Seminar: Künstliche neuronale Netze: Lösung einer Fragestellung aus der Praxis

Bagging, Random Forests, Boosting

Lars Langen

Mathematisches Institut

16. November 2018

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät

Universität zu Köln

Übersicht

1. Einleitung

2. Grundlegendes

3. Bagging

4. Random Forests

5. Boosting

6. Auswertung

7. Fazit

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Einleitung

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Motivation

- Wir betrachten Regressions- und Klassifizierungsmodelle
- Ein möglicher Ansatz: Entscheidungsbäume
- Vorteil: intuitiv und einfach (grafisch) interpretierbar
- Nachteil: nicht so genau wie andere Modelle
- $\rightarrow$ Bäume weisen oft hohe Varianz auf und sind nicht robust

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Motivation (II)

- Unser Ziel: Modellverbesserung
- Dafür betrachten wir die Methoden *Bagging*, *Random Forests* und *Boosting*

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Grundlegendes

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Problematik

- Wir differenzieren zwischen quantitativem und qualitativem Output
- Regressionsmodell vs. Klassifizierungsmodell
- Regressionsbäume vs. Klassifizierungsbäume
- Daten werden aufgeteilt in Trainingsset T und Testset U
- Ziel: $\hat{f}(x) := \hat{y}$

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Fehlerbeschreibung

- Hat man eine Regression bzw. Klassifizierung $\hat{f}(x)$ mit Hilfe des Trainingssets T trainiert, so kann man anhand des Testsets U eine Güte bestimmen
- Regression: MSE (Mean Squared Error)
- Klassifizierung: Klassifizierungsfehler

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Bagging

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Ziel des Bagging's

- Allgemein Anwendbar auf verschiedenste Modelle
- Ziel: Reduktion der Varianz (von Entscheidungsbäumen)
- Idee: Bootstrapping aus dem Trainingsset

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Wahrscheinlichkeitstheoretische Grundlage

Bemerkung

Sind X_1, X_2,...,X_B unabhänging, identisch verteilte Zufallsvariablen mit Varianz $\mathrm{Var}(X_i) = \sigma^2 \; \forall \; i \in \{1, ..., B\}$, so ist die Varianz von $\bar{X} := \frac{1}{B} \sum_i^B (X_i)$ gegeben durch

$$\mathrm{Var}(\bar{X}) = \frac{\sigma^2}{B}.$$

$\Rightarrow$ Von dieser Eigenschaft macht Bagging Gebrauch.

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Bagging I

- Generieren von B Mengen b_1, b_2, ..., b_B
- B-maliges Ziehen mit Zurücklegen aus dem Trainingsset T, sprich ein Element $x \in T$ kann mehrmals in eine Menge b_i gezogen werden
- Jede Menge b_i soll $|T|$ Elemente enthalten
- $1 - \frac{1}{e} \approx 63,21\%$ einzigartige Elemente in b_i, $\frac{1}{e} \approx 36,79\%$ Dubletten

