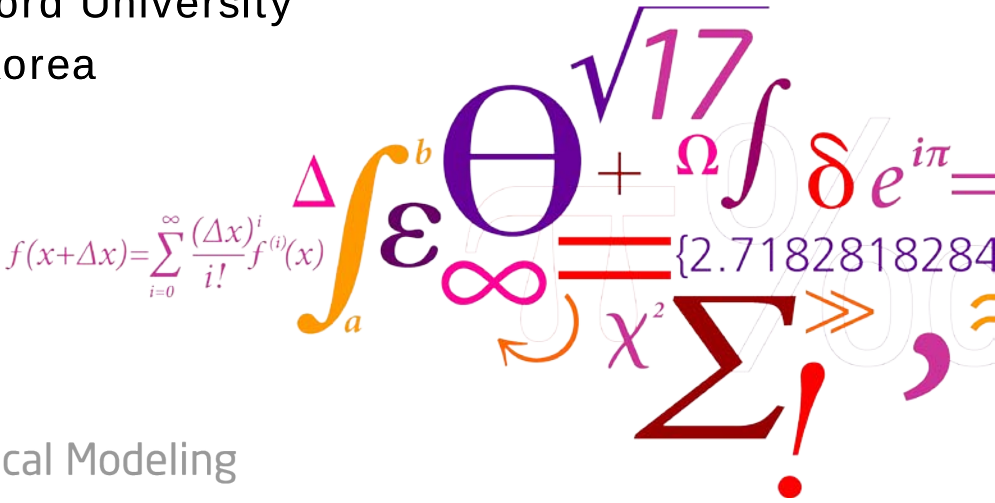
Sparse Supervised Analysis

Line H. Clemmensen
lhc@imm.dtu.dk

Contains joint work with:
Bjarne Ersbøll, DTU Data analysis
Trevor Hastie, Statistics Dept., Stanford University
Michael E. Hansen, Institut Pasteur Korea
Hildur Olafsdottir, DTU Informatics
Rasmus Larsen, DTU Informatics


$f(x+\Delta x)=\sum_{i=0}^{\infty}\frac{(\Delta x)^i}{i!}f^{(i)}(x)$
$\Delta \int_a^b \varepsilon \Theta^{\sqrt{17}} + \Omega \int \delta e^{i\pi} =$
∞
{2.7182818284
χ^2
Σ
!
≫

DTU Informatics
Department of Informatics and Mathematical Modeling

Outline

- The type of data we are considering (large p, small n problems)

- The theory behind such problems

- Methods and examples

The type of data

- Supervised

- Classification or regression

- Large *p*, small *n* problems (many features, few observations)

- Examples: Microarrays in genome research, (spectral) images of samples which are rare or expensive

- Next: What should we consider for this type of data?
  - *We need solutions which are sufficiently rich to answer the questions at hand and at the same time generalize well to yet unseen data instances*

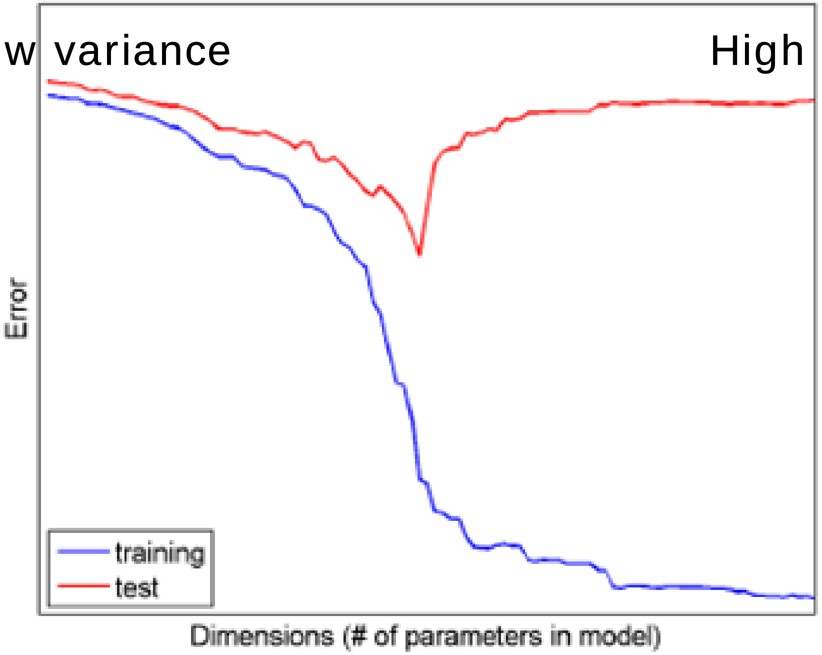
Considerations - Issue of overfitting in large *p*, small *n* problems (bias-variance trade off)

High bias

Low variance

Low bias

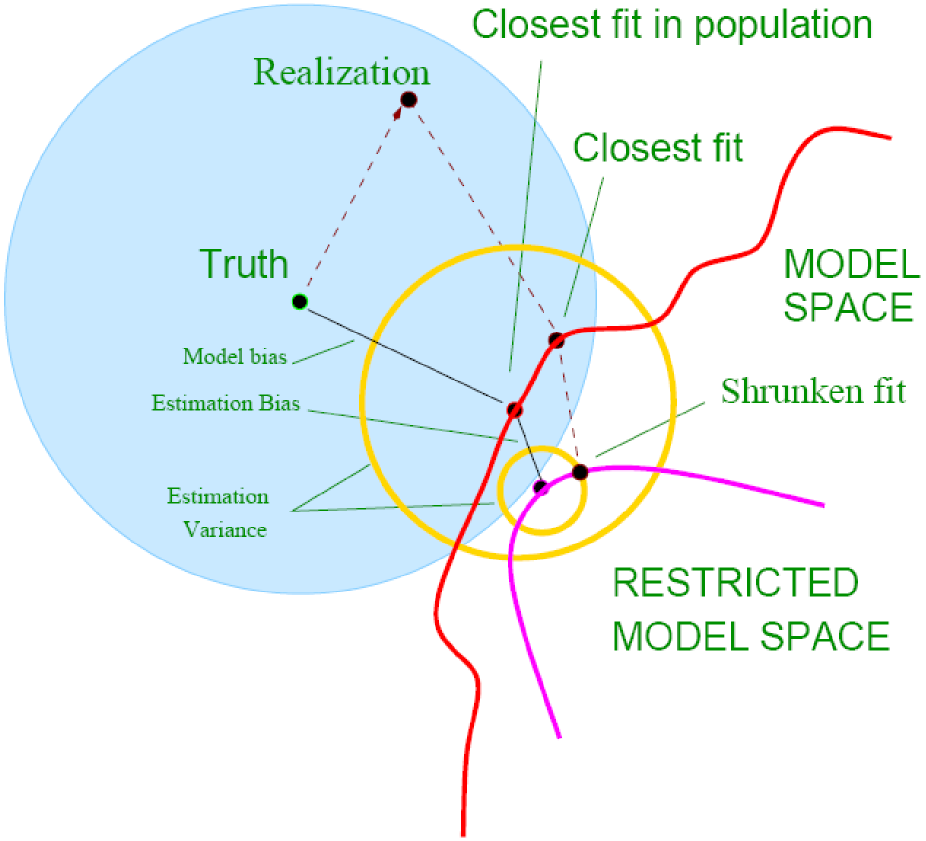
High variance



Low complexity

High complexity

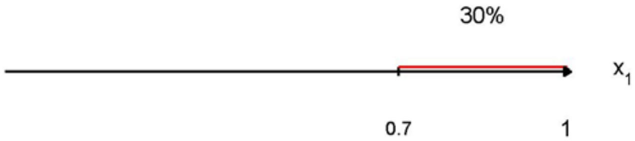
More on bias-variance trade-off



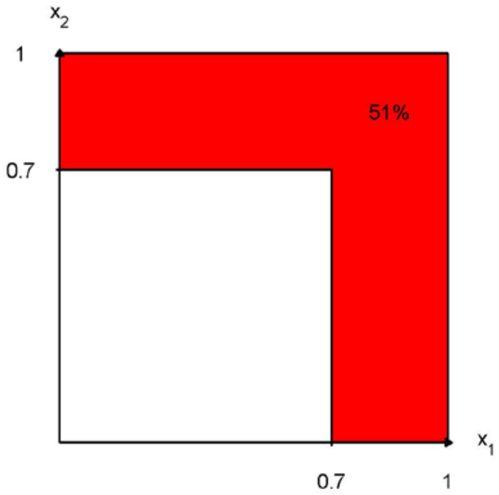
- Figure is from: Hastie et al., *Elements of Statistical Learning*, 2nd Ed.

