Machine Learning & Data Analytics

Mario V. Wüthrich
RiskLab, ETH Zurich

Schweizerische Aktuarvereinigung SAV

Zurich, April 4, 2017

New Lecture & Literature

▷ **New lecture at ETH Zurich**

- **Data Analytics for Non-Life Insurance Pricing**

  Wüthrich and Buser (AXA Winterthur) starting Spring 2018.

▷ **Lecture notes on SSRN preprint server**[1] (first draft)

- **Data Analytics for Non-Life Insurance Pricing**

  Manuscript ID 2870308.

[1] https://www.ssrn.com/en/

- **Section 1: Supervised and Unsupervised Learning**

Regression Structure

Basic Assumption:
There are structural differences which can be explained by a regression function

$$\mu : \mathcal{X} \to \mathbb{R}, \qquad \boldsymbol{x} \mapsto \mu(\boldsymbol{x}).$$

- $\mathcal{X}$ is called feature space, covariate space;
- $\boldsymbol{x} \in \mathcal{X}$ is called feature, covariate, explanatory variable, independent variable;
- $\mu(\cdot)$ is called regression function or classifier function.

Example of feature:

$$\boldsymbol{x} = (x_1, \ldots, x_d) = (\text{age},\ \text{gender},\ \text{type of car},\ \text{NOGA code},\ \text{income},\ \ldots)$$

Supervised Learning

Assumption: We have n independent (noisy) observations (data)

$$\mathcal{D} = \{(Y_1, \boldsymbol{x}_1), \ldots, (Y_n, \boldsymbol{x}_n)\},$$

satisfying for all $i = 1, \ldots, n$ the model assumption

$$\mathbb{E}[Y_i] = \mu(\boldsymbol{x}_i).$$

▷ **Supervised Learning** (regression problem):

Determine the (unknown) regression function

$$\mu : \mathcal{X} \to \mathbb{R}, \qquad \boldsymbol{x} \mapsto \mu(\boldsymbol{x})$$

from the given data $\mathcal{D} = \{(Y_1, \boldsymbol{x}_1), \ldots, (Y_n, \boldsymbol{x}_n)\}$.

Unsupervised Learning

Assumption: We have n (possibly noisy) features

$$\mathcal{F} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \;\subset\; \mathcal{X}.$$

▷ **Unsupervised Learning** (pattern recognition):

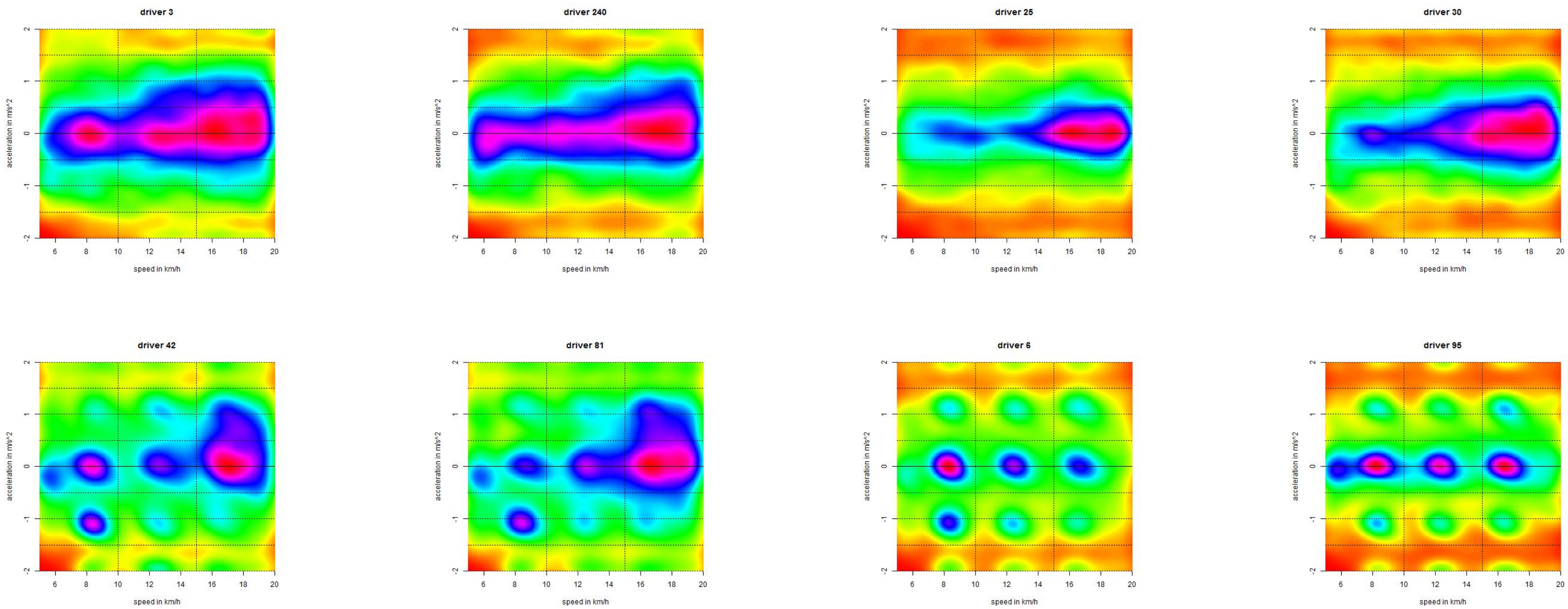
Find patterns and differences in these features $\mathcal{F} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.