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Bagging II

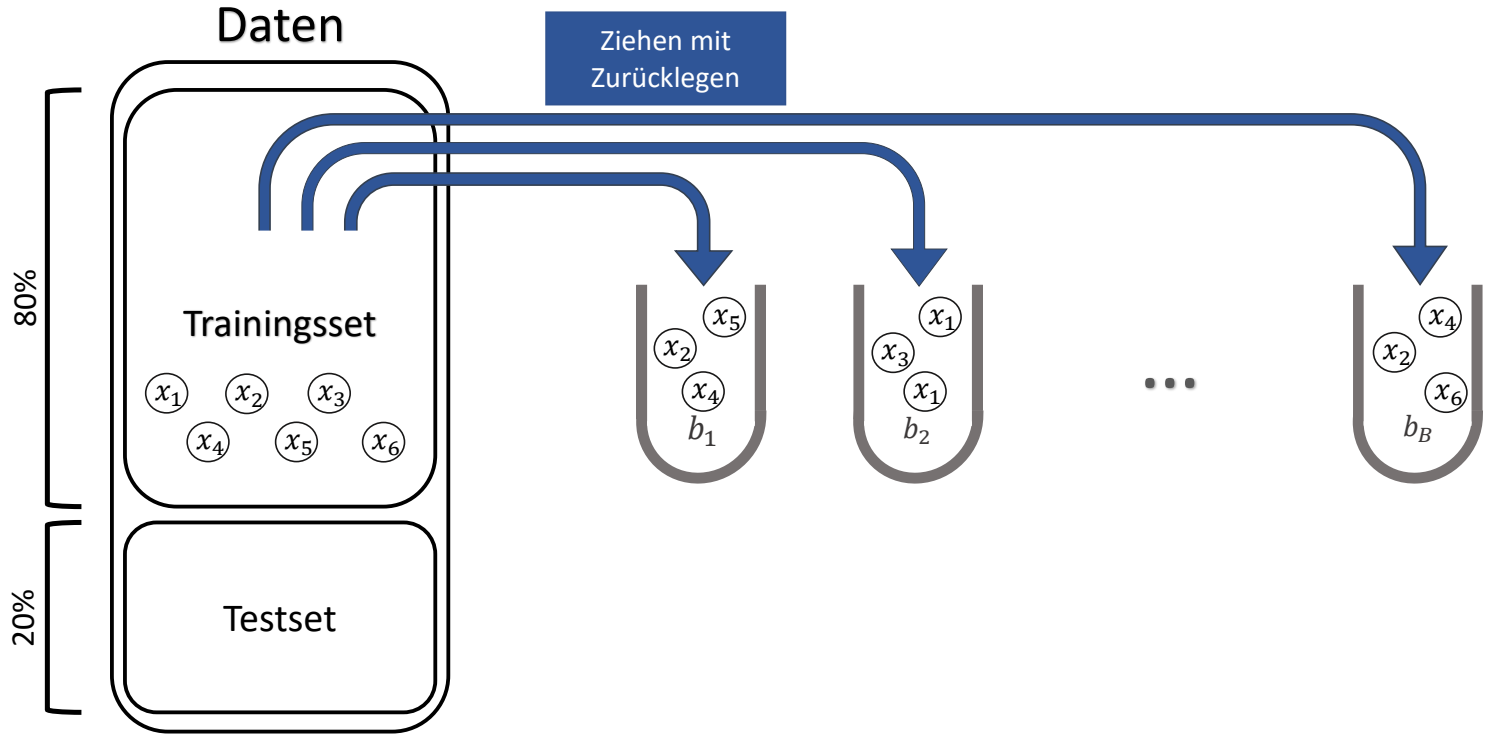

Abbildung: Die Idee des *Bagging's* durch Bootstrapping des Trainingssets.

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Bagging III

- Berechnen (bzw. fitten) für alle b_i, $i \in \{1, ..., B\}$
- Dadurch erhalten wir $\hat{f}^{b_1}$, $\hat{f}^{b_2}$,...,$\hat{f}^{b_B}$
- Darüber bilden wir den Durchschnitt:
  - Regression: $\hat{f}_{bag}(x) := \frac{1}{B} \sum_{i=1}^{B} \hat{f}^{b_i}(x) = \hat{y}$
  - Klassifizierung: Mehrheitsvotum über die vorhergesagten Klassen: Wähle die am häufigsten vorhergesagte Klasse