Curse of dimensionality

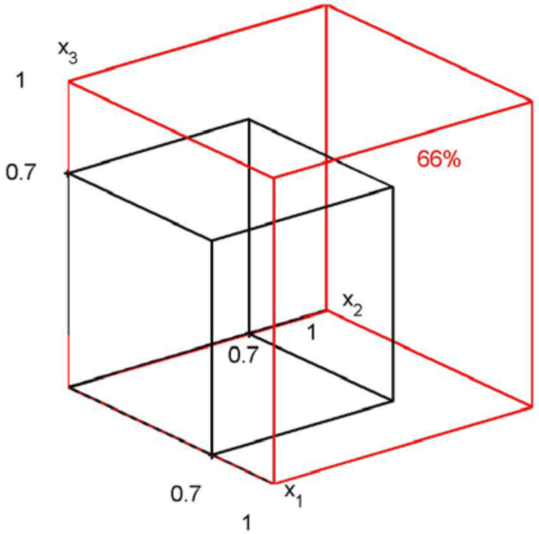
D=1: 70% of $\mathbf{x}_1$, 70% of data



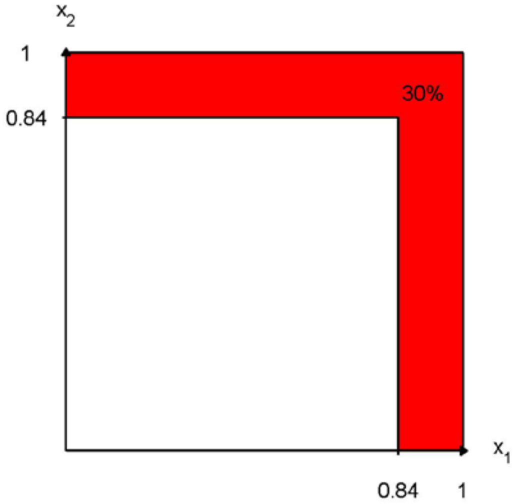
D=2: 70% of $\mathbf{x}_i$, 49% of data



D=3: 70% of $\mathbf{x}_i$, 34% of data



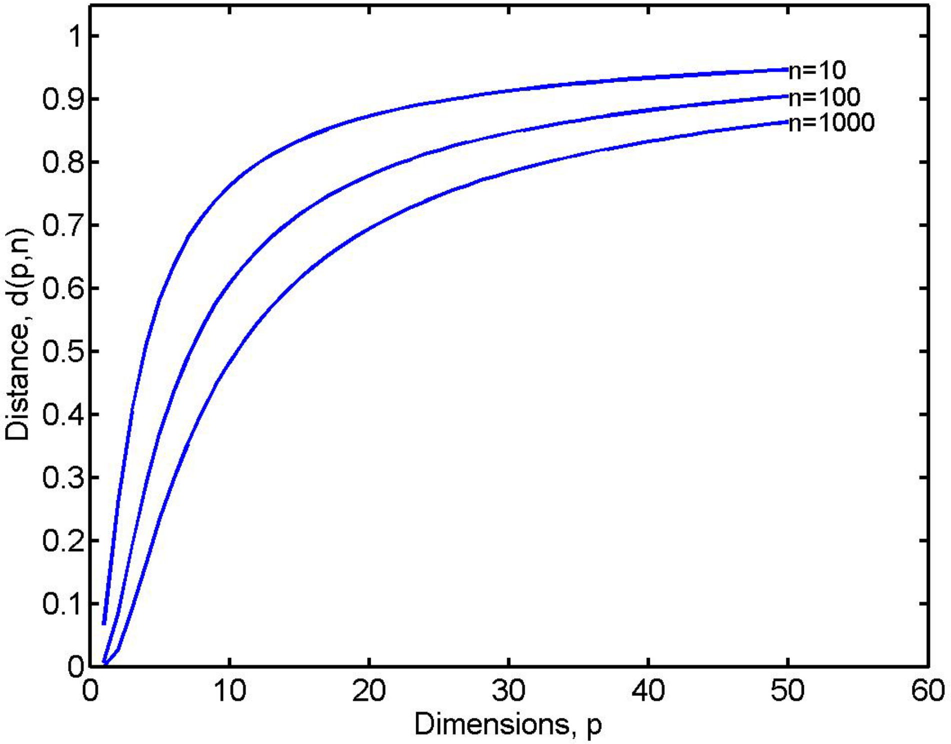
D=2: 84% of $\mathbf{x}_i$, 70% of data



Curse of dimensionality

- For data uniformly distributed in a unit sphere the median distance from the center of the sphere to the closest point is

$$d(p,n) = \left(1 - \frac{1}{2}^{\frac{1}{n}}\right)^{\frac{1}{p}}$$



- Interpolations become extrapolations

Considerations - The curse of dimensionality

- The no. of regions grow exponentially with the dimensionality *p* (Bellman 1961)

- When *p* increases it is necessary to cover a larger and larger range of each variable in order to cover the same fraction of data (exponential relation)
  - This means that local estimates become infeasible: (a) the estimates become global if we include more samples, (b) the variance of the estimate increases if we user fewer samples

- The median distance from the center of data to the closest point also grows with the dimensions – data points are all close to the boundary
  - This means that interpolations become extrapolations which have less generalization power

Considerations - Blessings of dimensionality

- It’s not all bad… (Donoho 2000)
- 1st blessing comes from probability theory and assumes that there are many similar (highly correlated) features which we can average over.
- 2nd blessing comes from the central limit theorem and says that there is an underlying limit distribution which is approached as the dimensions go to infinity = data lie on a low-dimensional manifold.
- 3rd blessing arises when measurements are taken from an underlying continuous process, e.g. images or spectra and says that for such data the underlying structure often gives an approximate finite dimensionality = data lie on a low-dimensional manifold.

Considerations - Dimension reduction is crucial

- Feature selection or extraction
  - Forward selection
  - Backward elimination

- Regularization of parameters (priors)
  - Ridge regression, lasso, elastic net, SDA, SPLS

- Projections to lower dimensions – latent variables - decompositions
  - PCA, PLS, MNF, ICA, multiway models

- Clustering of features

- Structuring parameter estimates

The theory behind - regularizations

- First we turn to regression

- Y: continuous response/output
- X: observations times predictors/features
- $\boldsymbol{\beta}$ **:** parameters in model

Sparsity in regression using l1-regularization

Ridge

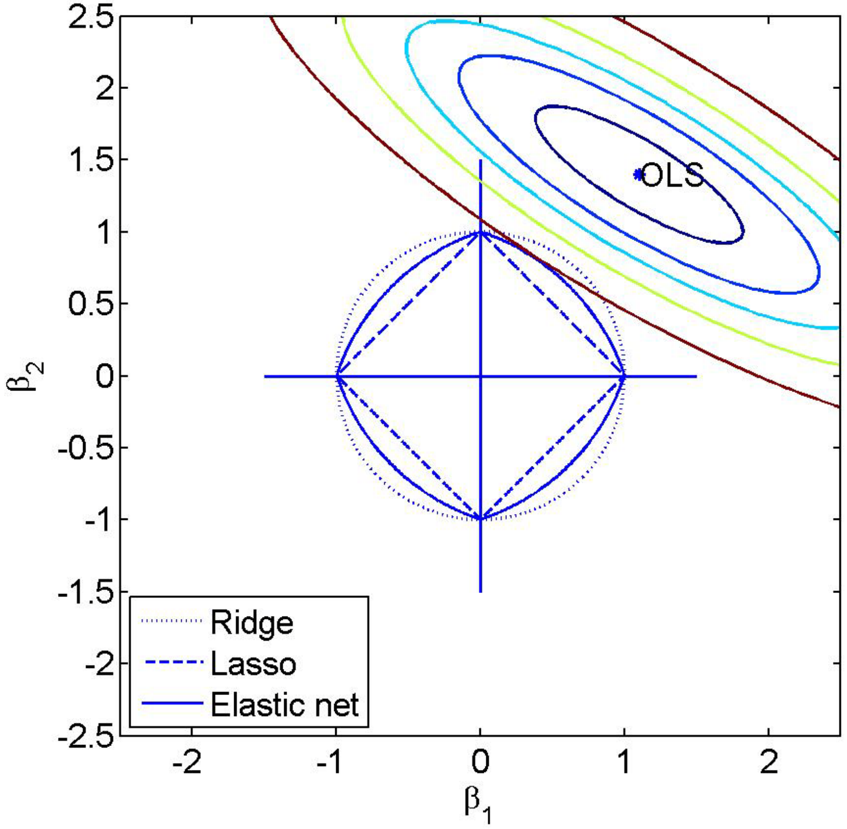
$$\boldsymbol{\beta}_{ridge} = \arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2$$

Lasso

$$\boldsymbol{\beta}_{lasso} = \arg\min_{\boldsymbol{\beta}} \|\,\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\,\|_2^2 + \lambda \|\,\boldsymbol{\beta}\,\|_1$$

Elastic net

$$\boldsymbol{\beta}_{en} = \arg\min_{\boldsymbol{\beta}} \|\,\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\,\|_2^2 + \lambda_2 \|\,\boldsymbol{\beta}\,\|_2^2 + \lambda_1 \|\,\boldsymbol{\beta}\,\|_1$$