- **Section 2: Unsupervised Learning**

Telematics Car Driving Data

▷ **Unsupervised Learning** (pattern recognition):

Find patterns and differences in these (noisy) features $\mathcal{F} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.



K-Means Algorithm

Select (desired) number K of categories and construct a "good" classifier

$$\mathcal{C} : \mathcal{X} \;\rightarrow\; \mathcal{K} = \{1, \ldots, K\}, \qquad \boldsymbol{x} \;\mapsto\; \mathcal{C}(\boldsymbol{x}).$$

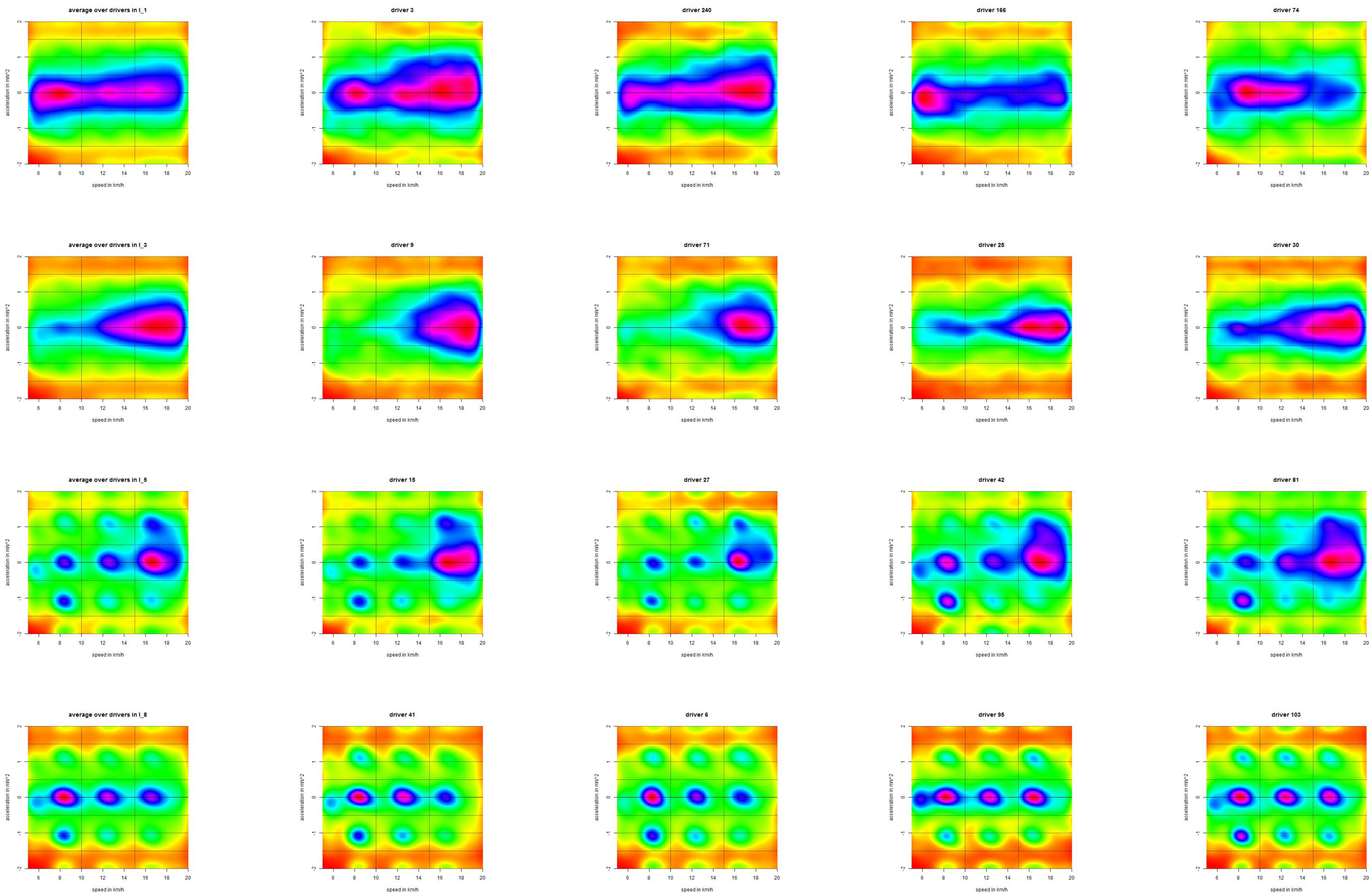
- Choose a distance function $d(\cdot, \cdot) \geq 0$ on $\mathcal{X} \times \mathcal{X}$.
- K-means algorithm determines iteratively K centers $z^{(1)}, \ldots, z^{(K)} \in \mathcal{X}$ such that

