Q&A

Computational Data Analysis
2018 May 11

Feedback

- Lectures/Slides
- Exercises
- Cases
- Presentations by Maersk and Leap Beyond Group

The 15 minutes exam

- Randomly draw a week number
  - ~ 5 minutes present the week (overview, what are the essential parts, comparison of methods, important details)
  - We may ask question during this presentation to help you along
- We ask questions to other subjects
  - ~ 5 minutes where you answer questions about other methods, theory, comparisons, and application use.
- Censor and examinators evaluate performance

Assessment criteria – Randomly drawn subject

Randomly drawn week/subject	Assessment			
	Exceptional	Effective	Less effective	Poor
Ability to evaluate the subject and its context				
Presents essentials of the given week				
Presents pros and cons of the methods				
Presents important theoretical details of the essentials/ methods				

Assessment criteria – other weeks

Questions to other weeks	Assessment			
	Exceptional	Effective	Less effective	Poor
Ability to evaluate the additional subject(s) and its/their context				
Ability to give theoretical details				
Ability to compare methods				
Makes solid argumentation for choice of method(s) given a problem at hand				

Maybe the gap between the huge amount of information in the book and the light slides is too big so there is no really easygoing material.

Thank you! We have worked hard to make the slides easier to follow than the book.

What we expect you to understand is covered by the slides and exercises. Read the book or go back to your lecture notes if something is unclear in the slides.

Could you maybe go through some pointers to what is expected regarding Mortens presentations?

Unsupervised decomposition

- 4 different factorization methods
- How are they defined mathematically?
- What structures in data do they find?
- Uniqueness, sparsity, other properties?

Multi-way models

- Two models – one is a special case of the other
- How are they defined mathematically?
- One test for evaluating the number of components
- What kind of data are they usefull for?
- Missing values

How much should we be able to go into details about constraint optimization and Lagrange multipliers in regards to the support vector machines?

You should,

- Know how we handle constraints using Lagrange multipliers
- Know about primal/dual formulation
- We will not ask you to prove why it works

For the OSH/SVM we expect that you understand the steps in the derivation. We do NOT expect you to show the proof on the blackboard.

Also a brief catch up on the SVM: the support vectors, the margin etc.

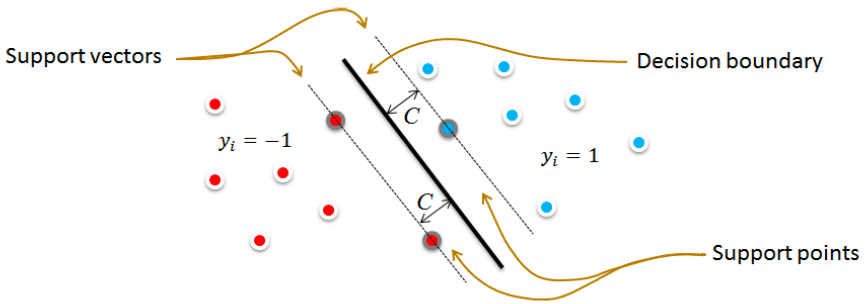
OSH as a maximization problem

We can now formulate a maximization problem

$$\arg\max_{\beta,\beta_0} C$$

such that

$$y_i \frac{x_i\beta + \beta_0}{||\beta||} \geq C \quad \forall i$$



SVM Cost Function

We got OSH from

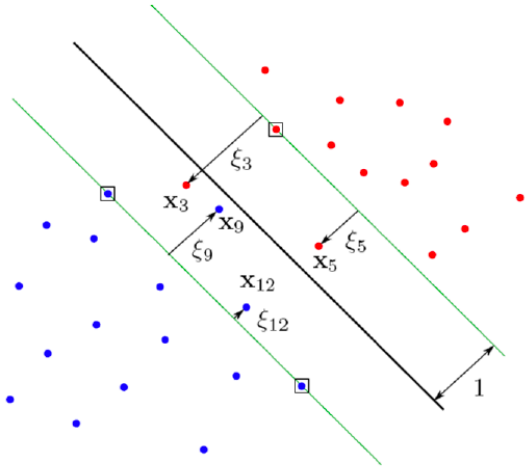
$$\begin{cases} \arg\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 \\ \text{such that} \\ y_i(x_i\beta - \beta_0) \geq 1 \quad \forall i \end{cases}$$

Now, allow some overlap

$$\begin{cases} \arg\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 + \lambda\sum_{i=1}^{n}\xi_i \\ \text{such that} \\ y_i(x_i\beta - \beta_0) \geq 1 - \xi_i \quad \forall i \\ \xi_i \geq 0 \quad \forall i \end{cases}$$

We give our self a **budget for overlap**.