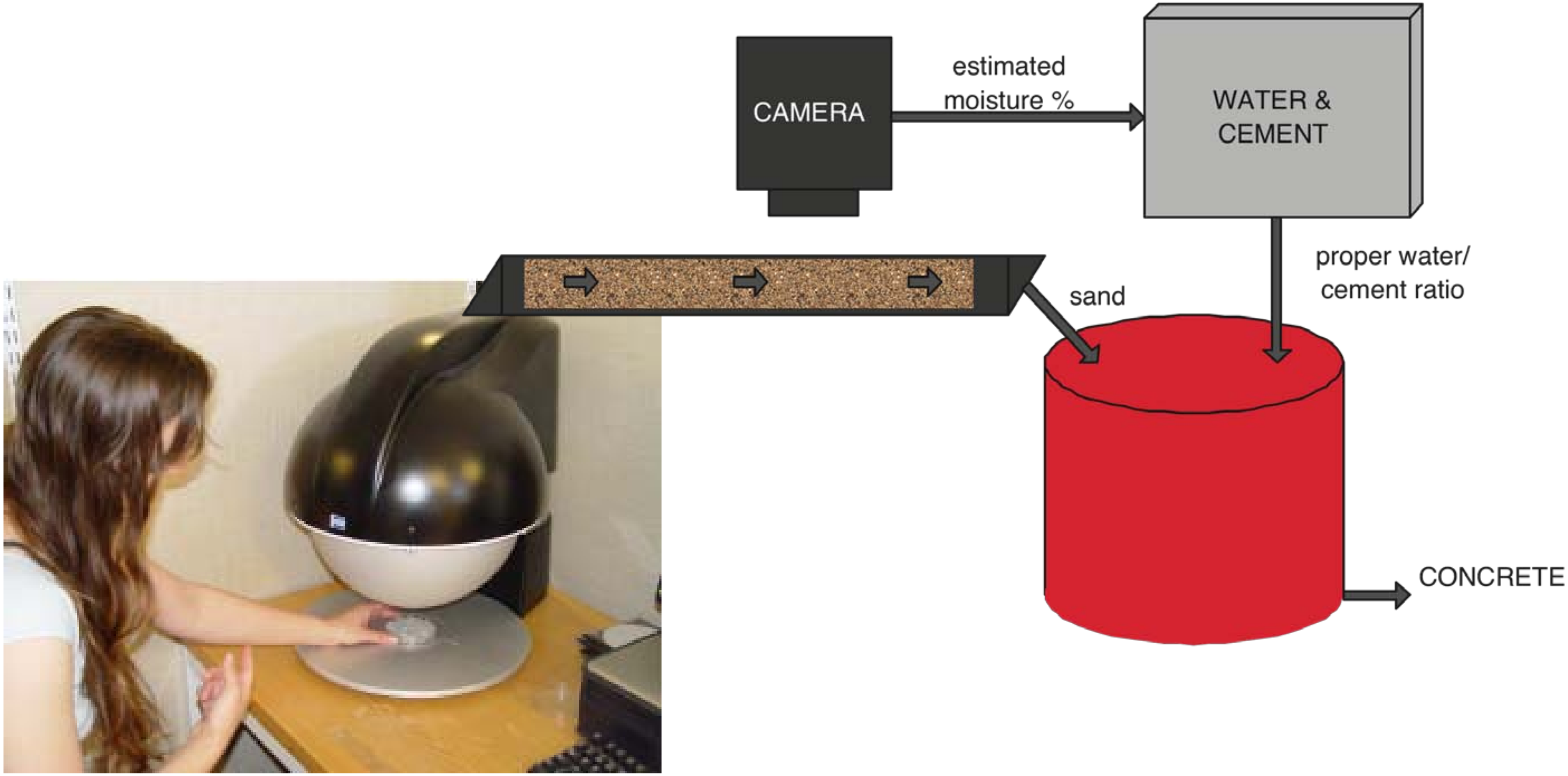
The elastic net

- Advantages: Combines the shrinkage of ridge and parameter selection of the lasso to obtain robust sparse estimates
  - Get rid of irrelevant variables/select important variables (lasso)
  - Ability to include more variables than there are observations (ridge)
  - Works well when covariates are highly correlated; allows us to "average" highly correlated features and obtain more robust estimates (grouping feature)

- Disadvantages: Issue of tuning two parameters

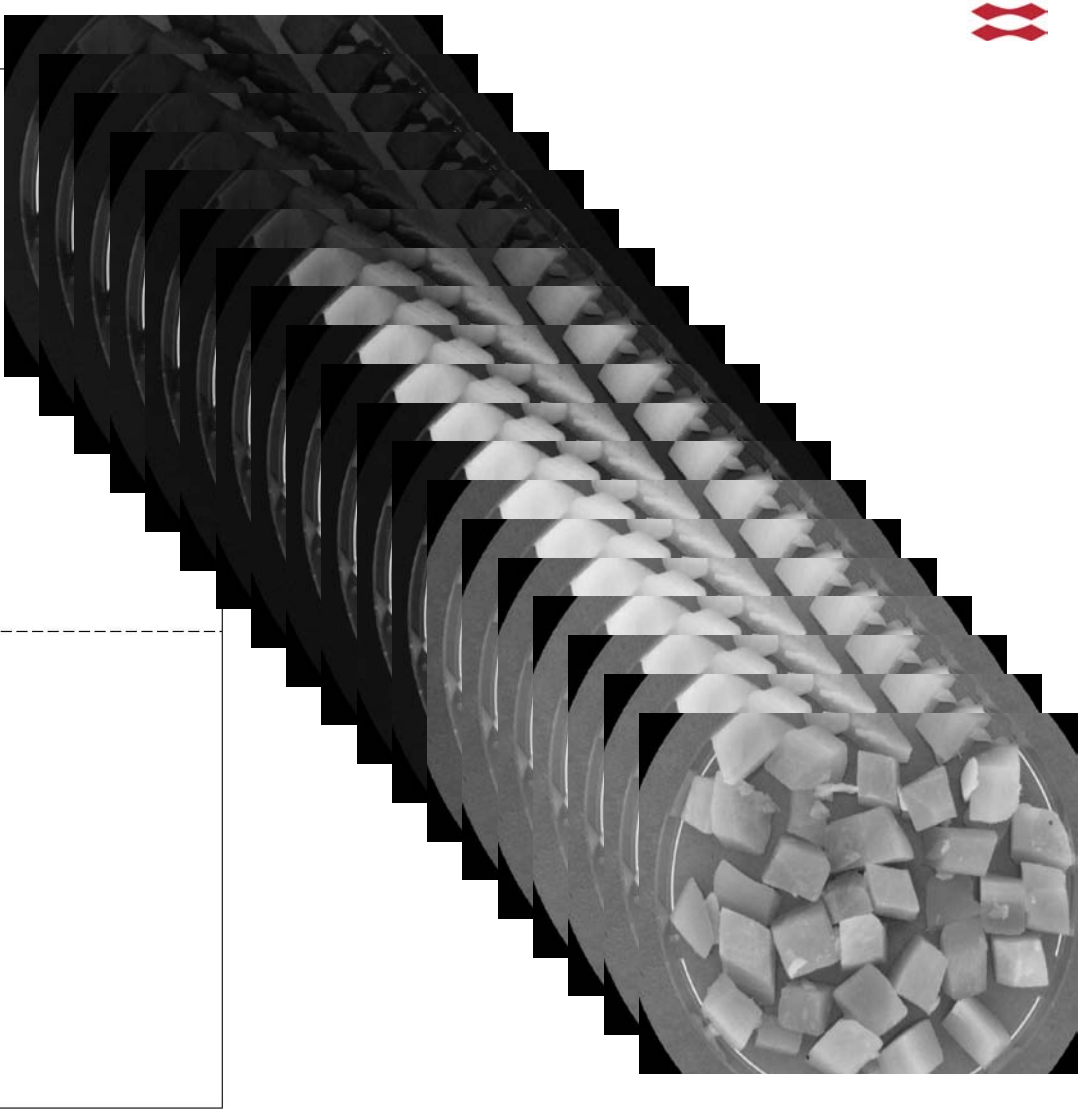
Example of sparse regression

- Multi-spectral images of sand used to make concrete



Examples - Spectra

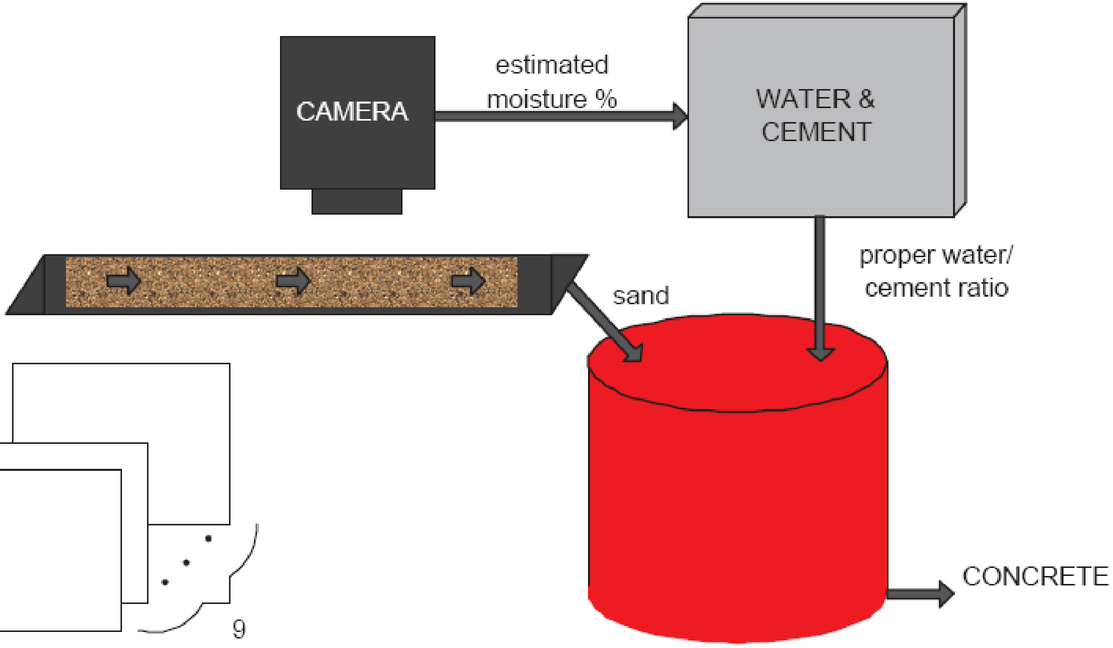
Wavelength (nm)	Color
430	Ultra blue
450	Blue
470	Cyan
505	Cyan
565	Amber
590	Orange
630	Red
645	Red
660	Red
700	Red
850	NIR
870	NIR
890	NIR
910	NIR
920	NIR
940	NIR
950	NIR
970	NIR