$$\sum_{k=1}^{K} \left(\sum_{i=1}^{n} \mathbb{1}_{\{\mathcal{C}(\boldsymbol{x}_i)=k\}} \, d(\boldsymbol{x}_i, z^{(k)}) \right) \;\overset{!!!}{=}\; \min,$$

with classifier $\mathcal{C}(\boldsymbol{x}_i) \;=\; \underset{k \in \mathcal{K}}{\operatorname{argmin}}\, d(\boldsymbol{x}_i, z^{(k)}) \;\in\; \mathcal{K}$.

- Algorithm converges but result may be non-optimal (depending on starting point).

Result of K-Means Algorithm for $K = 4$



- **Section 3: Supervised Learning**

Choice of Loss Function

▷ **Supervised Learning** (regression problem):

Infer the (unknown) regression function

$$\mu : \mathcal{X} \to \mathbb{R}, \qquad \boldsymbol{x} \mapsto \mu(\boldsymbol{x})$$

from the given data $\mathcal{D} = \{(Y_1, \boldsymbol{x}_1), \ldots, (Y_n, \boldsymbol{x}_n)\}$.

▷ This inference is done w.r.t. a given loss function $\mathcal{L}$. For simplicity, set

$$\mathcal{L}_{\mathcal{D}}(\mu(\cdot)) \;=\; \sum_{i=1}^{n} (Y_i - \mu(\boldsymbol{x}_i))^2.$$

▷ In general, one should/may use (scaled) deviance statistics as loss function.

Regression Problem

Aim: Find regression function $\mu : \mathcal{X} \to \mathbb{R}$ that "minimizes" in-sample loss

$$\mathcal{L}_{\mathcal{D}}(\mu(\cdot)) \; = \; \sum_{i=1}^{n} \left(Y_i - \mu(\boldsymbol{x}_i)\right)^2.$$

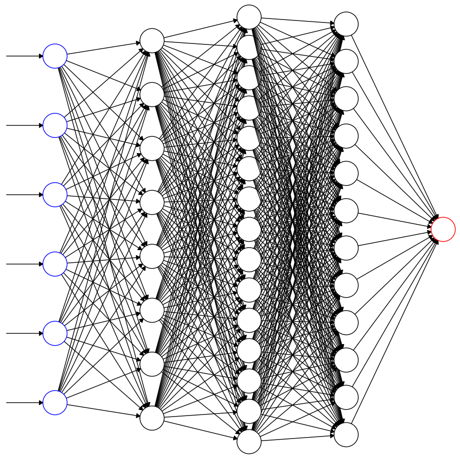
- The saturated model minimizes in-sample loss, but it is over-parametrized!
- What if we do not have any idea about a "low-parametrized" $\mu(\cdot)$?
- What if the feature space $\mathcal{X}$ is very high-dimensional?

