

Unüberwachte Lernverfahren

Prof. Dr.-Ing. Rüdiger Dillmann

Prof. Dr.-Ing. J. Marius Zöllner



Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft



Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

- Motivation & Einführung (3-5)
- k-means Clustering (6-13)
- Hierarchisches Clustering (14-20)
- Begriffliche Ballungen & COBWEB (22-37)
- Ausblick (38)

Warum unüberwachtes Lernen?

- Sammeln und Klassifizieren von Trainingsdaten kann sehr aufwändig sein (z.B. Spracherkennung)
- Engineering z.B: Merkmalsberechnung der Daten kann sehr aufwendig sein
- Data Mining
- Sich verändernde Charakteristika von Mustern
- Finden von neuen Eigenschaften
- Erste Erkenntnisse über Struktur von Daten

- Ausnutzen von Ähnlichkeiten in Trainingsdaten, um
 - die Klassen / Ballungen zu erschließen
 - oder um die wesentlichen Charakteristika= Merkmale aus den Daten zu verwenden – siehe Ausblick

- Analogie zum menschlichen Lernen:
 - Schüler lernt graduell Konzepte
 - Beobachtet eine Menge von Objekten / Ereignissen
 - Formt dabei (hierarchische) Konzepte, die seine Erfahrungen zusammenfassen und organisieren
 - Findet die richtige Repräsentation von Daten was ihm ermöglicht schnell zu lernen

Unüberwachte Lernverfahren (Schwerpunkt Cluster)

- Klassische Ballungen
 - k-means-Clustering
 - Agglomerative Hierarchical Clustering
- Begriffliche Ballungen („Conceptual Clustering“)
 - CLUSTER/2
- Bildung von Begriffshierarchien („Concept Formation“)
 - COBWEB
 - CLASSIT
- Lernen durch Entdeckung
 - BACON
 - ABACUS
- ...

k-means-Clustering (Lloyd, 1982)

- Sehr elementar aber populär
- Klassifiziert eine Datenmenge in eine a-priori vorgegebene Anzahl von Ballungen
- Grundidee:
 - Definieren eines Mittelpunkts für jeden Cluster
 - Iterative Anpassung / Verbesserung
 - Optimalitätskriterium: Minimierung der Abstände aller Datenpunkte von ihrem Ballungsmittelpunkt

k-means-Clustering Formal

■ Gegeben:

- Menge X von **unklassifizierten** Trainingsbeispielen mit jeweils d Attributen: $x_i = \langle \text{attr } 1_i, \text{attr } 2_i, \dots, \text{attr } d_i \rangle$
- Anzahl der gesuchten Ballungen k

■ Gesucht:

- Einteilung der Trainingsmenge in Ballungen X_1, \dots, X_k (mit Zentren c_1, \dots, c_k) unter Minimierung von

$$\sum_{j=1}^k \sum_{x_i \in X_j} |x_i - c_j|$$

k-means-Clustering Algorithmus

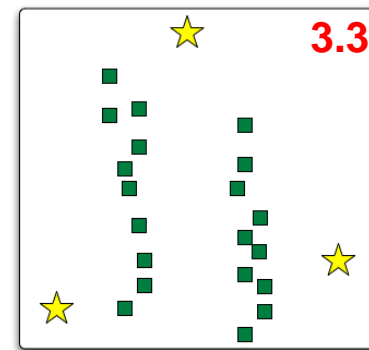
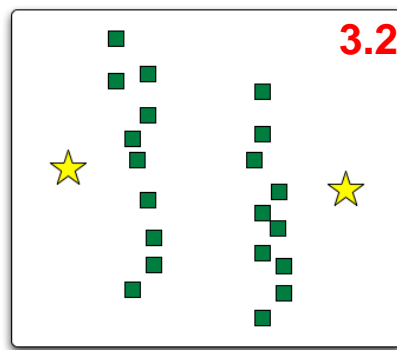
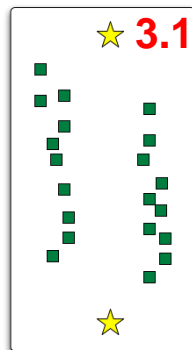
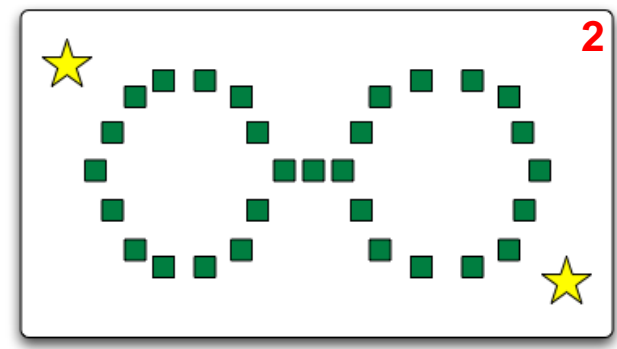
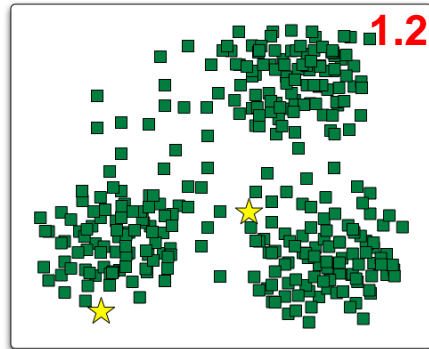
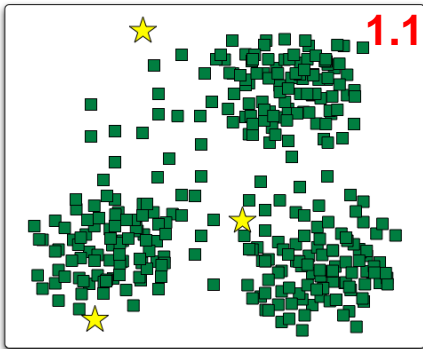
- **Algorithmenbeschreibung:**
- Platziere k Punkte c_j im d -dimensionalen Raum als initiale Mittelpunkte der Ballungen
- Bis sich die c_j nicht mehr ändern:
 - Klassifiziere jedes x_i gemäß der c_j neu:

$$l = \arg \min_{j=1}^k |x_i - c_j| \Rightarrow x_i \in X_l$$

- Berechne die Mittelpunkte c_j zu jeder Ballung neu:

$$c_j = \frac{\sum_{i=1}^{|X_j|} x_i}{|X_j|}$$

k-means-Clustering Beispiel(e)



- Resultate hängen stark von der initialen Belegung der c_j ab
 - Evtl. werden suboptimale Lösungen gefunden
 - Nochmal Beispiel 1.2: Fehlschlag!
 - Mögliche Lösung: Algorithmus mehrfach mit unterschiedlichen Startpunkten anstoßen

- Resultate hängen von der verwendeten Metrik $|x - c_j|$ ab
 - Curse of dimensionality! In hochdimensionalen Repräsentationen sind alle Daten unähnlich → schwer Cluster zu finden

- Resultate hängen von der korrekten Wahl von k ab
 - Keine fundierten theoretischen Lösungen
 - Ergibt sich k aus der Domäne?
 - Buchstabenerkennung $\Rightarrow k = 26$
 - Mehrmaliges Anstoßen des k-means-Verfahren mit unterschiedlichen k . Abbruch, wenn Ergebnis bestimmten Optimalitätskriterium genügt
 - Problem: Overfitting!

- Beim normalen k-means: jeder Datenpunkt in genau einem Cluster
- Abschwächung: jeder Datenpunkt x_i hat eine abgestufte / „unscharfe“ Zugehörigkeit zu jedem Cluster X_j
 - $p(X_j|x_i)$: „Wahrscheinlichkeitsmaß für die Zugehörigkeit“
 - $p(X_j|x_i) \sim 0$: Datenpunkt weit von Cluster entfernt
 - $p(X_j|x_i) \sim 1$: Datenpunkt nahe bei Cluster
 - p ist normiert über die Ballungen X_j
 - Neuberechnung der X_j durch Adaption von c_j unter Beachtung der unscharfen Zugehörigkeit aller Datenpunkte und von p für jedes x_i
- Problem: Laufzeit = $O(kn)$ je Iteration

- Cluster-Zugehörigkeit von Punkt x_i :

$$P(c_j|x_i) = \frac{\left(\frac{1}{d_{ij}}\right)^{\frac{1}{b-1}}}{\sum_{r=1}^k \left(\frac{1}{d_{ir}}\right)^{\frac{1}{b-1}}} \quad \text{mit } d_{ij} = |x_i - c_j|^2$$

- p ist normiert über die Ballungen X_j

$$\forall i = 1, \dots, n : \sum_{j=1}^k P(c_j|x_i) = 1$$

- Adaption von c_j unter Beachtung der unscharfen Zugehörigkeit aller Datenpunkte x_i

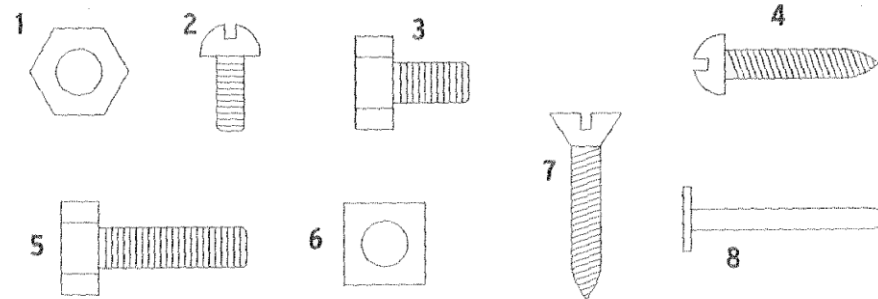
$$c_j = \frac{\sum_{i=1}^n [P(c_j|x_i)]^b \cdot x_i}{\sum_{i=1}^n [P(c_j|x_i)]^b}$$

- Parameter b kontrolliert, Abschwächung des Einflusses einzelner Punkte mit der Entfernung

Einordnung k-means-Clustering

Typ der Inferenz	<i>induktiv</i>	↔	<i>deduktiv</i>
Ebenen des Lernens	<i>symbolisch</i>	↔	<i>subsymbolisch</i>
Lernvorgang	<i>überwacht</i>	↔	<i>unüberwacht</i>
Beispielgebung	<i>inkrementell</i>	↔	<i>nicht inkrementell</i>
Umfang der Beispiele	<i>umfangreich</i>	↔	<i>gering</i>
Hintergrundwissen	<i>empirisch</i>	↔	<i>axiomatisch</i>

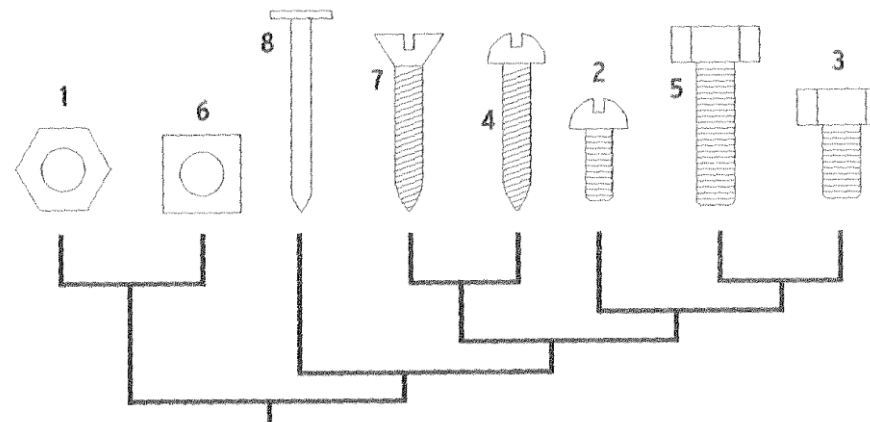
Hierarchische Ballungsanalyse - Motivation



Schraubologie: Das System der Werkzeugkiste

Bevor Sie sich an den Stammbaum der Caminalcula machen (...), können Sie Ihre taxonomischen Fähigkeiten an unserer Werkzeugkiste erproben. Gruppieren Sie obige Kleinteile nach abgestufter Ähnlichkeit, und erstellen Sie einen Stammbaum. Eine von mehreren möglichen Lösungen sehen Sie unten. Allerdings könnte man durchaus auch argumentieren, dass der

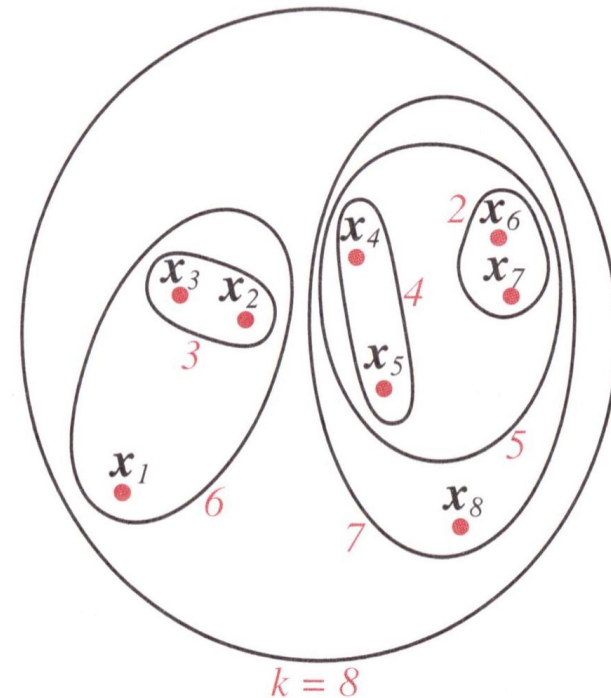
Nagel und die Schrauben 4 und 7 eine phylogenetische Gruppe bilden, wenn man Spitzheit höher bewertet als das Vorliegen eines Gewindes. Dieses würde damit zu einem konvergenten, unabhängig voneinander erworbenen Merkmal. Dafür spricht, dass Nummer 4 und 7 ein schräges, die übrigen Schrauben dagegen ein gerades Gewinde besitzen.



Quelle: Northern Arizona University/Lara Dickson

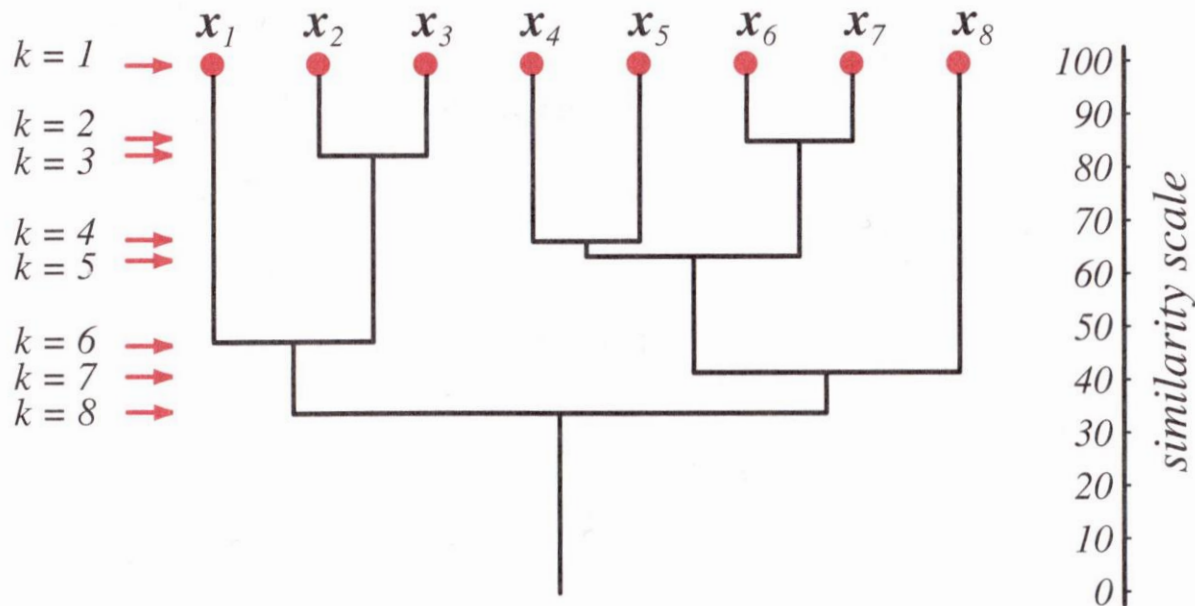
- k-means: „flache“ Datenbeschreibung
- Ballungen können Sub-ballungen und Sub-sub-ballungen ... haben

- Idee:
 - Iteratives Vereinen von (Sub-)Clustern zu größeren Clustern



Hierarchische Ballungsanalyse II

■ Andere Darstellungsweise (Dendrogramm)...



Agglomerative Hierarchical Clustering

$c := k$, (k : geg. Konstante)

$c' := n$, $D_i := \{x_i\}$, $i = 1 \dots n$

DO

$c' := c' - 1$

Find nearest Clusters D_i , D_j

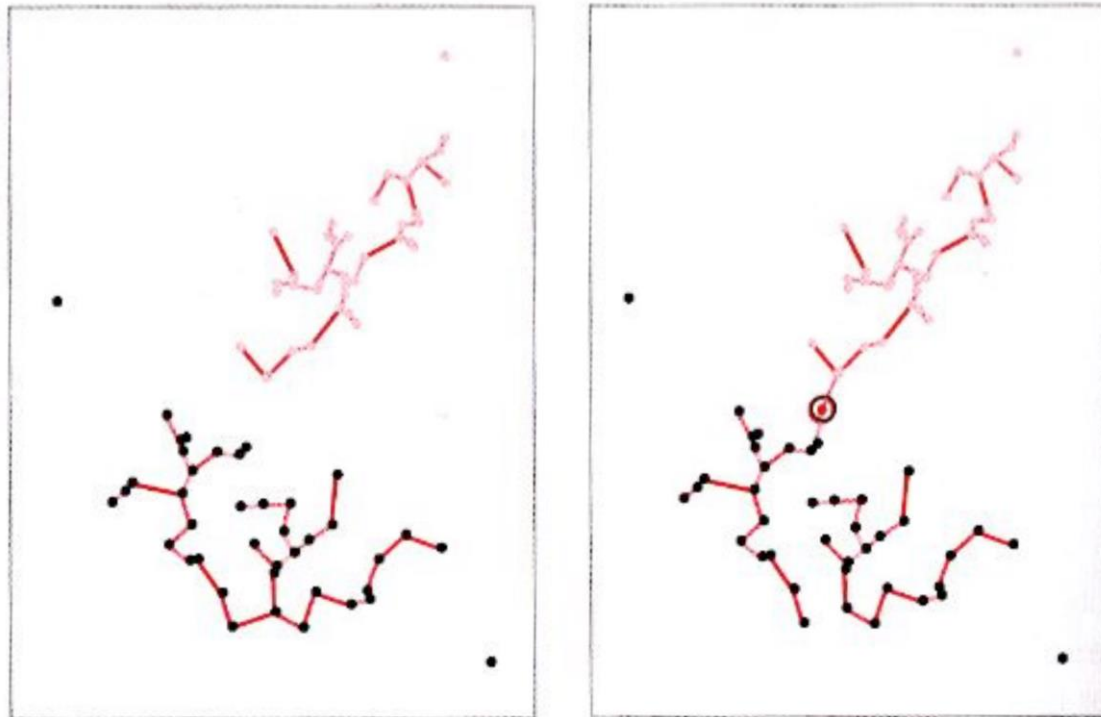
Merge D_i and D_j

UNTIL [$c = c'$ OR $d(D_i, D_j) > t$]

- „Nächster Cluster“ bzw. Abstand $d(.,.)$ definierbar
- Erzeugt Dendogramm wenn $t = \text{maxdist } x_i$
- $O(n^3)$!!!

AHC-Distanz: Nearest Neighbor

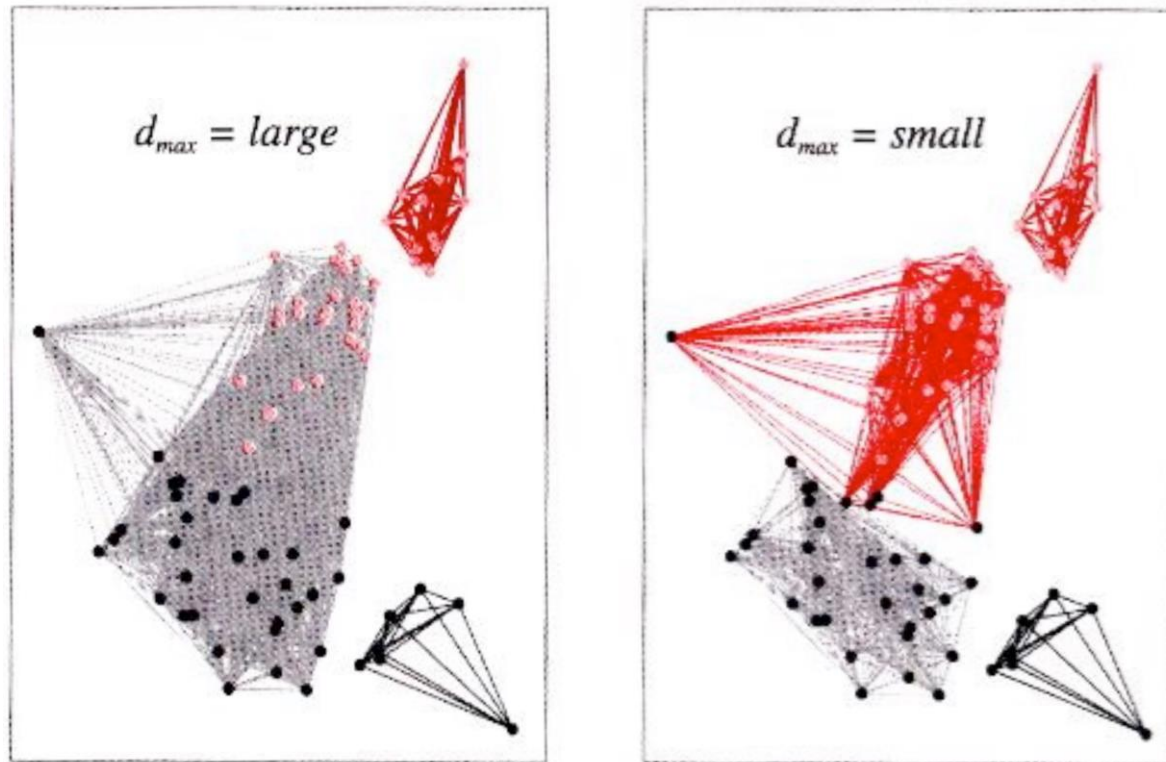
$$d(D_i, D_j) = \min_{\substack{x \in D_i \\ x' \in D_j}} |x - x'|$$



- 2 Cluster: rot und schwarz; links fast korrekt, rechts durch Rauschen → Fehler (rot und schwarz in einem Cluster)

AHC-Distanz: Farthest Neighbor

$$d(D_i, D_j) = \max_{\substack{x \in D_i \\ x' \in D_j}} |x - x'|$$



■ Fehler: links: zu hoher Grenzwert, rechts zu niedrig

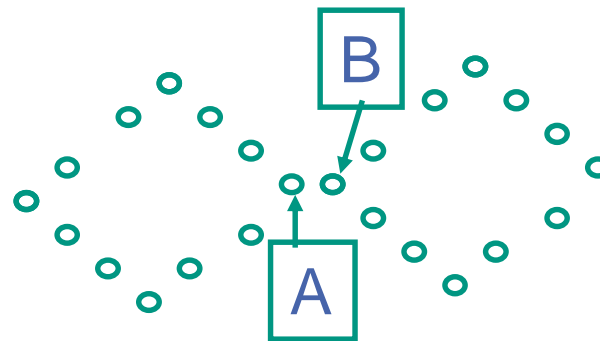
- Einfaches Verfahren
- Besser geeignet als k-means Clustering, wenn k nicht bekannt, aber gewisse Aussagen über die Form der Cluster getroffen werden können (d , t)
- Unabhängig von Initialwerten
- Finden der richtigen Parameter nötig
- Rauschanfällig

Einordnung AHC

Typ der Inferenz	<i>induktiv</i>	↔	<i>deduktiv</i>
Ebenen des Lernens	<i>symbolisch</i>	↔	<i>subsymbolisch</i>
Lernvorgang	<i>überwacht</i>	↔	<i>unüberwacht</i>
Beispielgebung	<i>inkrementell</i>	↔	<i>nicht inkrementell</i>
Umfang der Beispiele	<i>umfangreich</i>	↔	<i>gering</i>
Hintergrundwissen	<i>empirisch</i>	↔	<i>axiomatisch</i>

Klassische Ballung vs. Begriffliche Ballung

- Bei klassischen Ballungsverfahren:
 - Definition der Ähnlichkeit auf der Basis einer meist numerischen Ähnlichkeitsfunktion
 - Ähnlichkeitsmaß ist kontextfrei, d.h. Umgebung spielt keine Rolle
 - keine Ausnutzung konzeptueller Zusammenhänge
 - keine Verwendung von Gestalteigenschaften
 - Ähnlichkeit hängt nicht von der Einfachheit der resultierenden Beschreibungen ab



■ Ziel:

- Wie können Beispiele in Klassen bezüglich ihrer Ähnlichkeit geordnet werden?
- keine Klasseninformationen gegeben

■ Beispielhafte Algorithmen:

■ COBWEB:

- Lernen von Begriffen für Attribute mit symbolischen Wertebereichen (Fisher, 1987)

■ CLASSIT:

- Lernen von Begriffen für Attribute mit numerischen Wertebereichen (Gennari, 1989)