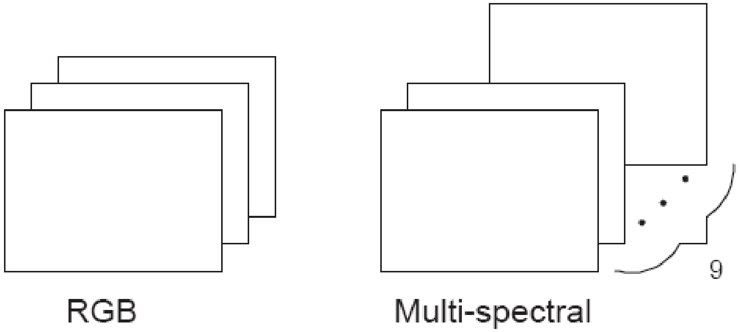


Example - Estimation of moisture content in sand used to make concrete

$$(cement + water) + aggregate = concrete$$

In-line approach:



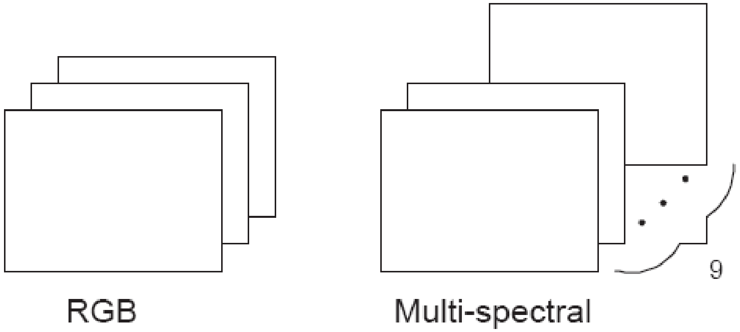


Example - Estimation of moisture content in sand used to make concrete

- Features: 1st, 5th, 10th, 30th, 50th, 70th, 90th, 95th, and 99th percentiles are extracted from the ROIs; resulting in 2016 features.
    - On each spectrum, on pairwise differences and pairwise ratios between spectra
- n = 21 images of sand
- p = 2016 features

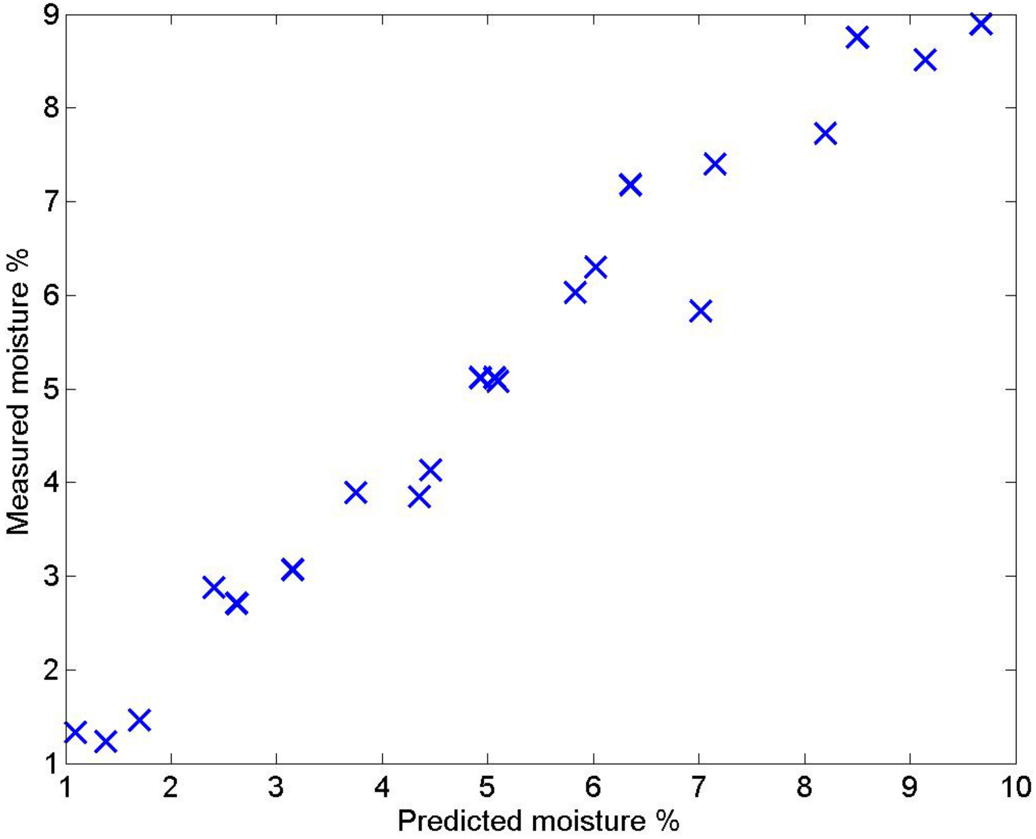
  which represent

  each image



Results - sand

- Sand (type 2), MSE = 0.2 moisture % (leave-one-out predictions)
- 109/2016 features were chosen, ridge regularization 10^{-1}



Classification

- In the next slides we will look at classification methods
- And how to make them sparse

- We consider:

- *K* normally distributed classes with means μ_j, $j = 1,\ldots,k$ and equal dispersion $\boldsymbol{\Sigma}$.

Linear discriminant analysis

- Maximize between-class sums of squares (variance)

$$\Sigma_B = \sum_{j=1}^{K} (\mu_j - \mu)^T (\mu_j - \mu)$$

- Minimize within-class sums of squares (variance)

$$\Sigma_W = \sum_{j=1}^{K} \sum_{i=1}^{n_i} (X_{ij} - \mu_j)^T (X_{ij} - \mu_j)$$

- Find the discriminating directions as, *j=1,…,k-1* (Fisher’s criterion)

$$\beta_{j,LDA} = \arg\max_{\beta_j} \beta_j^T \Sigma_B \beta_j$$

- Under the orthogonality constraint

$$\beta_j^T \Sigma_W \beta_l = \begin{cases} 0 & l = 1,\dots, j-1 \\ 1 & l = j \end{cases}$$

Optimal scoring – Classification via regression

- It can be shown that optimal scoring and LDA are equivalent (using the equivalence with canonical correlation analysis - CCA)

$$\arg\min_{\boldsymbol{\theta},\boldsymbol{\beta}} n^{-1} \| \mathbf{Y}\boldsymbol{\theta} - \mathbf{X}\boldsymbol{\beta} \|_2^2$$

$$\text{s.t.} \quad n^{-1} \| \mathbf{Y}\boldsymbol{\theta} \|_2^2 = 1$$

- **Y** is a matrix of dummy variables of the classes.
- $\boldsymbol{\theta}$ assigns a score $\boldsymbol{\theta}_{ij}$ for each class *i* and each parameter vector $\boldsymbol{\beta}_j$. The scores give a continuous ordering of the samples. Thus, we regress these quantitative responses on the predictors **X**.

Methodical development - Sparse discriminant analysis

- The ridge and lasso penalties are added to the parameter estimates

$$\arg\min_{\boldsymbol{\theta},\boldsymbol{\beta}} \left(\| \mathbf{Y}\boldsymbol{\theta} - \mathbf{X}\boldsymbol{\beta} \|_2^2 + \lambda_2 \| \boldsymbol{\Omega}^{1/2}\boldsymbol{\beta} \|_2^2 \boxed{+\lambda_1 \| \boldsymbol{\beta} \|_1} \right)$$

$$\text{s.t. } n^{-1} \| \mathbf{Y}\boldsymbol{\theta} \|_2^2 = 1$$

- $\boldsymbol{\beta}$ is *p* times *K*

1. Fix $\boldsymbol{\theta}$ and update $\boldsymbol{\beta}$ (elastic net)
2. Next fix $\boldsymbol{\beta}$ and update $\boldsymbol{\theta}$ (singular value decomposition)
3. Repeat step 1 and 2 until convergence or maximum no. of iterations is reached
4. Remove trivial directions using the singular values

Examples of sparse classification

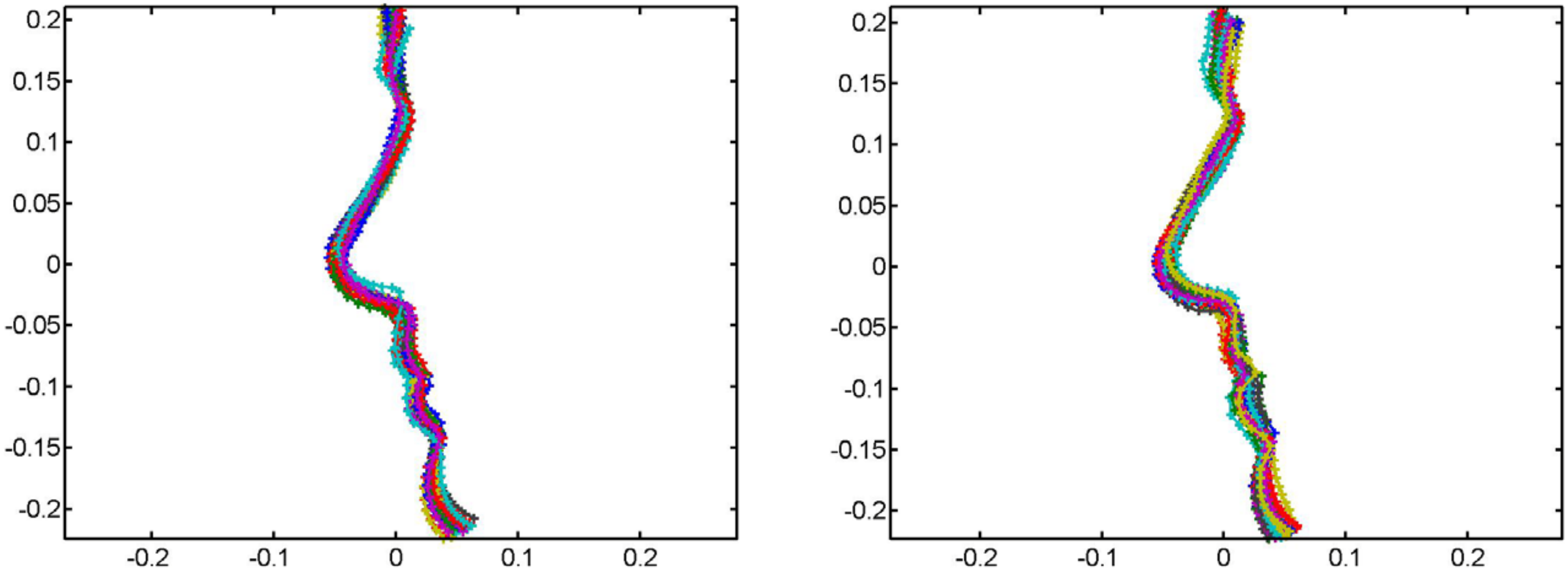
- Example 1 (Matlab): Silhouette profiles: male vs. female (landmarks).