▷ Machine learning methods help to find $\mu(\cdot)$.

▷ **Crucial:** Trade-off between small in-sample loss and over-parametrization.

Supervised Machine Learning

Supervised Machine Learning Categorization:



▷ deep learning

- ∗ e.g. deep artificial neural networks
- ∗ deep: use many hidden layers (more like black-box)
- ∗ often very powerful, but difficult to calibrate

▷ shallow learning

- ∗ e.g. regression & classification trees, boosting machines, shallow neural networks
- ∗ shallow: analysis remains at the surface (more transparent)
- ∗ also powerful and easy to use
- ∗ useful to improve parametric statistical models

Classification and Regression Trees (CART)

Classification and regression trees (CART) provide regression functions that

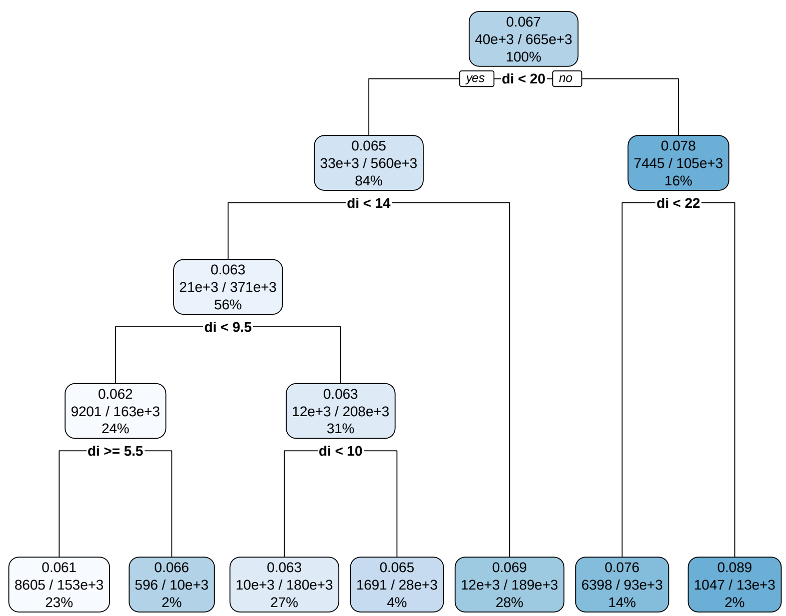
- are non-parametric,
- learn an underlying structural form of $\mu(\cdot)$ from the data $\mathcal{D}$, and
- which can deal with high dimensional feature spaces $\mathcal{X}$.

CART go back to the seminal work of Breiman, Friedman, Olshen and Stone (1984).

Regression Tree Algorithm (1/2)

Idea: Group observations $(Y_i, \boldsymbol{x}_i)$ that are similar into the same basket, i.e. grouping is done such that the observations in the same basket are "more similar".

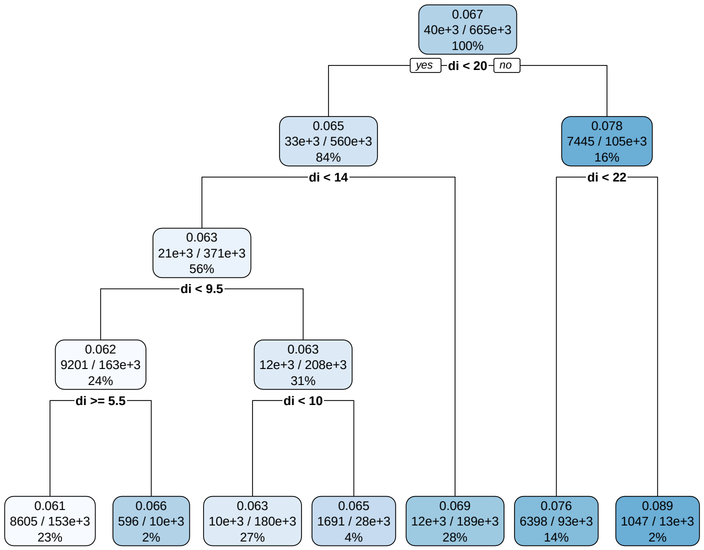
▷ Binary split regression tree algorithm builds at every step 2 baskets:



