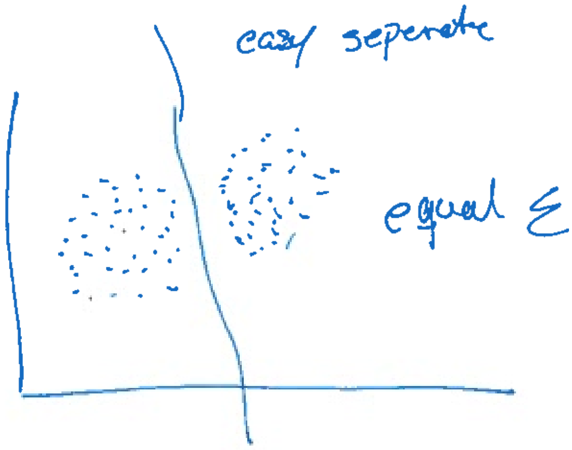
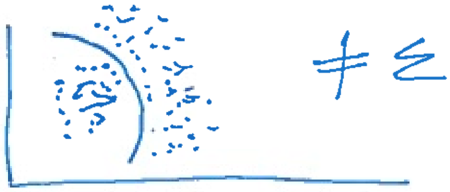
Lecture 1

LDA as in OLS finds best possible split

- Main theme:
  - Supervised vs. unsupervised and classification vs. regression.
- Mentioned methods:
  - Supervised:
    - Classification:
      - LDA, KNN, SVM
    - Regression:
      - Error: Expected Prediction Error
      - OLS, Ridge-regression, KNN
  - Unsupervised:
    ↳ KNN
  - The bias and variance tradeoff
    - High bias: under-fitting data
    - High variance: over-fitting the data





- Selected method:
  - LDA: Maximizing the separability among the known classes:
    - Class → equal covariant structure → linear decision bound **else** QDA
    - Maximize between-class variance
    - Minimize between-class variance

$$\max_{a} \frac{a^T \Sigma_B a}{a^T \Sigma_w a}$$

- Highlights
  - Class separation: Flexibel vs. Linear
- Comparison of methods:

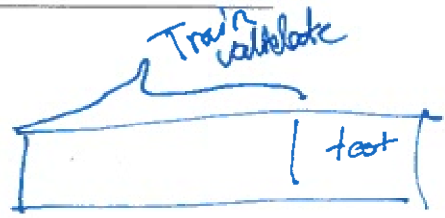
Methods	Assump				
LDA	outliers Bad				
KNN	Suggest another cluster (average within variance)				
SVM	outliers ok				
OLS					

Ridge-

Ridge regression

$$\beta = (X^T X + \lambda I)^{-1} X^T$$

Lecture 2



- Main theme:
  - Model development
- Mentioned methods:
  - Model complexity — linear vs. non-linear, no. parameters vs no. samples
  - Model selection (train and validation set) → hyperparameter selection
    - regression: lowest MSE
    - classification: lowest false discovery rate
  - Model assessment (test set)
    - Bootstrap (and out-of-back OOB) with replacement → create multiple fits → create confidence intervals → many data replicates
      ↳ estimates, parameters, ↳ predictions } → confidence int. (20,000 - 50,000 samples)
    - Performance metrics
- Selected method:
  - CV: Hyperparameter selection
    - Why: easy and simple to implement
    - How: Shuffle data before splitting into, preprocess in each fold
    - When: lots of data, NOT time dependent,
    - NB: observations are assumed to be **independent → otherwise information leak → overfitting**, Always use the "One standard error rule" → choose less complex model
- Highlights — Fold 5 or 10 → ok generalize
  - Hyperparameter selection
  - Supervised vs. unsupervised
    - Supervised:
      - CV → on standard error rule → chose a less complex model
      - Classification metrics: False discovery rate (proportion of positive which are incorrectly predicted) → use a tuning parameter — FDR
      - Regression: MSE
    - Unsupervised
      - Dissimilarity metrics e.g. KMs → average within distance
- Comparison of methods:

Methods		
CV	independent obs.	MSE misclassification rate / FDR
Bootstrap	→ uncertainty measures	
train test	times series	AIC, BIC, log likelihood