- Example 2: Classification of three fish species (RGB images).

- Example 3: Mixture models – nonlinear boundaries and subgroups within classes. A simulation example.

Example 1 (Matlab) – the data

- load('Silhouettes')
  - % Xa (data) , Fem (female indices), Male (male indices)
- figure, plot([Xa(Fem,:)].','-+'); axis equal; % 19 female shapes
- figure, plot([Xa(Male,:)].','-+'); axis equal; % 20 male shapes



Shape = 65 landmarks of (x,y) coordinates = 130 features

Example 1 (Matlab) – predictors and response

- X(:,1:65) = real(Xa); % the first 65 features are the x-coordinates
- X(:,66:130) = imag(Xa); % the last 65 features are the y-coordinates
- Yc(Fem) = 1; % female = class 1
- Yc(Male) = 2; % male = class 2
- Y = double([Yc-1, -(Yc-2)]); % Y dummy (zeros and ones)

```
Y =
1      0
0      1
1      0
0      1
0      1
.
.
```

Example 1 (Matlab) – train and test

- I = randperm(N);
- Itr = I(1:22);
- Itst = setdiff(1:N,Itr);

```
Itr = 31    20    13    16
14    21     5    35    29
8    18    27     9    30     1
24    36    17    26    28
15     4
```

```
Itst = 2     3     6     7    10
11    12    19    22    23
25    32    33    34    37
38    39
```

- Ytr = Y(Itr,:);
- Xtr = X(Itr,:);
- Xtst = X(Itst,:);

Example 1 (Matlab) – normalizing data

- [Xtr,mx,vx] = normalize(Xtr);
- Xtst = normalize_test(Xtst, mx, vx);

- Every feature gets zero mean and standard deviation one, X^TX is now the correlation matrix of X.

Example 1 (Matlab) – running sparseLDA

- Setting parameters:
- lambda = 1e-2; % L2-norm parameter
- stop = -10 % L1-norm parameter (number of non-zero loadings)
- maxiter = 30; % parameter: max. number of iterations

- Running the function:
- [sl theta rss] = slda(Xtr, Ytr, lambda, stop, maxiter,1);

```
ite: 1,          ridge cost: 22.0037,     |beta|_1: 1.1229
ite: 2,          ridge cost: 22.0293,     |beta|_1: 3.1761
ite: 3,          ridge cost: 22.0293,     |beta|_1: 3.1761
final update,ridge cost: 5.2001,          |beta|_1: 3.1761
```

Example 1 (Matlab) – the selcted parameters

- Which parameters are active?
- [ActV,J]=find(sl);
- disp(ActV')

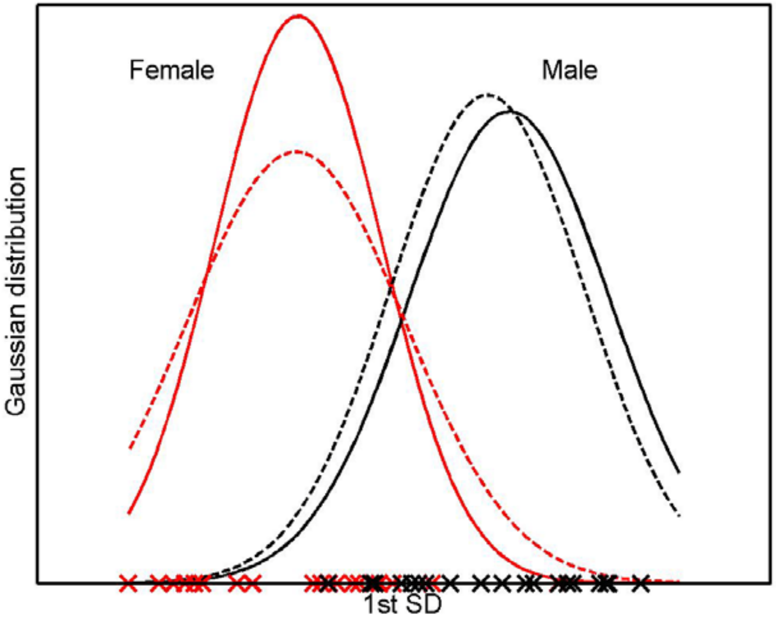
```
10    13    14    56    87    88   105
114   115   130
```

Example 1 (Matlab) – the errors

- Prediction:
- [err_tr, err_tst, DCtr, DCtst] = predictSLDA(Xtr, Yc(Itr), Xtst, Yc(Itst), sl);

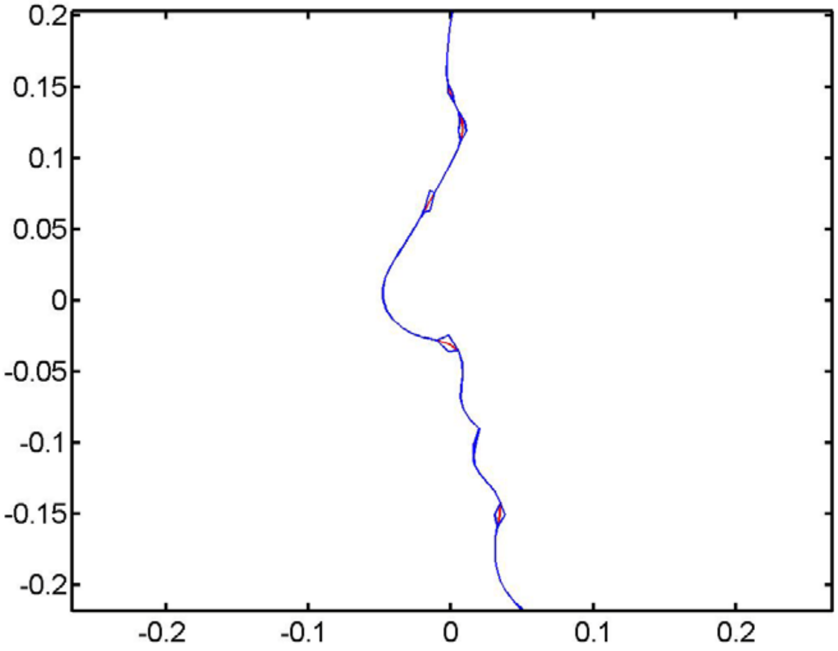
- disp([err_tr*100, err_tst*100])

18.18% 17.65%



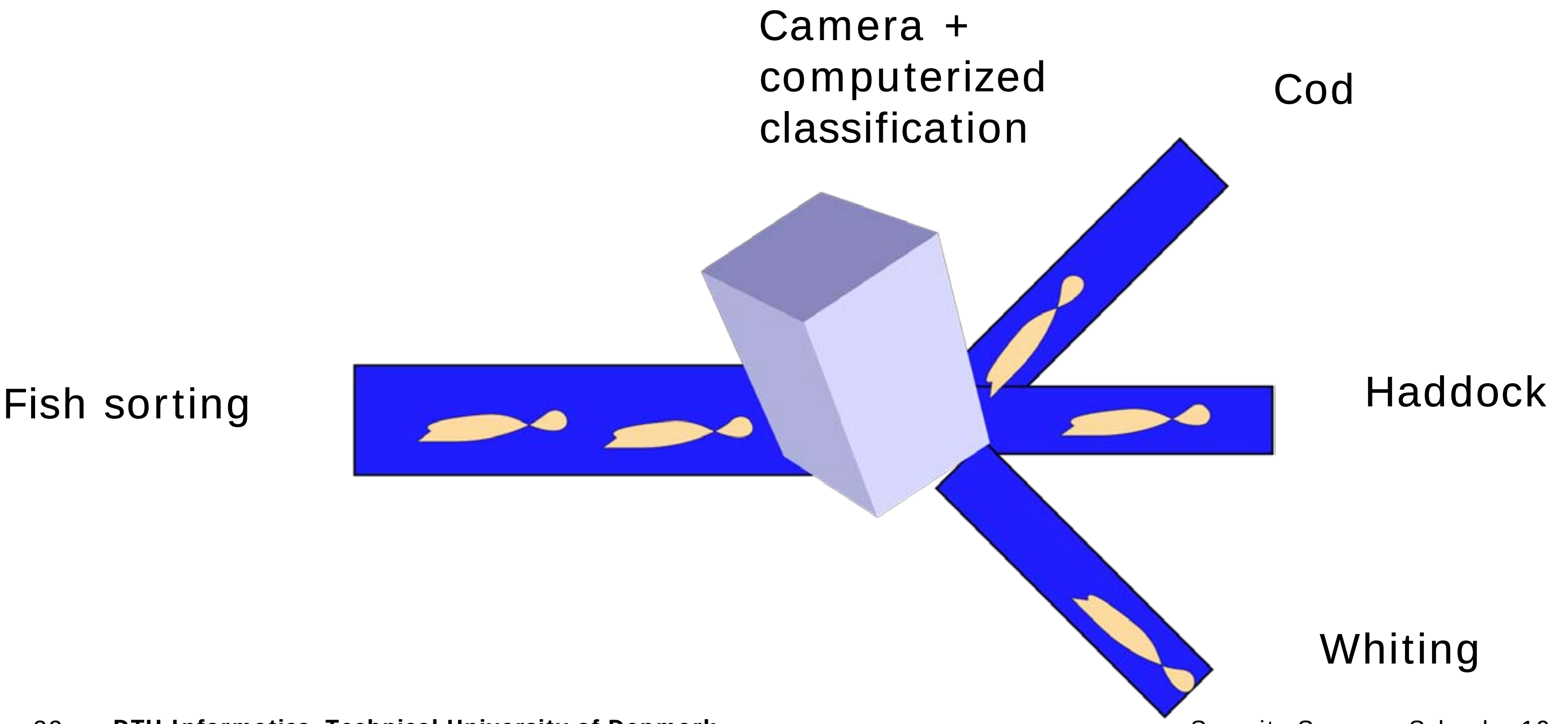
Exmample 1 (Matlab) – illustrating the model

```
• figure, plot(mean(Xa,1).','-r','MarkerSize',10,'linewidth',1), hold on; axis equal;
• Xa_pred = [sl(1:65,1)+i*sl(66:end,1)];
• plot(mean(Xa,1).'+(std(Xa,[],1).'.*Xa_pred*2),'-b','MarkerSize',10,'linewidth',1);
• plot(mean(Xa,1).'-(std(Xa,[],1).'.*Xa_pred*2),'-b','MarkerSize',10,'linewidth',1);
```



