Data Mining and Visualization – Introduction

August 29, 2019

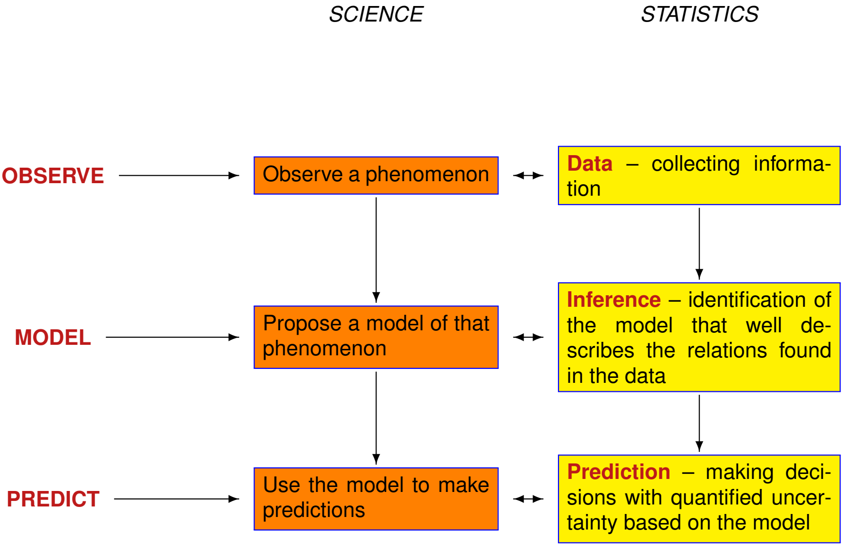
Motto

Nothing is more practical than a good theory.

Vladimir Vapnik[*]

[*] in *Statistical Learning Theory*. John Wiley, New York (1998)

What is statistical learning?



SCIENCE

STATISTICS

OBSERVE

Observe a phenomenon

Data – collecting information

MODEL

Propose a model of that phenomenon

Inference – identification of the model that well describes the relations found in the data

PREDICT

Use the model to make predictions

Prediction – making decisions with quantified uncertainty based on the model

Examples of successful statistical learning

- Analyzing the relation of volatility of stock price vs. daily shocks on financial markets.
- Predicting the occurrence of high concentrations of harmful algae in rivers based on chemical properties of water samples, the river characteristics and related historical frequencies of occurrences of algae.
- Predict whether a patient, hospitalized due to a heart attack, will have a second heart attack. The prediction is to be based on demographic, diet and clinical measurements for that patient.
- Predict the price of a stock in 6 months from now, on the basis of company performance measures and economic data.
- Identify the numbers in a handwritten ZIP code, from a digitized image.
- Estimate the amount of glucose in the blood of a diabetic person, from the infrared absorption spectrum of that persons blood.
- Identify the risk factors for prostate cancer, based on clinical and demographic variables.

How statistical data mining different from statistics?

Similarities

- Statistical data mining can be viewed as a part of statistics since it is based on the same fundamental scheme of inference:
  Data → Model → Prediction
- Data having some uncertainty due to randomness are at the center of interest, and the goal is to obtain some model that quantifies this uncertainty
- Statistical data mining uses probability modeling as its methodological foundation and for quantification of the conclusions – this differs it from data mining as understood by a computer analyst

Differences

- Statistics assumes a **mathematical model** in advance (either by understanding the mechanism that governs 'production' of the data or from experiences with similar data).
- Statistical data mining does not assumes any model but runs an algorithm that is searching for a model – **learning process**.
- In statistics, the model is verified by comparing properties of the data relatively to the fit with the **mathematical properties** of the model.
- In **data mining aka machine learning**, algorithmically found models are often not described analytically but they are verified by running them on **additional data sets** to get validation of the obtained fits.
- The conclusions following from the data mining are more driven by particular data set and frequently do not easily generalize to other even similar situations.

Big Data techniques: data mining/machine learning

- **Big Data** is a term that is very frequently (ab)used, when describing the data mining techniques
- What is big about data analyzed by data mining?
- Big because there is a lot of them – large sample size
- Big because the data have high dimension – the curse of dimensionality
- Big because there are complicated – unknown and complex structure

Often there are big for all these reasons

Learning from the data – data mining jargon

The following may help to read texts on data mining:

- Data: outcomes (we wish to predict) and features (prediction will be based on)
  - outcome: heart attack/no heart attack
  - features: diet/clinical measurements
- Training set: available data set for which both outcomes and features are given based on them a prediction model will be searched for
- Learner: the proposed prediction model for predicting outcomes for new objects based on their features

Similarity with regression terminology

- There is a strong conceptual similarity to a classical regression although the jargon is different
- Data: outcomes vs. responses, dependent variables and features vs. regressors, explanatory variables, independent variables pause
- learner vs. regression fit

The essential difference is not in the different terminology but in that the statistical learning methods need a training set and validation sets to assess the fitted models.

Supervised vs. non-supervised learning – jargon continued

- The previous scheme is referred to as a supervised learning as the presence of outcome variable can guide ('supervise') learning
- If there is no outcome and the aim is for more descriptive analysis of the data (creating clusters, organizing data), then we talk about non-supervised learning.
- Non-supervised learning is methodologically less developed.
- Our focus will be on supervised learning.

What we will not do.

- The true big data analytics works with huge data sets that require special computing techniques to even access them (cloud computing).
- **INTERNET OF THINGS**
- There are dedicated programs to handle such problems: APACHE SPARK.
- We will not work with this important part of Big Data Analytics due to time constraints.
- We focus on principles of analysis not on principles of handling and accessing data.
- Consider this course as an umbrella course which overviews the main methodological topics that constitute the core of this newly emerging field of data mining and machine learning.

Email spam – a classification problem

- Training data: 4601 email messages the true outcome (email type) email or spam is available, along with the relative frequencies of 57 of the most commonly occurring words and punctuation marks
- Objective: automatic spam detector – predicting whether the email was junk email aka spam
- Supervised problem: the outcome is the class (categorical) variable `email/spam`.
- Classification problem: the outcomes are discrete (bi-) valued

Classifier: which features to use and how

- Average percentage of words or characters in an e-mail message:

	george	you	your	hp	free	hpl	!	our	re	edu	remove
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	0.51	0.13	0.01	0.28
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	0.18	0.42	0.29	0.01

- Learning method has to decide which features to use and how
- We might use the rule:

```
if (%george < 0.6) & (%you > 1.5) then spam
else email.
```

- Another rule might be:

```
if (0.2  %you  0.3  %george) > 0 then spam
else email.
```

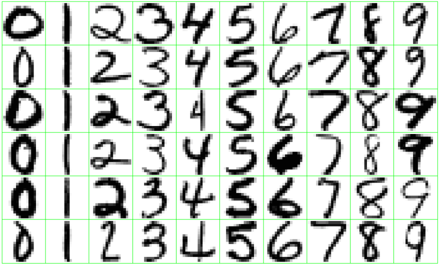
- The problem is not ‘symmetric’: we want to avoid filtering out good email, while letting spam get through is not desirable but less serious in its consequences

Prostate cancer – regression problem

- Correlation between the level of prostate specific antigen (PSA) and a number of clinical measures in 97 men
- Objective: predict the log of PSA (lpsa) from a number of measurements (features)
- Features: log cancer volume (lcavol), log prostate weight lweight, age, log of benign prostatic hyperplasia amount (lbph), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason), and percent of Gleason scores 4 or 5 (pgg45)
- Supervised learning problem, known as a regression problem – the outcome is quantitative.