- Lernen durch inkrementelles Aufbauen und Anpassen eines Strukturbaums
- Repräsentation der Begriffshierarchie als Baum
 - Jede Verzweigung innerhalb des Baumes steht für eine Einteilung der Unterbäume in verschiedene Kategorien
 - Blätter sind die speziellsten Begriffe (Kategorien)
- Es werden nominale Attributwerte gestattet

■ Auswahl geeigneter Kategorien:

- Maß für die Ballungsnützlichkeit (category utility)
- Eine Ballung c_i besitzt eine hohe Nützlichkeit, wenn man
 - Falls x zu c_i gehört, die Attributwerte von x mit hoher Wahrscheinlichkeit vorhersagen kann ($p(v|c)$, [predictability/Vorhersagbarkeit](#))

■ UND

- Falls die Attributwerte v von x gegeben sind, die Zugehörigkeit von x zu c_i mit hoher Wahrscheinlichkeit bestimmt werden kann ($p(c|v)$, [predictiveness/Vorhersagekraft](#))

- Maximiere die Ähnlichkeit zwischen den Instanzen einer Klasse und gleichzeitig die Unterschiede zwischen den Klassen
- Man erhält ein Maß für die Vorhersagekraft und die Vorhersagbarkeit jedes Attributes in einem Konzept.
(Category Utility)

$$CU = \sum_{k=1}^K \sum_{i=1}^I \sum_{j=1}^{J(i)} P(A_i = V_{ij}) \cdot P(A_i = V_{ij} | C_k) \cdot P(C_k | A_i = V_{ij})$$

C_1, \dots, C_K = Partitionierung in Unterkonzepte

K = Anzahl der Klassen (Nachfolgeknoten)

I = Anzahl der Attribute

$J(i)$ = Anzahl der Attributwerte des i -ten Attributes

V_{ij} = j -ter möglicher Wert für Attribut i

$P(A_i = V_{ij} | C_k)$ = Vorhersagbarkeit (**predictability**) eines Attributwertes

$P(C_k | A_i = V_{ij})$ = Vorhersagekraft (**predictiveness**) eines Attributwertes

Beispiel für Kategorienbaum I

- Domäne: Geometrische Objekte
- Attribute:
 - Größe: [klein, mittel, groß]
 - Farbe: [rot, blau, grün]
 - Oberfläche: [eben, rauh]
 - Form: [rechteck, kreis]



Beispiel für Kategorienbaum II

P[C0] = 1.0		P[V C]
Größe	klein	1
	mittel	0
	groß	0
Farbe	rot	1
	blau	0
	grün	0
Oberfläche	eben	1
	rauh	0
Form	Rechteck	0.5
	Kreis	0.5

$$CU = 0,5 + 0,5 = 1$$

P[C1] = 0.5		P[V C]
Größe	klein	1
	mittel	0
	groß	0
Farbe	rot	1
	blau	0
	grün	0
Oberfläche	eben	1
	rauh	0
Form	Rechteck	1
	Kreis	0

$$CU = 1 + 1 = 2$$

P[C2] = 0.5		P[V C]
Größe	klein	1
	mittel	0
	groß	0
Farbe	rot	1
	blau	0
	grün	0
Oberfläche	eben	1
	rauh	0
Form	Rechteck	0
	Kreis	1

Der COBWEB-Algorithmus I

- Siehe [2]
- Eingabe:
 - Aktueller Knoten N in der Konzepthierarchie
 - Ein Unklassifiziertes Attribute-Werte-Paar I
- Ausgabe:
 - Konzepthierarchie, die die Instanz I klassifiziert
- Top-level call: COBWEB(Top-node, I)
- Variablen:
 - C, P, Q, R: Knoten in der Hierarchie
 - W, X, Y, Z: Category-Utility Werte

Der COBWEB-Algorithmus II

Füge I in N ein

IF !(N Terminalknoten)

FOR C Nachfolger von N

Berechne Category Utility-Funktion, wenn man I in C platziert

$(P, W) :=$ (Knoten mit dem höchsten CU, dessen CU)

$R :=$ Knoten mit zweithöchsten CU

$X :=$ CU für einen neuen Knoten Q mit I

$Y :=$ CU für Vereinigung von P und R

$Z :=$ CU für Aufteilen von P in seine Nachfolger

SWITCH MAX (W, X, Y, Z):

CASE W: COBWEB(P, I) *//Platzieren von I in Knoten P*

CASE X: *Erzeuge neuen Knoten Q aus I unter N*

CASE Y: $O :=$ *Vereinigung (P, R)*

COBWEB(O, I) // Platzieren von I in vereintem Knoten O

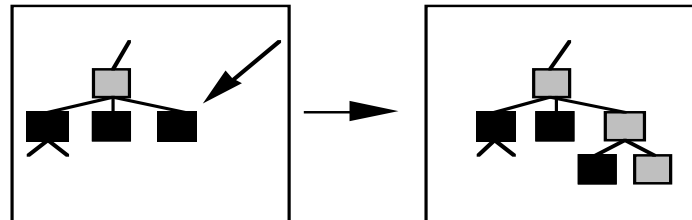
CASE Z: *Teile P in seine Nachfolger auf*

COBWEB(N, I) // erneuter Platzierungsversuch in Knoten N

■ Einfügen des Beispiels I in Knoten N

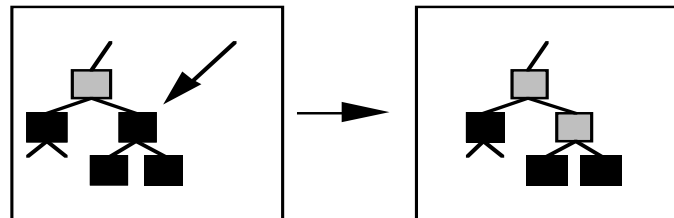
■ N ist Blatt:

- Erzeuge neuen Knoten mit N und I als Nachfolger

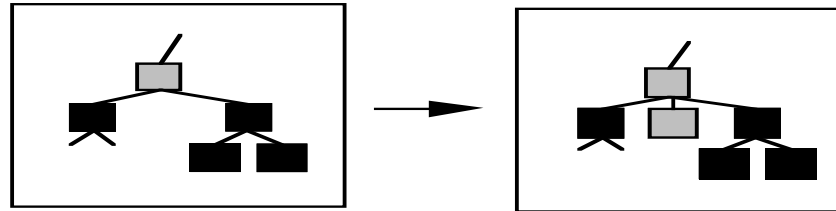


■ N hat Nachfolger:

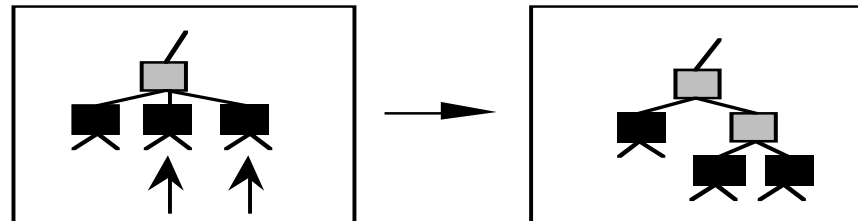
- Nur Parameteranpassung



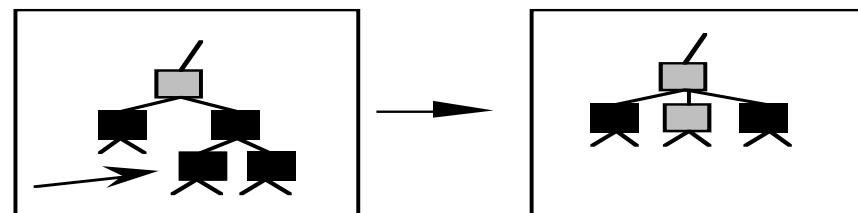
- Neuen Knoten erzeugen (I passt in keine Klasse)



- Vereinigung zweier Knoten: Verallgemeinertes Konzept



- Knoten aufteilen: Konzept war zu allgemein



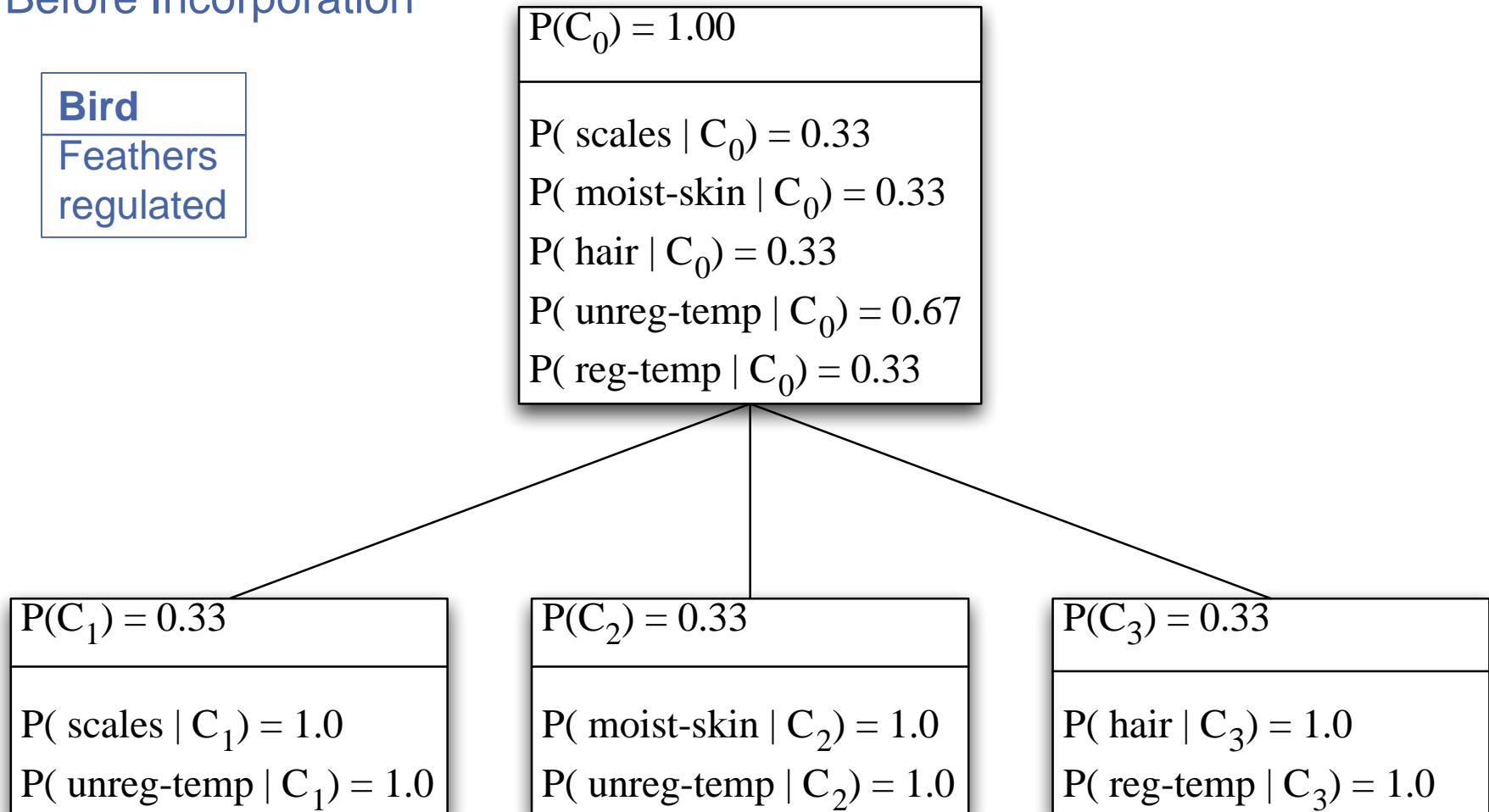
COBWEB: Beispiel I

Name	Body cover	Body temp
Fish	scales	unregulated
Amphibian	moist-skin	unregulated
Mammal	hair	regulated
Bird	feathers	regulated

COBWEB: Beispiel II

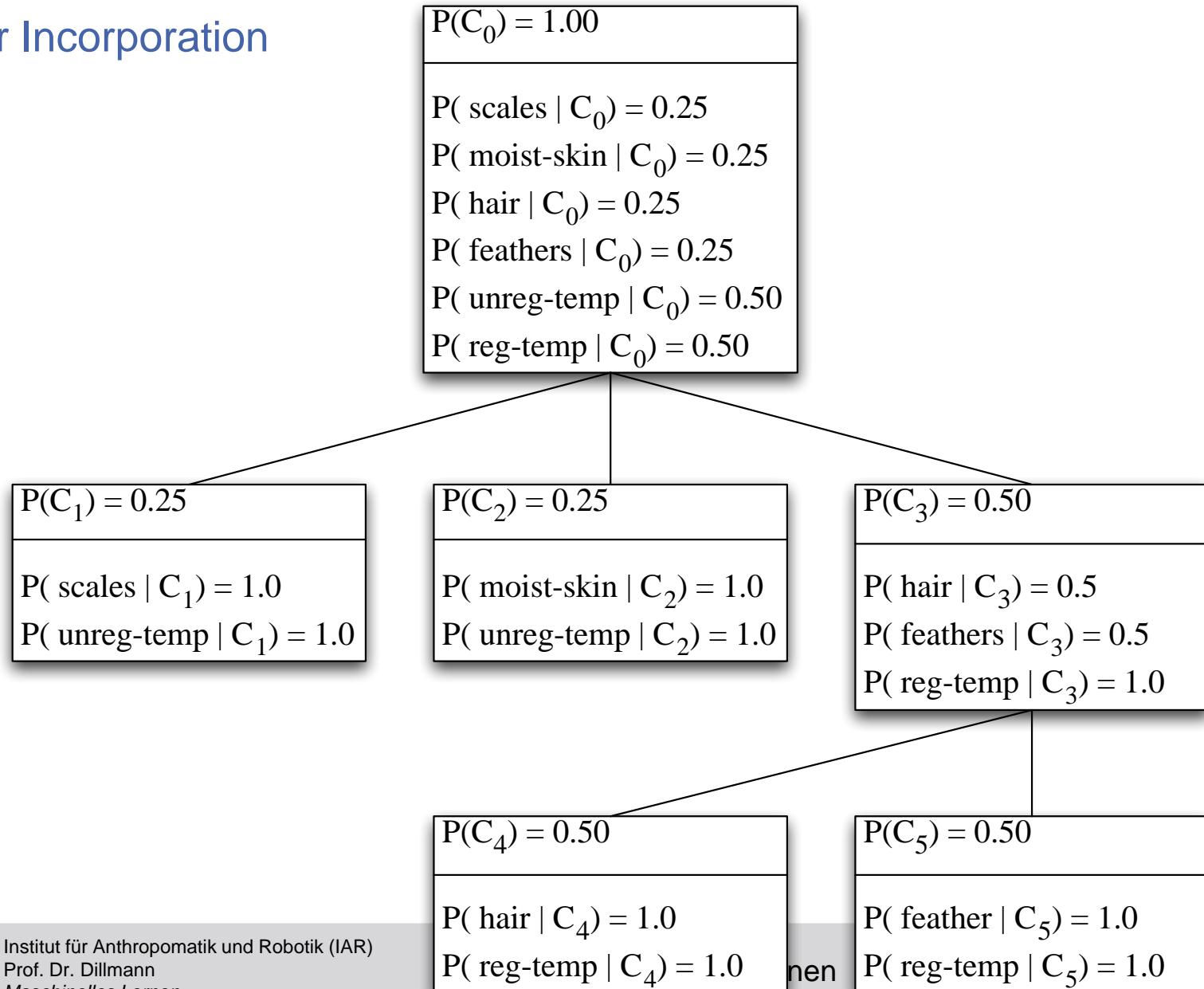
Before Incorporation

Bird
Feathers regulated



COBWEB: Beispiel III

After Incorporation



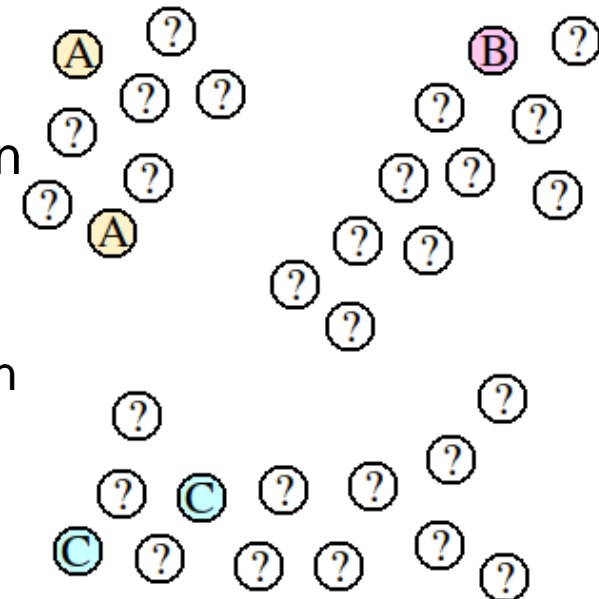
Einordnung COBWEB

Typ der Inferenz	<i>induktiv</i>	↔	<i>deduktiv</i>
Ebenen des Lernens	<i>symbolisch</i>	↔	<i>subsymbolisch</i>
Lernvorgang	<i>überwacht</i>	↔	<i>unüberwacht</i>
Beispielgebung	<i>inkrementell</i>	↔	<i>nicht inkrementell</i>
Umfang der Beispiele	<i>umfangreich</i>	↔	<i>gering</i>
Hintergrundwissen	<i>empirisch</i>	↔	<i>axiomatisch</i>

Ausblick auf fortgeschrittene Erweiterungen (ML2)

- Constrained k-mean Clustering [3]
 - zusätzliche *must-link*- und *cannot-link*-Beschränkungen
 - Clustering-Algorithmus muss Beschränkungen einhalten

