Computational Statistics - Summary

Lorenz Walthert

Contents

Chapter 1

Hello World

This is a summary of the class computational statistics at ETH Zurich.

Chapter 2

Multiple Linear Regression

Chapter 3

Nonparametric Density Estimation

Chapter 4

Nonparametric Regression

Chapter 5

Cross Validation

Chapter 6

Bootstrap

- Bootstrap can be summarized as “simulating from an estimated model”
- It is used for inference (confidence intervals / hypothesis testing)
- It can also be used for estimating the predictive power of a model (similarly to cross validation) via out-of-bootstrap generalization error

6.1 Motivation

Consider i.i.d. data.

$$Z_1, ..Z_n \sim \; P \; \; with \; \; Z_i = (X_i, Y_i)$$

And assume a statistical procedure

$$\hat{\theta} = g(Z_1, ..., Z_n)$$

$g(\cdot)$ can be a point estimator for a regression coefficient, a non-parametric curve estimator or a generalization error estimator based on one new observation, e.g.

$$\hat{\theta}_{n+1} = g(Z_1, ..., Z_{new}) = (Y_{new} - m_{Z_1, ..., Z_{new}}(X_{new})^2$$

To make inference, we want to know the distribution of $\hat{\theta}$. For some cases, we can derive the distribution analytically if we know the distribution P. The central limit theorem states that the sum of random variables approximates a normal distribution with $n \to \infty$. Therefore, we know

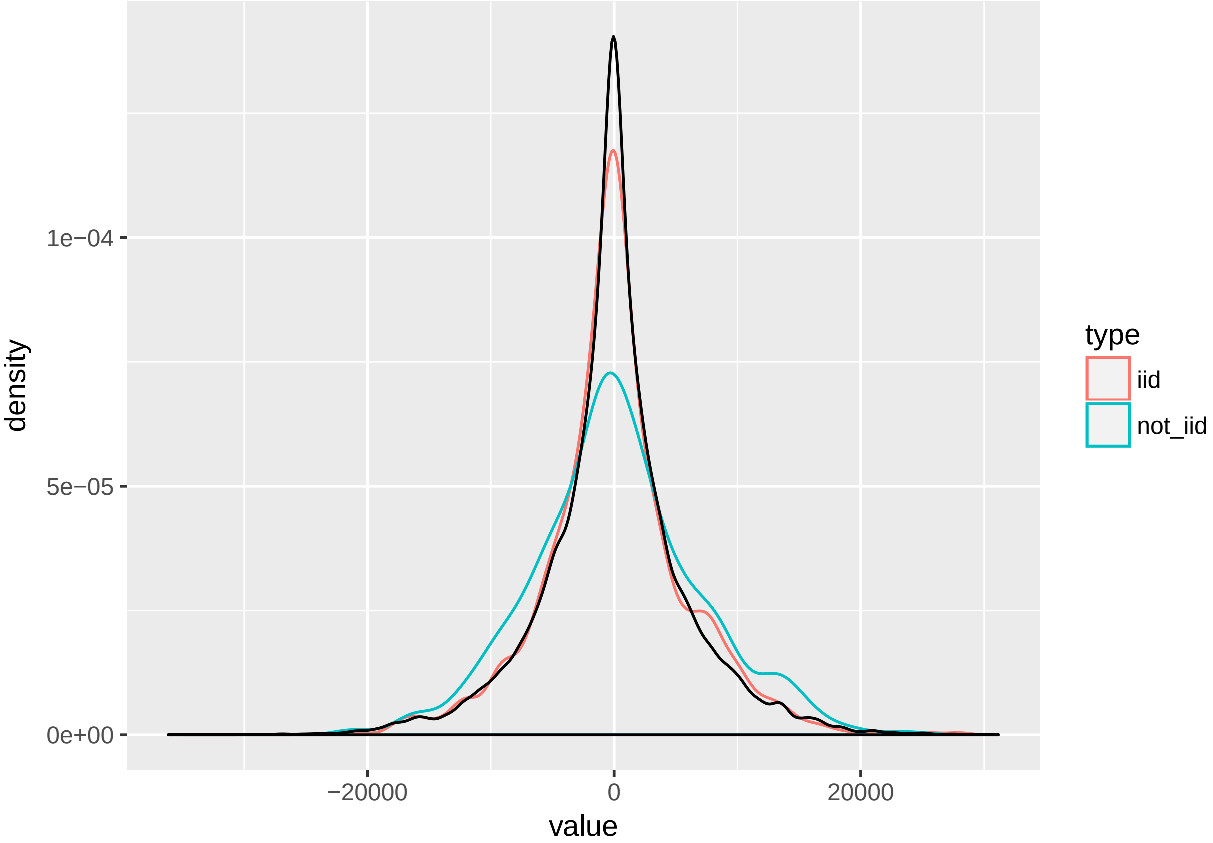
$$\hat{\theta}_{n \to \infty} = n^{-1} \sum x_i \sim N(\mu_x, \sigma_x^2/n)$$