Lecture 3

- Main theme: — High dimension ⇒ regularization
  - The general topic here is when there is more variables and observations (p >> n) → Sparse regression → forcing the coefficients towards zero
  - *Multiply testing*
- Mentioned methods:
  - "The curse of dimensionality": there exist a lower dimensional manifold which captures the structures, correlation between variables → Always a lower dimensional representation of the data
  - Dimension reduction: Regularization methods

    L2 L1 L0,5

    - Ridge-regression, Lasso, Elastic net
  - Multiple hypothesis testing

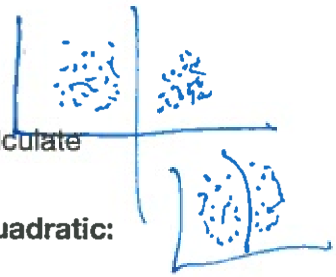
- Selected method:
  - Ridge-regression: min b st. (y - Xb)^T (y - Xb)+L * b^T * b
    - Closed form shrinkage method → computational efficient
    - NOT feature selected → only variables towards not complete zero
- Highlights
  - Properties of high dimensional problems: "Interpolation becomes extrapolation in high dimensions" → every observation is far away
  - Best practices:
    - subtract the mean and standardize the variance → no pen. of the intercept and creates equal importance of the variables
    - Use CV for hyperparameter estimation
- Comparison of methods:
  - Lasso and elastic net:
    - not closed form solution estimated by: **LARS**: correlated variables in high dimensional → no go. **Coordinate decent** → similar to gradient decent, but only optimizing for one parameter at the time.
    - As feature selectors → forcing coefficients towards zero
  - elastic net: the pros of both methods..

Methods					
Ridge-regression					
Lasso					
Elastic net					

Lecture 4

- Main theme:
  - Supervised classification: Linear classifiers and basis expansion
- Mentioned methods:
  - Linear Discriminant Analysis (LDA)
  - Logistic regression — probabilistic model
  - Basis expansion → linear estimate → non-linear decision boundaries
- Selected method:

  ↗ always found some "line"

  - Linear discriminate analysis: ~~probabilistic density function~~
    - Classes are gaussian distributed → **stochastic model** for data to calculate ~~probabilities~~, with different mean-values
    - Decision boundary: **Linear:** Common covariance matrix structure, **Quadratic:** different covariance matrix structure
    - LDA does not weight the observation far from the decision line.. This means that LDA is more prone to bad ass outliers which may affect the decision line(s)!
    - **Regularization:** Make a compromise between LDA and QDA → Shrink the covariance towards its diagonal, Shrink the covariance towards a scalar covariance structure.

don't need to be well separated

- Highlights
  - Data from different classes will overlap the
  - Logistic: boolean
    - Focus on boundary cases vs. LDA — out-lier poor
    - **NO** distribution assumptions
    - Optimize linear log-odds function directly → likelihood function → numerical solution to estimate the best set of parameters: iteratively re-weighted least squares solution.
    - In the probability domain, interpretable coefficients **(log odds)** → variable importance for separating the classes
  - Basis expansion → when data is not linear → transformation of the linear input data → Basis expansion opens for non-linear modeling of data using linear methods.

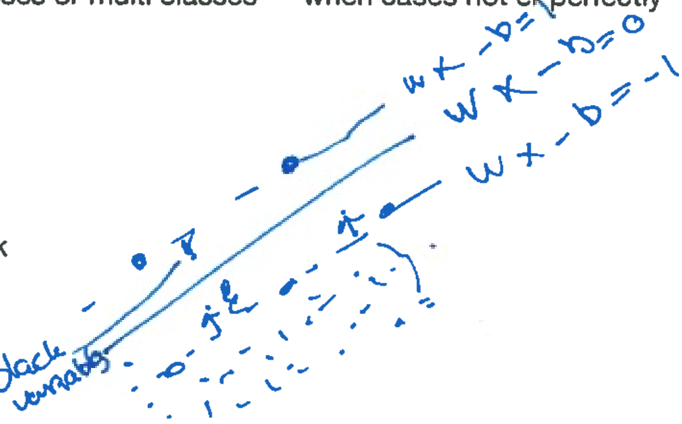
"weight" obs close decision line

- Comparison of methods:

