

Skript zur Vorlesung  
Knowledge Discovery in Databases II  
im Sommersemester 2010

# Kapitel 5: Ensemble Techniken

Skript KDD II © 2009/10 Arthur Zimek

[http://www.dbs.ifi.lmu.de/Lehre/KDD\\_II](http://www.dbs.ifi.lmu.de/Lehre/KDD_II)

## Übersicht

---

1. Einleitung und Grundlagen
2. Aspekte der Diversität
3. Methoden der Konstruktion von Ensembles

## Einleitung und Grundlagen

---

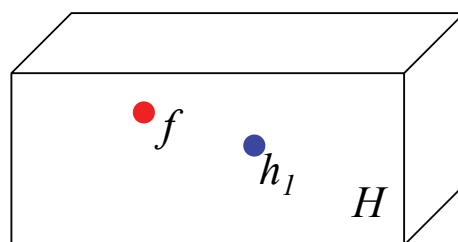
- Annahme: Elemente  $x$  aus einem Raum  $D$  gehören zu einer Klasse  $c_i$  aus einer Menge von möglichen Klassen  $C$ .
- Es gibt eine Funktion  $f: D \rightarrow C$ , die einen eindeutigen Zusammenhang zwischen einem gegebenen Element  $x$  und seiner Klasse  $c_i$  beschreibt.
- Aufgabe eines Lern-Algorithmus' ist es, diesen Zusammenhang zu "lernen".
- Im Allgemeinen stellt ein Klassifikator (das Ergebnis eines Lern-Algorithmus') eine Approximation der Funktion  $f$  dar, auch eine "Hypothese" genannt.

277

## Einleitung und Grundlagen

---

- Die "wahre" Funktion  $f$  ist unbekannt.
- Es gibt nur eine Menge von Beispielen: Tupel  $(x, c_i) \in f \subseteq D \times C$ , die Trainingsdaten.
- Ein konkreter Lernalgorithmus sucht diejenige Hypothese  $h_i$  als Klassifikator aus einem Raum  $H \subseteq D \times C$  möglicher Hypothesen, die optimal zu den Trainingsdaten passt.



- Achtung: die Zielfunktion  $f$  ist nicht zwangsläufig Element von  $H$ !

278

## Einleitung und Grundlagen

---

- Ein Klassifikator (eine erlernte Hypothese  $h$ ) kann auf Elemente  $x \in D$  angewendet werden, um die Klasse  $c_i = f(x)$  vorherzusagen.
- Die Genauigkeit eines Klassifikators ist die Wahrscheinlichkeit (oder statistisch gemessen: die Häufigkeit), mit der seine Vorhersage korrekt ist.

$$Acc(h) = P(h(x)=f(x))$$

- Entsprechend ist die Fehlerrate das Komplement:

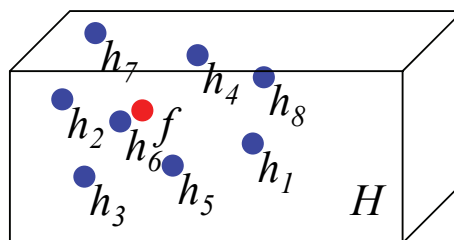
$$Err(h) = P(h(x) \neq f(x)) = 1 - Acc(h)$$

279

## Einleitung und Grundlagen

---

- Idee der Ensemble-Technik: Reduktion der Häufigkeit von Fehlurteilen durch Bilden einer “Jury von Experten” und Abstimmung über die richtige Vorhersage.
- mathematisch: bessere Approximation von  $f$  durch Mittelung über mehrere Hypothesen



280

## Einleitung und Grundlagen

---

- Einfacher Abstimmungsmodus für ein Zwei-Klassen-Problem mit  $C=\{-1, 1\}$ :
  - Bilde Menge von Hypothesen  $\{h_1, \dots, h_k\}$  mit Gewichten  $\{w_1, \dots, w_k\}$ .
  - Ensemble-Klassifikator  $\hat{h}$  ist gegeben durch

$$\hat{h}(x) = \begin{cases} w_1 h_1(x) + \dots + w_k h_k \geq 0 \rightarrow 1 \\ w_1 h_1(x) + \dots + w_k h_k < 0 \rightarrow -1 \end{cases}$$

- Häufig  $w_1 = \dots = w_k = 1$  (bzw. ungewichtete Abstimmung).
- Gewichte können aber auch auf der (gemessenen) Zuverlässigkeit der einzelnen Klassifikatoren (Hypothesen) basieren.
- Komplexeres Abstimmungsverhalten möglich (und bei mehr als zwei Klassen auch nötig)  $\rightarrow$  verschiedene Ensemble-Methoden

281

## Einleitung und Grundlagen

---

$$\hat{h}(x) = \begin{cases} w_1 h_1(x) + \dots + w_k h_k \geq 0 \rightarrow 1 \\ w_1 h_1(x) + \dots + w_k h_k < 0 \rightarrow -1 \end{cases}$$

- Error-Rate eines Ensembles abhängig von der Error-Rate der Base-Classifier und ihrer Anzahl:

die Häufigkeit, mit der mindestens die Hälfte der Ensemble-Mitglieder falsch abstimmt:

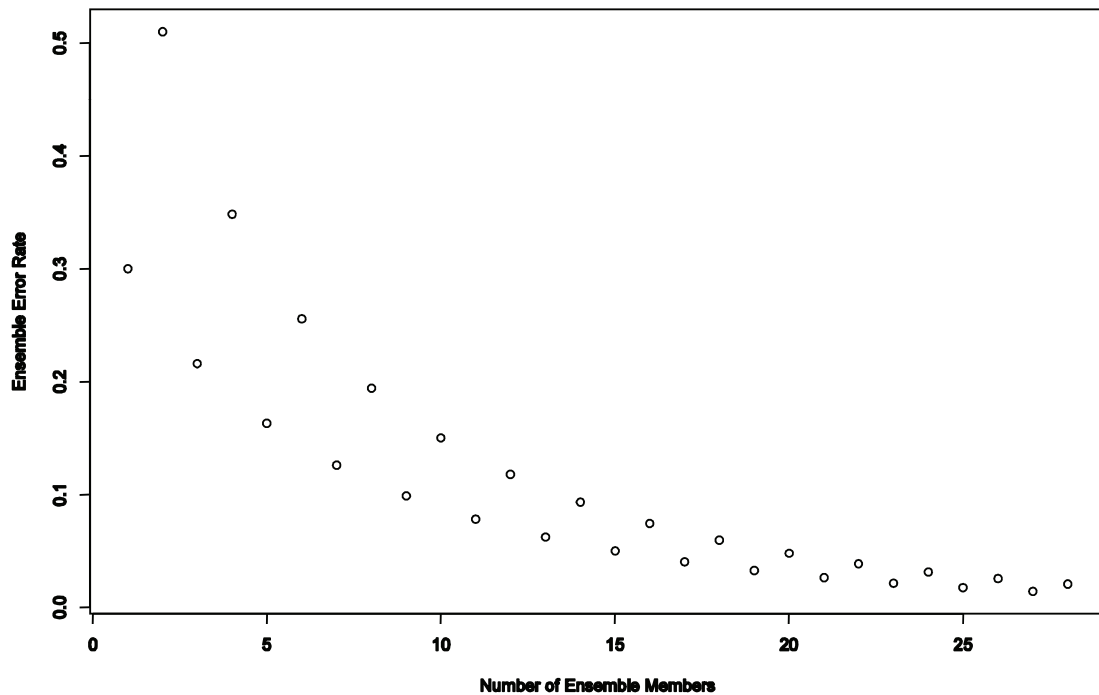
$$Err(\hat{h}) = \sum_{i=\lceil \frac{k}{2} \rceil}^k \binom{k}{i} e^i (1-e)^{k-i}$$

- (Annahme:  $Err(h_1) = \dots = Err(h_k) = e$ )

282

# Einleitung und Grundlagen

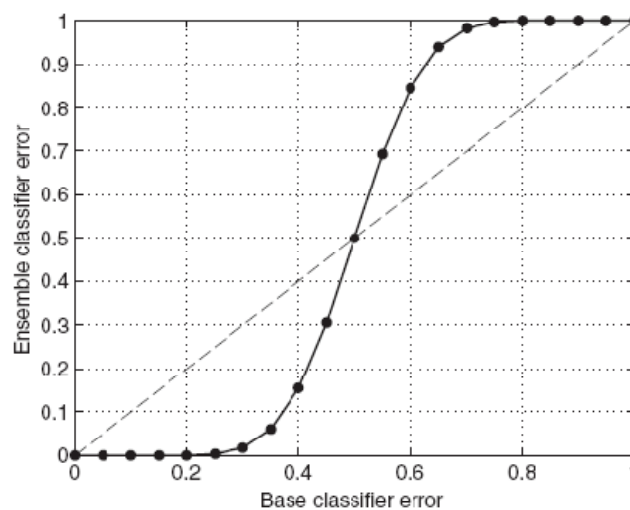
- Abhängigkeit der Gesamt-Error-Rate von der Anzahl der Base-Classifer (bei Fehlerrate der Base-Classifier von 0,3):



283

# Einleitung und Grundlagen

- Error-Rate für ein einfaches Abstimmungs-Ensemble mit 25 Basis-Klassifikatoren:



(aus: Tan, Steinbach, Kumar: Introduction to Data Mining)

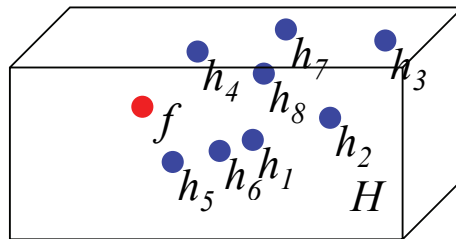
284

## Einleitung und Grundlagen

---

- Notwendige Annahme für diese Verbesserung: Unabhängigkeit der Fehler der einzelnen Base-Classifier

$$Err(\hat{h}) = \sum_{i=\lceil \frac{k}{2} \rceil}^k \binom{k}{i} e^i (1-e)^{k-i}$$



- einseitige Fehler: keine oder nur wenig Verbesserung durch Ensemble

285

## Einleitung und Grundlagen

---

- Schlussfolgerung:

Notwendige Bedingungen für Verbesserung der Gesamt-Fehlerrate:

1. Alle Base-Classifier sind "genau" (accurate).
2. Die einzelnen Base-Classifier sind "unterschiedlich" (diverse).

- Genauigkeit: milde Bedingung (besser als Zufall)
- Diversität: keine (oder wenigstens keine starke) Korrelation der Vorhersagen
- Ist gleichzeitige Optimierung von Genauigkeit und Diversität möglich?