Regression Tree Algorithm (2/2)

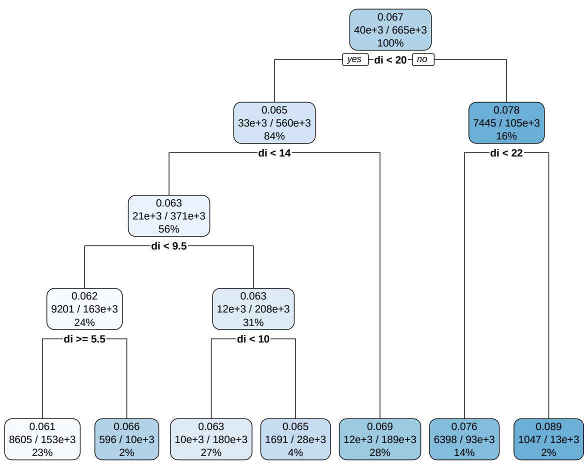
Main Questions:

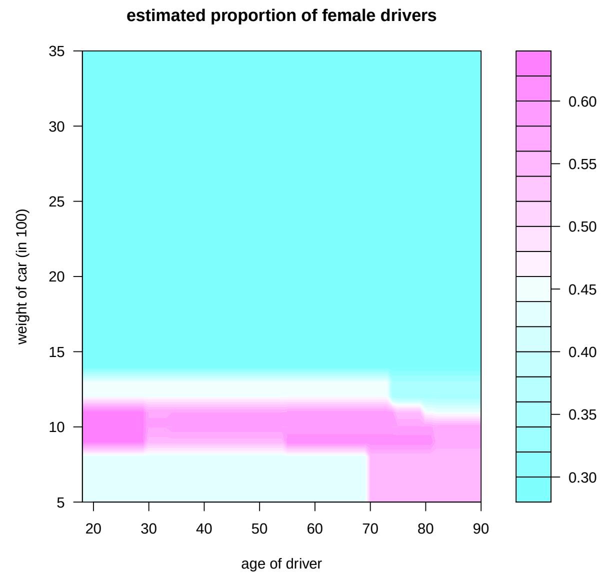
- measure of dissimilarity $\Rightarrow$ loss function $\mathcal{L}_{\mathcal{D}}$
- choice of potential splits, in particular, for high dimensional $\mathcal{X}$
- stopping rule for algorithm (statistics)



Regression Tree Estimator (1/2)

Successive application of binary splits provides partition $\mathcal{X}_1, \ldots, \mathcal{X}_K$ of $\mathcal{X}$.





Define the regression tree estimator (of complexity K) in $\boldsymbol{x} \in \mathcal{X}$ by

$$\widehat{\mu}(\boldsymbol{x}) = \sum_{k=1}^{K} \widehat{\mu}_k \, \mathbb{1}_{\{\boldsymbol{x} \in \mathcal{X}_k\}},$$

with $\widehat{\mu}_k$ being the sample mean on $\mathcal{X}_k$.

Regression Tree Estimator (2/2)