Methods	**Linear?**	**Robust**	**N classes**
LDA	Yes		N
Logit	Yes	Better than LDA	2 (can be multiple N)
Basis	No	higher degree polynomials	

Lecture 5

- Main theme:
  - Supervised classification: Two classes or multi classes → when cases not er perfectly separately
- Mentioned methods:
  - Optimal separating hyperplanes
  - Support vector machine (SVM)
  - Basis expansion → The kernel trick
- Selected method:
  - SVM
    - Do picture on the blackboard…



    - SVM is based upon the structure of the optimal separating hyperplanes
      - Maximizing the distance between the points from either class to the decision boundary -> but allow room for overlapping in ~~the~~ margin.
      - introduce slack variables for overlapping data points.
    - The basis expansion is applicable to the SVM as well → which is the same as happened in the LDA.
      - Most common choice is the Radial Basis Functions
- Comparison of methods:

Methods	Assumptions	Theory	p > n	Features
OSH	Separable	a_i bound by lambda		
SVM	room for overlap			Basis expansion → non linear
LDA	separable			

minimize

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1-y_i(w\cdot x_i-b))\right]+\lambda\|w\|^2$$

small → hard classifier

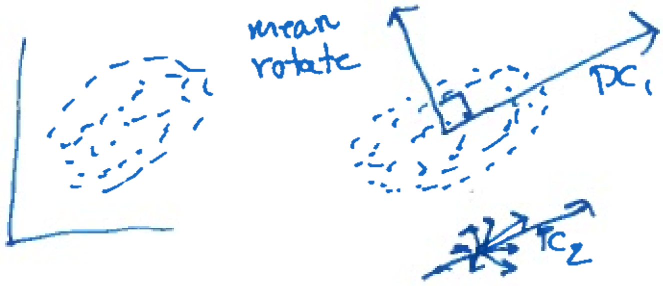
$$\min \frac{1}{n}\sum \xi_i+\lambda\|w\|^2 \quad \Rightarrow \text{primal} \xrightarrow{\text{transform}} \text{dual (max)}$$

s.t. $y_i(w x_i - b)\geq 1-\xi_i$, $\xi_i \geq 0$

when cons. convex → primal = dual

Lecture 6

- Main theme:
  - This lecture is about principal component analysis: classification, regression, dimension reduction, exploratory analysis, structure in data, outlier detection
- Mentioned methods:
  - Principal component analysis
  - Principal component regression
  - Partial least squares
  - Canonical correlation analysis



- Selected method:
  - PCR → $[U\ S\ L] = SVD(X)$
    - The principal components are used as regressors → removes the issue with multicollinearity: two or more of the explanatory variables are close to collinear (correlation is +-1).
    - Dimension reduction → you chose the amount of PC.
    - Similar performance to ridge.
    - Equivalent to OLS when choosing all PCs

scores PCA

$$Y = \beta_0 + [S_1, \dots, S_N]\beta + e$$

- Highlights
  - PCA → hard to understand the data representation
  - PCA → removes multi-colinearity (→ linearly uncorrelated)
- Comparison of methods:

Methods	Assumptions		p > n		
PCA			Yes → NB: cor		
PCR			Yes, but you to ↓ the dimensions		
PLS					
CCA	Between datasets				

↳ NMF for images

PLS: find the multi dimensional direction in X which explains the maximum multidim. variance direction in Y.

Lecture 7

- Main theme:
  - Cluster analysis → Unsupervised classification → grouping observations with same similarity → a reflection of the distance between observations.
  - Dimensionality reduction and outlier detection
- Mentioned methods:
  - Similarity measures
  - K-means (and k-medoids)
  - Hierarchical clustering: Single-linkage, average-linkage and complete-linkage.
  - Gaussian mixture models: estimated of latent variable by the EM: two step procedure → E.: defines the expectation value (conditional probabilities) of belonging to a given cluster. M.: parameter estimates of distributions (mean variance) and updating the mixing coefficients
- Selected method: iterative method
  - K-means (and k-medoids)
    - Pick K —domain knowlegde
    - Initial starting points for K
    - Selecting dissimilarity measures

- Highlights
  - Pick "K"
    - NOT cross-validation → splitting does not make sense
    - Finding the elbow, choose dissimilarity measure inside cluster (maybe log)→ statistical heuristic → Gab statistics: Within cluster dissimilarity and uniform simulations.