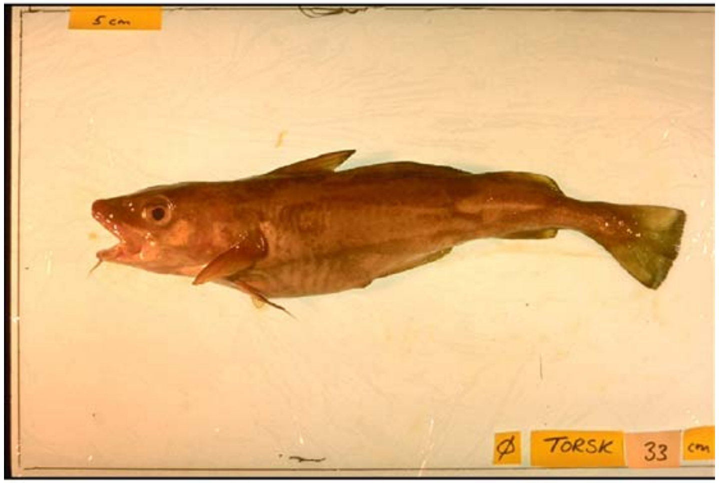
Example 2 – sorting fish species

- Classify three fish species objectively (can eventually be implemented in a construction line)
- The fish species: Cod, haddock, and whiting
- Image analysis of standard color images -> Classification

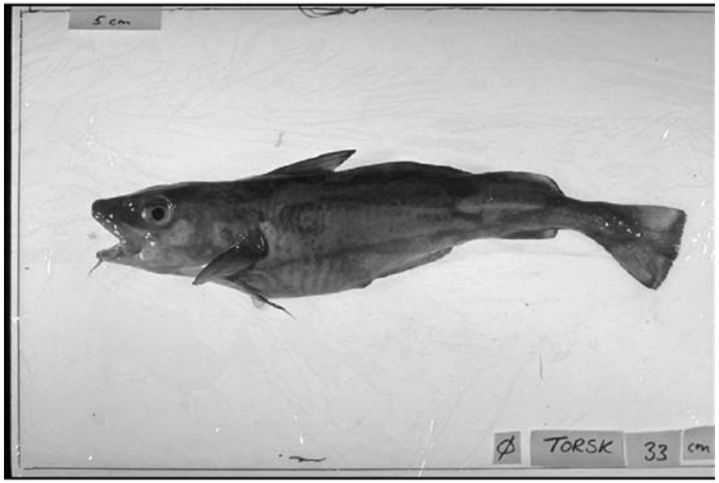


Example 2 - The images

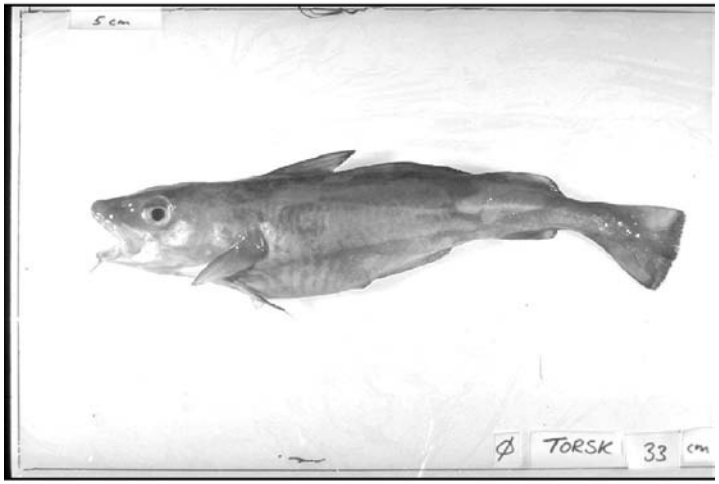
Color



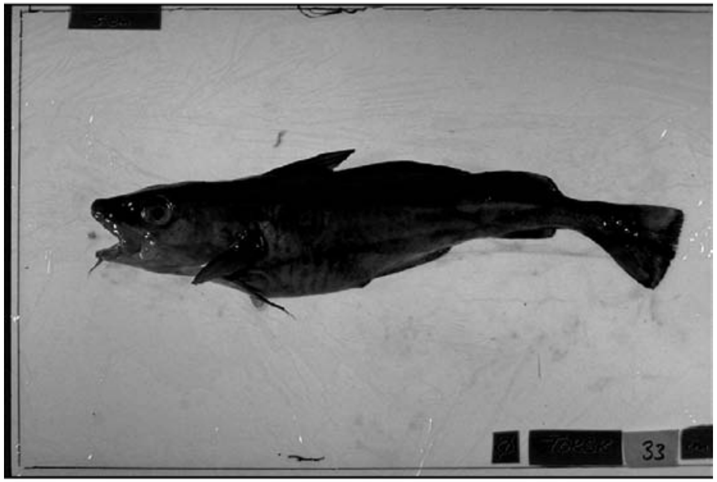
Green



Red

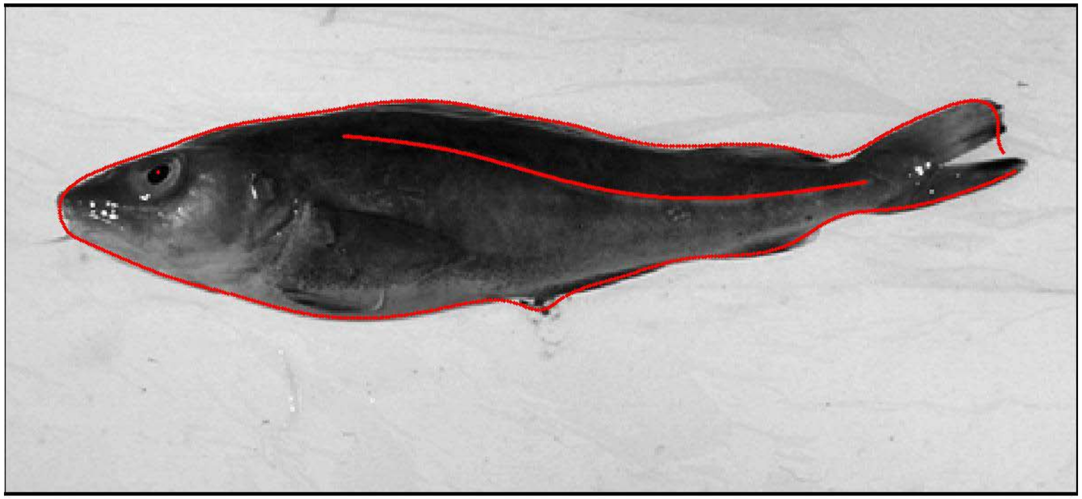


Blue



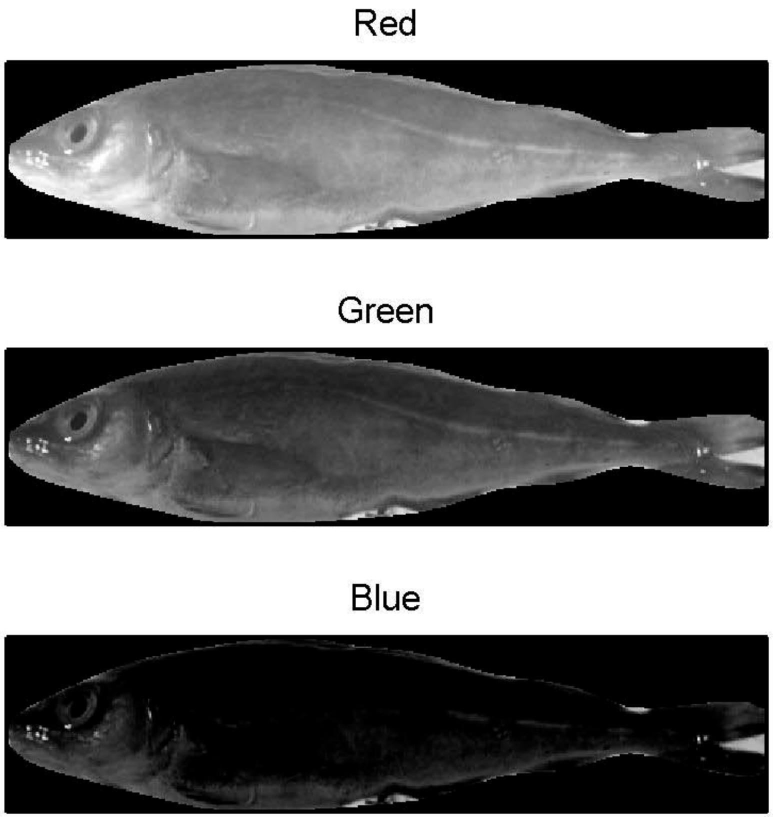
Example 2 - The shape features

- Minimum description length (MDL) landmarks: 700 points for the contour, 300 for the midline, and 1 for the eye of *(x,y)*-coordinates
- Correspondence of landmarks between fish obtained using Procrustes alignment (translation and rotation removed)



