

5.2 Unüberwachte Lernverfahren

Erinnerung Folie 228

- **Unüberwachtes Lernen:** Gegeben Merkmalvektoren $\langle \mathbf{In} \rangle$, leite Fkt. f ab, die Regularitäten/Einteilungen beschreibt

Hier behandelte Verfahren

- Induktives Logisches Programmieren (ILP)
(nach Russell/Norvig)
- Clustering-Verfahren

ILP: Lernen in Logik

Beispiel: Aus Lernbeispielen (und evtl. Weiterem) erzeuge z.B.

$$\begin{aligned}\forall r. \text{Warten}(r) \Leftrightarrow & \text{Gäste}(r, \text{Einige}) \\ & \vee [\text{Gäste}(r, \text{Voll}) \wedge \text{Hungrig}(r) \wedge \text{Typ}(r, \text{Französisch})] \\ & \vee [\text{Gäste}(r, \text{Voll}) \wedge \text{Hungrig}(r) \wedge \text{Typ}(r, \text{Thai}) \wedge \text{Frei/Sams}(r)] \\ & \vee [\text{Gäste}(r, \text{Voll}) \wedge \text{Hungrig}(r) \wedge \text{Typ}(r, \text{Burger})]\end{aligned}$$

... und bevorzuge „einfache“ Formeln (*Ockham's Razor*)!

Vorteile

- Überwachtes Lernen (wie mit DTL) ist echte Untermenge davon
- Lerne beliebige Prädikate/Relationen (nicht nur 1-stell.)
- Kann Regeln induzieren ohne explizite Lernbeispiele
- Kann „Hintergrundwissen“ einbeziehen bzw. ausbauen

(Potenzieller) Nachteil

- Größere Ausdrucksfähigkeit macht Lernen komplexer

Struktur des ILP-Lernproblems

... am Beispiel von Klassifikation:

Wissensbasiertes induktives Lernen (**KBIL**):

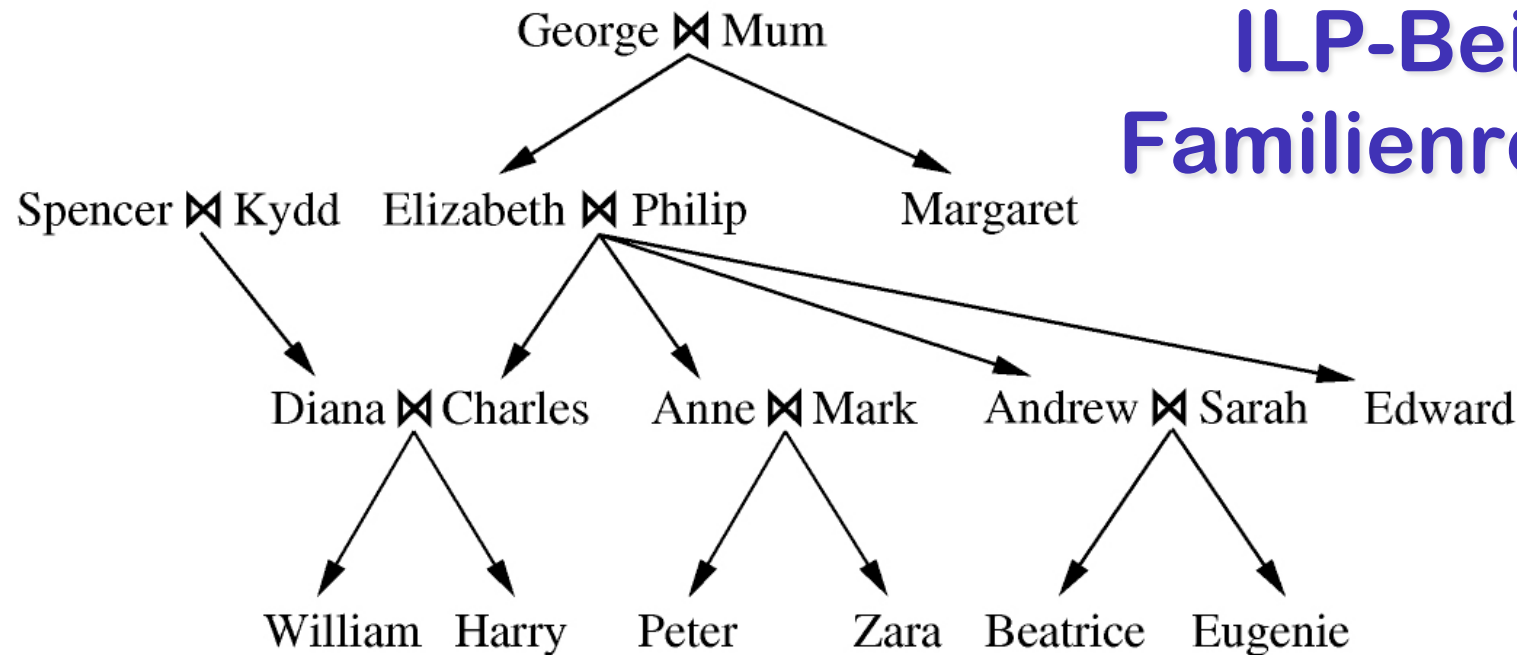
$$KB \wedge Hypothese \wedge Lernbeispiele \models Klassifikation$$