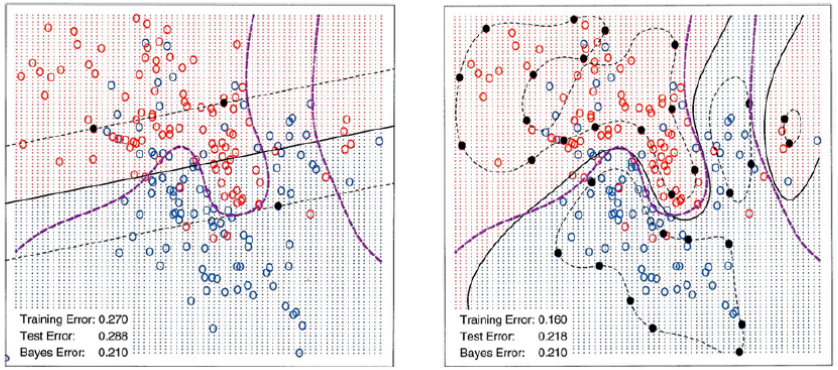
Smaller budget - larger λ - noisier solution



Also a brief catch up on the SVM: the support vectors, the margin etc.

Example

Linear SVM and enlarged feature space using RBF kernel



Maybe go into depth with the important parts instead of delving into some unnecessary specifics (eg. constrained optimization techniques for SVM's etc.) - especially if it's not essential for the course material.

- Constrained optimization is a key component in many machine learning techniques.
- We are actually using it in at least three completely different settings during the course. Which? (Two in the lectures and one in the exercises)
- This will be a great exam question – thank you!

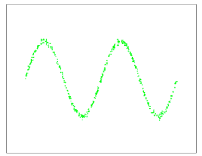
SOM, self organizing map, which were gone through very brief during the lectures.

Background

- Observations are grouped into clusters much like K-means clustering
- The clusters have neighbors in a one or two dimensional grid.
- Neighbor clusters are enforced to lay close to each other also i feature space.
- This creates a mapping of data down to one or two dimensions.

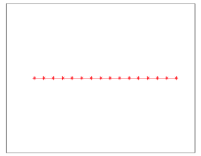
- Great for visualization
  - Exploratory data analysis
  - Overview of large amounts of text documents

Example, data on one dimensional curve



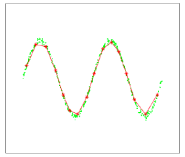
- 400 data points in a 2D-feature space
- Data are laying in a one dimensional curve (plus some noise).

Example, data on one dimensional curve



- Define a one dimensional grid
- We want to map this grid onto data
- Each node is one cluster center
- Data that are close should be clustered to the same node or nodes that are close

Example, data on one dimensional curve



- This is the Self Organizing Map
- Nodes are plotted as cluster centers in **feature space**

SOM, self organizing map, which were gone through very brief during the lectures.

Blessings of dimensionality

It's not all bad...

In 2000, Donoho pinpointed **3 blessings of dimensionality.**

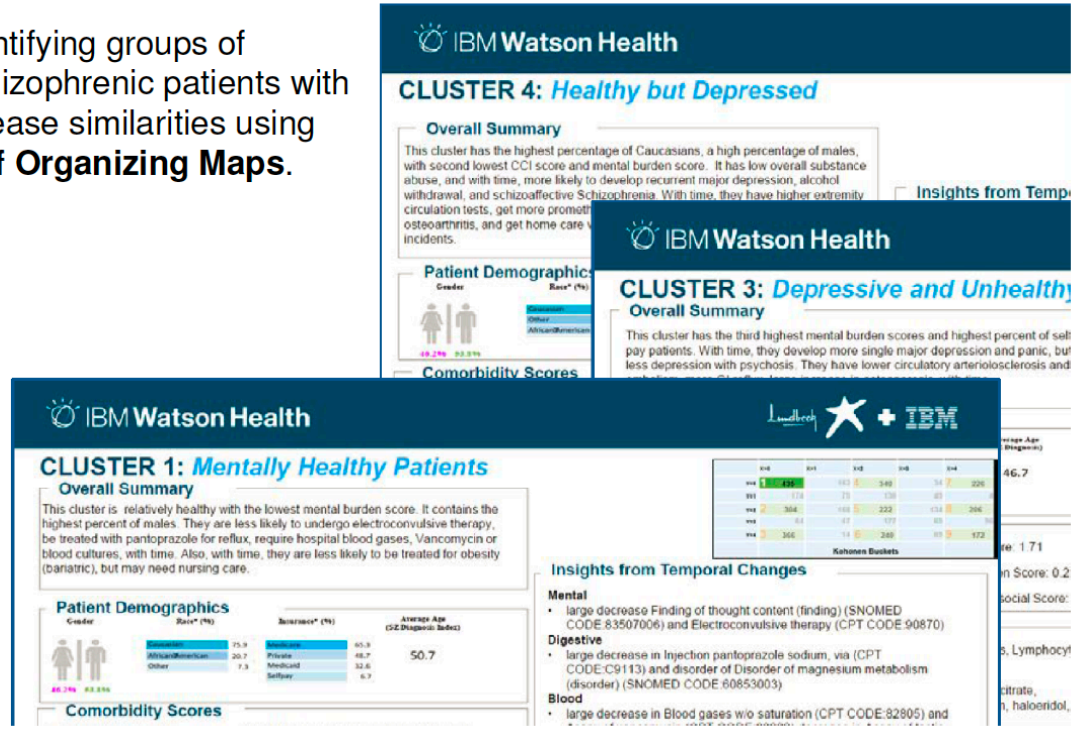
1. Several features will be correlated and we can average over them
2. Underlying distribution will be finite, informative data will lay on a low-dimensional manifold
3. Underlying structure in data (samples from continuous processes, images etc) will give an approximate finite dimensionality.