for *any* P. However, if $\hat{\theta}$ is not a sum of random variables, and the CLT does not apply, it’s not as straightforward to obtain the distribution of $\hat{\theta}$. Also, if P is not the normal distribution, but some other distribution, we can’t find the distribution of $\hat{\theta}$ easily. The script mentions the median estimator as an example for which the variance already depends on the density of P. Hence, deriving properties of estimators analytically, even the asymptotic ones only, is a pain. Therefore, if we knew P, we could simply simulate many times and get the distribution of $\hat{\theta}$ this way. That is, draw many (X_i, Y_i) from that distribution and compute $\hat{\theta}$ for each draw.

The problem is that we don’t know P. But we have a data sample that was generated from P. Hence, we can instead take the **empirical** distribution $\hat{P}$that places probability mass of $1/n$ on each observation, draw a sample from this distribution (which is simply drawing uniformly from our sample with replacement) and compute our estimate of interest from this sample.

$$\hat{\theta}^* = g({Z_1}^*, ..., {Z_{new}}^*)$$

We can do that many times to get an approximate distribution for $\hat{\theta}$. A crucial assumption is that $\hat{P}$ ressembles P. If our data is not i.i.d, this may not be the case and hence bootsrapping might be missleading. Below, we can see that i.i.d. sampling ressembles the true distribution quite well, wherease biased sampling obviously does not. We produce a sample that places higher probability mass to the large (absolute) values.



We can summarize the bootstrap procedure as follows.

- draw a bootstrap sample ${Z_1}^*, ..., {Z_{new}}^*$
- compute your estimator $\hat{\theta}$ based on that sample.
- repeat the first two steps B times to get bootstrap estimators $\hat{\theta}_1, ..., \hat{\theta}_B$ and therefore an estimate of the distribution of $\hat{\theta}$.

Use the B estimated boostrap estimators as approxmimations for the bootstrap expectation, quantiles and so on. $\mathbb{E}[\hat{\theta}^*_n] \approx B^{-1} \sum\limits_{j=1}^{n} \hat{\theta}^{*j}_n$

6.2 The bootstrap distribution

With P^*, we denote the boostrap distribution, which is the conditional probability distribution introduced by sampling i.i.d. from the empirical distribution $\hat{P}$. Hence, P^* of $\hat{\theta}^*$ is the distribution that arrises from sampling i.i.d. from $\hat{P}$ and applying the transformation $g(\cdot)$ to the data. Conditioning on the data allows us to treat $\hat{P}$ as fixed.

6.3 Bootstrap Consistency

The bootstrap is is called consistent if

$$\mathbb{P}[a_n(\hat{\theta} - \theta) \leq x] - \mathbb{P}[a_n(\hat{\theta} - \theta) \leq x] \to 0$$

Chapter 7

Classification

7.1 Indirect Classification - The Bayes Classifier

In classification, the goal is to assign observations to a group. Similar to regression, where we have $m(x) = E[Y|X = x]$, we want to assign class probabilities to the observations

$$\pi_j(x) = P[Y = j|X = x] \quad (j = 0, 1, ..., J - 1)$$

Def: A classifier maps A multidimensional input vector to a class label. Or mathematically: $C : \mathbb{R}^p \to \{0, ..., J - 1\}$ The quality of a classifier is measured via the zero-one test-error.

$$\mathbb{P}[C(X_{new}) \neq Y_{new}]$$

The optimal classifier with respect to the zero-one Error is the Bayes Classifier. It classifies an observation to the group for which the predicted probability was the highest.

$$C_{bayes}(x) = \arg \max_{0<j<J-1} \pi_j(x)$$

Hence, the Bayes Classifier is a point-wise classifier. For the Bayes Classifier, the zero-one test error is known as the *Bayes Risk*.

$$\mathbb{P}[C_{Bayes}(X_{new}) \neq Y_{new}]$$

In practice, $\pi_j(\cdot)$ is unknown (just as the MSE in regression is unknown) and hence, the the Bayes Classifier and Risk is unknown too. However, we can estimate $\pi_j(\cdot)$ from the data and plug it in the Bayes Classifier.

$$\hat{C}(X) = \arg \max_{0<j<J-1} \hat{\pi}_j(x)$$

This is an indirect estimator, since we first estimate the class probabilities $\pi_j(\cdot)$ for each observation x and then assign the class to it for which the probability was the highest. Question how is that more indirect than Discriminant analysis? Don't we use the Bayes classifier in the end?