Example 2 - The texture features

- Red, green, and blue intensities (standard in computer vision)
- A total of 3 x 33782 = 101346 texture features
- The pixels were matched/annotated using a Delauney triangulation



Example 2 - Results on fish species

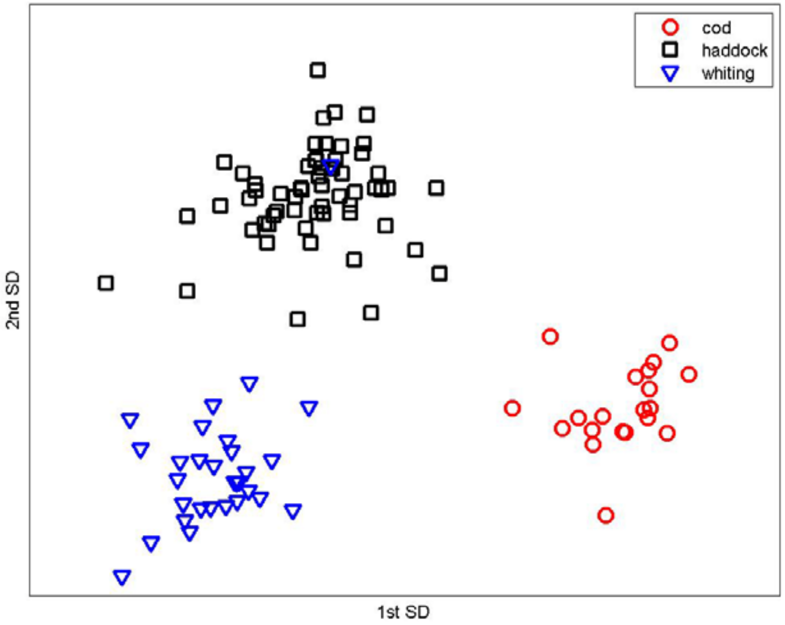
- Originally: PCA and LDA gave 76% resubstitution rate
- Comparisons with shrunken centroids regularized discriminant analysis (RDA) and sparse partial least squares (SPLS)

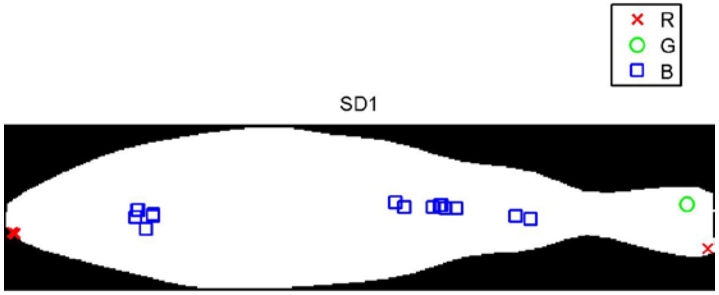
Method	Train	Test	Non-zero loadings
RDA(n)	100%	41%	103084
RDA(n)	100%	94%	103348
SPLS	100%	81%	315
EN	100%	94%	90
SDA	100%	97%	60

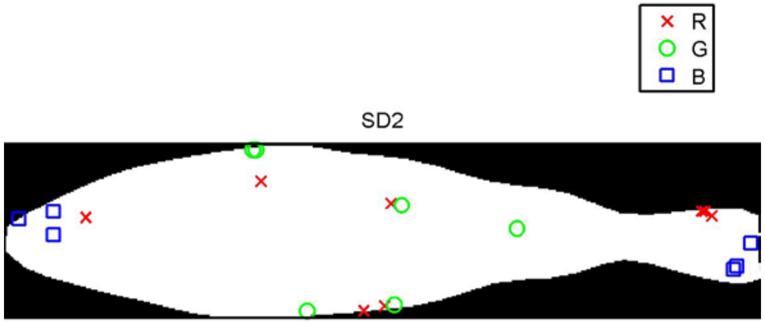
- RDA builds on the same underlying model as SDA, but a different algorithmic approach to obtaining stable and sparse solutions.
- SPLS does not build on the same underlying model, but uses the same algorithmic approach as SDA.

Example 2 - Interpretability

- The 1st and 2nd sparse directions (SDs)
- With data projected onto them
- With the selected features for each of them







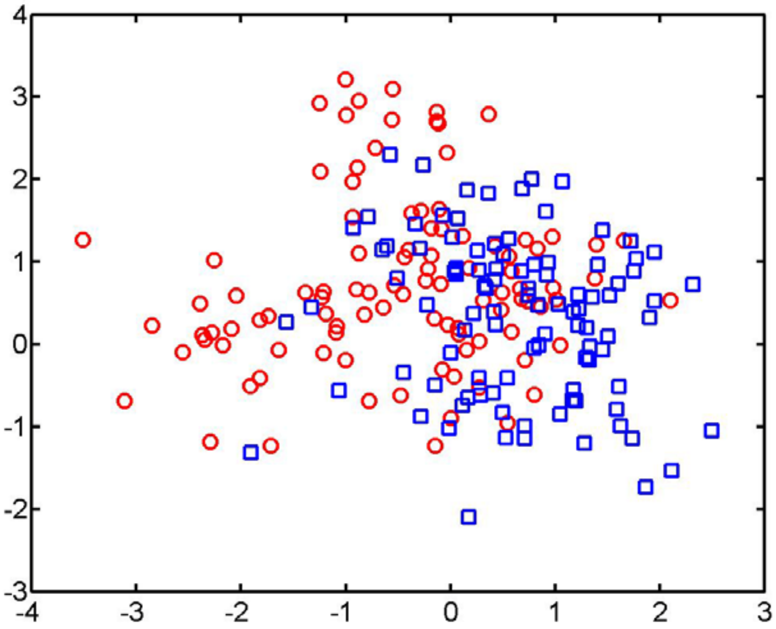
Example 3: Nonlinear boundaries and subgroups within classes

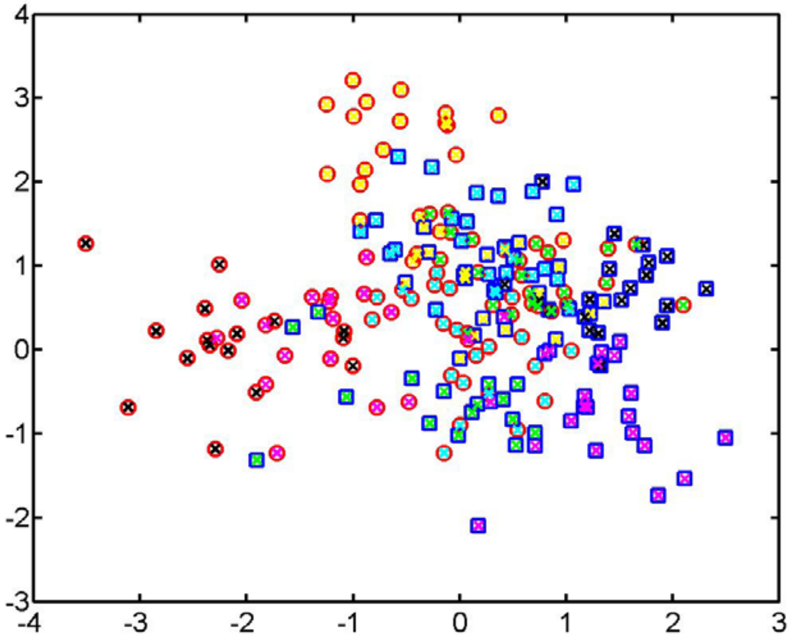
- Simulate two groups in four dimensions (each group is a mixture of five Gaussians)
  - First generate 5 means m_k from a multivariate Gaussian distribution $N((0.4,0,0.4,0)',I)$ => BLUE .
  - Another 5 means μ_k from $N((0,0.4,0,0.4)',I)$ => RED.
  - Pick at random an m_k *or* μ_k with probability 1/5, and then generate a $N(m_k,I/5)$ or $N(\mu_k,I/5)$ which leads to a mixture of Gaussians for each class.

  - Add another 196 randomly distributed variables to all observations
  - Data is now 200 by 200
  - With 2 groups and 5 subgroups for each group.

Example 3: sparse linear discriminant analysis (4 non-zero parameters, 2 SDs)

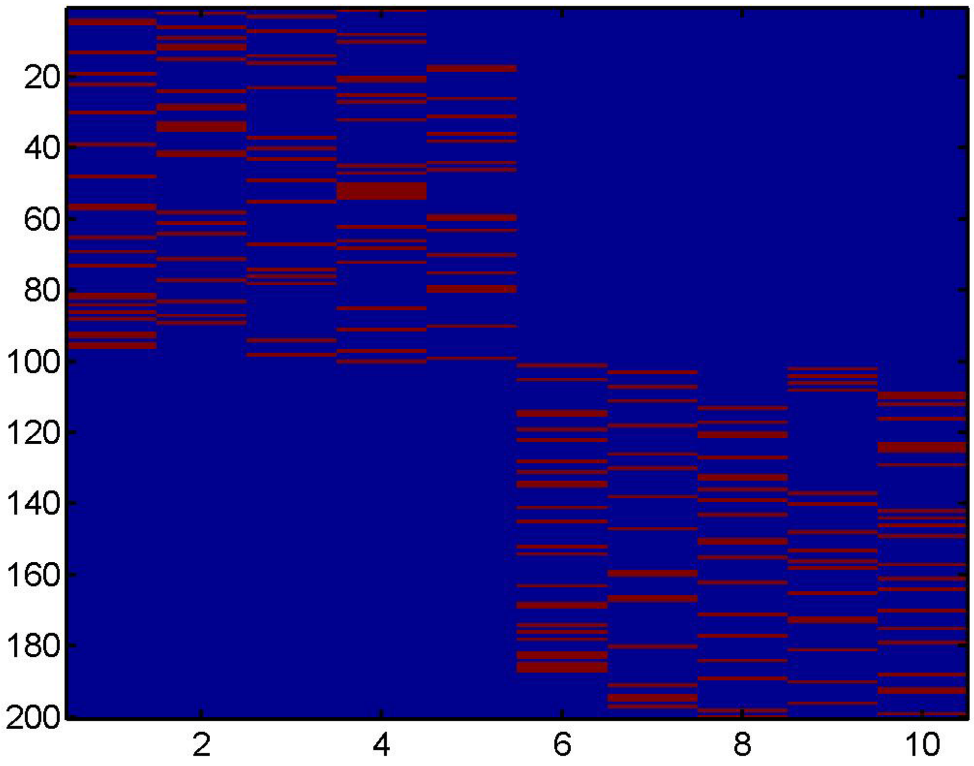
- Misclassification: 29%





Example 3: sparse mixture discriminant analysis

- Instead of dummy Y, use subprobabilities of classes (Z)
- For example:
  - true subclass = 0.96
  - true class, not subclass = 0.01



Example 3: sparse mixture discriminant analysis

- Rj = [5,5]; % number of subgroups in each class
- lambda = 1e-6; % L2-norm penalty weight
- stop = -2; % L1-norm penalty (no. of nonzero loadings)
- maxiter=20;

- [sl, theta, Znew, mu, Cov, Dp] = smda(X,Z,Rj,lambda,stop,maxiter,1,1e-8);

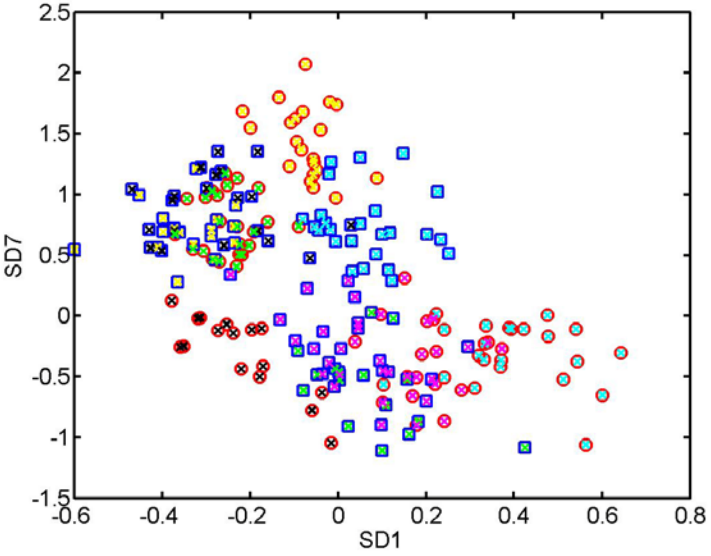
- The algorithm uses the Expectation Maximization algorithm to update Z and otherwise works like slda.

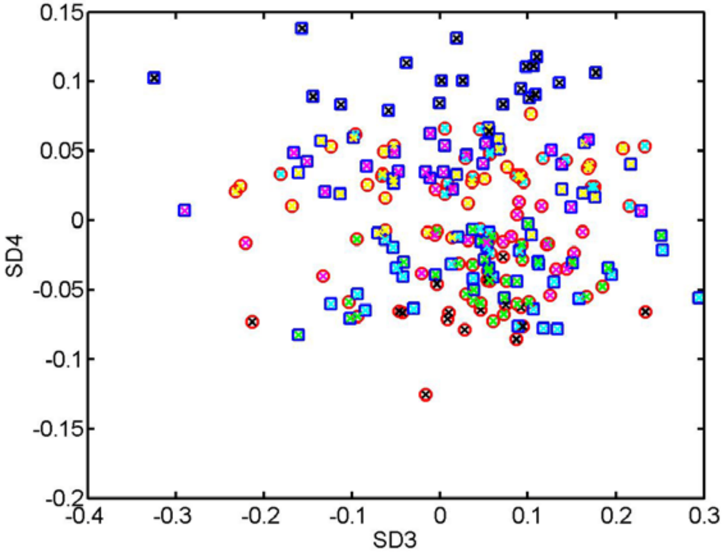
Example 3 – smda prediction

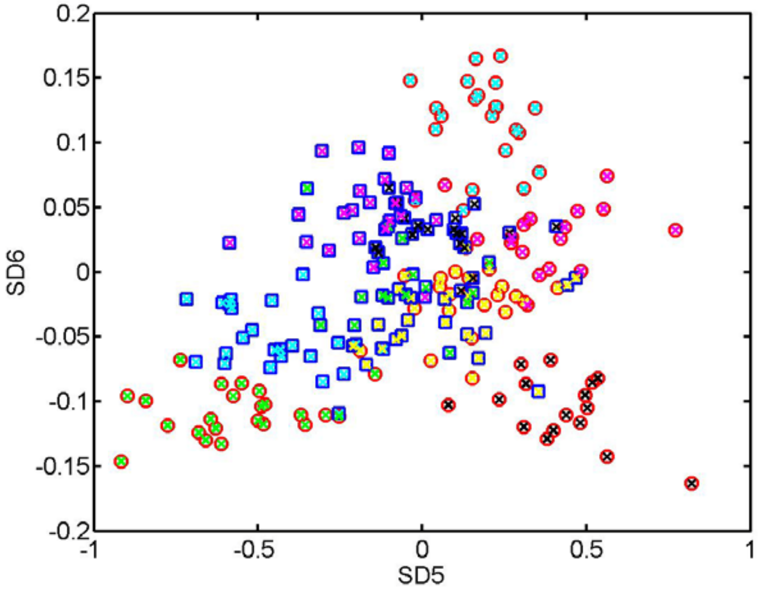
```
Rz = [0,Rj];

for ii=1:K % probability for class ii
    pr(:,ii) = sum(Znew(:,(sum(Rz(1:ii))+1):(sum(Rz(1:ii))+Rz(ii+1))),2);
end

[G, Yhat] = max(pr,[],2); % max. prob.
 err_tr = sum(Yhat~=(y+1))/length(y)
```

- Misclassification: 6%

Example 3 – smda directions







Sparse partial least squares

- Developed by Chun and Keles, University of Wisconsin, published in 2010
- Latent variables (like in PCA but correlated with **Y**)
- With sparse loadings (like in SPCA)
- Can be used for regression or classification
- Builds on the elastic net and the same algorithmic approach as both SPCA and SDA

Sparse partial least squares

- SPLS promotes the lasso zero property onto a surrogate direction vector **c** instead of the original latent direction vector **a**, while keeping **a** and **c** close (like in SPCA).

$$\min_{\mathbf{a},\mathbf{c}} \quad -\kappa\mathbf{a}^T\mathbf{M}\mathbf{a} + (1-\kappa)(\mathbf{c}-\mathbf{a})^T\mathbf{M}(\mathbf{c}-\mathbf{a}) + \lambda_1\|\mathbf{c}\|_1 + \lambda_2\|\mathbf{c}\|_2$$

$$\text{s.t.} \qquad \mathbf{a}^T\mathbf{a} = 1$$

- Where $\mathbf{M}=\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$, $0 \leq \kappa \leq 1$, and λ_2 and λ_1 are the weights on the ridge and lasso constraints, respectively.

- Solved by iterating over an elastic net regression step with **a** fixed, and an svd step with **c** fixed.

- In fact: when $\kappa=0.5$ and $\mathbf{M}=\mathbf{X}^T\mathbf{X}$ this is the same as SPCA.

References

- SparseLDA downloads are available in R and Matlab from www.imm.dtu.dk/~lhc .

- Clemmensen, Hastie and Ersbøll, Sparse discriminant analysis, DTU Informatics Technical report, 2008 (to appear in Technometrics).
- Hastie, Tibshirani and Buja, Flexible discriminant and mixture models, *In: Neural networks and statistics conference, Oxford Press*, 1995.
- Zou and Hastie, Regularization and variable selection via the elastic net, *J. R. Statist. Soc*. B(67), Part 2, 2005.
- Hastie, Tibshirani and Friedman, *The elements of statistical learning*, 2nd Ed., Springer, 2009. (Chapter 18: High-Dimensional Problems: *p>>N*)
- Chun and Keles, Sparse partial least squares regression for simultaneous dimension reduction and variable selection, *J. R. Statist. Soc.* B(72), Part 1, 2010.
- Chung and Keles, Sparse partial least squares classification for high dimensional data, *Statistical applications in genetics and molecular biology*, Vol. 9(1), 2010.