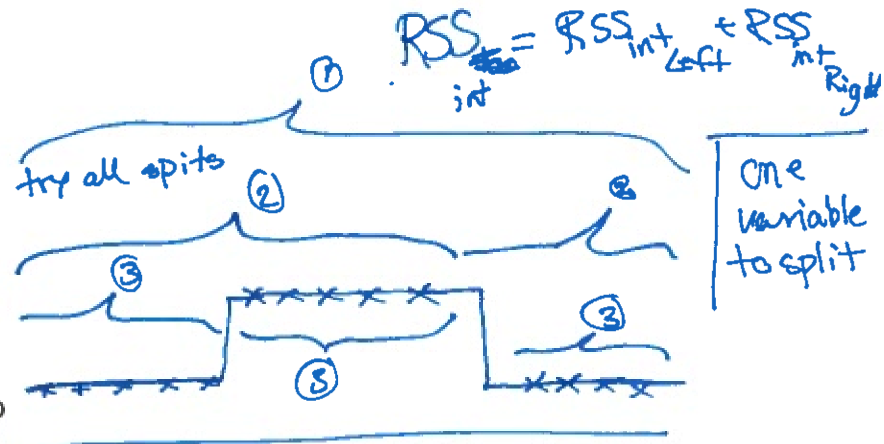
    - Gaussian mixture models: use AIC or BIC, you have the likelihood.
  - Works best for numerical attributes ->
    - Categorical values: using Hamming distance as distance metric
- Comparison of methods:

Methods	Assumptions	Theory	Features	BIG O
K-means	Numerical		p > n → OK	
Hierarchical clustering	Numerical		p > n → OK	
Gaussian mixture models	Numerical		p > n → OK	

k-medoids ⇒ pick point not average point

Lecture 8

- Main theme:
  - This lecture was about CART → classification and regression tree
- Mentioned methods:
  - Regression
  - Classification
- Selected method:
  - regression
    - Use the blackboard
    - A good split ①
    - Grow the tee and when to stop
      - 1. Stop when nodes contains < X
      - 2. Build full tree → prune the tree: e.g. Weakest-link pruning: prune branches that contribute the least to lowering RSS.
        - stop?? use iid test set and CV
    - Bias variance
      - full tree → high variance → low bias
      - Pruned tree → lower variance → low bias ③
      - Small tree → low variance → high bias ①



cat.: $2^{k-1}-1$ possible splits

min parent

you need know what the next split will provide

CV: grow full tree
fit "prune" value

- Highlights
  - Missing values:
    - if categorical: add "missing" to the categories
    - if numeric: impute mean or median or other sophisticated methods.
    - Or use a surrogate variable for the split, maybe the next most important variable.
  - Huge tree → large memory footprint
  - Interpretability is very high!! → But new data might completely change the shape
- Comparison of methods:
  - Concept is the same between classification and regression but the model error is determined different.

Methods	Assumptions	p > n	Split critation	Prune → iid test → CV
regression	Same	Yes	RSS	RSS
Classification	Same	Yes	Misclassification rate, **Gini**, cross-entropy	Misclassification rate

Lecture 9

- Main theme:
  - Multiple model fitting (ensemble methods)
- Mentioned methods:
  - (Bootstrap) → confidence intervals
  - Bagging → multiple trees → average
  - Boosting → serial models (XGboost kaggle)
  - Random forest

parallel ✓ ✓ ÷ ✓

- Selected method:
  - Random forest → an improvement of bagged trees
    - Fit many models of the bootstrap replicated data. Outputs will be aggregated.
    - High-variance and low-bias → de-correlating the tree → increase in variance!
    - The key:
      - de-correlating the trees without increasing the variance
      - random subset of variables as random candidates for splitting → reducing the number of candidates for each split will reduce the correlation between trees!
      - Few hyperparameters: pruning ceof., m_try, n_trees
    - Model selection:
      - CV for iid. test → remember to use the OOB for each tree.
      - Lots of data iid test set but the metrics are not comparable
    - NB: p > n does work → troubles when the proportion of noise variables is to high! garbage in garbage out.
- Highlights
  - Easy to use and can be parallelized
- Comparison of methods:
  - The boosting trees handles better a large number of noise variables!

Methods	
Bagging (trees)	averaging → low bias → reduce variance
Boosting trees	small trees low variance high bias
Random forrest	low variance low bias

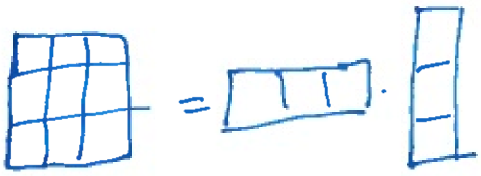
→ still correlation between trees

→ creates different splits

Lecture 10

- Main theme: NMF ≈
  - Unsupervised learning for data decomposition → find the hidden latent structure in data
- Mentioned methods:
  - Non-negative Matrix Factorization
  - Archetypal Analysis
  - Independent component analysis
  - Sparse coding

$V_{4,3} \approx H_{4,r} \cdot W_{r,3} + E$ $\quad W, H \geq 0$ $\quad r < p$



- Selected method:
  - NMF: choose "r" w.r.t. reconstruction error
    - An alternate to PCA → works perfectly for non-negative problems such as images.
    - The design parameter "r": reduction parameter.
      - if "r" = "p" → perfect reconstruction → no dimensionality reduction
    - GOAL:
      - reduce the feature space → represent it with less representable components
    - NB:
      - The non-negativ constraint and only additive operations
      - There exists many solutions: → Multiplicative updates for NMF or coordinate descent
- Highlights
  - Dimensionality reduction
  - Estimated the latent feature space
- Comparison of methods:

Estimate H, W with
↳ Multiplicative updates
↳ coordinate decent

Methods	Assumptions	p > n	App.		Design
NMF	non-negative	Yes	Images, text	Numerical	"r"
AA		Yes	Biology	Numerical	
ICP	obs. Mutual independent, NON-gauss	Yes		Numerical	
Sparse c.	D_i < 1,	Yes		Numerical	D and h

Sparse coding: optim $c(s) = |x^{(t)} - Dh^{(t)}|^2 + |h^{t}|_1$

↳ linear combination D and $h^{(t)}$ → reconstruction

Lecture 11

- Main theme:
  - Tensor decomposition → high dimensional decomposition → reduction of feature space
- Mentioned methods:
  - Tucker Decomposition
  - PARAFAC aka. SD → Tucker model → Tucker(L = M = N) → diagonal
- Selected method:
  - Tucker
    - This is a higher order SVD → SVD as n-mode multiplication
    - The solution is not unique because there can be added on invertible matrix Q
      - If the components of the Tucker decomposition are constraint to orthogonal or orthonormal → decompression of feature space
- Highlights
  - Expresses a tensor as a linear combination of simple tensors.
  - Core Consistency Diagnostic: A heuristic for evaluating the number of components.
  - Tensor vs. matrix decomposition
    - Pros:
      - Uniqueness
      - component identification even when only a relatively small fraction of all the data is observed → handle missing data
      - multi-way decomposition techniques can explicitly take into account the multi-way structure of the data that would otherwise be lost when analyzing the data by matrix factorization approaches by collapsing some of the modes
    - Cons:
      - Its geometry is not yet fully understood ?
      - The occurrence of so-called degenerate solutions → not existing solution
      - NB. Lack of guarantee of finding the optimal solution
- Comparison of methods:

Methods	Assumptions	Theory	p > n	Propreties
Tucker		HIgh dimensional SVD	Yes	
SD	Special case of Tucker(L = M = N)	HIgh dimensional SVD	Yes	uniqueness or identifiability