Donoho, D. L., August 2000. High-dimensional data analysis: The curses and blessings of dimensionality. In: Conf. Math Challenges of the 21st Century, Los Angeles.

SOM, self organizing map, which where gone through very brief during the lectures.

Application - Patient Segmentation

Identifying groups of Schizophrenic patients with disease similarities using **Self Organizing Maps**.



Is there another purpose of Auto Encoders than data compression?

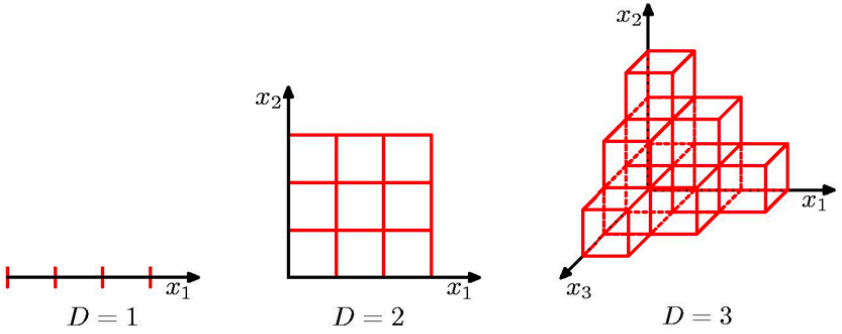
- I don’t know. I use it for data compression.

On the topic of the curse of dimensionality: If we take for example the case for a cube, it is perfectly clear why we need a larger part of the side lengths to capture a certain fraction of the data, when the number of dimensions increase. However, can you elaborate a bit on why this is so important? Why do we need to capture a certain fraction of the data?

Curse of dimensionality

What happens when the dimension of the solution space grows, ie the number of variables grows?

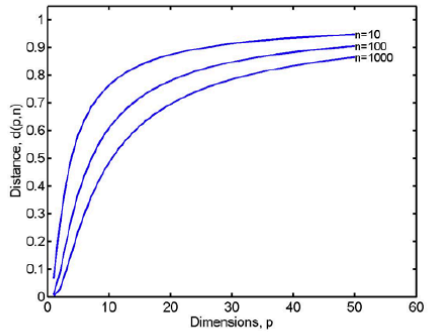
- The number of regions grows exponentially with the dimensionality D



Curse of dimensionality

For data fitted to a unit sphere the median distance from the center of the sphere to the closest point is

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



Interpolation becomes extrapolation in high dimensions

The key thing is that we need much more data to capture the structure in data when data is high dimensional.

When using information criteria, why do we use low bias models for sigma^2-estimate? Why is low bias more important than low variance here?

C_p-statistic

Given a squared error loss and a model with d variables we can calculate the so-called C_p-statistic

$$C_p = \overline{\text{err}} + 2\frac{d}{N}\hat{\sigma}_e^2$$

Using this metric we can select the best model by choosing the model that minimizes C_p.