Handwritten digit recognition

- Handwritten ZIP codes from U.S. postal mail – images of a single digit isolated from a ZIP code.
- The images are 1616 eight-bit grayscale maps, with each pixel ranging in intensity from 0 to 255

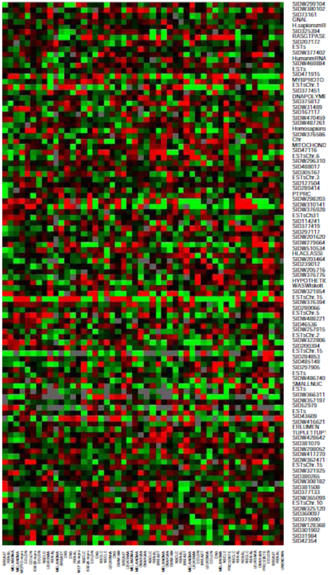


- The task is to predict the identity of each image (0, 1, . . . , 9) quickly and accurately to sort automatically envelops.
- A classification problem – the error rate needs to be very low.

DNA microarrays – genomic technology

- Nucleotide sequences for thousands of genes – printed on a glass slide.
- A target sample and a reference sample DNA are labeled with red and green dyes, and each are hybridized (put into active state) with the DNA on the slide.
- Log (red/green) intensities of hybridizing at each gene site is measured – relative activity of genes.
- The result – a few thousand numbers measuring the expression level of each gene in the target relative to the reference sample.
- The numbers are presented in the heat map (red for large values and green for small values – thus coloring coresponds to the orignal dyes) A gene expression dataset collects the expression values from a series of DNA microarray experiments:
    - each column – an experiment,
    - several thousand rows – individual genes,
    - tens of columns – samples

Example – human tumor data



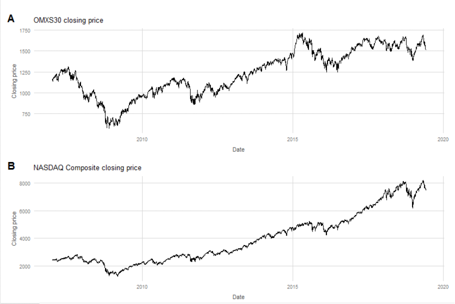
- In the particular example there are 6830 genes (rows) and 64 samples (columns), only a random sample of 100 rows are shown.
- The figure displays the data set as a heat map, ranging from green (negative) to red (positive).
- The samples are 64 cancer tumors from different patients.

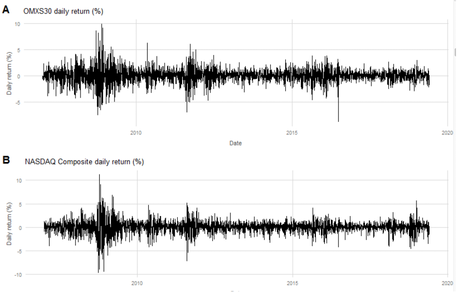
Example of unsupervised learning problem – human tumor data

The new technology can be utilized for various purposes in genomics studies. For example in a genetic disease example one can investigate the role of gene groups.

- The challenge can be to understand how the genes and samples are organized. Typical questions:
  - which samples are most similar to each other, in terms of their expression profiles across genes?
  - which genes are most similar to each other, in terms of their expression profiles across samples?
  - do certain genes show very high (or low) expression for certain cancer samples?
- These kind of problems can be viewed as unsupervised learning – no prediction variable

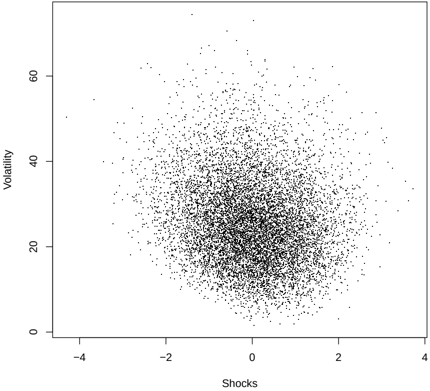
Financial data studying ‘volatility smile’





- What is volatility? The daily price shocks enter the return data multiplied by a random scale called volatility.
- GARCH and similar models allow to disentangle volatility from shocks.
- Studying their relation is very important for a financial analyst.