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Bagging IV

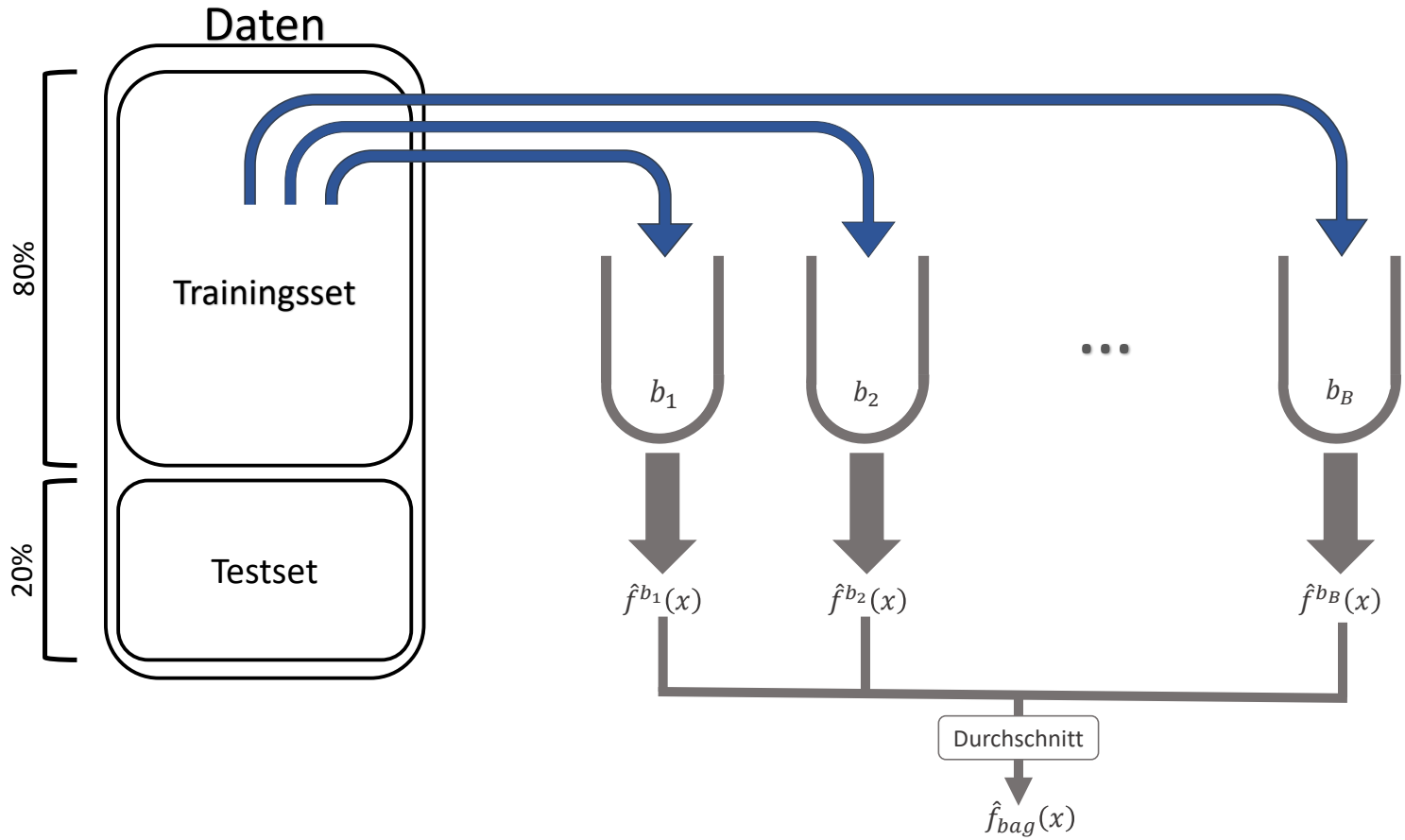

Abbildung: Das gesamte Prinzip des *Bagging's*

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Bagging V

Algorithmus: *Bagging*

Input: Trainingsset T, Testset U, Anzahl der zu ziehenden Mengen B

Output: Regression bzw. Klassifizierung $\hat{f}_{bag}$

(1) Initialisiere: $b_1 = b_2 = b_B = \emptyset$, $\hat{f}_{bag}(x) \leftarrow 0$

(2) for $(i = 1,2,...,B)$ **do**

  (2.1) for $(j = 1,2,...,|T|)$ **do**

    (2.2) Ziehe beliebiges Element $x \in T$

    (2.3) Füge dieses Element zu b_i hinzu:
    $b_i \leftarrow b_i \cup x$

  end

end

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Bagging VI

Algorithmus: *Bagging* (Fortsetzung)

(3) for $(i = 1,2,...,B)$ **do**

  (3.1) Berechne $\hat{f}^{b_i}$

end

(4) if *(Regressionsmodell)* **then**

  $\hat{f}_{bag}(x) \leftarrow \frac{1}{B}\sum_{i=1}^{B} \hat{f}^{b_i}(x) = \hat{y}$

else

  Bestimme $\hat{f}_{bag}(x)$ per Mehrheitsvotum

end

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Bagging VII

- Schon bei verhältnismäßig kleinem B wird oft eine deutliche Verbesserung erzielt
- Hohe Wahl von B führt nicht zu Overfitting
- Sollte mit vorhandener Rechenkapazität abgewägt werden

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Out-of-Bag Error I

- Ziehen der Mengen b_i erfolgt mit Zurücklegen
- Eine Menge b_i erwartet $1 - \frac{1}{e} \approx 63,21\%$ einzigartige Inputs x und $\frac{1}{e} \approx 36,79\%$ mehrfach vorkommende Inputs x
- Damit sind circa $\frac{1}{3}$ der Werte aus T nicht in b_i enthalten und somit *Out-of-Bag* (OOB)

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Out-of-Bag Error II

- Wir können nun einen OOB-Fehler für ein x berechnen:
- Sei dafür $O := \{b_i : x \text{ ist OOB in } b_i\}$
- Regression: $\hat{f}_{bag}(x) = \frac{1}{|O|} \sum_{b_i \in O} \hat{f}^{b_i}(x)$
- Klassifizierung: Mehrheitsvotum aus den Funktionen $\hat{f}^{b_i \in O}$
- Daraus lässt sich ein OOB-MSE (OOB-Klassifizierungsfehler) berechnen
- Vorteil: vergleichsweise geringer Rechenaufwand und kein Testset U benötigt

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Random Forests

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Random Forests I

- Bagging verringert Varianz der Entscheidungsbäume
- Angenommen, es existiert eine sehr wichtige erklärende Variable und mehrere etwas weniger wichtige Variablen
- Wie würden die B Entscheidungsbäume beim Bagging aussehen?
- Bäume würden hohe Korrelation aufweisen

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Random Forests II

- Sei p die Anzahl der erklärenden Variablen, so stehen in jedem Split p Splitkandidaten zur Verfügung
- Idee: Bei jedem Split m zufällig ausgewählte Splitkandidaten in Betracht ziehen
- Häufig wird $m \approx \sqrt{p}$ gewählt
- Gedankenexperiment von eben: $\frac{p-m}{p}$ der Splits haben die eine wichtige Variable gar nicht zur Auswahl

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Random Forests III

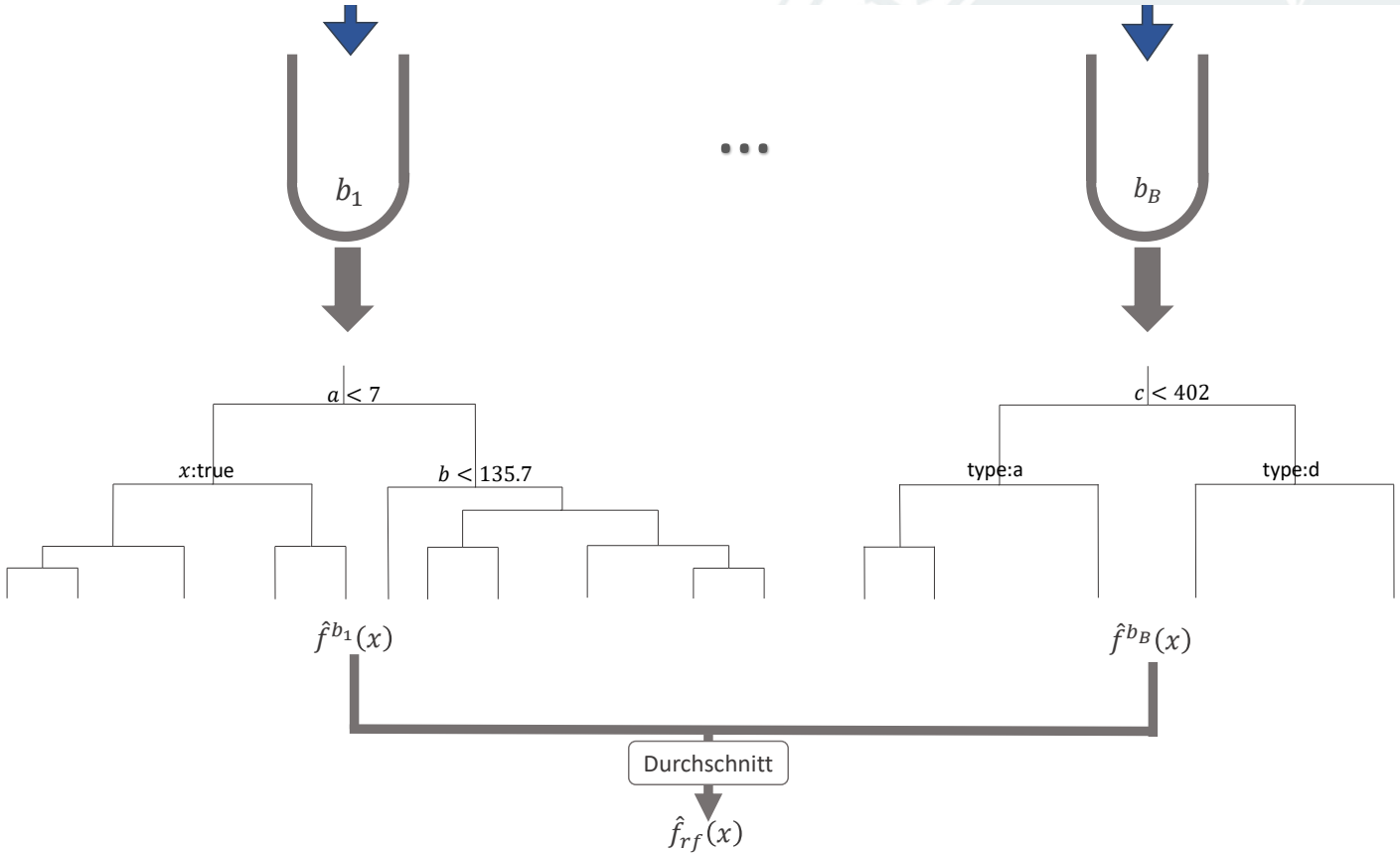

Abbildung: *Random Forests*: Durch die zufällige Auswahl von Splitkandidaten werden die Entscheidungsbäume zunehmend unkorreliert.

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Random Forests IV

- Random Forests machen Entscheidungsbäume unkorrelierter
- Offensichtlich nicht allgemeine Methode, sondern nur auf Entscheidungsbäume anwendbar
- Optimale Wahl von m ist abhängig von der Problemstellung
- Je höher die Korrelation der Bäume, desto kleiner sollte m gewählt werden
- Bagging ist Spezialfall von Random Forests mit $m = p$

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Boosting

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Boosting I

- Wie Bagging eine allgemeine Methode
- Idee: Statt Konstruktion vieler Bäume diesmal nur ein Baum
- Sequentielle Verbesserung
- Fitten nach Residualen $\rightarrow$ Baum wird da verbessert, wo er noch nicht genau ist
- B Durchläufe, in jedem Durchlauf wird ein Baum berechnet und per Linearkombination dem bestehenden Baum hinzugefügt

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Boosting II

- 3 Tuningparameter
- Anzahl der Bäume B
  - Zu hohes B kann zu Overfitting führen
- Dämpfungsparameter λ
  - Sorgt dafür, dass Modell nicht zu schnell lernt
  - Meistens wird $\lambda \in \{0.01, 0.001\}$ gewählt
- Anzahl der Splits d pro Baum
  - Meist genügt $d = 1$ oder $d = 2$
  - d wird auch als Interaktionstiefe bezeichnet

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Boosting III

Algorithmus: Boosting, vgl. [1, Algorithmus 8.2]

Input: Trainingsset T, Anzahl der Bäume B,
Dämpfungsparameter λ, Anzahl der Splits pro Baum d

Output: Regression bzw. Klassifizierung $\hat{f}$

(1) Initialisiere $\hat{f}(x) \leftarrow 0$ und $r_i \leftarrow y_i \; \forall \; i \in T$

(2) for $(b = 1,2,...,B)$ **do**

(2.1) Fitte einen Baum $\hat{f}^b$ mit d Splits ($\Leftrightarrow d+1$ Endknoten) zu den Trainingsdaten (T, r).

(2.2) Aktualisiere $\hat{f}$ durch Hinzufügen der gedämpften Version des neuen Baums:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$$

(2.3) Aktualisiere die Residuale:

$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$$

end

(3) return $\hat{f}(x)$

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Boosting IV

- Damit entspricht $\hat{f}(x)$ einer Linearkombination aus den einzelnen Bäumen einer jeden Iteration
- $\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$
- Funktioniert bei Klassifizierungsmodellen prinzipiell genauso, lediglich Aktualisierung komplexer

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Auswertung

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Heart-Daten I

- 303 Patienten mit Brustschmerzen
- 13 verschiedene erklärende Variablen
- binärer Output: Herzkrankheit ja/nein
- Bagging und Random Forests mit $m \approx \sqrt{p}$ angewandt

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Heart-Daten II

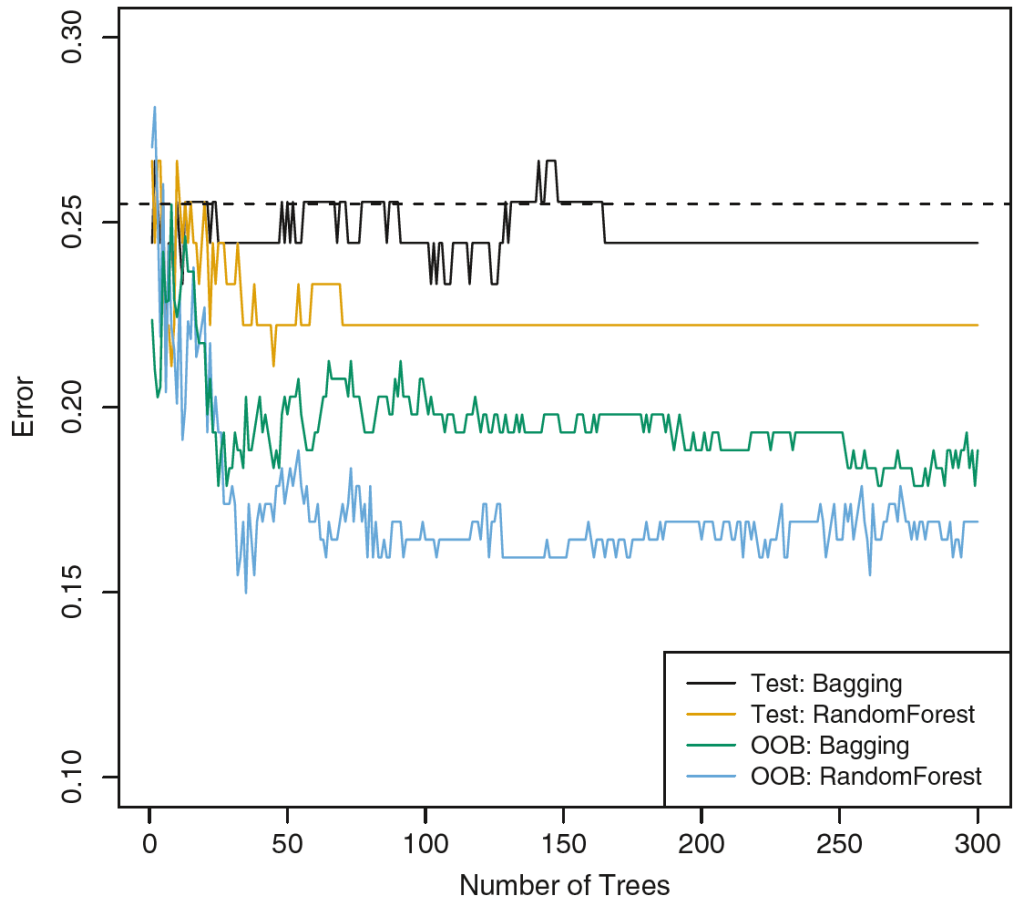

Abbildung: *Bagging* und *Random Forests* angewandt auf die „Heart“-Daten, siehe [1, Section 8.2, Figure 8.8].