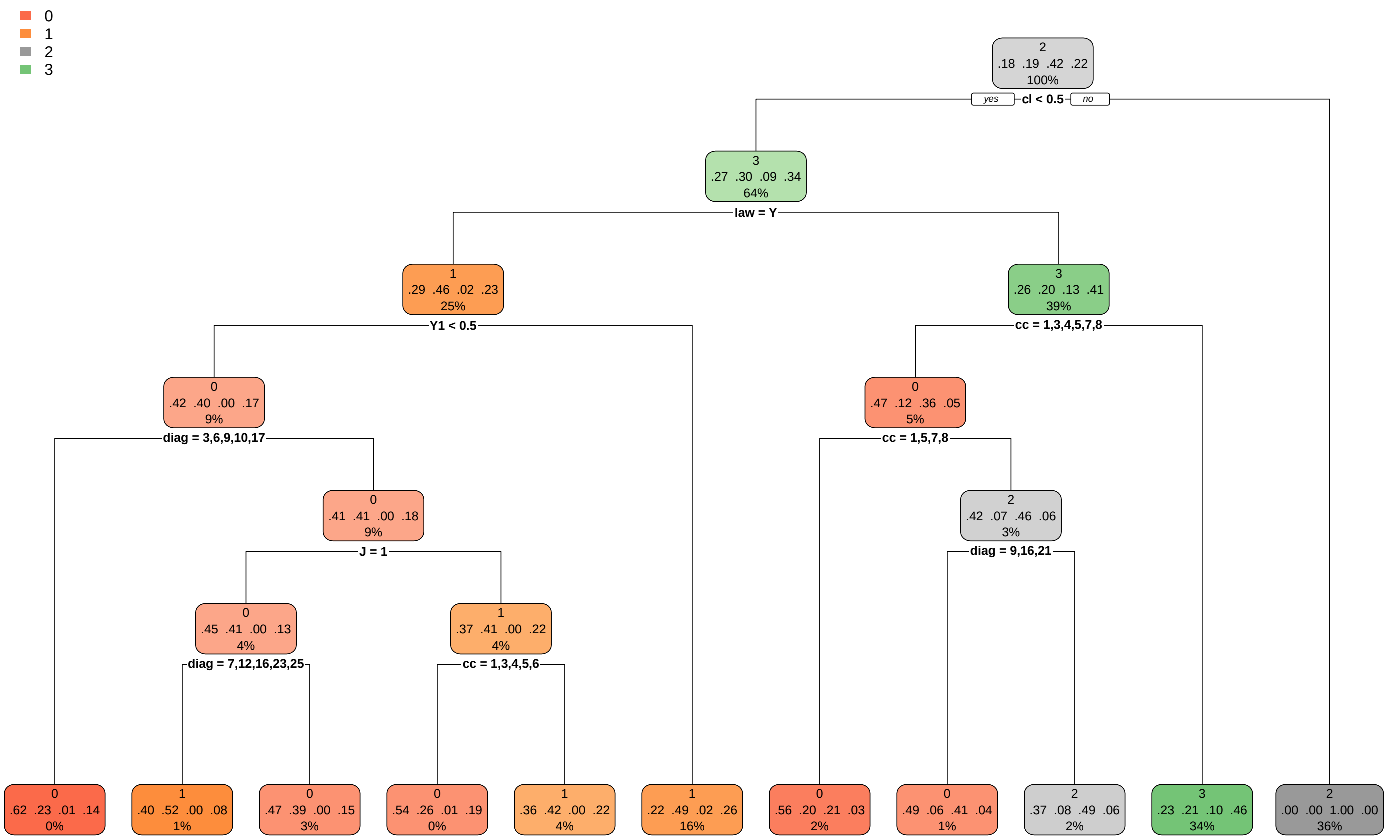
Define the regression tree estimator (of complexity K) in $\boldsymbol{x} \in \mathcal{X}$ by

$$\widehat{\mu}(\boldsymbol{x}) \;=\; \sum_{k=1}^{K} \widehat{\mu}_k \; \mathbb{1}_{\{\boldsymbol{x} \in \mathcal{X}_k\}}.$$

- Regression tree estimator is non-parametric (similarity and loss function driven).
- Regression tree estimator works for high dimensional feature spaces $\mathcal{X}$.
- To be discussed:
  - ⋆ choice of feature space $\mathcal{X}$ and potential splits affect results (and dependencies);
  - ⋆ stability of the results under slight changes in observations (different noise);
  - ⋆ choice of sensible stopping rule (tree pruning);
  - ⋆ stability under different choices of loss functions;
  - ⋆ more advanced methods than regression trees;
  - ⋆ weak learning, stage-wise adaptive regression, boosting machine.