7.2 Direct Classification - The Discriminant View

7.2.1 LDA

One example for a direct classification is discriminant analysis. Using Bayes Theorem

$$\mathbb{P}[Y = j|X] = \frac{\mathbb{P}[X = x|y = j]}{\mathbb{P}[X = x]} * \mathbb{P}[Y = j]$$

And assuming

$$(X|Y) \sim N_p(\mu_j, \Sigma); \quad \sum_{k=0}^{J-1} p_k = 1$$

We can write

$$\mathbb{P}[Y = j|X = x] = \frac{f_{x|Y=j} * p_j}{\sum_{k=0}^{J-1} f_{x|Y=k} * p_k}$$

Note that there is no distributional assumption on Y so far. You can estimate

$$\mu_j = \sum_{i=1}^{n} x_i * 1_{Y_i=j} / 1_{Y_i=j}$$

and

$$\Sigma = \frac{1}{n-j} \sum_{j=0}^{J-1} \sum_{i=1}^{n} (x_i - \mu_j)(x_i - \mu_j)' \, 1_{Y_i=j}$$

Note that the means of the response of the groups are different, but the covariance structure is the same for all of them. We now also need to estimate p_j. A straight-forward way is

$$\hat{p}_j = n^{-1} \sum_{i=1}^{n} 1_{[Y_i=j]} = \frac{n_j}{n}$$

From here, you can easily compute the classifier (as done in the exercise) by maximizing the log-likelihood. Then, you can derive the decision boundary by using $\delta_j - \delta_k = 0$. In a two dimensional predictor space with two classes, the decision boundary is a line. Every combination of the two predictors on one side of the line will result in a prediction of class one, everything on the other side of the line of class two. Note that both the decision function (and hence the decision boundary) are linear in x.

7.2.2 QDA

Quadratic discriminant analysis loosens the assumption of shared covariance matrices, namely each group has their own covariance matrix. This leads to quadratic decisions functions δ and hence to non-linear decision boundaries. QDA is more flexible but for high p, the problem of over-fitting can occur, since the number of variables to estimate is $J * p(p+1)$ variable for the covariance matrix only (instead of $p * (p+1)$ for LDA.

Chapter 8

Flexible regression and classification methods

Chapter 9

Bagging and Boosting

Chapter 10

Introduction

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 10. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter **??**.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 10.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 10.1.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the **bookdown** package [@R-bookdown] in this sample book, which was built on top of R Markdown and **knitr** [@xie2015].

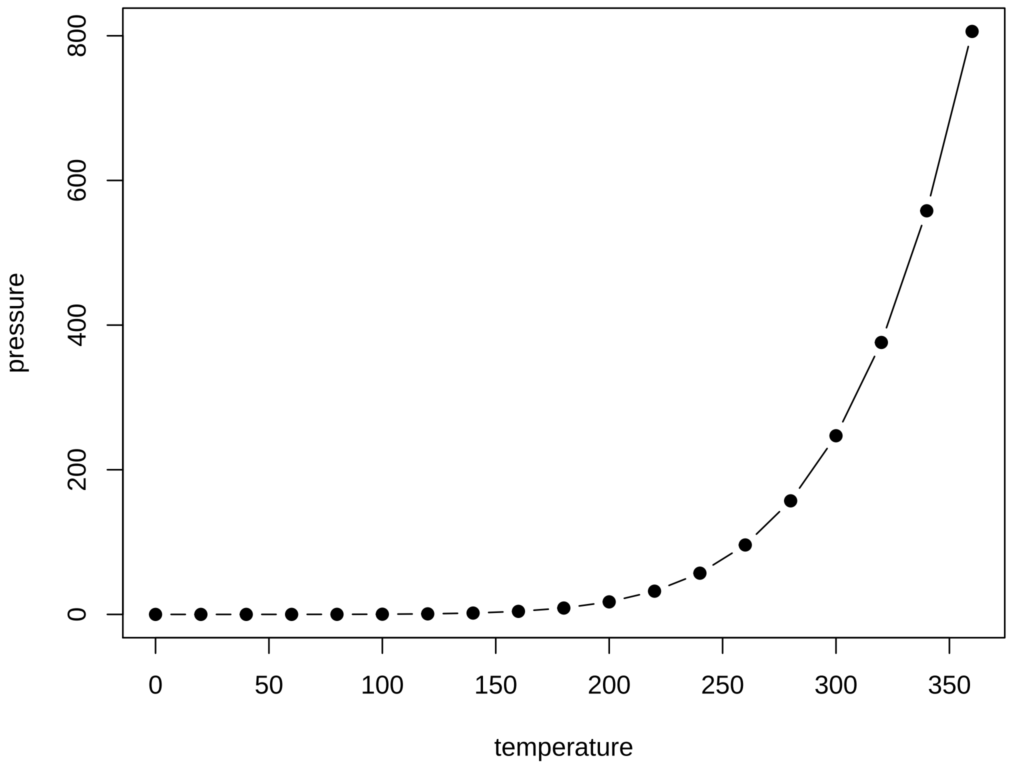


Figure 10.1: Here is a nice figure!

Table 10.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa