technische universität dortmund

# Vorlesung
# Wissensentdeckung in Datenbanken

## Graphische Wahrscheinlichkeitsmodelle

Kristian Kersting, (Katharina Morik), Claus Weihs

LS 8 Informatik
Computergestütze Statistik
Technische Universität Dortmund

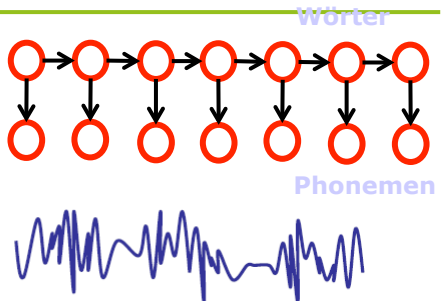
20. Mai 2014

## Ziele der heutigen Vorlesung

**Judea Pearl hat für seine Beiträge zu Graphischen Modellen den ACM Turing Award 2012 erhalten**

- Einblick in
  1. eine der **spannendsten Entwicklungen** im Data Mining in den letzen zwei (oder mehr) Jahrzenten: **Graphische Modelle**
  2. Die **Grundlagen des Schlussfolgerns** unter Unsicherheit aus Sicht von Graphischen Modellen
- Achtung! Notwendigerweise unvollständig

## Anwendung: Spracherkennung

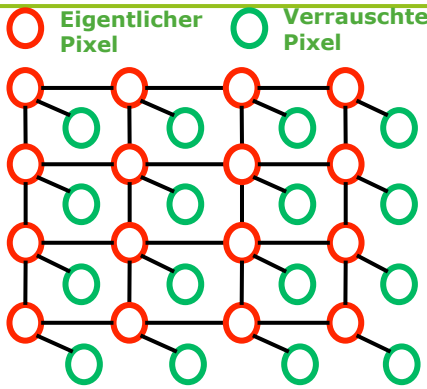




„He ate the cookies on the couch"

- Schließe auf die Wörter aus dem gesprochenen Signal
- Hidden Markov Model



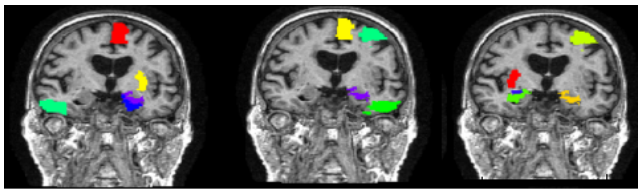
## Anwendung: Image Denoising




- Schliesse auf das Original aus dem verrauschten Abbild
- Markov Netzwerke, hier Gitterstrukturen

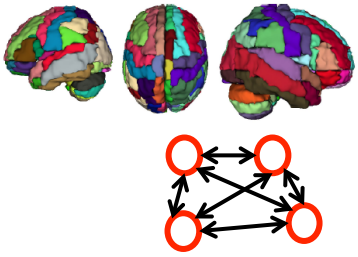
[Natarajan, Joshi, Saha, Edwards, Khot, Moody, **Kersting**, Whitlow, Maldjian ICMLA 2012]

## Anwendung: Alzheimer-Vorhersage



**Alzheimer** | **milde kognitive Einschränkungen** | **kognitiv normal**

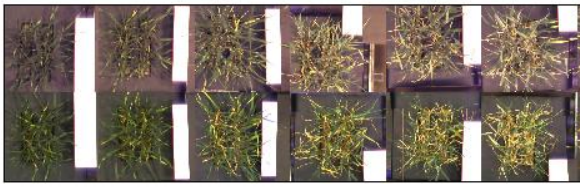
- Welche Areale sagen gut verschiedene Stadien von Alzheimer vorher?
- Relationale „Gitter“ mittels Abhängigkeitsnetzwerken über Hirn-Atlas und medizinischem Hintergrundwissen

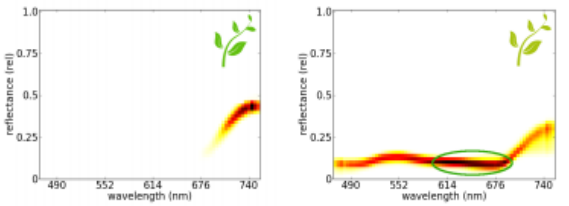


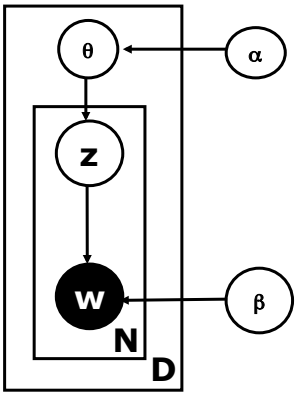


[Wahabzada, Kersting, Bauckhage, Römer, Ballvora, Pinto, Rascher, Leon, Plümer UAI 2012]

## Anwendung: Präzisionslandwirtschaft






- Wie reagieren Pflanzen auf Stress?
- Latent Dirichlet Allocation: Modellierung von Themen in Textdokumenten

## Graphische Modelle sind omnipräsent

Informationsdienste, Suche, Kollaboratives Filtern,
Expressionsanalyse, Sprachverarbeitung, Bioinformatik,
… und viele,

viele,

viele,

viele

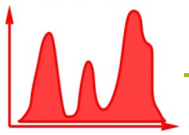
mehr !

**OK, aber was sind denn nun Wahrscheinlichkeiten und Graphische Modelle?**



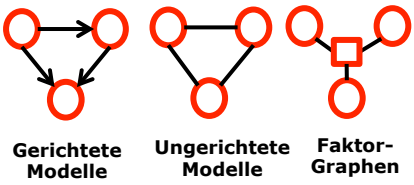
## WAS SIND DENN NUN GRAPHISCHE MODELLE?

## Kurz gesagt, Graphische Modelle sind …




… eine einfache, **graphische Notation** für (bedingte) Unabhängigkeitsannahmen und somit zur **kompakten Spezifikation** von **Wahrscheinlichkeitsverteilungen**

**Knoten =**
**Zufallsvariablen (ZVn)**
**Kanten =**
**Abhängigkeiten zwischen ZVn**



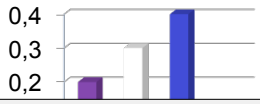



## Diskrete Zufallsvariablen

- Endliche Menge X an möglichen Zuständen

$$X \in \{x_1, x_2, x_3, \ldots, x_n\}$$

$n$




**OK, aber zum Beantworten der Frage sind wir an Verbünden von und nicht an einzelnen Zufallsvariablen interessiert!**

**Was ist die Wahrscheinlichkeit, dass Rauchen zu Krebs führt?**

**Rauchen**

| Nein | Gelegentlich | Stark |
|---|---|---|
| 0.800 | 0.150 | 0.050 |

**Krebs**

| Nein | Benigne | Maligne |
|---|---|---|
| 0.935 | 0.046 | 0.019 |

## Verbundwahrscheinlichkeit

- Wahrscheinlichkeit, dass X=x und Y=y eintreffen

$$P(x, y) \equiv P(X = x \wedge Y = y)$$

| | | Krebs | | |
|---|---|---|---|---|
| | | Nein | Benigne | Maligne |
| Rauchen | Nein | 0.768 | 0.024 | 0.008 |
| | Gelegentlich | 0.132 | 0.012 | 0.006 |
| | Stark | 0.035 | 0.010 | 0.005 |

- Von der Verbundwahrscheinlichkeit können wir alles ablesen! Aber wie?



## Dafür nutzen wir die Regeln der Wahrscheinlichkeitsrechnung

- **Marginalisierung**

$$P(Y) = \sum_{i=1}^{n} P(Y, x_i)$$

summiere

| | | Krebs | | | |
|---|---|---|---|---|---|
| | | Nein | Benigne | Maligne | TOTAL |
| Rauchen | Nein | 0.768 | 0.024 | 0.008 | 0.800 |
| | Gelegentlich | 0.132 | 0.012 | 0.006 | 0.150 |
| | Stark | 0.035 | 0.010 | 0.005 | 0.050 |
| | TOTAL | 0.935 | 0.046 | 0.019 | |

P(Rauchen)

P(Krebs)

- Produktregel & **Bedingte Wahrscheinlichkeit**

$$P(X, Y) = P(X \mid Y)P(Y) = P(Y \mid X)P(X)$$

Wahrscheinlichkeit für X=x falls wir Y=y wissen (P(y)>0)

## Vielleicht die wichtigste Regel: Bayes

$$P(R,K) = P(R \mid K)P(K) = P(K \mid R)P(R)$$

$$P(R \mid K) = \frac{P(K \mid R)P(R)}{P(K)} = \frac{P(K,R)}{P(K)}$$

| Rauchen \ Krebs | Nein | Benigne | Maligne |
|---|---|---|---|
| Nein | 0.768 | 0.024 | 0.008 |
| Gelegentlich | 0.132 | 0.012 | 0.006 |
| Stark | 0.035 | 0.010 | 0.005 |
| TOTAL | 0.935 | 0.046 | 0.019 |

**P(Krebs)**

**Wir hatten ja schon die Verbundwahrscheinlichkeit P(R,K) und auch P(K)! Also teile entsprechend.**


## Vielleicht die wichtigste Regel: Bayes

$$P(R,K) = P(R \mid K)P(K) = P(K \mid R)P(R)$$

$$P(R \mid K) = \frac{P(K \mid R)P(R)}{P(K)} = \frac{P(K,R)}{P(K)}$$


| Rauchen \ Krebs | Nein | Benigne | Maligne |
|---|---|---|---|
| Nein | 0.768/0.935 | 0.024/ 0.46 | 0.008/ 0.019 |
| Gelegentlich | 0.132/0.935 | 0.012/ 0.46 | 0.006/ 0.019 |
| Stark | 0.035/0.935 | 0.010/ 0.46 | 0.005/ 0.019 |
| TOTAL | 0.935 | 0.046 | 0.019 |

**P(Krebs)**

## Vielleicht die wichtigste Regel: Bayes

$$P(R,K) = P(R \mid K)P(K) = P(K \mid R)P(R)$$

$$P(R \mid K) = \frac{P(K \mid R)P(R)}{P(K)} = \frac{P(K,R)}{P(K)}$$

Krebs= ... )

| P(Rauchen) | Nein | Benigne | Maligne |
|---|---|---|---|
| Nein | 0.821 | 0.522 | 0.421 |
| Gelegentlich | 0.141 | 0.261 | 0.316 |
| Stark | 0.037 | 0.217 | 0.263 |

**So lange alle Einträge >0 sind, können wir alles berechnen! Mission completed?**

## Nein!! Unsere Mission hat gerade erst begonnen!

- Die Verbundwahrscheinlichkeit zählt alles nur auf
  - Worst-case Laufzeit: $O(2^n)$
    - n = # der ZVs
  - Platzkomplexität ist auch $O(2^n)$
    - Größe der Verbundwahrscheinlichkeit

**Nutze zur kompakten Darstellung Unabhängigkeiten aus**

## Unabhängigkeit

- Alter und Geschlecht sind unabhängig

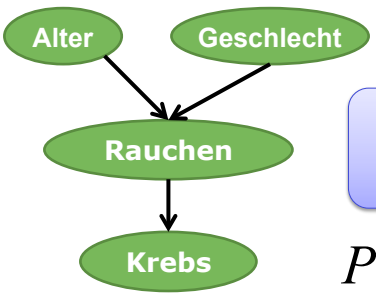

$$P(G, A) = P(G) \cdot P(A)$$
$$P(A \mid G) = P(A)$$
$$P(G \mid A) = P(G)$$

**Ihr würdet mir kein Geld für die Information über das Alter einer Person geben, wenn Ihr ihr Geschlecht wissen wollt!**



## Bedingte Unabhängigkeit

- Krebs ist unabhängig vom Geschlecht und dem Alter, wenn man raucht
- Wenn man nichts weiß, sind Alter und Geschlecht immer noch unabhängig.



