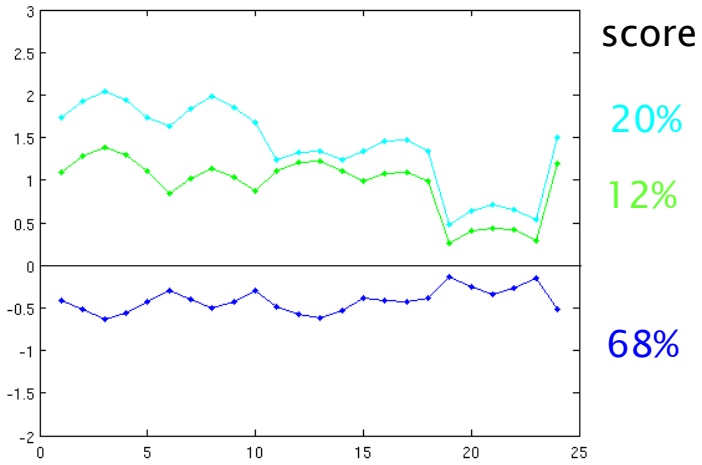
- LVQ1 exaggerates differences between the classes
- here: almost identical performance



several prototypes per class

- increased complexity
- **improved performance**
- possible over-fitting (?) (low training, high test error) due to highly specialized w^i

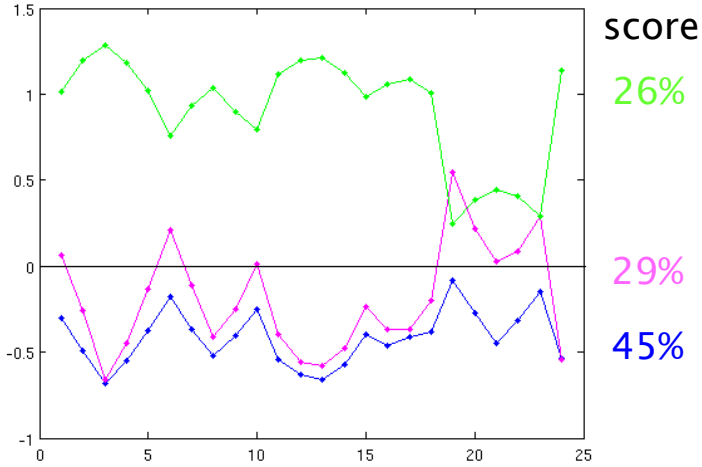
2+1 prototypes



	all	class 0	class 1
test	0.125	0.056	0.247

1+2 prototypes

class 0 class 1



all	class 0	class 1
0.134	0.020	0.326

(→ place more prototypes in class with greater variability)

Adaptive Distance Measures – Relevance Learning

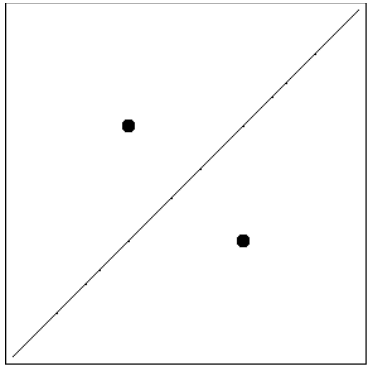
scaled features, e.g. modified Euclidean (or Manhattan,…) distances:

$$d_\lambda\left[w^i,\xi\right]=\sum_{j=1}^{N}\lambda_j\left(w_j^i-\xi_j\right)^2 \quad \text{global relevances} \quad \lambda_j\geq 0,\ \sum_{j=1}^{N}\lambda_j=1$$

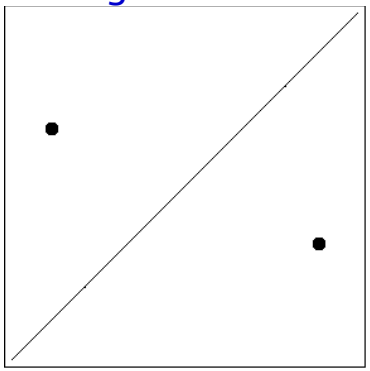
$$d_\lambda^i\left[w^i,\xi\right]=\sum_{j=1}^{N}\lambda_j^i\left(w_j^i-\xi_j\right)^2 \quad \text{local relevances} \quad \lambda_j^i\geq 0,\ \sum_{j=1}^{N}\lambda_j^i=1\ \ (1\leq i\leq K)$$

$$d_\lambda^{s^i}\left[w^i,\xi\right]=\sum_{j=1}^{N}\lambda_j^{s^i}\left(w_j^i-\xi_j\right)^2 \quad \text{class-wise relev.} \quad \lambda_j^c\geq 0,\ \sum_{j=1}^{N}\lambda_j^c=1\ \ (1\leq c\leq C)$$

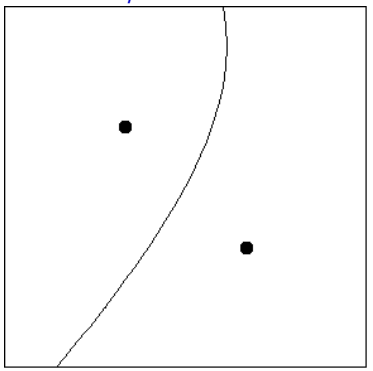
Euclidean



global rel.



local/class-wise



Adaptive Distance Measures – Relevance Learning

LVQ-training + **adaptation of relevances,** e.g. heuristic **RLVQ**
[Bojer, Hammer et al., 2001]

ξ^t → determine winning prototype $\quad w^{i^*} = \underset{j}{\operatorname{argmin}} \left\{ d_\lambda \left[w^j, \xi^t \right] \right\}$

update winner as in LVQ1

update (global) relevances $\quad \lambda_j \rightarrow \lambda_j - \eta_\lambda \ \psi(S^{i^*}, \sigma^t) \ \underbrace{\left| \xi_j^t - w_j^{i^*} \right|}_{\delta_j}$

enforce $\quad \sum_{j=1}^{N} \lambda_j = 1, \ \ \lambda_j \geq 0$

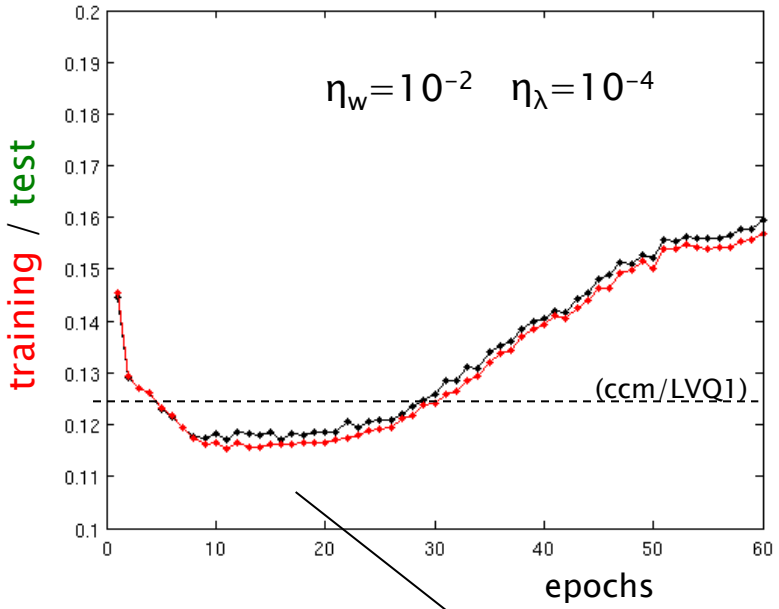
winner is **correct**, **contribution** δ_j **is** { large / small → λ_j { increases / decreases

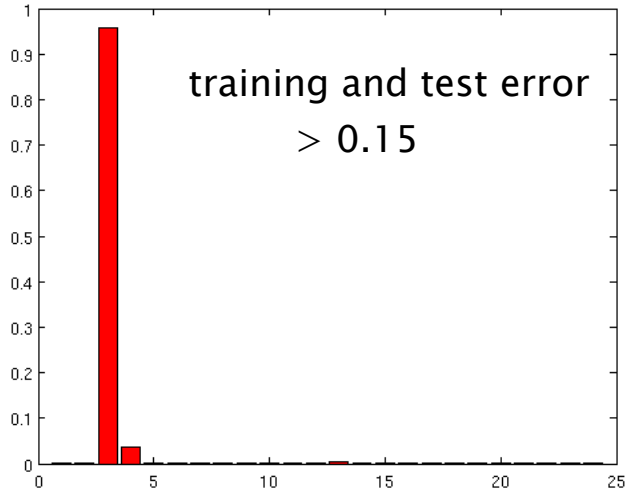
winner is **wrong**, **contribution** δ_j **is** { large / small → λ_j { decreases / increases

- **weighting/ranking** of features → better **performance**
- **elimination** of noisy/irrelevant features → reduced **complexity**

insight into the data / classification problem

(1+1) prototypes, global relevance learning





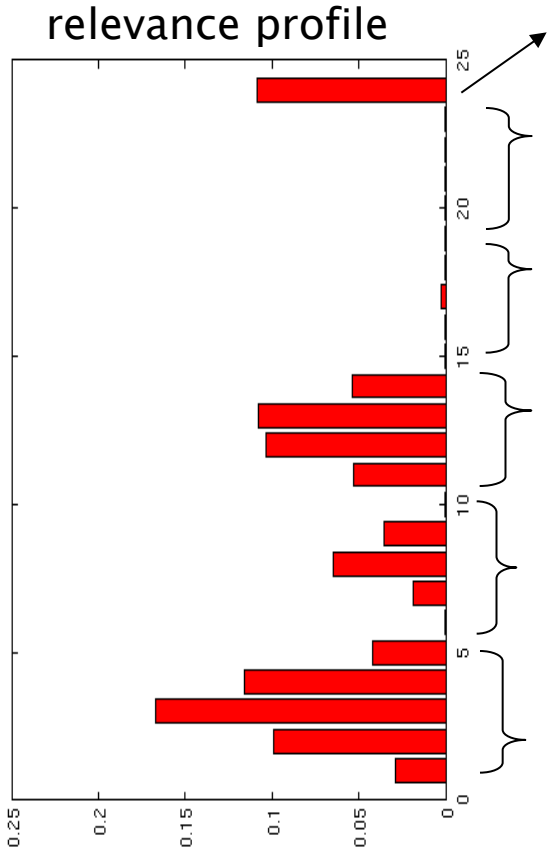
over-simplified classification

(≠ overfitting)

improved performance by weighting and selection of features

successful learning requires $\eta_\lambda \ll \eta_w$

the data revisited: global relevances



significance of strain and stage effects

melting temperature (G-C content)
of probe and its neighbors

m.m. correlations of
probe with neighbors

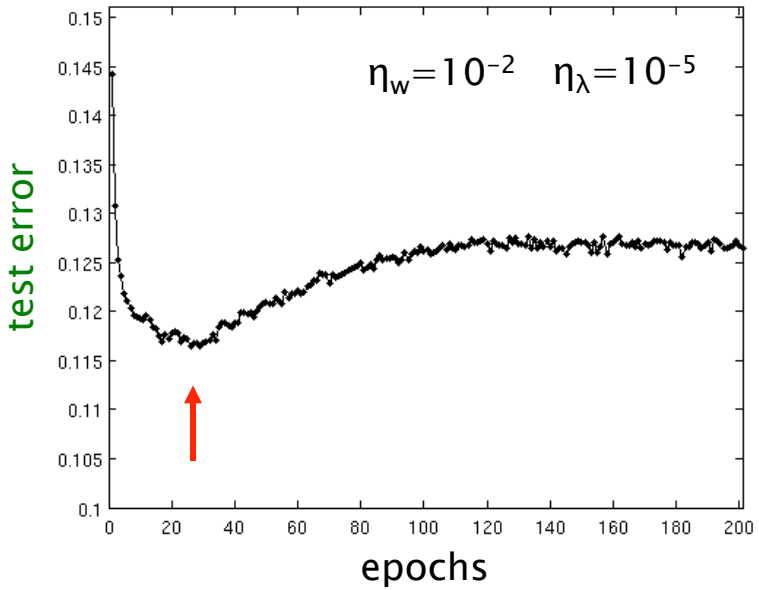
p.m. correlations of
probe with neighbors

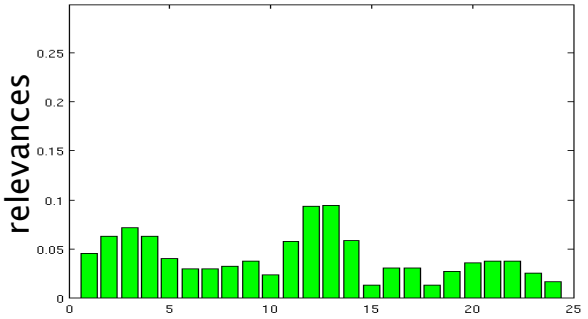
mismatch (m.m) intensities

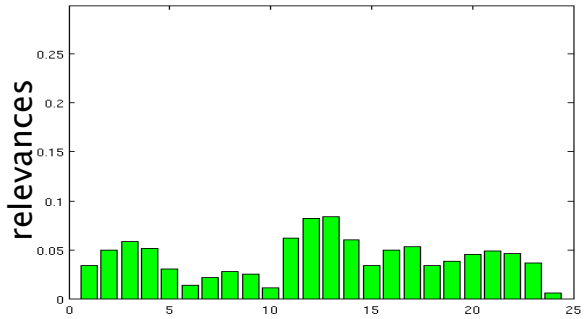
perfect match (p.m.) intensities
of the probe itself (3) and
of neighboring probes

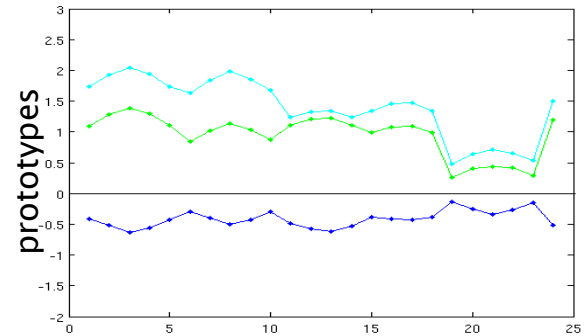
(median over all samples)