286

# Übersicht

---

1. Einleitung und Grundlagen
2. Aspekte der Diversität
3. Methoden der Konstruktion von Ensembles

287

## Aspekte der Diversität

---

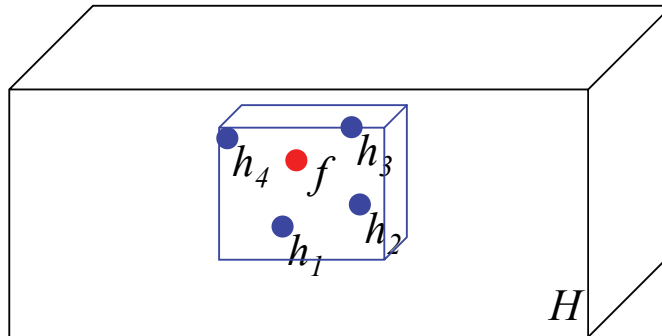
- Gründe für die Diversität von Classifiern für das selbe Klassifikationsproblem:
  - Statistische Varianz
  - Berechnungs-Varianz
  - Darstellungsproblem

288

# Aspekte der Diversität

---

- Statistische Varianz:
  - Der Raum möglicher Hypothesen ist zu groß, um anhand der begrenzten Trainingsdaten eine beste Hypothese zu bestimmen.



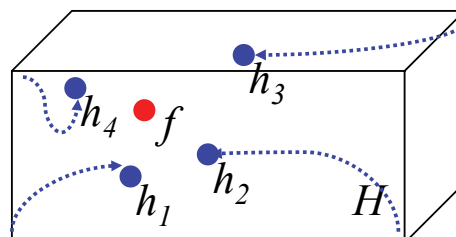
- Kombination mehrerer Hypothesen reduziert das Risiko, sehr stark daneben zu liegen.

289

# Aspekte der Diversität

---

- Berechnungs-Varianz:
  - Manche Lern-Algorithmen können nicht garantieren, die beste Hypothese aus dem Raum möglicher Hypothesen zu finden, da dies zu Berechnungsaufwändig wäre.
  - Z.B. werden beim Lernen Heuristiken verwendet, die in lokalen Optima gefangen bleiben können.



- Kombination mehrerer Hypothesen reduziert das Risiko, das falsche (lokale) Optimum gewählt zu haben.

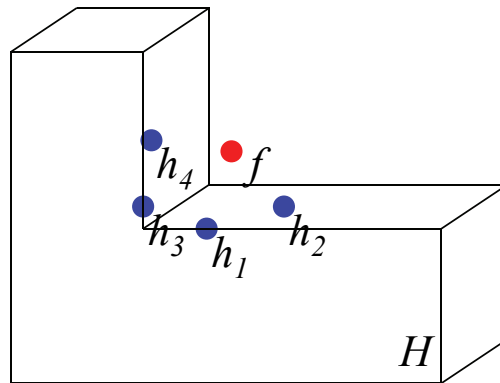
290



# Aspekte der Diversität

---

- **Darstellungsproblem:**
  - Der Hypothesenraum enthält gar keine guten Approximationen an die “wahre” Funktion  $f$ .



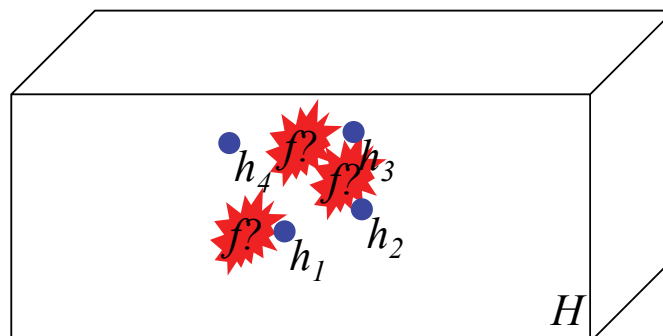
- Kombination mehrerer Hypothesen kann den Raum darstellbarer Hypothesen erweitern.

291

# Aspekte der Diversität

---

- **Unschärfe Zielfunktion:**
  - Die Lernbeispiele (Trainingsdaten) erlauben keine eindeutigen Rückschlüsse auf die Zielfunktion (z.B. wegen widersprüchlicher Beispiele oder nicht-deterministischer Klassenzugehörigkeit).



- Kombination mehrerer Hypothesen reduziert das Risiko, eine fehlerhafte Zielfunktion zu approximieren.

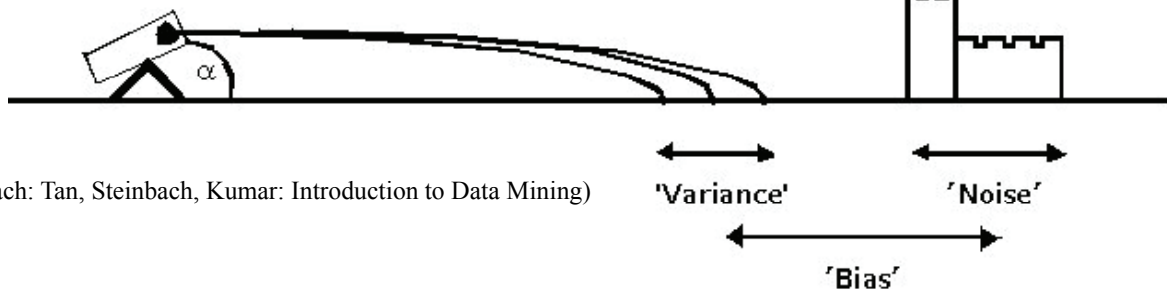
292

# Aspekte der Diversität

---

- Begriffe Varianz, Bias, Noise:

- Veranschaulichung: Beispiel aus der Ballistik



(nach: Tan, Steinbach, Kumar: Introduction to Data Mining)

- Varianz, Bias und Noise sind verschiedene Komponenten des Fehlers

$$err = Bias_{\alpha} + Variance_f + Noise_t$$

- Varianz: abhängig von der aufgewendeten Kraft  $f$
- Noise: Unschärfe des Ziels
- Bias: abhängig vom Abschusswinkel

293

# Aspekte der Diversität

---

- Begriffe Varianz, Bias, Noise in der Klassifikation:

- Varianz:

Abhängig von Variationen in den Trainingsdaten oder der Parametrisierung des Klassifikators werden unterschiedliche Hypothesen gebildet.

- Noise:

Klassenzugehörigkeit ist nicht deterministisch oder anderweitig uneindeutig (z.B. widersprüchliche Trainingsbeispiele).

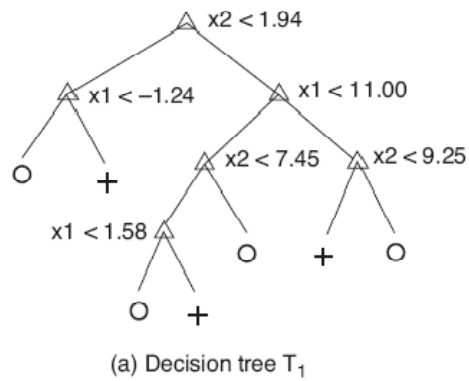
- Bias:

Ein bestimmter Lernalgorithmus hat immer auch bestimmte Annahmen über das zu erlernende Konzept (z.B. Annahme der Möglichkeit linearer Trennbarkeit verschiedener Klassen).

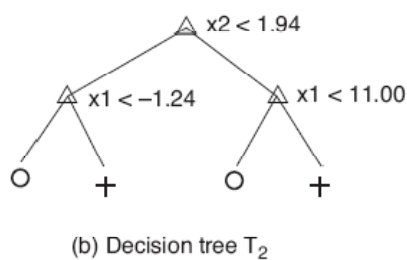
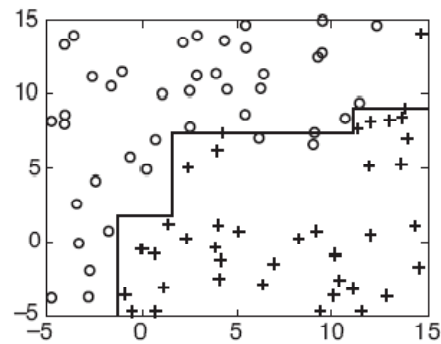
Ein Lernen ohne jede konzeptionelle Annahme wäre nur ein Auswendiglernen → "Bias-free learning is futile."

294

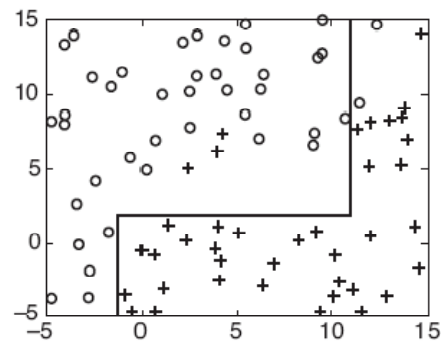
# Aspekte der Diversität



(a) Decision tree  $T_1$



(b) Decision tree  $T_2$



(aus: Tan, Steinbach, Kumar: Introduction to Data Mining)

295

# Aspekte der Diversität

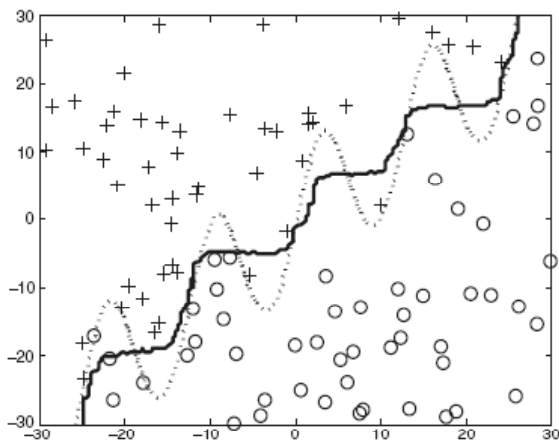
- Bias am Beispiel von Decision Trees:
  - $T_1$  und  $T_2$  wurden auf den gleichen Daten trainiert
  - $T_2$  wurde durch Pruning auf maximale Tiefe 2 aus  $T_1$  erzeugt
  - $T_2$  hat stärkere Annahmen bezüglich der Trennbarkeit der Klassen, also stärkeren Bias

296

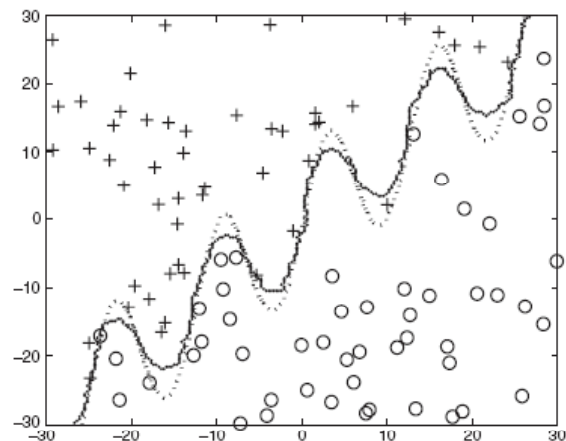
## Aspekte der Diversität

---

- relativer Beitrag von Bias und Variance zum Error ist unterschiedlich für verschiedene Klassifikationsmethoden



(a) Decision boundary for decision tree.



(b) Decision boundary for 1-nearest neighbor.

(aus: Tan, Steinbach, Kumar: Introduction to Data Mining)

297

## Aspekte der Diversität

---

- Beispiel:
  - Durchschnittliche Entscheidungsgrenzen über 100 Klassifikatoren, trainiert auf 100 unterschiedlichen Trainingsdatensätzen mit jeweils 100 Beispielen.
  - gestrichelt: wahre Entscheidungsgrenze, die zur Erzeugung der Daten benutzt wurde
  - Beobachtung:
    - geringerer Abstand der gemittelten Entscheidungsgrenze von der wahren Entscheidungsgrenze bei 1-NN Klassifikatoren  
→ niedrigerer Bias
    - größere Variabilität der einzelnen Entscheidungsgrenzen innerhalb der 100 1-NN Klassifikatoren  
→ höhere Varianz

298

1. Einleitung und Grundlagen
2. Aspekte der Diversität
3. Methoden der Konstruktion von Ensembles

---

## Methoden der Konstruktion von Ensembles

---

- Wie kann man Unterschiedlichkeit von Klassifikatoren erreichen?
  - Variieren des Training Sets
    - Methoden: Bagging und Boosting
  - Manipulieren der Input-Features
    - Lernen auf unterschiedlichen Unterräumen
    - Verwendung verschiedener Repräsentationen (MR-learning: nächstes Kapitel)
  - Manipulieren der Klassenlabel
    - Verschiedene Arten von Abbildungen auf Zwei-Klassen-Probleme
  - Manipulieren des Lernalgorithmus'
    - Einführen von Zufallselementen
    - Unterschiedliche Startkonfigurationen
    - Kombination unterschiedlicher Lernalgorithmen

## Variieren der Trainings-Menge

---

- Eine wichtige Eigenschaft von Lernalgorithmen ist die Stabilität.
- Ein Lernalgorithmus ist umso stabiler, je weniger sich die auf unterschiedlichen Trainingsdaten (für das gleiche Klassifikationsproblem) erzeugten Klassifikatoren unterscheiden.
- Bei einem instabilen Lernalgorithmus haben kleine Änderungen in der Trainingsmenge starke Änderungen der gelernten Hypothese zur Folge.
- Um Ensembles basierend auf Variationen der Trainingsmenge zu bilden, sind **instabile** Lernalgorithmen vorteilhaft, z.B.:
  - Decision Trees
  - Neuronale Netze
  - Regel-Lerner

301

## Variieren der Trainings-Menge

---

- Bootstrap:  
bilden einer Trainingsmenge aus einer gegebenen Datenmenge durch Ziehen mit Zurücklegen.
  - jedes Sample hat die gleiche Größe wie die ursprüngliche Trainingsmenge
  - ein Sample enthält durchschnittlich 63% der Ausgangsbeispiele (einige mehrfach, etwa 37% gar nicht):
    - ein einzelnes Beispiel in einem Datensatz mit  $n$  Beispielen hat bei jedem Ziehen die Chance  $1/n$  gezogen zu werden, wird also mit Wahrscheinlichkeit  $1-1/n$  **nicht** gezogen
    - nach  $n$ -mal Ziehen ist ein bestimmtes Element mit Wahrscheinlichkeit  $\left(1-\frac{1}{n}\right)^n$  nicht gezogen worden
    - für große  $n$  ist  $\left(1-\frac{1}{n}\right)^n \approx e^{-1} \approx 0.368$
  - daher auch der Name “0.632 bootstrap” für diese Sampling-Methode (als solche auch eine Alternative zur Kreuzvalidierung)

302

## Variieren der Trainings-Menge

---

- Bagging (**B**ootstrap **A**ggregating):  
bilden unterschiedlicher Trainingsmengen durch wiederholtes bootstrapping
- Bagging aggregiert mehrere Bootstraps (Samples nach obigem Muster) und trainiert auf jedem Bootstrap einen eigenen Classifier.
- Bei instabilen Lernalgorithmen werden hinreichend unterschiedliche Hypothesen erlernt.
- Ein neuer Datensatz wird durch einfache Abstimmung über alle erlernten Hypothesen klassifiziert.

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

303

## Variieren der Trainings-Menge

---

- Während der 0.632 Bootstrap unter Gleichverteilung gezogen wird, weist **Boosting** jedem Datensatz ein Gewicht zu.
- Datensätze, die schwierig zu klassifizieren sind, erhalten ein höheres Gewicht.
- Verwendung der Gewichte:
  - Angabe der Ziehungswahrscheinlichkeit im bootstrap sample der nächsten Runde
    - ➔ schwierige Beispiele sind in der nächsten Runde häufiger in der Trainingsmenge und erhalten daher automatisch ein höheres Gewicht beim Training des Klassifikators

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

- Manche Lernalgorithmen können Gewichte von Datensätzen direkt benutzen
  - ➔ Bias der erlernten Hypothese auf die höher gewichteten Beispiele hin

304

# Manipulation der Input-Feature

---

- Manipulieren der Input-Features:
  - Lernen auf unterschiedlichen Unterräumen oder kombinierten Features
    - Beispiel: Random Forests  
Menge von Decision Trees, deren Training durch Zufallsvektoren bestimmt wird, z.B.:
      - a) zufällige Auswahl von Features für den Split an jedem Knoten des Baumes
      - b) an jedem Knoten Erzeugen eines neuen Features als Linearkombination einer zufällig ausgewählten Teilmenge der Features
      - c) an jedem Knoten zufällige Auswahl aus den F besten Splits
  - Kombination von Klassifiern, die auf unterschiedlichen Repräsentationen der Daten trainiert wurden: siehe nächstes Kapitel

305

# Manipulieren der Klassenlabel

---

- Zahlreiche Methoden bilden ein Multi-Klassen-Problem auf mehrere Zwei-Klassen-Probleme ab.

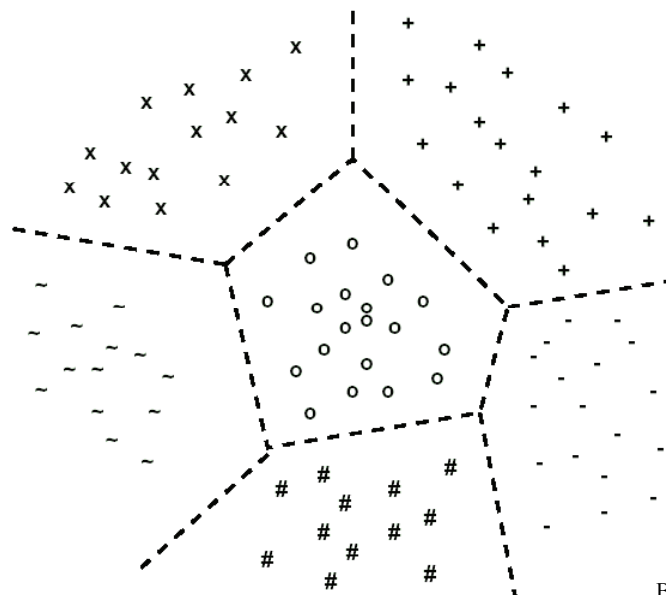


Bild aus: Fürnkranz 2002

306



# Manipulieren der Klassenlabel

---

- Die Entscheidungen der auf den einzelnen Zwei-Klassen-Problemen trainierten Klassifikatoren werden geeignet kombiniert, um auf die ursprüngliche Klasse zurückzuschließen.
- Dies entspricht dem Einführen von Unterschiedlichkeit in Klassifikatoren durch Manipulieren der Klassenlabel.
- Gängige Methoden:
  - one-versus-rest
  - all-pairs
  - error correcting output codes

307

# Manipulieren der Klassenlabel

---

- *one-versus-rest* (auch: *one-versus-all*, *one-per-class*):  
Bei  $n$  Klassen, werden  $n$  Klassifikatoren trainiert, die jeweils eine Klasse von allen anderen unterscheiden sollen.

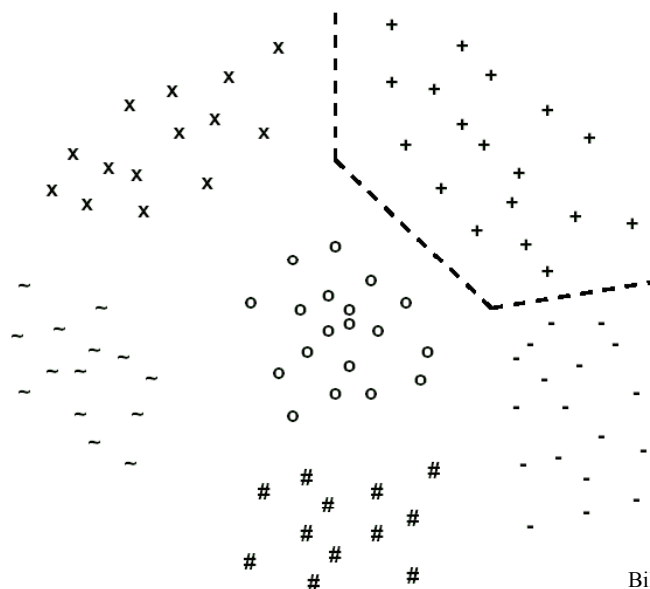


Bild aus: Fürnkranz 2002

308

## Manipulieren der Klassenlabel

---

- *all-pairs* (auch: *all-versus-all*, *one-versus-one*, *round robin*, *pairwise*):

Für jedes Paar von Klassen wird ein Klassifikator trainiert, der diese beiden Klassen unterscheiden soll.

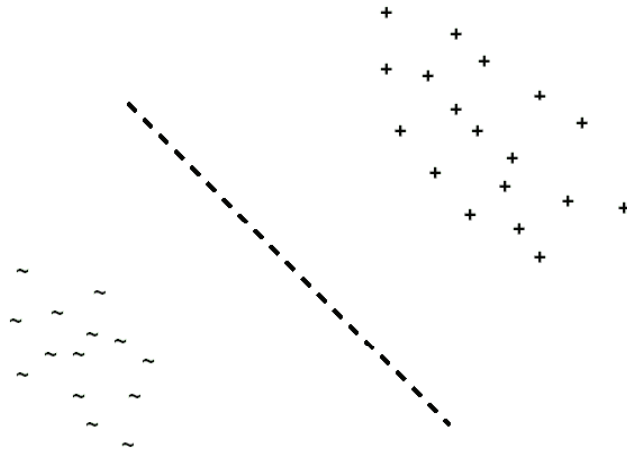


Bild aus: Fürnkranz 2002

309

## Manipulieren der Klassenlabel

---

- Error Correcting Output Codes (ECOC):
  - Die Menge  $C$  der Klassen wird  $k$ -mal zufällig in zwei Teilmengen  $A+B$  aufgeteilt.
  - Datensätze, die zu Klasse  $A$  gehören, erhalten das neue Label  $-1$ , die anderen (Klasse  $B$ ) das neue Label  $1$ .
  - Auf den entstehenden  $k$  Zwei-Klassen-Problemen werden  $k$  Klassifikatoren trainiert.
  - Stimmt Klassifikator  $i$  für Klasse  $A$ , erhalten alle Klassen aus  $C$ , die zu  $A$  gehören, eine Stimme.
  - Die Klasse  $c \in C$ , die die meisten Stimmen erhalten hat, ist die Klassifikationsentscheidung des Ensembles.

310

## Manipulieren der Klassenlabel

---

- Beispiel:  $C = \{c_1, c_2, c_3, c_4\}$ , 7-bit Kodierung

Klasse	Code-Wort						
$c_1$	1	1	1	1	1	1	1
$c_2$	0	0	0	0	1	1	1
$c_3$	0	0	1	1	0	0	1
$c_4$	0	1	0	1	0	1	0

- Für jedes Bit der Code-Wörter wird ein Klassifikator trainiert, hier also 7 Klassifikatoren.
- Angenommen, ein Klassifikationsergebnis ist  $(0,1,1,1,1,1,1)$  – für welche Klasse entscheidet das Ensemble?

311

## Manipulieren der Klassenlabel

---

- Der Name “Error Correcting Output Codes” steht für die Idee, dass beim Lernen eine gewisse Redundanz der Klassengrenzen eingeführt wird.
- Die “Code-Wörter”, die die Zugehörigkeit zu den Klassen binär codieren, können zufällig gewählt werden.
- Für eine gute Diversität sollten die Code-Wörter aber gut separieren:
  - Row-Separation: Jedes Paar von Code-Wörtern sollte eine große Hamming-Distanz (=Anzahl der unterschiedlichen Bits) aufweisen.
  - Column-Separation: Die einzelnen Binär-Klassifikatoren sollten unkorreliert sein.

312

## Manipulieren der Klassenlabel

---

Klasse	Code-Wort						
$c_1$	1	1	1	1	1	1	1
$c_2$	0	0	0	0	1	1	1
$c_3$	0	0	1	1	0	0	1
$c_4$	0	1	0	1	0	1	0

- Große Hamming-Distanz zwischen den Zeilen erlaubt möglichst eindeutige Klassifikationsentscheidung des Ensembles.
- Welche Hamming-Distanz weist das Klassifikationsergebnis  $(0,1,1,1,1,1,1)$  zu den Codes für  $c_1$ ,  $c_2$ ,  $c_3$  und  $c_4$  jeweils auf?

313

## Manipulieren des Lernalgorithmus

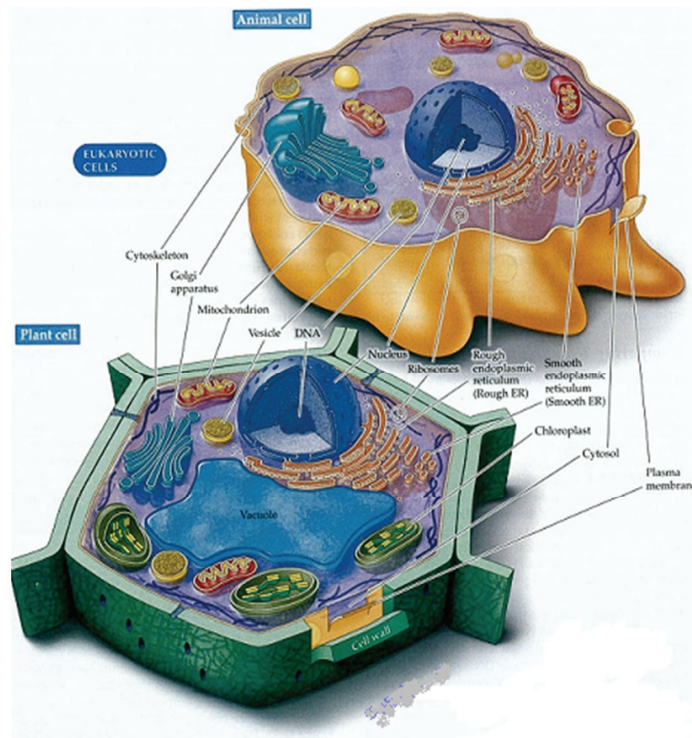
---

- Manipulieren des Lernalgorithmus durch Zufallselemente:
  - Randomisierte Entscheidungen in Decision Trees beim Split-Kriterium (vgl. Random Forests)
- Starten von unterschiedlichen Konfigurationen aus
  - Start-Gewichte für Backpropagation
  - Unterschiedliche Parametersettings

314

# Manipulieren des Lernalgorithmus

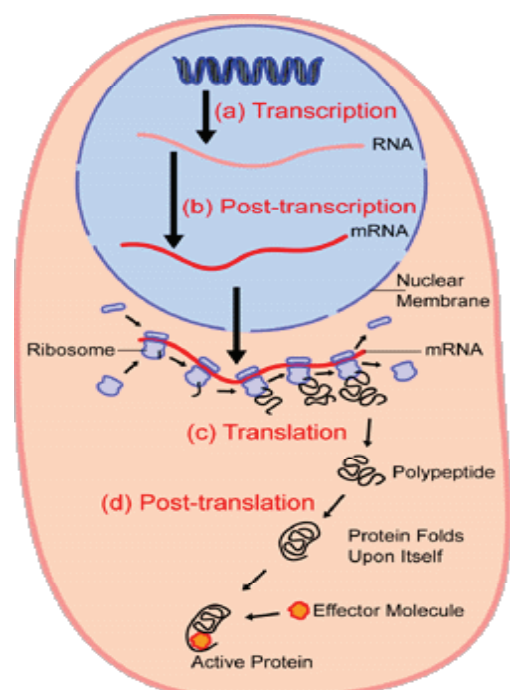
- Kombination unterschiedlicher Lernalgorithmen
  - Beispiel: Vorhersage der subzellulären Lokalisation von Proteinen (Abfalg et al. 2009)



315

# Manipulieren des Lernalgorithmus

- Ein Protein muß sich an der richtigen Stelle in der Zelle befinden, um seine Funktion erfüllen zu können.
- Die Zelle hat eine komplexe Apparatur für den Transport von Proteinen (co-translational oder post-translational).
- Vorhersagemethoden nutzen verschiedene Eigenschaften des Zell-Systems:
  - Anpassung des Proteins an bestimmtes Milieu spiegelt sich in der (Oberflächen-) Aminosäurezusammensetzung wieder.
  - Peptid-Signale aktivieren Transportproteine
  - Homologie von Proteinen legt ähnliche Funktion nahe

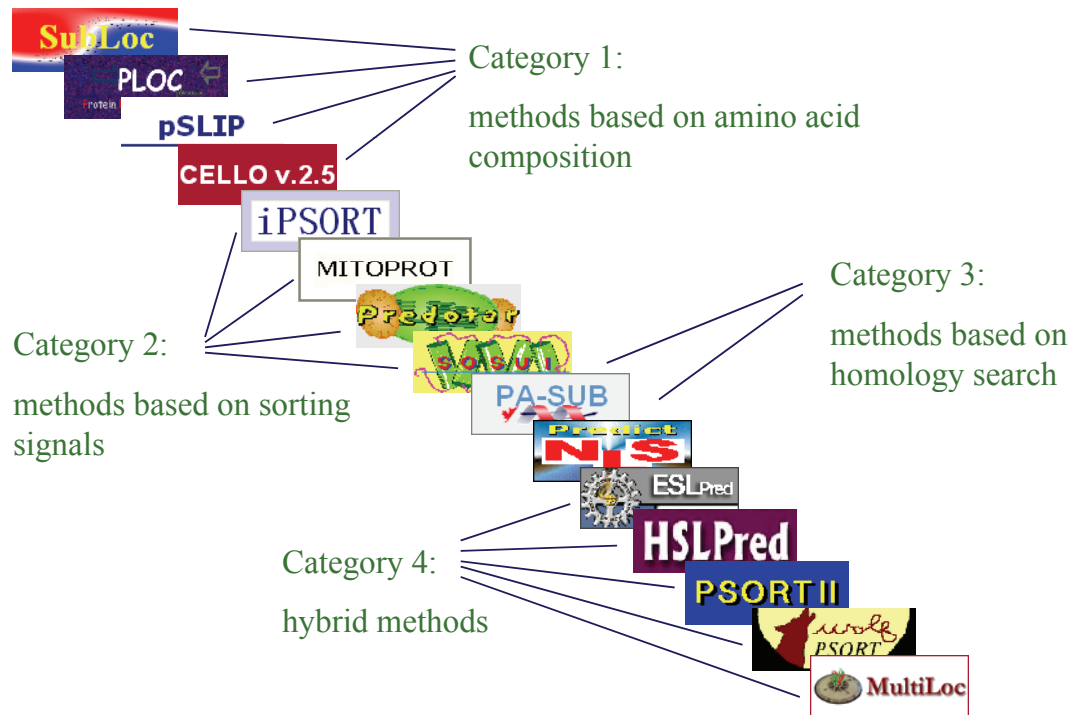


316

# Manipulieren des Lernalgorithmus

---

Zahlreiche Vorhersagemethoden nutzen unterschiedliche dieser Eigenschaften



317

# Manipulieren des Lernalgorithmus












---

- Verschiedene Vorhersagemethoden nutzen unterschiedliche Lernparadigmen:
  - naïve Bayes
  - Bayes Netzwerke
  - k-nächste Nachbarn
  - SVM
  - Neuronale Netze
  - Regel-Lerner

318

# Manipulieren des Lernalgorithmus









- Ensemble basiert auf eine Auswahl diverser Methoden

Category	Method	Foundation	Algorithm
1		Aa	SVM
		Dipeptide	SVM
		n-peptide	SVM
2		detecting sorting signals	AA-index
		detecting sorting signals	NN
3		BLAST against Swiss-Prot	Naive Bayes
4		aa+signal+motif+structure	k-NN
		aa+length+signal	k-NN
		aa+signal+motif+structure	SVM
		aa+di+properties+psi-BLAST	SVM
		aa+di+gap+properties+psi-BLAST	SVM

319

# Manipulieren des Lernalgorithmus

- Ensemble basiert auf eine Auswahl diverser Methoden

Category	Method	Foundation	Algorithm
too simple foundation, lower rank in preliminary tests <sup>SVM</sup>			
1		dipeptide	SVM
		n-peptide	SVM
2		detecting sorting signals	AA-index
		detecting sorting signals	NN
based on virtually all SWISSPROT entries that provide a localization <sup>yes</sup>			
PSORT II extension WoLFPSORT is used			
4		aa+length+signal	k-NN
		aa+signal+motif+structure	SVM
		aa+di+properties+psi-BLAST	SVM
		aa+di+gap+properties+psi-BLAST	SVM

320

# Manipulieren des Lernalgorithmus

---

- Ein Entscheidungsbaum lernt die Zuverlässigkeit der Vorhersagen der diversen Methoden für verschiedene Lokalisationen.
- Mapping eines Vorhersagevektors (Vorhersagen der verschiedenen Methoden) auf eine Vorhersage durch den Entscheidungsbaum.
- Beispiel:  
*“If CELLO predicts class 6 and WoLFPSORT predicts class 4, then class 4 is correct.”*

321

---

## Literatur

---

- J. Aßfalg, J. Gong, H.-P. Kriegel, A. Pryakhin, T. Wei, A. Zimek: **Supervised Ensembles of Prediction Methods for Subcellular Localization**. In: J. Bioinformatics and Computational Biology 7(2): 269-285 (2009)
- T. G. Dietterich: **Ensemble methods in machine learning**. In: Multiple Classifier Systems, First International Workshop, MCS 2000, Cagliari, Italy, 2000.
- T. G. Dietterich: **Ensemble learning**. In: M. A. Arbib, editor, The Handbook of Brain Theory and Neural Networks. MIT Press 2003.
- J. Fürnkranz: **Round robin classification**. In: Journal of Machine Learning Research, 2:721-747, 2002.
- P.-N. Tan, M. Steinbach, and V. Kumar: **Introduction to Data Mining**, Addison-Wesley, 2006, Kapitel 5.6+5.8.
- G. Valentini and F. Masulli: **Ensembles of learning machines**. In: Neural Nets WIRN Vietri 2002.

322