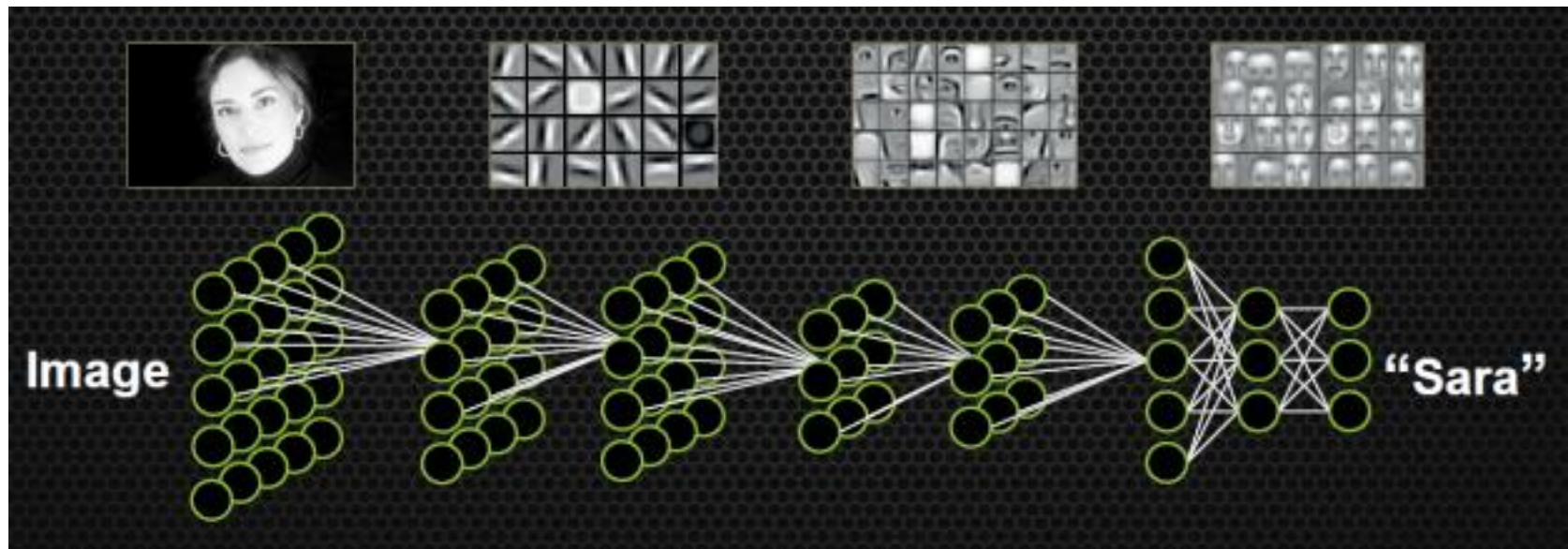
- Semi supervised learning (SSL, 2000er)
 Transduktion (Vapnik, 1990er) [4]
 - Folgern aus speziellen Beispielen und ggf. Verteilung der anderen, ungelabelten Daten
 - Beispiel:
 - Trainingspunkte, nur wenige gelabelt
 - Klassifizieren/Clustern und Label bestimmen beim Lernen
 - z.B S³VM etc... (ML2)



Ausblick auf fortgeschrittene Erweiterungen

■ Deep Learning (ML2)

- Tiefe (mehrschichtige) Netze z.B. Convolution Neural Nets (Faltungsnetze) und Deep Belief Netze (bzw. Kaskadierte eingeschränkte Boltzmann Netze)
- Lernen von Merkmalsrepräsentation und Interpretation der Daten



[Nvidia]

Ausblick auf fortgeschrittene Erweiterungen

■ Deep Learning (ML2)

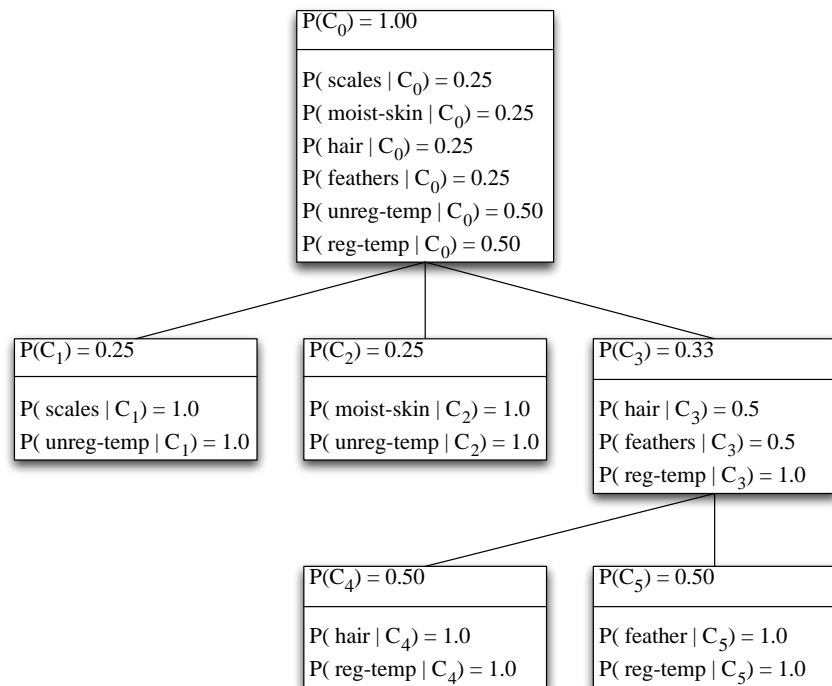
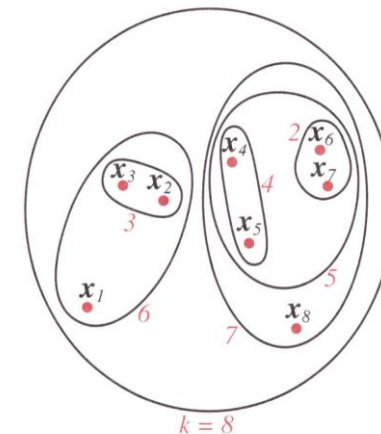
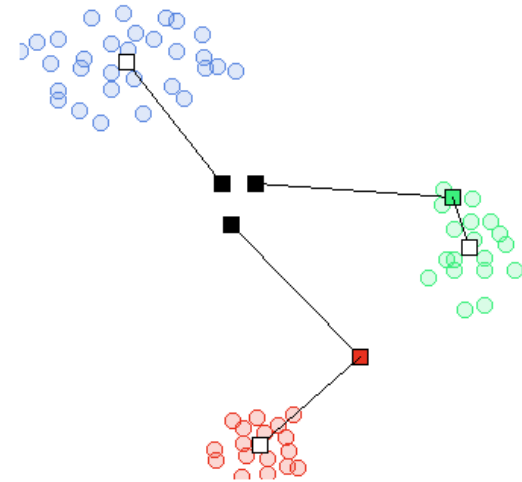
- Tiefe (mehrschichtige) Netze z.B. Convolution Neural Nets (Faltungsnetze) und Deep Belief Netze (bzw. Kaskadierte eingeschränkte Boltzmann Netze)
- Lernen von Merkmalsrepräsentation und Interpretation der Daten



[<http://www.cs.toronto.edu/~hinton/adi/index.htm>]

Zusammenfassung

- k-means Clustering
- Hierarchisches Clustering
- COBWEB



- [1] *Duda, Hart, Stork: **Pattern Classification***. John Wiley & Sons, 2001, Kapitel 10.
- [2] *Gennari, Langley, Fisher: **Models of incremental concept formation***. Artificial Intelligence, vol. 40, pp. 11-61, 1989.
- [3] *Wagstaff, Cardie, Rogers, Schroedl: **Constrained K-means Clustering with Background Knowledge***. Proceeding of the 8th Int. Conference on Machine Learning, pp. 577-584, 2001.
- [4] *Vapnik: **Statistical learning theory***. Wiley, pp. 339-371, 1998.
- [5] ML2 → Sommersemester