Data: 10000 volatility-shock pairs

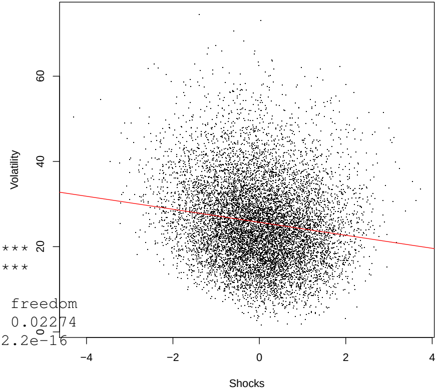


- Is there any visible relation between the two?
- Is the volatility larger when shocks are large or when they are small?
- When the volatility is larger: for positive or negative shocks?
- Which of the classical statistical methods could be applied here?

Regression analysis

Making a regression fit:

```
rf=lm(VDVolatility˜VDShocks)

summary(rf)
Call:
lm(formula = VDVolatility ˜ VDShocks)

Residuals:
    Min       1Q   Median       3Q      Max
-24.045  -7.182  -0.964    5.913  47.357

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 25.72440    0.09967  258.09   <2e-16 ***
VD$Shocks   -1.52567    0.09980  -15.29   <2e-16 ***
---
Residual standard error: 9.967 on 9998 degrees of freedom
Multiple R-squared:  0.02284,Adjusted R-squared:  0.02274
F-statistic: 233.7 on 1 and 9998 DF,  p-value: < 2.2e-16

abline(rf,col='red')
```



Does the regression line really represent the relation? Is it a reasonable relation?
People react nervously on negative news and thus volatility (risk) is higher

Regression tree: a data mining/machine learning approach

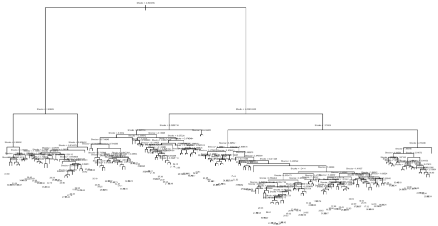
Let us choose three subsets from our data

- **Training sample** - the one on which we will learn something: 50% of the data.
- **Validating sample** - the one on which the choice of the method will be validated: 25% of the data.
- **Testing sample** - the one on which the chosen method will be evaluated: 25% of the data.

```
n=nrow(as.matrix(VD))
```

Choosing **regression tree** as a Machine Learning approach

```
install.packages("tree") #Only if it was not insta
library(tree)

help(tree)

#Training phase
tree.volatility=tree(Volatility~Shocks,VD,mindev = 0.0003,subset=1:floor(n/2))
plot(tree.volatility)
text(tree.volatility,adj=c(0.5,5.5),cex=0.5)
```

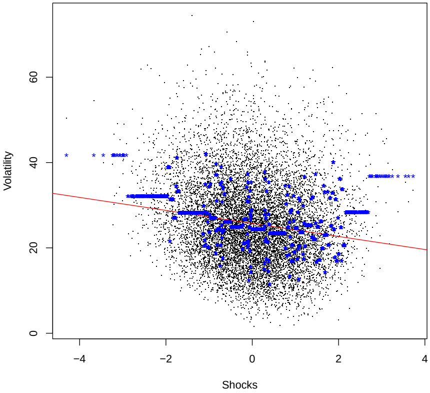


What do the trees tell us?

- We used **training sample** to bild a tree.
- What does this tree tell us?

```
yhat=predict(tree.volatility,newdata = VD)
sqrt(mean((VD$Volatility-yhat)^2))
9.884492
```

- The above standard error compares favorably to the standard error to the simple regression fit, which was 9.967.
- Is it a fair comparison?
- We see that our tree rather overfit the data. How to compare it without having an overfit criticism?
- Check it on a ‘fresh’ set of the data, i.e. the one that was not used to create the fit. For example, the validating sample.