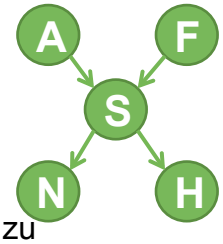

**Weniger darzustellen, also geringere Komplexität**

$$P(K \mid R, G, A) = P(K \mid R)$$

[Pearl 1989]

## Allgemein: Bayes‘sches Netzwerk (BN)

- Menge von Zufallsvariables $\{X_1,\ldots,X_n\}$
- **Gerichtete, azyklische Graphen**
- Zu jedem Xi assoziieren wir **die bedingte Wahrscheinlichkeit:** $P(X_i \mid \mathrm{Pa}(X_i))$




- Die **Verbundwahrscheinlichkeit** ergibt sich zu

$$P(X_1,\mathrm{K}\ ,X_n) = \prod_{i=1}^{n} P(X_i \mid \mathrm{Pa}(X_i))$$

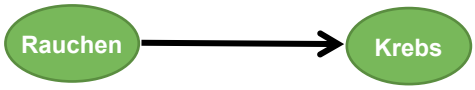
- **Lokale Markov Annahme**

**Eine ZV X ist unabhängig von ihren „Nicht-Nachfahren“ gegeben ihre Eltern und nur ihre Eltern: ($Xi \perp NichtNachfahren_{Xi} \mid Pa_{Xi}$)**



## Beispiel

$R \in \{nicht, gelegentlich, stark\}$




| | |
|---|---|
| P(R=n) | 0.80 |
| P(R=g) | 0.15 |
| P(R=s) | 0.05 |

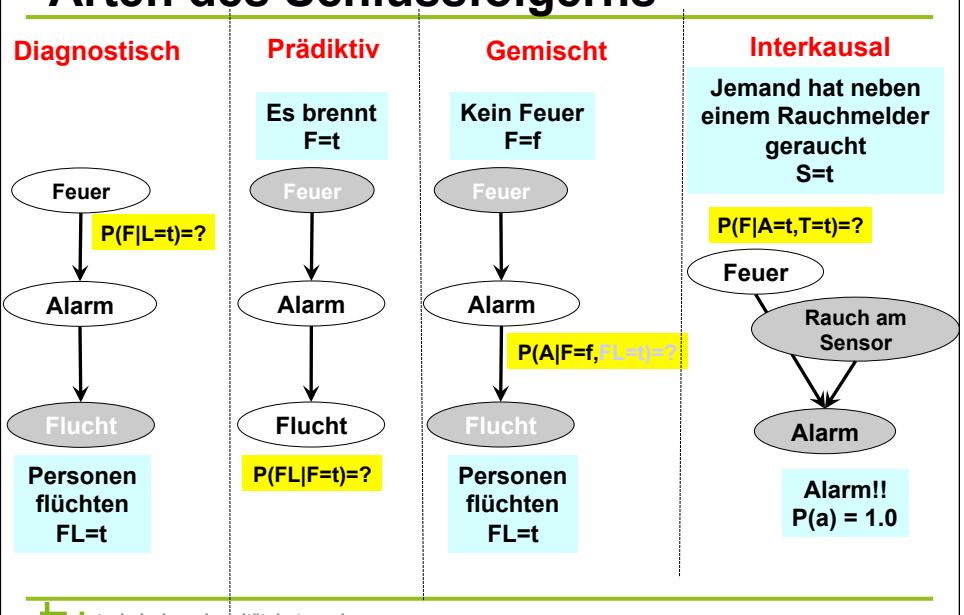
$K \in \{kein, benigne, maligne\}$

| Rauchen= | n | g | s |
|---|---|---|---|
| P(K=k) | 0.96 | 0.88 | 0.60 |
| P(K=b) | 0.03 | 0.08 | 0.25 |
| P(K=m) | 0.01 | 0.04 | 0.15 |

# UND WIE NUTZEN WIR DIESE JETZT ZUM SCHLUSSFOLGERN?



# Arten des Schlussfolgerns



## Was verstehen wir unter Schlussfolgern?

- **Anfrage:** $P(X \mid e)$
- **Definition der bedingten Wahrscheinlichkeit**

$$P(X \mid e) = \frac{P(X, e)}{P(e)}$$

- **Also bis auf Normalisierung**

$$P(X \mid e) \propto P(X, e)$$

das heisst

$$P(\mathbf{Y}) = \sum_{X_i \notin \mathbf{Y}} \prod_{i=1}^{n} P(X_i \mid \mathrm{Pa}(X_i))$$