Lecture 12

weights = latent represent

- Main theme:
  - Artificial Neural Networks and SOM (unsupervised clustering)
- Mentioned methods:
  - Artificial Neural networks (ANN)
  - Auto-encoders (Data compression) → pre train weights.
  - Self organizing maps (SOM)

weight which are "most" similar

- Selected method:
  - Self organizing maps
    - Unsupervised clustering quite similar to the k-means algorithm. Projecting of date onto 1D or 2D feature space → dimensional reduction
    - Standardize the data : zero mean unit variance
    - SOMs are capable of doing online learning and batch-learning
    - Projection of data to a low dimensional space (Neighbor clusters are enforced to lay close to each other also in feature space).
    - How it works:
      - Determine the grid size e.g. 4x4 → 16 neurons in the hidden layer..
      - Do training in epochs -> ~~increase~~ decrease the radius for each epoch.
      - For each observation
        - Find the node closet to the given observations. compare the "weights" w.r.t. the "column" features. the distances are found by the euclidian-formulation.
        - Assign the node number to the observation and update the "weights" to match the "column" features. update the "weights" of the nodes within the radius
      - ~~Increase~~ decrease the radius and perform another epoch
- Highlights
  - Dimensionality reduction
  - Clustering and exploratory data analysis
- Comparison of methods:

Methods					

Auto encoder: lots of unsupervised data
↳ Learn the latent representations

Case 1

- Main theme:
  - The task was to build a model which can predict the response given the features.
- Ensemble methods → chosen three models
  - Models:
    - Ridge-regression
    - Lasso
    - Elastic net

    ensemble method
  - An weighted average of each prediction.. the weight for each model are derived by its relative R2 performance.
  - TRANING:
    - Hyperparameter selection: **one std. error rule** for the hyper parameters
    - 5-fold CV → 30% for test and 70% for training and validation → parameter selecting based on iid test set

    compared to 10-fold
  - ISSUES → wrong most of the time
    - Overfitting the training set → high variance → low bias → too complex model
    - Did a poor job in describing the 1000 unknown responses
- Highlights – Co working with another student → compare results
  - 100 (103) variable for 100 observations → needs dimension reduction → regularized methods
  - How to handle missing data: I did chose impute mean **but outside** the 5-fold CV!!
  - How to handle different kinds of features → one-hot-encoding aka. creating dummy variables
- Future work
  - Track the effective number of variables
  - Do some explorative analysis to begin with → PCA.
  - Use Lasso → which do parameter selection → shrinks parameters towards
  - **Bootstrap replicate** → **many model fits** → after parameter selection → create confidence intervals for performance and parameter estimates
- Comparison of methods:

Methods	**Assumptions**	**p > n**	**Pen.**		
Ridge-regression		Yes	L2		
Lasso		Yes	L1		
Elastic net		Yes	0.5 L2 + L1		

NB: did not test for duplicates ; X

Case 2

- Main theme: Our group tried two different methods:
  - "automatic feature extraction" → ANN → deep learning → Convolutional nets
  - ★ **Manuel feature extraction** → see how well the features generalizes to a different location
- Manuel feature extraction using random forest:
  - Manuel feature extraction: **Dark channel** (mean value), **Sobel filter** (variance and squared sum), **Laplace** (abs sum and variance) and pct. of **overexposed pixels**.
  - Training on Skive images → great job — m. dim. hyperparameter space
    - 5-fold CV → randomized grid search → grid search → optimal hyperparameters
    - Does a great job of describing foggy images from Billund but poor clear images
  - Random forest → an improvement of bagged trees
    - Fit many models of the bootstrap replicated data. Outputs will be aggregated.
    - High-variance and low-bias → de-correlating the tree → increase in variance!
    - The key:
      - de-correlating the trees without increasing the variance
      - random subset of variables as random candidates for splitting → reducing the number of candidates for each split will reduce the correlation between trees!
      - Few hyperparameters: pruning ceof., m_try, n_trees
    - Model selection:
      - CV for iid. test → remember to use the OOB for each tree.
    - NB: p > n does work → troubles when the proportion of noise variables is to high!
- Feature work
  - Analysis of variable importance → night images??
    ↳ discover misclassified images
- Highlights
  disclaimers
  - Images assumed NOT to be time depended
  - Duplicates are not considered → entail error in CV
- Comparison of methods:

Methods	**Assumptions**	**Complexity**	**Number of variables**	**Hyperparameters**
CONVnet	Automatic feature extraction	High	HUGE	- learning rate - Conv filter size, stride,
RF	Manuel feature extraction	Lower	4	- Pruning - m_try - n_trees