- **Section 4: Examples of Supervised Learning**

Individual Claims Reserving (1/2)



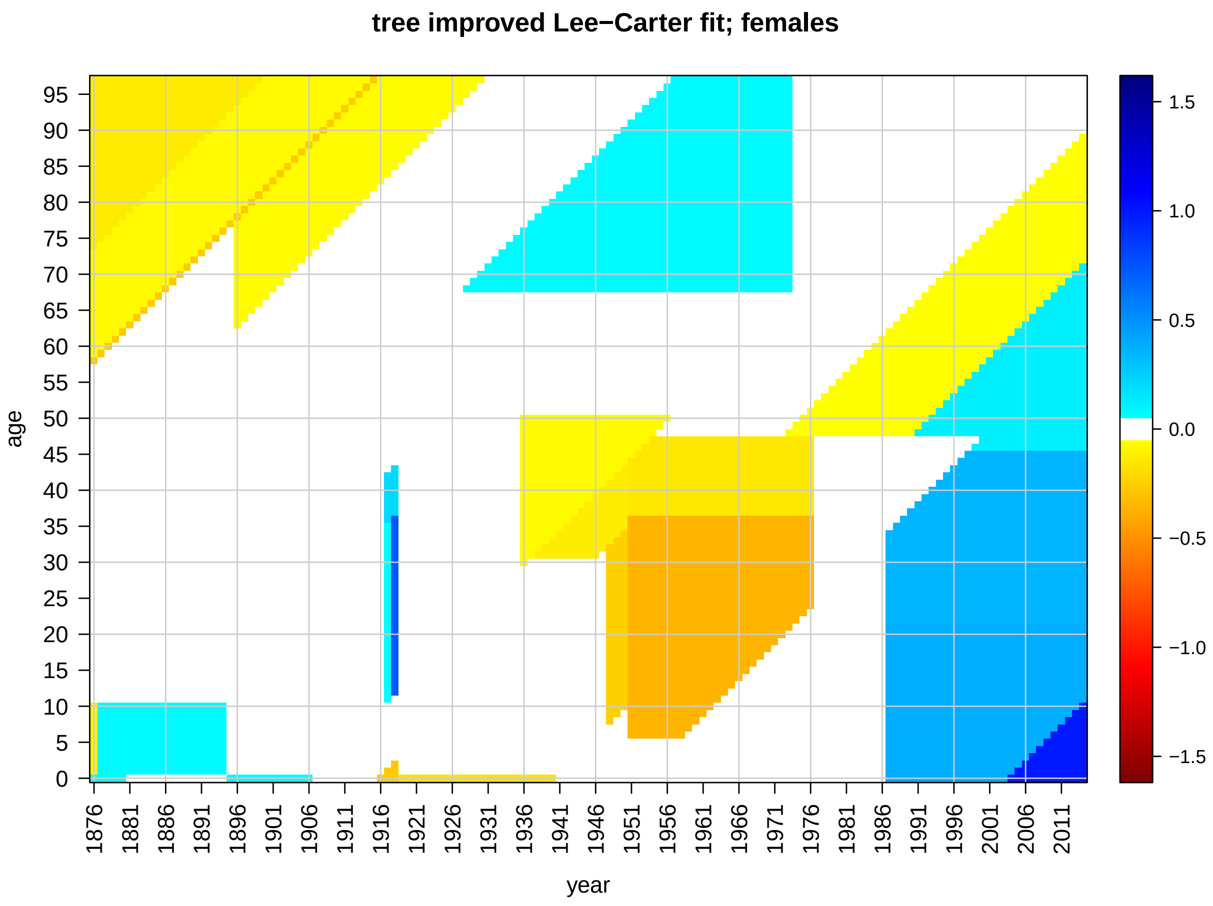
Individual claims development (regression tree for one-step ahead $t \to t+1$) based on feature components `cl`, `diag`, `cc`, `law`, `j`, individual claims history.

Individual Claims Reserving (2/2)

for time lag $t+1$:	Y1	Y2	Y3	Y4	Y5	Y6
numbers of leaves	8	11	18	12	4	4
	components used for split questions					
claim closed	cl	cl	cl	cl	cl	cl
lawyer involved		law	law	law		
claims code	cc	cc	cc	cc	cc	
claims diagnosis	diag	diag	diag	diag		diag
reporting delay		j	j	j		
previous payments	Y0	Y1	Y2	Y3	Y4	Y5
previous payments			Y1	Y2		

Relevant feature information and Markov condition.

Boosting the Lee-Carter Model



Iterated weak learning applied to residuals is known as a Boosting Machine.

Conclusions should be here ...

... and your remarks!