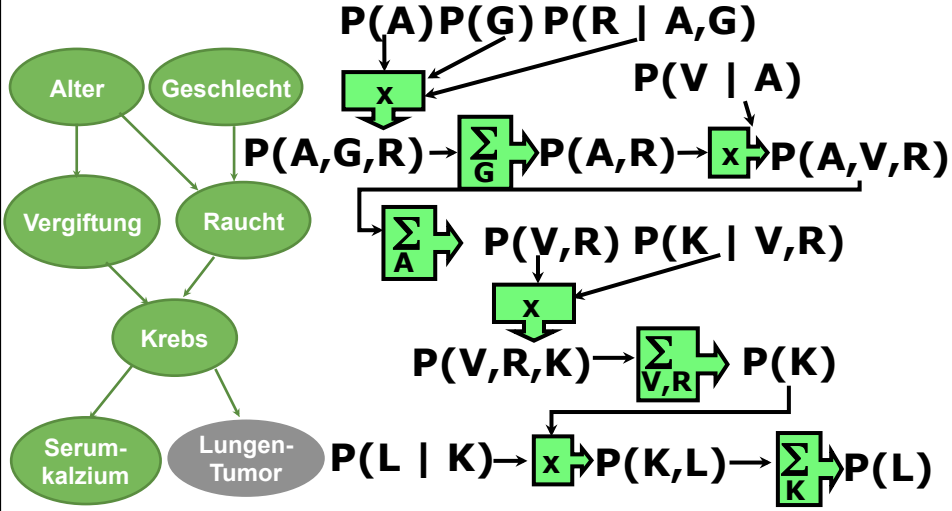
BN Semantik

Marginalisierung

**Zentrale Einsicht: Σ und ∏ vertauschbar**

- $\Sigma_a(P_1 \times P_2) = (\Sigma_a P_1) \times P_2$ falls A nicht in $P_2$

## Ein komplexeres Beispiel (G,A,V,R,K)




**Exponentiell in der Größe der induzierten Faktoren (Baumweite): $2^3$ vs $2^7$**

## I.A.: Variablen-Elimination

[Zhang, Poole 1994]

- Gegeben ein BN und eine Anfrage P(X|e) / P(X,e)
- Initialisiere die *Faktors* $\{f_1,\ldots,f_n\}$: $f_i = P(X_i|\mathbf{Pa}_{Xi})$
- Instanziiere die Evidenz **e** **(Operation 1)**
- **Wähle Ordnung über die ZVn, z.B., $X_1, \ldots, X_n$**

- **For i = 1 to n, if $X_i \notin \{X,E\}$ eliminiere $X_i$**
  - Sammle alle $f_1,\ldots,f_k$ ein, die $X_i$ beinhalten
  - Berechne eine neuen Faktor g durch das Eliminieren von $X_i$ aus diesen Faktoren **(Operationen 2 und 3)**
  - $X_i$ ist jetzt eliminert! Füge g zu den restlichen Faktoren hinzu
- Normalisiere P(X,**e**) um P(X|**e**) zu erhalten **(Operation 4)**

$$g = \sum_{X_i} \prod_{j=1}^{k} f_j$$



## "Σ und ∏ vertauschbar" für Potentiale

$\Sigma_a(P_1 \times P_2) = (\Sigma_a P_1) \times P_2$ **falls A nicht in $P_2$**

| | b | ¬b |
|---|---|---|
| a | .1 | .5 |
| ¬a | .2 | .8 |

X

| | c | ¬c |
|---|---|---|
| b | .2 | .4 |
| ¬b | .3 | .5 |

=

| | a | | ¬a | |
|---|---|---|---|---|
| | c | ¬c | c | ¬c |
| b | .02 | .04 | .04 | .08 |
| ¬b | .15 | .25 | .24 | .40 |