- Hypothese ist zu finden \rightarrow **induktives** Lernverfahren
- Hypothese muss konsistent sein mit $KB \wedge Lernbeispiele$

Hier: **Induktives Logisches Programmieren (ILP)**

Verwendet CWA-Folgerung \models_{cwa} wie Prolog (Folie 110)

ILP-Beispiel: Familienrelationen



Gegeben:

- Vollständige Repräsentation des Stammbaums mittels Prädikaten *Vater*, *Mutter*, *Paar*, *Mann*, *Frau*
- einige der 20x20 Instanzen (pos., neg.) des Prädikats $Opa(x,y)$

Gesucht:

- Definition von $Opa(x,y)$ in Termini der anderen Prädikate
- Form der Definition:
Disjunktion von Hornklauseln

Beispiel: Opadefinitionskandidaten

Gegeben: $Opa(\text{George}, \text{Anne})$, $Opa(\text{Philip}, \text{Peter})$, $Opa(\text{Spencer}, \text{Harry})$,
 $\neg Opa(\text{Anne}, \text{Anne})$, $\neg Opa(\text{Harry}, \text{Zarah})$, $\neg Opa(\text{Charles}, \text{Philip})$

Kandidaten f. Definitionen („PROLOG-artige“ Notation, aber „ \Rightarrow “)

- $\Rightarrow Opa(x,y)$.
Richtig für alle Beispiele; falsch f. a. Gegenbeispiele
↳ spezialisieren (durch „Raten“)! Zufällige Kandidaten:
- $Vater(x,y) \Rightarrow Opa(x,y)$. (Falsch für alle Beispiele)
- $Paar(x,z) \Rightarrow Opa(x,y)$. (Falsch für einige (CWA!) Gegenbeispiele)
- $Vater(x,z) \Rightarrow Opa(x,y)$. (Falsch für weniger Gegenbeispiele)
- ... wähle #3 zum Spezialisieren etc.

... bis Klauseln gefunden, die alle Positiv-, keine Negativbeispiele implizieren ↳ deren Disjunktion ist die Definition!

Klauselkopf
in Prolog

FOIL (*First Order Inductive Learner*)

```
function FOIL(examples, target) returns a set of Horn clauses
inputs: examples, set of examples
         target, a literal for the goal predicate
local variables: clauses, set of clauses, initially empty

while examples contains positive examples do
    clause ← NEW-CLAUSE(examples, target)
    remove ✓ examples covered by clause from examples
    add clause to clauses
return clauses
```

✓positive

- Potenziell jede Klausel der Sprache f. NEW-CLAUSE möglich, die kein Gegenbeispiel wahr macht
- Heuristiken/Bedingungen zu Klauseln: # Variablen, Länge, ...
- Mehr bei Russell/Norvig, Kap. 19.5

Inverse Resolution

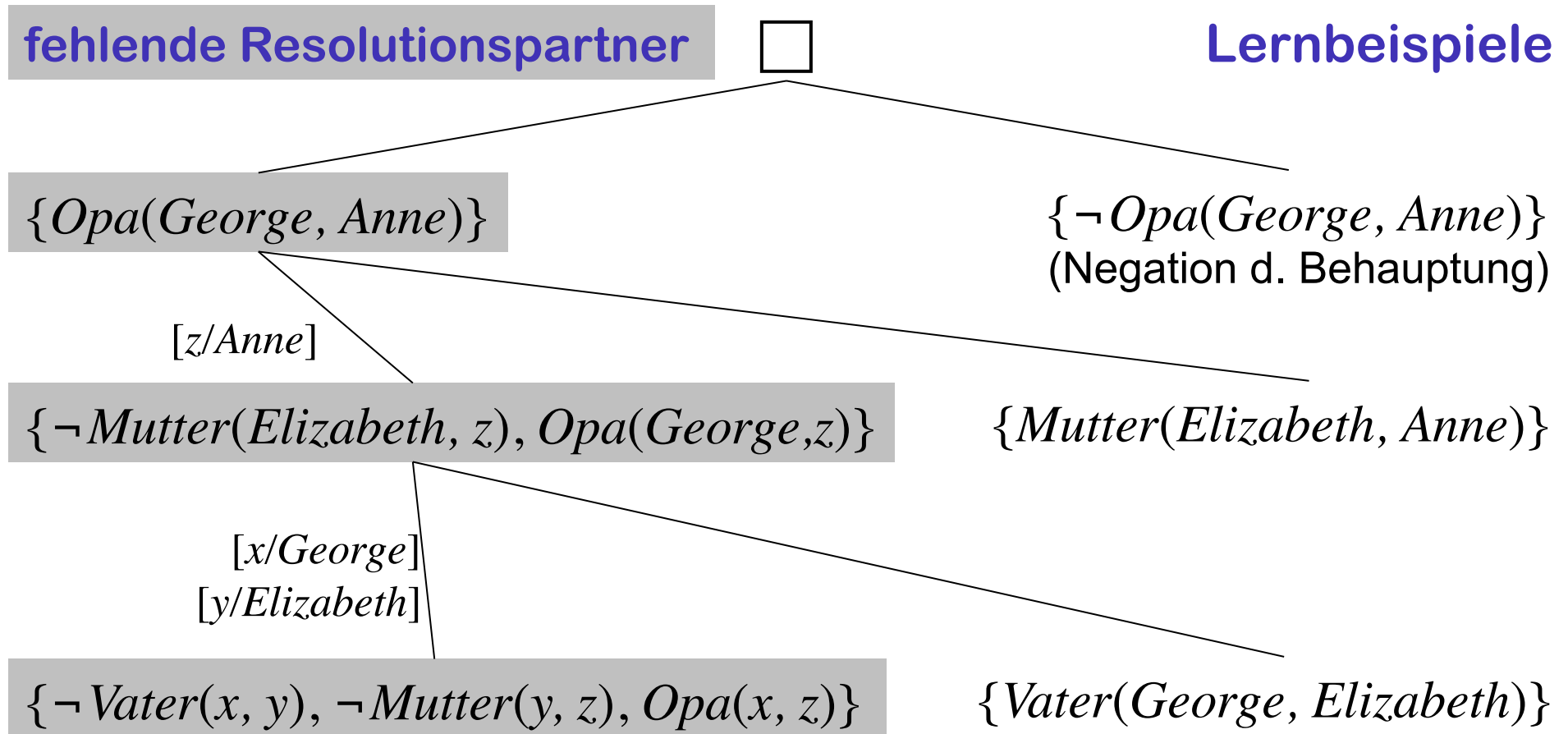
- Der Suchraum beim „Klauselraten“ à la FOIL ist riesig!
- Wenn das Gelernte die Form
 $KB \wedge Hypothese \wedge Lernbeispiele \models Klassifikation$
haben soll, muss
 $KB \wedge Hypothese \wedge Lernbeispiele \wedge \neg Klassifikation$
durch Resolution widerlegbar sein
- Könnte man dann nicht statt Klauselraten gezielt solche Klauseln als *Hypothese* nehmen, die für die Widerlegung von
 $KB \wedge Lernbeispiele \wedge \neg Klassifikation$
fehlen?

➡ Inverse Resolution

Konstruiere „rückwärts“ Klauseln, die zur Resolution fehlen, bis zu einer „Definition“ (Hornklausel) des gesuchten Prädikats

Beispiel für Inverse Resolution

Ziel: widerlege Lernbeispiel $Opa(George, Anne)$ mittels anderer Lernbeispiele, um Definition von $Opa(x, z)$ zu hypothetisieren



ILP und die Praxis

- Um Klauselraten/FOIL oder Inverse Resolution praxistauglich zu machen, gibt es viel an Theorie und Heuristiken
- Es ist sogar möglich, neue, Sinn tragende Prädikate zu „entdecken“, welche die Klauselmenge „kompakter“ machen
- Mehr bei Russell/Norvig, Kap. 19.5
- ILP-Systeme werden praktisch beim *data mining* eingesetzt



Logo der Firma von Ross Quinlan, Entwickler von FOIL, www.rulequest.com/

Clustering-Verfahren

Ziel

- Finde neue „Ballungen“/Gruppierungen in vorhandenen Daten
- Unterschied zu Klassifikation: Klassen sind hier nicht vorher bekannt/gegeben (unüberwachtes Verfahren!)

Beispiel

- In Volltextdatenbank sortiere Dokumente, die unterschiedlichen Bedeutungen desselben Worts entsprechen (z.B. „Decke“)
- Daten: Lexikon mit z.B. 50.000 Wörtern (einschl. Flexionsformen und Ableitungen: z.B. „Decke“, „Decken“, „decken“, „Zudecke“ sind unterschiedliche Wörter)
- Jedes Dokument repräsentiert durch 50.000-stell. Vektor, x_i , $1 \leq i \leq 50.000$, ist Häufigkeit von Wort i im Dokument
- Gruppiere Dokumente mit „Decke“ nach häufigen Kontexten („schlafen“, „Bett“, „Kissen“, ... vs. „Stuck“, „Farbe“, „Lampe“, ...)

k-Means Clustering

Idee

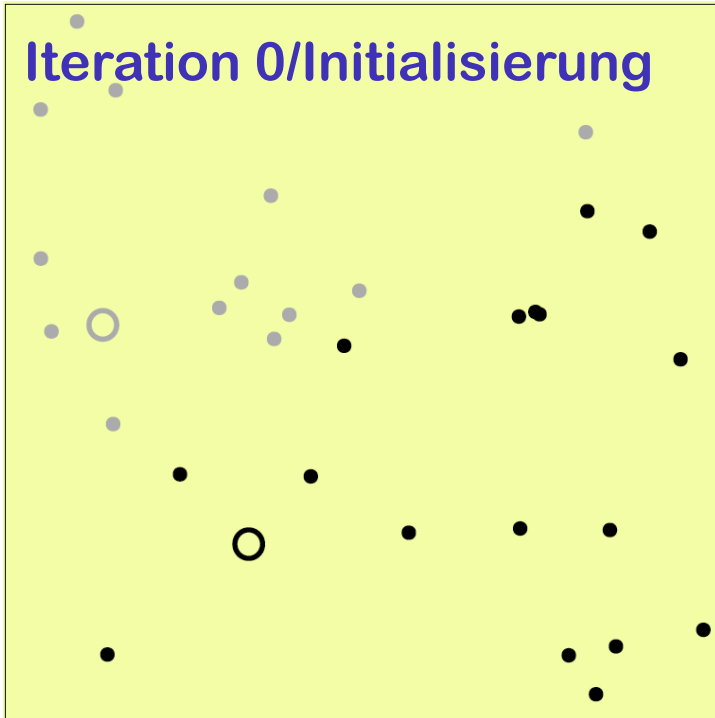
- Ein Cluster entspricht Datenpunkten, die „benachbart“ sind
- Abstandsmaße s. Folie 232, z.B. Euklidischer Abstand im \mathcal{R}^n

k-Means Clustering

- Voraussetzung: Es ist bekannt/erwartet, dass in den Daten k Cluster zu finden sind!
- Verfahren: wähle initial (z.B. zufällig) k Punkte μ_1, \dots, μ_k im Datenraum als Clustermittelpunkte;
dann wiederhole bis Konvergenz:
 - ordne jeden Datenpunkt dem nächsten (\hookrightarrow Abstandsmaß!) der k Clustermittelpunkte μ_i zu
 - aktualisiere alle μ_i als Mittelwerte der Datenpunkte, die ihnen zugeordnet sind

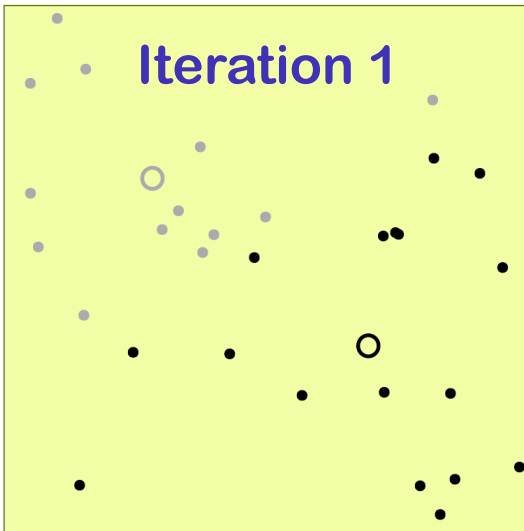
Beispiel im \mathcal{R}^2

Iteration 0/Initialisierung

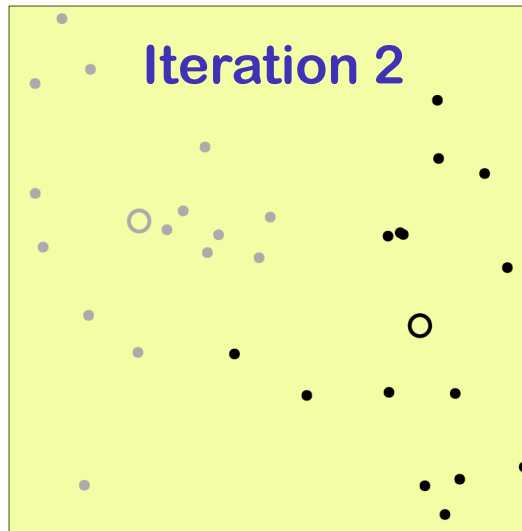


- initial μ_i (Kringel) zufällig gewählt; zugehörige Datenpunkte gleich eingefärbt
- in Iterationen: μ_i (Kringel) sind Mittelpunkte aller Punkte gleicher Farbe in voriger Iteration

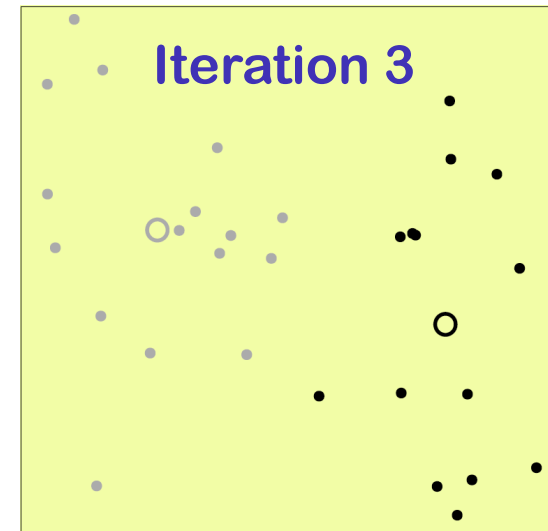
Iteration 1



Iteration 2



Iteration 3



k-Means Clustering Algorithmus

K-MEANS($\mathbf{x}_1, \dots, \mathbf{x}_N, k$)

initialisiere μ_1, \dots, μ_k (z.B. zufällig)

zuordne

Repeat

Klassifiziere $\mathbf{x}_1, \dots, \mathbf{x}_N$ zum jeweils nächsten μ_i

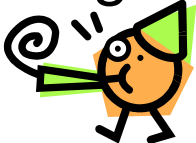
Berechne μ_1, \dots, μ_k neu

Until keine Änderung in μ_1, \dots, μ_k

Return(μ_1, \dots, μ_k)

= keine Änderung der Zuordnung
von Datenpunkten zu μ_i

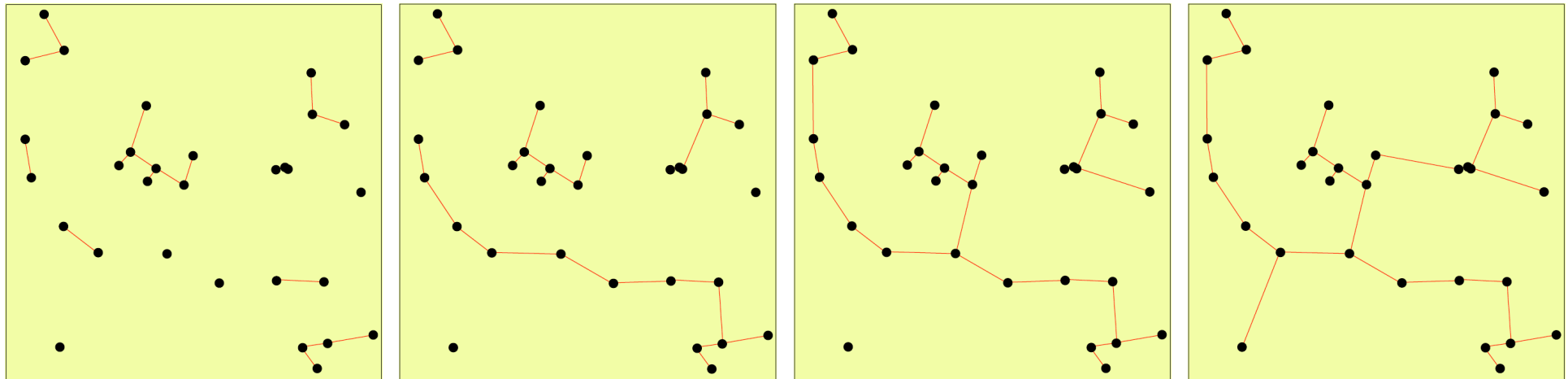
Eigenschaften von k-means

- Ertel sagt „keine Konvergenzgarantie“, gibt aber eine Komplexität an: ???! 
- Optimales k -Clustering zu finden, ist NP-vollständig
- Algorithmus konvergiert in der Regel schnell, aber nicht notwendig mit optimalen k Clustern (kleinste Abstandssumme)
- Startpositionen der μ_i beeinflussen Ergebnis und effektive Laufzeit
- Unterschiedliche k können in Clustern von drastisch unterschiedlicher Qualität resultieren!
- Einzelne Cluster können leer laufen (und bleiben dann leer)
- Da optimale/tatsächliche Clusterzahl k i.A. unbekannt: Restart des Algorithmus mit Variation von k und Start- μ_i

Hierarchisches Clustering

Idee

- Cluster „wachsen“ aus einzelnen Datenpunkten, die mit ihren Nachbarpunkten zusammengefasst werden
- Abbruch nach vorgegebenem Kriterium (Clusterzahl, Clusterweite, , ...)



- Selber Datensatz wie bei k-Means vorher;
beachte völlig unterschiedliche Gestalt der gefundenen Cluster!

Algorithmus Hierarchisches Clustering

HIERARCHISCHES-CLUSTERING($\mathbf{x}_1, \dots, \mathbf{x}_N, k$)

initialisiere $C_1 = \{\mathbf{x}_1\}, \dots, C_n = \{\mathbf{x}_N\}$

Repeat

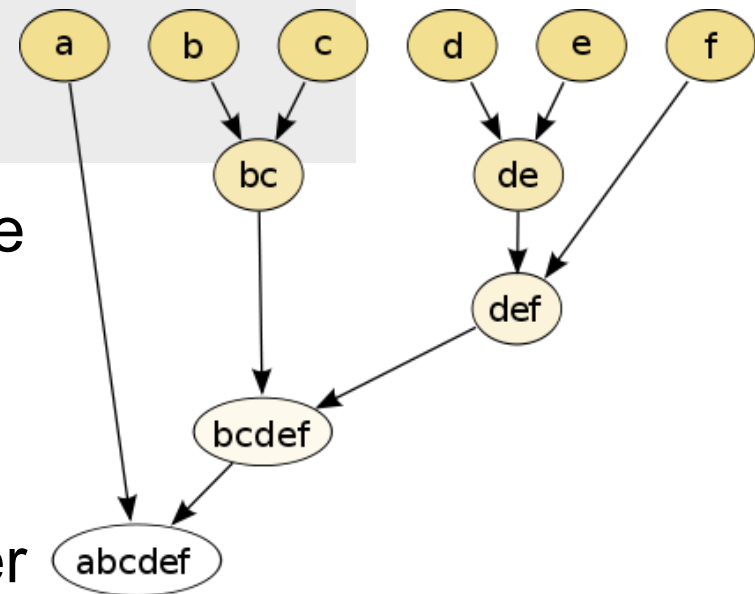
Finde zwei Cluster C_i und C_j mit kleinstem Abstand

Vereinige C_i und C_j

Until Abbruchbedingung erreicht

Return(Baum mit Clustern)

- Abstand d. Cluster sei Abstand (Maße s. Folie 232) der nächstgelegenen (Rand-)Punkte aus beiden Clustern
- Alternativen: Betrachte Abstand der Clustermittelpunkte oder Minimum der entferntesten Clustermittglieder



Eigenschaften des Hierarchischen Clusters

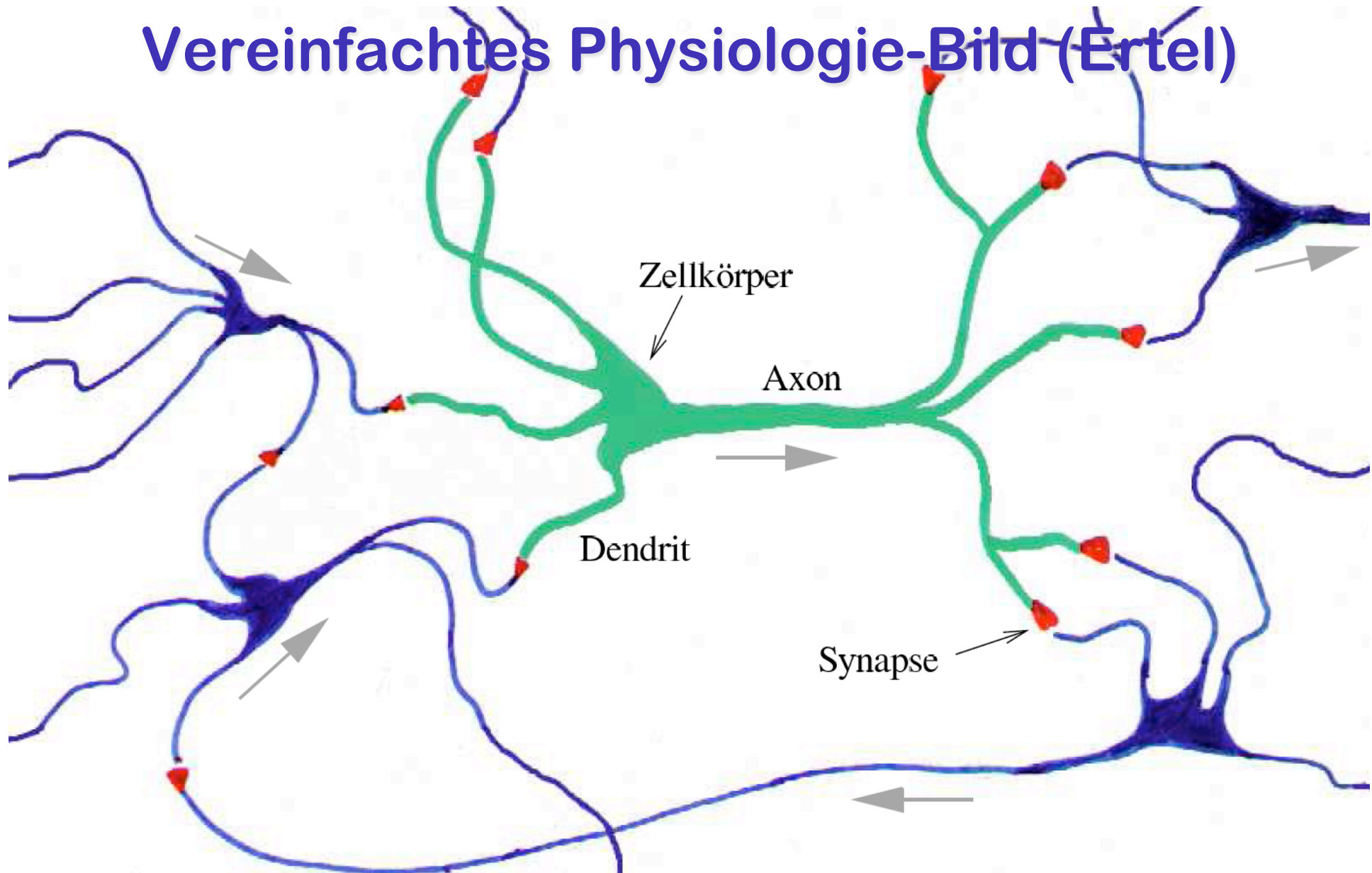
- Terminiert garantiert
- gefundene Cluster hängen ab von Abstandsmaß und Abbruchbedingung
- Abstandsermittlung implementiert z.B. über Adjazenzmatrix: Speicher $O(N^2)$
- Für Clustern mach max. $N-1$ Durchläufe, also Zeit $O(N^3)$