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

15-class gene expression I

- Genausprägungen bei verschiedenen Krebstypen
- Vorhersage von Krebstyp anhand Genausprägung

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

15-class gene expression II

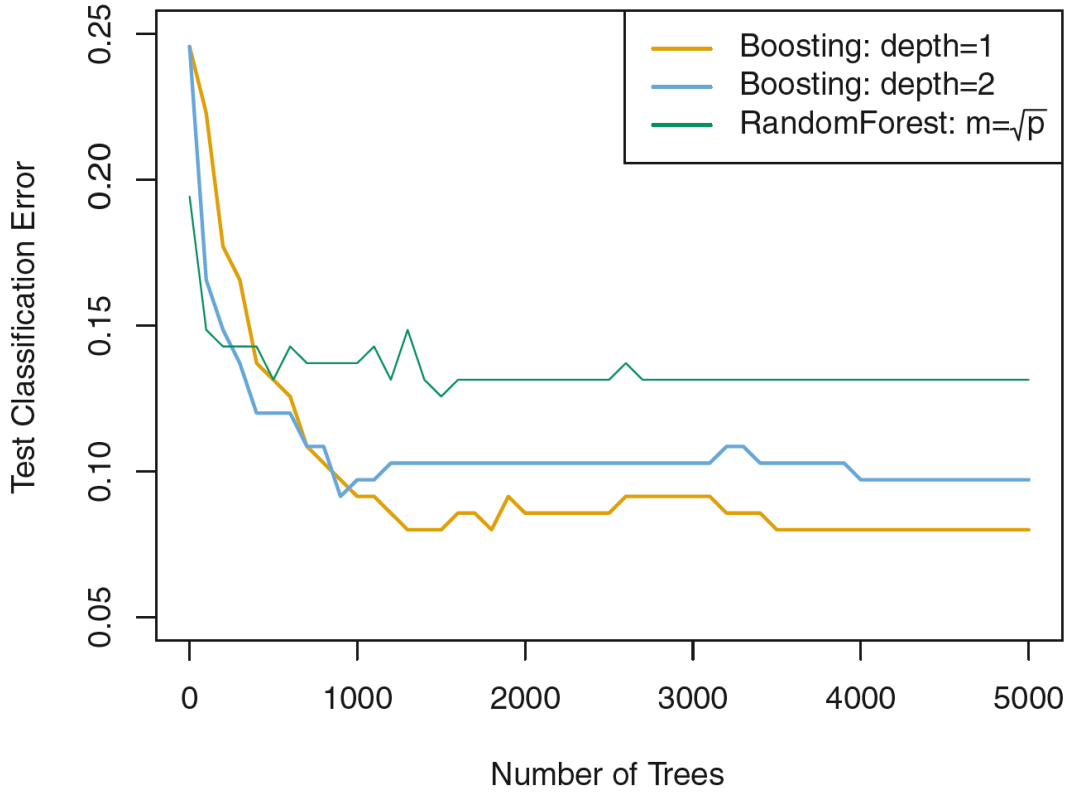

Abbildung: *Random Forests* und *Boosting* (mit $\lambda = 0.01$) angewandt auf die „15-class gene expression“-Daten, siehe [1, Section 8.2, Figure 8.11].

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Fazit

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Fazit I

- Alle 3 Methoden verbessern die Genauigkeit von Entscheidungsbäumen
- Bagging verringert die Varianz durch Konstruktion mehrere Bäume mittels Bootstrapping des Trainingssets
- Random Forests machen Entscheidungsbäume unkorrelierter durch zufällige Auswahl von Splitkandidaten

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Fazit II

- Boosting generiert einzelnen Baum, der in mehreren Durchläufen sequentiell verbessert wird, indem nach den Residualen gefittet wird

- Bagging und Boosting sind allgemeine Methoden und somit auf eine Vielzahl von Regressions- bzw. Klassifizierungsmodellen anwendbar

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln

Literatur

- Gareth James, Daniela Witten,Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning*, Springer, 2017.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, `http://www.deeplearningbook.org`, 2016.
- Shai Shalev-Shwartz, Shai Ben-David *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.
- Jeffrey S. Strickland *Predictive Analytics using R*, Lulu Press, 2014.

Vielen Dank für eure Aufmerksamkeit.

Lars Langen
Mathematisches Institut
Mathematisch-Naturwissenschaftliche Fakultät
Universität zu Köln