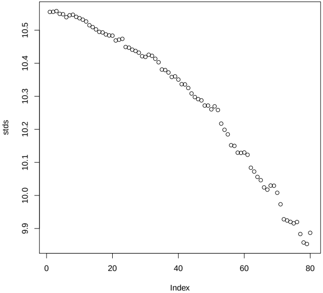
```
VVD=VD[validate_ind,]
yhat2=predict(tree.volatility,newdata =VVD )
sqrt(mean((VVD$Volatility-yhat2)^2))
11.08066
```

- We see that the performance is actually worse!
- Why? Explain!

Some trimming/prunning of the tree is needed

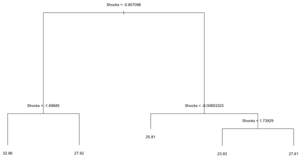
We trim the original tree hoping to get the better fit (pruning and checking if the tree produces better 'fruits').

```
pt=prune.tree(tree.volatility)
K=length(pt$k); stds=vector('numeric',K-3)
for(j in 2:(K-2))
{
  pts=prune.tree(tree.volatility,pt$k[j])
  yhat2=predict(pts,VVD)
  stds[j-1]=sqrt(mean(( VVD$Volatility-yhat2)^2))
}
plot(stds)
```



The smallest standard deviation (9.853157) with $k = k[80]$, use this pruned tree.

```
j=80 #optimal
pts=prune.tree(tree.volatility,pt$k[j])
plot(pts)
text(pts,adj=c(0.5,0.5),cex=0.5)
```



The above standard error for such a simple tree compares again favorably to the standard error to the simple regression fit, which was 9.967. Is it a fair comparison?
Test it on a 'fresh' set of the data: the testing sample.

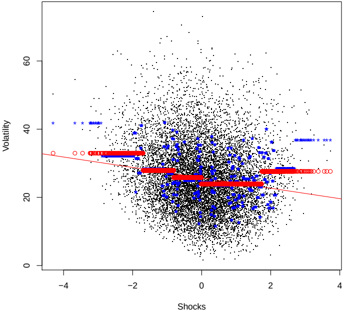
```
#Testing phase
test_ind=(floor(3*n/4)+1):n; TVD=VD[test_ind,]
yhat3=predict(pts,TVD)

sqrt(mean((TVD$Volatility-yhat3)^2))
9.900784
```

We see that the performance is actually better than the simple regression fit.

Final fit comparison – volatility smile

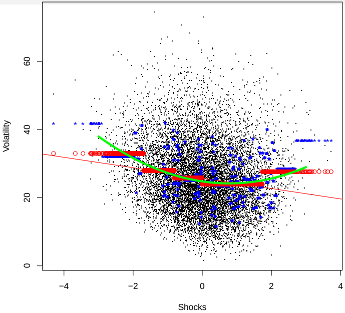
The illustration of the fits: 'volatility smile' with the negative news having larger volatility (risk).

```
plot(VD,pch='.')
abline(rf,col='red')
points(VD$Shocks,yhat,pch='*',col='blue')
points(VD$Shocks,yhat2,col='red')
```



One can try regress against the quadratic function.

```
X=cbind(VD$Shocks,VD$Shocks^2)
rf1=lm(VD$Volatility~X)
a=rf1$coefficients
lines(xx,a[1]+a[2]*xx+a[3]*xx^2,col='green')
```



```
summary(rf1)
lm(formula = VD$Volatility ~ X)
Coefficients:
            Estimate Std. Error  Pr(>|t|)
(Intercept) 24.76337    0.12122  <2e-16 ***
X1          -1.50051    0.09890  <2e-16 ***
X2           0.96377    0.07049  <2e-16 ***
---
Residual standard error: 9.875
```

We see that the performance by the quadratic function is better 9.875 vs 9.900 but the machine learning has learned itself about the shape, while the quadratic function form had to be guessed.