5.3 Vertiefung: Neuronale Netze

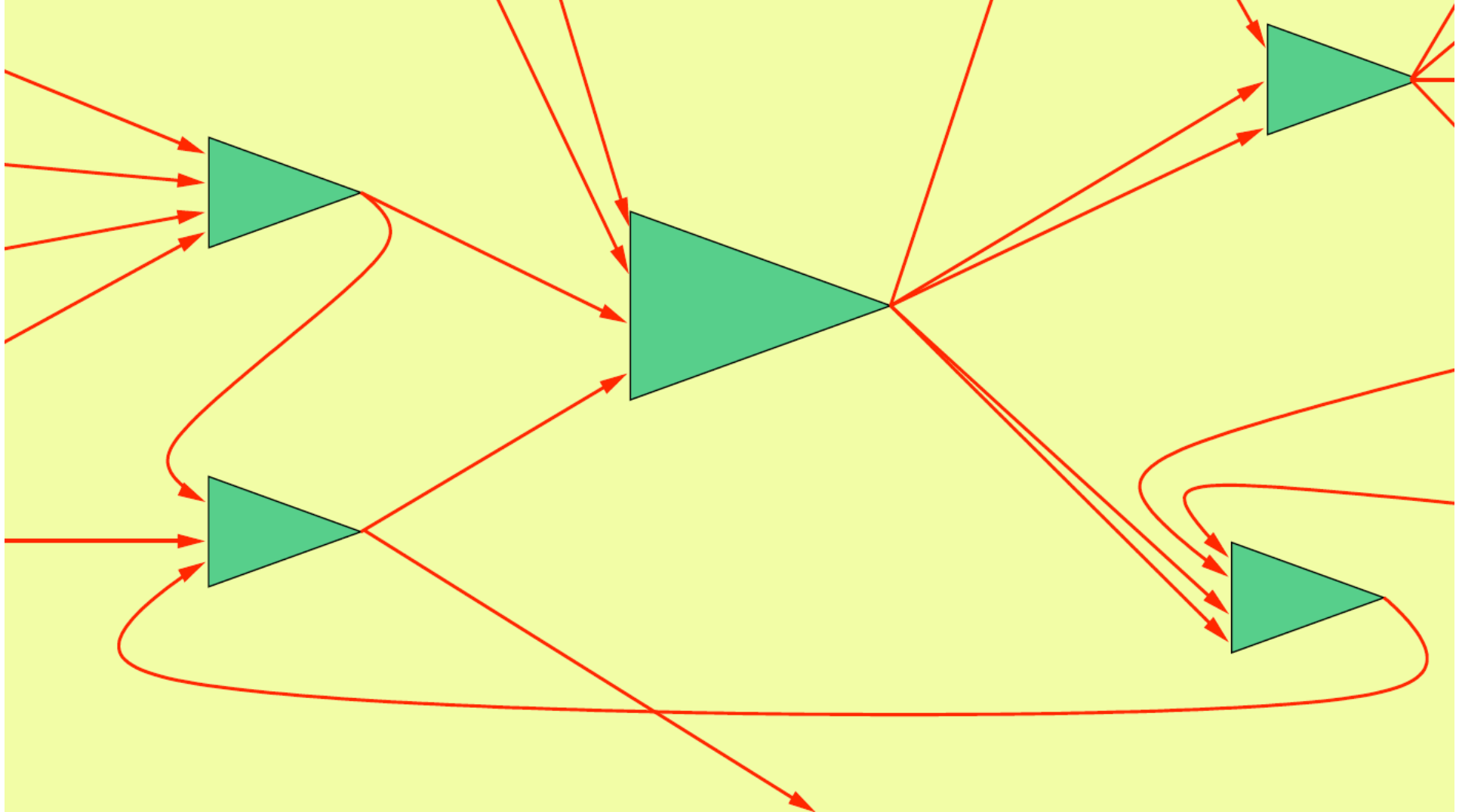
Ertel schreibt uns

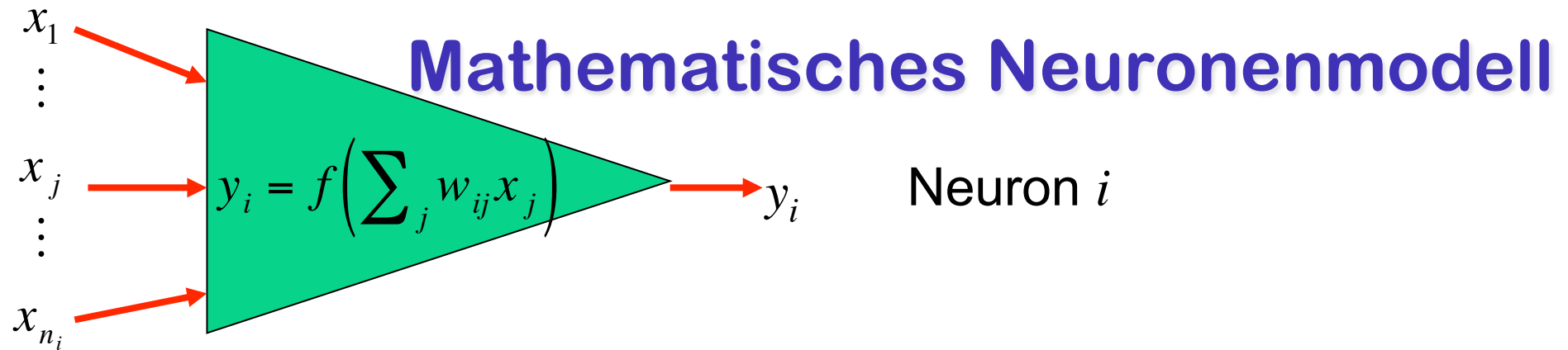
- Menschliches Gehirn hat ca. 10-100 Milliarden Nervenzellen (Neuronen)
- Jedes Neuron verbunden mit ca. 1.000–10.000 anderen, also Größenordnung: über 10^{14} Verbindungen
- Erstes mathematisches Modell der Signalverarbeitung durch Neuronen: McCulloch/Pitts 1943
- Dieses Modell aufgenommen durch „Bionik-Zweig innerhalb der KI“ (Ertel) (s. dazu Kritik auf Folie 303!!)

Vereinfachtes Physiologie-Bild (Ertel)



Schematisierung der Neuronen





- **Aktivierung** des Neurons i ist gewichtete Summe der Inputs
- **Ausgabe** y_i definiert über Schwellwertfunktion, z.B.

$$f(z) = H_\theta(z) = \begin{cases} 0 & \text{falls } z < \theta \\ 1 & \text{sonst} \end{cases}$$

(alternativ: Sigmoid-Funktion; vgl. Perzeptron mit Schwelle θ , Folie 234)

- Wert y_i ist Eingabe für andere Neuronen
- Aktivierungen und Ausgaben aller Neuronen für Zeit $t+1$ werden synchron aus Werten für t berechnet