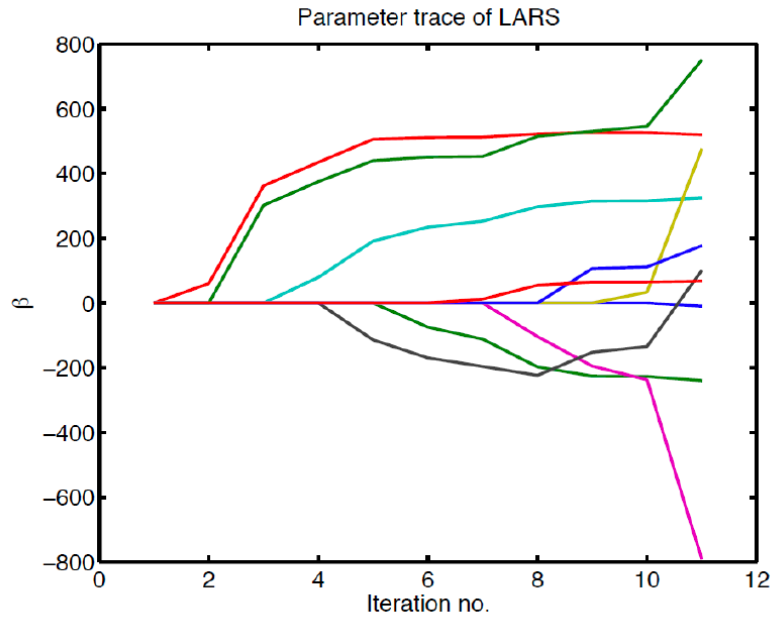
Expected training error,
use the actual training error $\frac{1}{N}\sum_{i=1}^{N}(y - x_i\hat{\beta})^2$ as an estimate.

Noise variance,
use the mean squared error (MSE) of a low bias model (OLS or KNN) as the estimate $\hat{\sigma}_e^2$.

A high bias model will have systematic error in the predictions. Systematic error in the predictions means that the MSE of the residuals will be higher than the true noise variance.

In the LARS-algorithm we increasingly introduce more features. How does this relate to the lambda-path?

Parameter trace for Diabetes example



- LARS and Lasso are not identical. A feature that enters in the LARS algorithm stays present. In Lasso a feature can enter and drop out again.
- The results are in practice very similar especially when it comes to performance of the selected models.

What is the point of the various regularizations of LDA?

Regularized discriminant analysis

It takes a lot of observations to estimate a large covariance matrix with precision. Three increasingly harsh regularizations are available

1. Make a compromise between LDA and QDA,

$$\hat{\Sigma}_k(\alpha) = \alpha\hat{\Sigma}_k + (1-\alpha)\hat{\Sigma}$$

2. Shrink the covariance towards its diagonal

$$\hat{\Sigma}_k(\gamma) = \gamma\hat{\Sigma} + (1-\gamma)\,\mathrm{diag}(\hat{\Sigma})$$

3. Shrink the covariance towards a scalar covariance structure

$$\hat{\Sigma}_k(\gamma) = \gamma\hat{\Sigma} + (1-\gamma)\hat{\sigma}^2 I$$

- Method 1 has more free parameters than method 2 which has more free parameters than method 3.
- The fewer observations we have in our data the better it is with fewer free parameters to estimate.

In SVM do we always have support points exactly on the margins? While clear in OSH, I cannot conclude it from the mathematical formulation for SVM.

SVM Cost Function

We got OSH from

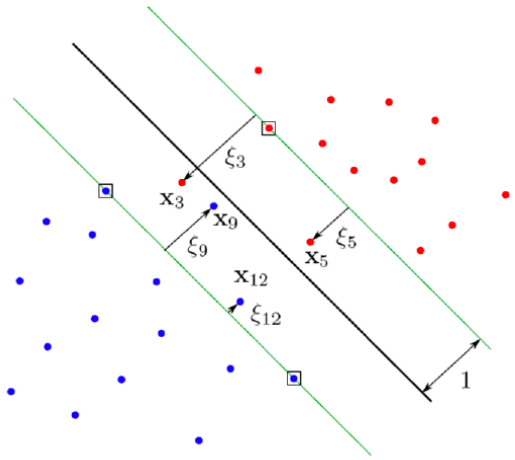
$$\begin{cases} \arg\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 \\ \text{such that} \\ y_i(x_i\beta - \beta_0) \geq 1 \quad \forall i \end{cases}$$

Now, allow some overlap

$$\begin{cases} \arg\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 + \lambda \sum_{i=1}^{n} \xi_i \\ \text{such that} \\ y_i(x_i\beta - \beta_0) \geq 1 - \xi_i \quad \forall i \\ \xi_i \geq 0 \quad \forall i \end{cases}$$

We give our self a **budget for overlap**.

Smaller budget - larger λ - noisier solution



- Yes, both OSH and SVM have support points on the support vectors.
- Some ξ will be equal to zero and for those we will have a problem identical to the OSH.

In Random Forest how/why does the OOB method for variable importance work?

Random forests and variable importance

- Two measures (the same can be done for boosting)
  - The Gini index
  - An OOB estimate
- **Gini:** The improvement in the split-criterion at each split is accumulated over all the trees for each variable.
- **OOB:** Measures prediction strength by first dropping the OOB sample down the tree, then permuting the values for the *j*th variable and computing the prediction accuracy again. An average of the difference in accuracy for all the trees gives the measure of importance for variable *j*.

We test how much worse the preditcions becomes when we scramble one feature at a time. This gives us the importance of that feature.

We use OOB samples since these were not used for building the tree.