ML Bootcamp 2017-08-16/1

Dr. Mike Ashcroft, ESDA, Sweden

- Root Cause Analysis
- FinTech

Using statistics with modern IT

Features = structure file (tidy) taken from unstructured data, like pictures, Web pages, Audio, Video, Text

Select Features → Transform Features
↓ ↓
Extract Features → Select Features

Supervised : target variable known and in the data

Unsupervised : target variable not known and in the data

X = Input Features, Y = Target Variable

U

ML Bootcamp 2017-08-16/2

Regression

Target variable is a real number.

Residuals = errors of a given model

= how far off are the real values

MSE = Mean Square Errors

$$= \frac{1}{n} \sum_{i=1}^{n} (\bar{y}_1 - y_1)^2$$

n = cases = rows in dataset

→ average of the errors

Ordinary Least Square (OLS) Regression

lm = Regression line that ~~sum~~ minimizes the sum of the square residuals

Console:

? lm

?? "second phrase"

No guaranty is that the mathematics is right.

Python has a significant advantage in deep learning.

Pandas.data.frame slightly more powerful than R data.frame!

ML Bootcamp 2017-08-11/3

In klassischer Statistik würde man manuelle versuchen, die Werte zu einer linearen Regression fassen.

In Data Science, der Rechner macht das.

Eine Möglichkeit ist

POISSON Regression

(Generalized Linear Models)

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \dots$$

$$x_0 = 1$$

$$\hat{y} = b_0 x_0 + b_2 x_0 + b_3 x_0 + \dots$$

$$\hat{y} = \sum_{x=1}^{n} b_n x_n$$

$$\log(\hat{y}) = \sum_{x=1}^{n} b_n x_n$$

$$\Downarrow$$

$$f(\hat{y}) = \sum_{x=1}^{n} b_n \cdot x_n$$

$$=$$

just some sort of transformation function

ML Bootcamp 2017-08-16 / 4

the transformation is called „link function".

⇓

an aid to predict the values of y

to get the real values parameter

type = „response"

must be used.

Polynomial Regression

transform the input features

x	y
3	7
2	6
4	11
6	6

=>

x	x^2	x^3	y
3	9	27	7
2	4	16	6
4	16	64	11
6	36		

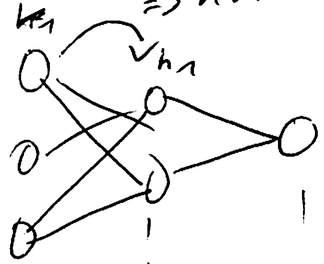
projection from 2 Dim to 3 Dim
linear model and transform it
back again to 2 Dim

ML Bootcamp 2017-08-16/5

Neural Networks

are used to get the functions for the transformations for linear models

=> non-linear function of the input features



$$h_1 = \phi\left(\sum_{x=1}^{n}\right.$$

input hidden target

$f(x)$ linear regression

weights = parameters of the transformation

coefficients = coefficients of the linear regression

weights are taken ~~by the~~ from the data!

Error Distributions

$$\hat{y} = \underbrace{f(x)}_{\substack{\text{point estimate} \\ \text{from NNET()} \\ \text{or LinReg,} \\ \text{Poisson Reg,} \\ \text{PolyReg}}} + \underbrace{N(0.6)}_{\substack{\text{Error Distribution} \\ \text{Function}}}$$

Example x = wind = 20
$f(x)$ = Point Estimate of Ozone = 7

$$\hat{P}(y) = f(x) + N(0.6)$$

Because we know that the point estimators are not correct. Is there a way to give a range of possible correct values? yes: Error Distribution around the estimate.
The error distribution is just the distribution of the residuals. We take the standard deviation (SD) of the residuals.

ML Bootcamp 2017-08-16/17

How to control complexity? Use Regularization

- make it difficult for parameters to take all values
- just bind the to a range of values
- this will prevent us from the need to throw away variables / features

$$\hat{y} = f(X, \beta)$$

β = minimize the MSE (Mean Squared Error) + sum of the absolute values

$\Downarrow$

regularisation function

L1

Do: penalise the parameter values

L1 = continuous feature selection

L1 = some parameters go to zero and stick, thus they can be thrown away

L2 = some parameters go to zero but rise again, thus all variables needs to be kept

ML Bootcamp 2017-08-16/18

Complexiness = Unregelmäßigkeit der Kurve

BSP:

nnet

decay = l (lambda) = L1/L2 regularisation für Neural Networks

How do we find the right model?

Build different models with different regularisation!

See ~~how~~ how do the models perform on the evaluation ~~and test~~ test data.

Wenn der Suchraum gefunden wurde kann Grid Search angewendet werden.

Lambda is not a parameter of the model itself but of the function that call it. It is therefore a hyperparameter. Functions will produce different models if lambda is adjusted.

ML Bootcamp 2017-08-16/9

max it => early stopping in iterations for neural networks

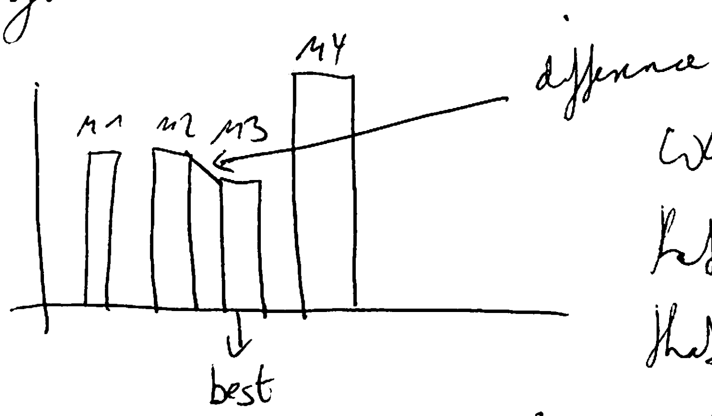
If we have more data, the best models become the more complex models.

How to split your data?

Is there a objective way to split the data?

1. Use learning graphs

2. Significance test for evaluating model performance.



What is the probability that M3 is better assuming that in the long run all models perform best.

Is M3 the best model?

ML Bootcamp 2017-08-16 / 10

ML Bootcamp 2017-08-17/1

Today: Feature Selection
Feature Transformation } Pre-Processing

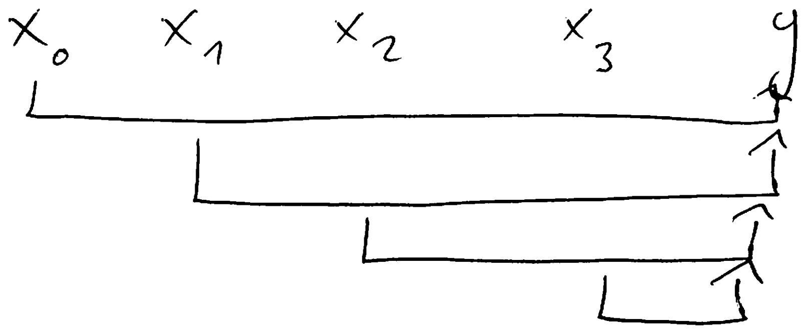
Target: ~~tti~~ Find minimum of feature with maximum information
= smallest number of most informative variables

Ways to do it:

1. Business Expert Knowledge
~~2~~ Be cautious: Experts can be wrong!
2. Pairwise statistical analysis
3. Model validation, in extreme 1 model for every ~~subset~~ subset of features
   - theoretical ideal
   - computational difficult, thus

ML Bootcamp 2017-08-17/2

Pairwise Statistical Analysis



What is the information the x_n gives to explain y?

Case	feature	target	
1	Num	Num	→ 1 Approach
2	Num	Cat	2. Approach
3	Cat	Cat	→ 3. Approach
4	Cat	Num	

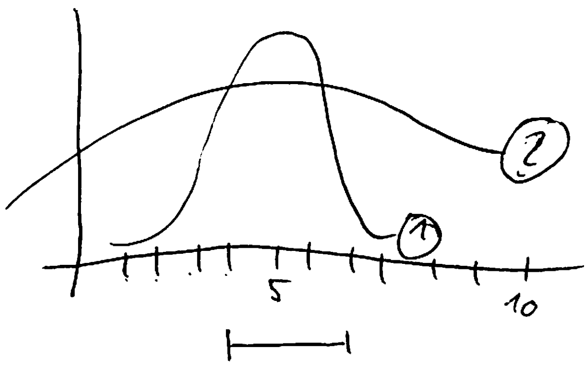
<u>Case 1</u> : (Num - Num) <u>simple correlation</u> = Pearson's corr coeff

$x \to y$
$x^2 \to y$ — for not linear correlation, ~~not dimensional~~ and take the maximum score
$x^3 \to y$

Use corr.test() in R!

Case (3): Mutual Information (Cat - Cat)

Entropy = measure of randomness



Intervall of 3

- measure the information that x and y share
- if we know (1) is weight ~~of~~ of all ~~pe~~ people and (2) is weight of Chinese people, we can reduce the value range dramatically.

Case (2) + Case (3): (Cat - Num / Num - Cat)

Conditional Distribution Divergence (CDD)

- the less the distributions overlap, the better we can estimate what the value is
- if normal distribution: mean, sd, variance
- Method: Bhattacharyya distance

ML Bootcamp 2017-08-17/4

Exercise: „Feature Selection.R"

Duncan dataset

X: Type (cat)
Income (num)
Education (num)

Y: Prestige (num)

1. first only numerical
2. bin ~~income~~ input feature
3. bin target variable

ML Bootcamp 2017-08-17/5

Feature Transformation

changing the coordinate system

(1) Center and Scaling Variables

it's important because the algorithms pay attention to the range of values which might be by accident wider or narrower.

SSD: - mean is „0".

↓ - the value is how many SD ~~the~~ the real value is away from the mean.

R::scale()

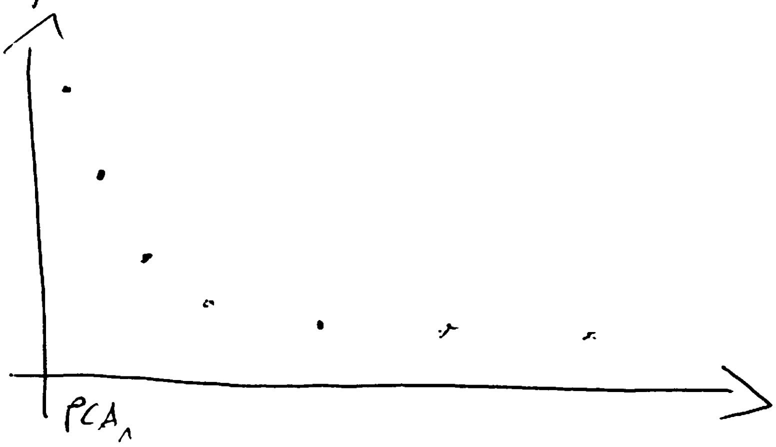
- the data cloud is the same as with original values

- all values are around „0"

ML Bootcamp 2017-08-17 / 6

② PCA = Principle Component Analysis

- Scale and center all values before using PCA !!!
- gives us the maximum variance of the variables
- R:: prcomp()
- Assumption: the more spread out the variable values are, the more information there is
- Throw away all but the first few variables of the PCA.
- PCAs contain the information how many variance = information each component has



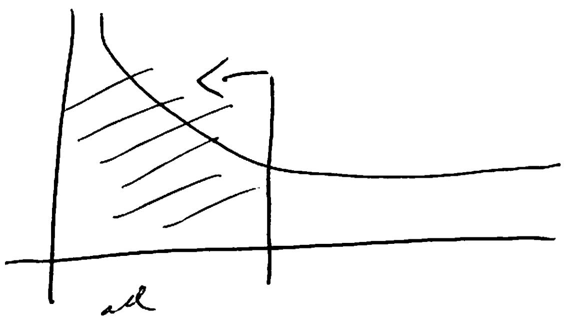
PCA_1 is the direction where to find more variance.

- Most of the time the first 10 PCA's should be chosen cause there is a quick drop out of components.

ML Bootcamp 2017-08-17 / 7

How to decide how many features to keep?

Rule of thumb: all value left of an angle of 45° of PCA curve



prcomp(scale = TRUE)

scale should always be TRUE

center = TRUE by default

predict can transform all new data according to the transformed data:

predict(myPCA, newData)

newData is transformed the same way as myPCA. Instead of myPCA any other transformation can be used.

ML Bootcamp 2017-08-17/18

Hastie, et al.:

CIT

- Elements of statistical learning
  supervised + unsupervised
- Neural Networks and learning machines
  supervised
- Pattern recognition
  unsupervised

Exercise: Feature Transformation

„pca.R“

Methods Attributes

type of input variable (nom, cat, ord, num)

type of target variable (dto.)

Assumptions   subject to overfitting

basic shape of data

concept (theory), implementations names (in R, Python, Julia, Scala, Java),

calculation procedure

outcome(s)   ggf. algorithms or links to them in other books

interpretation of values

target of the question / use case

white box / black box method

at least interpretable, could be filled also with business knowledge

when calling

ML Bootcamp 2017-08.17/9

Classification

on categorial ~~to~~ target variable

MSE = Misclassification Error

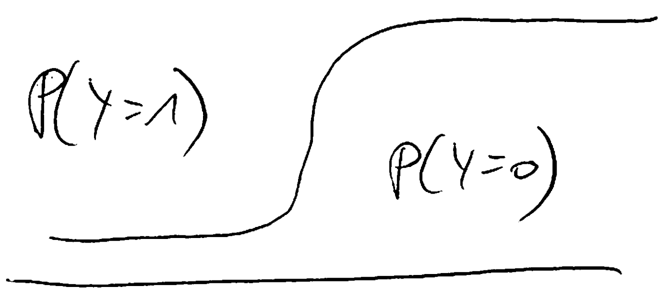
1. Technique: QDA Quadratic Discrimination Analysis

2. Technique: LDA Linear Discriminant Analysis

LDA + QDA work well on small data

3. Technique: Logistic Regression

linear classifier: where



only for use cases where y can be only „0" or „1".

ML Bootcamp 2017-08-17/10

Neural Networks: ~~&~~ sequence of linear regression

~~SVM~~ Neural Networks for binomial target variables : sequence of logistic regression

GLM : link function „binomial" instead of „poisson" for linear regression

---

Neural Networks for Classification

To Do

- Build a landscape of methods
- Craft a matrix ~~ak~~ of ~~meth~~ methods using the attributes on page 8 of 2017-08-17

ML Boot Camp 2017-08-17/11

SVM Support Vector Machines

It is easier to draw a ~~di~~ line to build a linear separal groups if there are more dimensions.

If the data is not linear separable we introduce a cost factor „c" and optimize the data from there taking into account the cost „c".

A linear kernel delivers to a separation on the original data.

Bisher habe wir „hyperparameters" behandelt habe, wo wir die Sequenze von mögliche Werte definiert habe und die Modelle mit alle ~~möglich~~ definierten Werte zu sehen und die Performance zu testen.

Da die SVM's ~~hyperpara~~ hyperparameter benötigen, kann man auf vorberechnete SVM-Modelle z.B. von Google zurückgreifen.

ML Boot Camp 2017-08-17 / 12

It is essential to select the hyperparameters very carefully.

Everything that's going on is using Euclydian distance.

In SVM is choosing the Kernel is very important as well as choosing values for the hyperparameters of the choosen kernel. It is uncommon to change the distance function for Euclydian to something ~~ds~~ else although it would be possible.

ML Bootcamp 2017-08-17/18

Exkurs:

Techniques für Time Series Data

simple

1. ARIMA
2. HIDDEN MARKOV MODELS (HMMS)

3. Dynamic Baysian Network → white box technique
4. Recurrent Neural Nets → black box techniques

complex

If you want to be sure to work right on real data use 3 or 4. If you need to put ~~it~~ in business expert knowledge ~~…~~ to it or need to interpret the results the only model is 3 Dynamic Baysian Networks.

ML Bootcamp 2017-08-17/14

Confusion Matrices, Precision, Recall

used ~~to~~ to evaluate how well a model is doing as an alternative with MSE Mean ~~Squan~~ Error.

Confusion matrix, Balanced ~~Accuracy~~ Accuracy + ROC Curves

"classification. Ra"

ML Bootcamp 2017-08-17/15

Decision Trees

- Decision Trees could not be used on real-world data.
- Cause they perform too bad.
- There are more useful methods like

  a) bagging

  b) boosting

- These other methods build on "Decision Trees" though.

(+) for trees we do not need to do any feature selection

Tree bagging

- We reduce the variance and not ~~effecting~~ effecting the bias.
- If we to reduce the correlation between trees we are pushing down the variance in all trees.

⇓

Random Forest (Forest = bunch of decision trees)

- Trees only on a small random sample of variables.
- This way we get very different trees.

ML Bootcamp 2017-08-17/16

- Random Forest perform pretty well.
- By using more trees, Random Forest does not overfit.

Evaluation:
~~Test~~s _ Out pret-bag error estimate to ~~&~~ test data

↳ we do not need to split between training and ~~&~~ evaluation data

(+)
- only one parameter to tune, which can be done using a function ↳ tuneRF()
- simple to do and understand
- much more simple than Neural Networks or SVM, although NN and SVM might perform better, but fine tuning takes a long long time.
- can be massively ~~parallisted~~ parallized

(−)
- you need feature selection cause it looks only at a small subset of feature. So if not done a lot of trees will be build of data with no information, thus producing a lot of ~~noice~~ noise.

(!)

ML Bootcamp 2017-08-17/17

Boosting with AdaBoost

- increasing the number of trees will lead to overfitting !
- can not be parallelized, cause it works sequentially

- a good package for boosting is "gbm" package