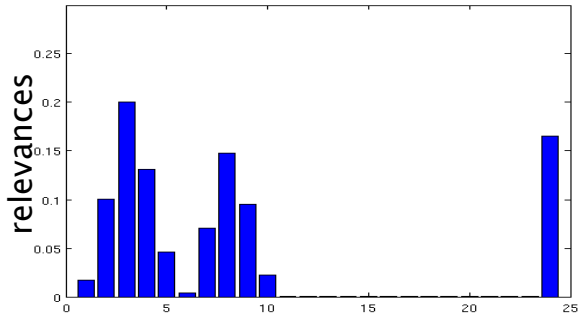
(2+1) prototypes, local relevances



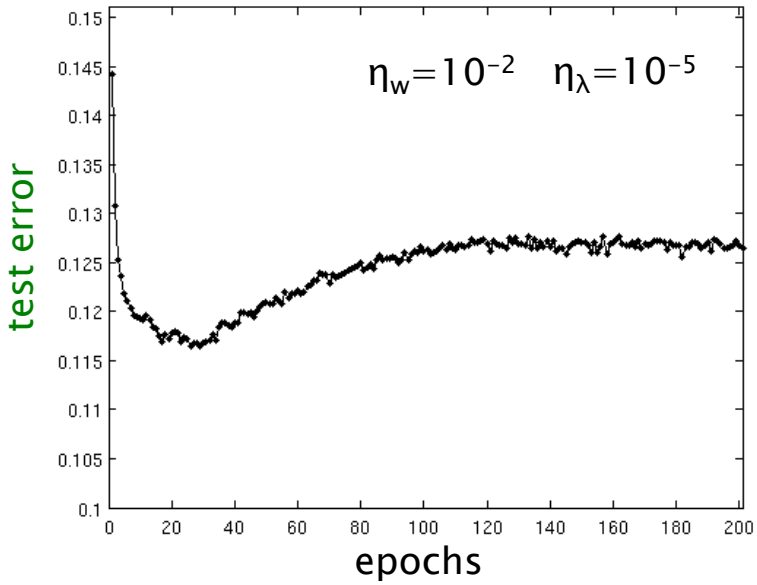


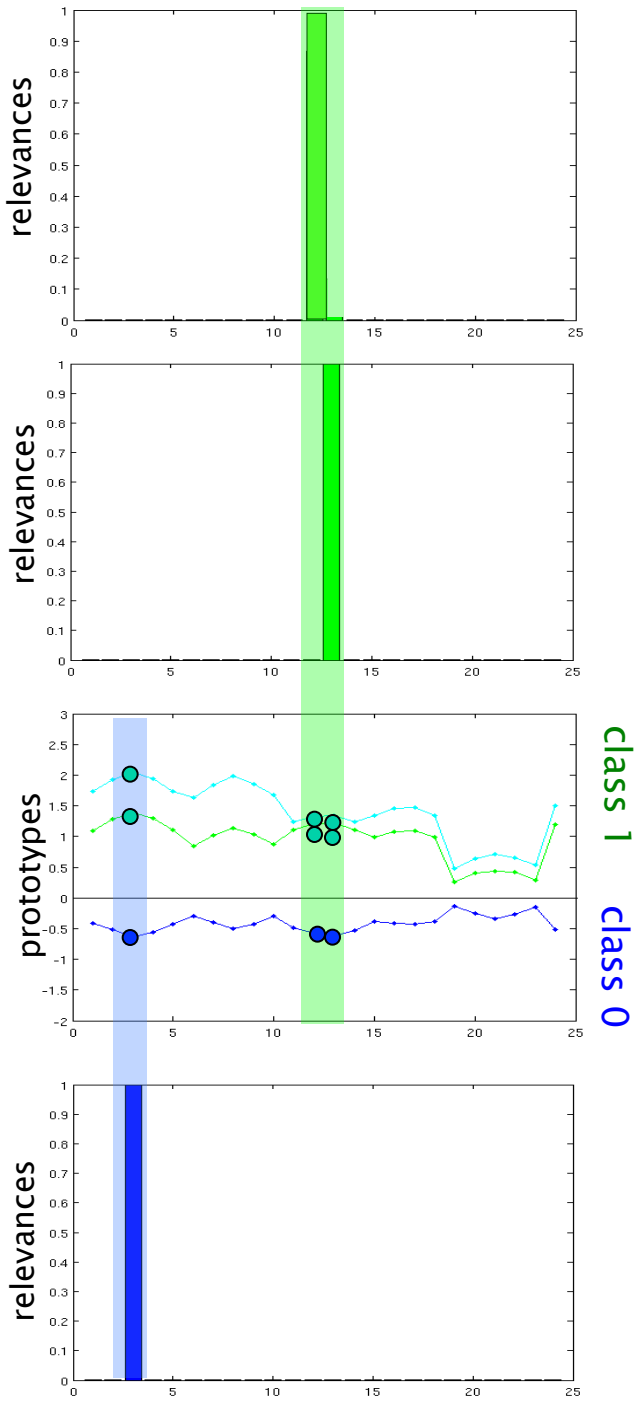






(2+1) prototypes, local relevances



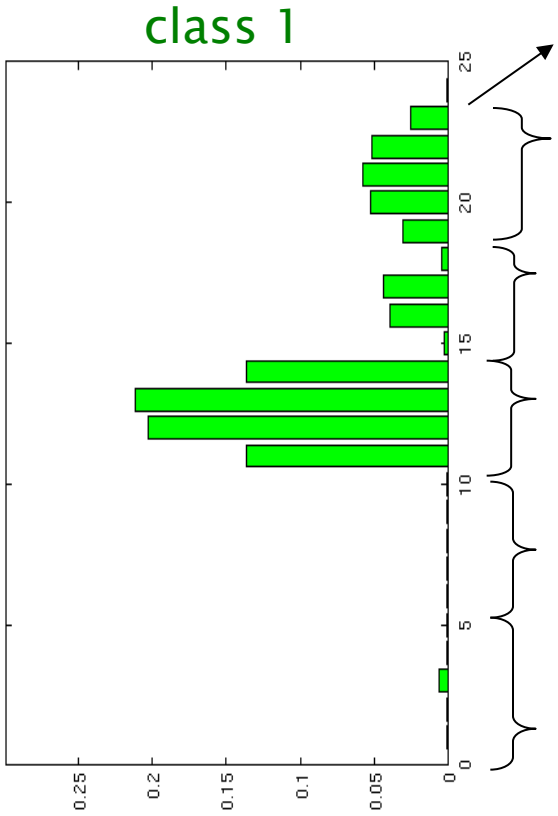


very simple classifier:

determine the minimum of…

$$\underbrace{\left(w_{12}^1 - \xi_{12}\right)^2, \left(w_{13}^2 - \xi_{13}\right)^2}_{\rightarrow \text{class 1}}, \left(w_3^3 - \xi_3\right)^2 \quad \rightarrow \text{class 0}$$

the data revisited: local or class-wise relevances



significance of strain and stage effects

melting temperature (G-C content)
of probe and its neighbors

m.m. correlations of
probe with neighbors

p.m. correlations of
probe with neighbors

mismatch (m.m) intensities

perfect match (p.m.) intensities
of the probe itself (3) and
of neighboring probes

(median over all samples)

important
for exon
identification!

Summary（LVQ）

LVQ classifiers

+ easy to interpret, distance based schemes
+ parameterized in terms of typical data
+ natural tool for multi-class problems
+ suitable for large amounts of data
- standard problems of model selection, parameter tuning,...
- choice of appropriate metrics

Relevance Learning

+ adapts distance measure while training prototypes
+ facilitates significant improvement of performance
+ can simplify the classifier drastically
+ Matrix RLVQ can take into account correlations
- may suffer from *over-simplification* effects

Summary (biology)

classification of exonic/intronic gene sequences

- weighting / selection of features
  leads to improvement and/or simplification of classifier
- plausible results when forced to over-simplify
- importance of p.m. correlations for exon identification
  (novel set of features suggested by Breitling et *al*.)

Outlook (biology)

- systematic study of matrix method (correlations between features)
- extension to whole-genome tiling data (millions of probes!)
- different organisms and technological platforms
- analysis of *raw* data before heuristic construction of features
- investigation of **false introns**

Further reading:

on LVQ:
Kohonen, T., "Self-Organizing Maps," Springer, 2nd ed., Berlin, Heidelberg, 1
on relevance LVQ:
T. Bojer, B. Hammer, D. Schunk, K. Tluk von Toschanowitz, "R elevance deter
B. Hammer and T. Villmann, "Generalized relevance learning vector quantizat

If you are interested to read further:
Generalized Matrix LVQ (GMLVQ):
P. Schneider, M. Biehl and B. Hammer, "Relevance Matrices in LVQ," Proc. of E

Limited Rank Matrix Learning Vector Quantization:
K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villmann and M. Biehl, "D
P. Schneider, K. Bunte, B. Hammer, T. Villmann and M. Biehl, "Regularization
K. Bunte, B. Hammer, P. Schneider and M. Biehl, " Nonlinear Discriminative Da

Thanks!