**VE implementiert man am Besten mittels Potentialen, die Tupeln reelle Zahlen zuweisen**

$\Sigma_a$

| | c | ¬c |
|---|---|---|
| b | .06 | .12 |
| ¬b | .39 | .65 |

| | b | ¬b |
|---|---|---|
| a | .1 | .5 |
| ¬a | .2 | .8 |

$\rightarrow \Sigma_a$

| | b |
|---|---|
| b | .3 |
| ¬b | 1.3 |

X

| | c | ¬c |
|---|---|---|
| b | .2 | .4 |
| ¬b | .3 | .5 |

=

| | c | ¬c |
|---|---|---|
| b | .06 | .12 |
| ¬b | .39 | .65 |

## Operation 1: Instanziierung

- Wir weisen einer (oder mehrer) Variablen einen Wert zu

**Was ergibt sich, wenn wir X= t setzen?**

f(X,Y,Z):

| X | Y | Z | val |
|---|---|---|---|
| t | t | t | 0.1 |
| t | t | f | 0.9 |
| t | f | t | 0.2 |
| t | f | f | 0.8 |
| ~~f~~ | ~~t~~ | ~~t~~ | ~~0.4~~ |
| ~~f~~ | ~~t~~ | ~~f~~ | ~~0.6~~ |
| ~~f~~ | ~~f~~ | ~~t~~ | ~~0.3~~ |
| ~~f~~ | ~~f~~ | ~~f~~ | ~~0.7~~ |

$f(X, Y, Z)_{X=t}$

| Y | Z | val |
|---|---|---|
| t | t | 0.1 |
| t | f | 0.9 |
| f | t | 0.2 |
| f | f | 0.8 |

Faktor über Y,Z



## Operation 2: Marginalisierung

- Das “Herausintergrieren” einer Variable X aus einem Faktor $f(X_1,\ldots,X_n)$ resultiert in einen neuen Faktor über $\{X_1,\ldots,X_n\} \setminus \{X\}$

$$\left(\sum_{X_1} f\right)(X_2,\ldots,X_j) = \sum_{x \in dom(X_1)} f(X_1 = x, X_2,\ldots,X_j)$$

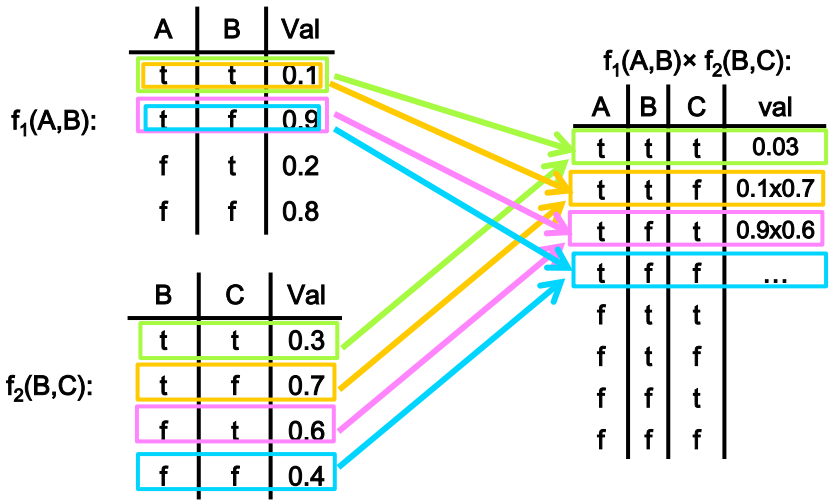
$f_3$=

| B | A | C | val |
|---|---|---|---|
| t | t | t | 0.03 |
| t | t | f | 0.07 |
| f | t | t | 0.54 |
| f | t | f | 0.36 |
| t | f | t | 0.06 |
| t | f | f | 0.14 |
| f | f | t | 0.48 |
| f | f | f | 0.32 |

$(\sum_B f_3)(A,C)$

| A | C | val |
|---|---|---|
| t | t | 0.57 |
| t | f | 0.43 |
| f | t | 0.54 |
| f | f | 0.46 |

## Operation 3: Multiplikation






## Operation 4: Normalisierung

- Überführe ein Potential in eine Wahrscheinlichkeitsverteilung (**alle Einträge summieren sich zu 1**)

| | *b* | ¬*b* |
|---|---|---|
| a | .1 | .5 |
| ¬*a* | .2 | .8 |

| | *b* | ¬*b* |
|---|---|---|
| a | .0625 | .3125 |
| ¬*a* | .125 | .5 |

## Mission Completed! Aber Achtung …

- … im Allgemeinen bleibt exaktes Schlussfolgern unter Unsicherheit schwierig!!!

> **Theorem**: Schlussfolgern (auch approximativ) in Bayes'schen Netzwerken ist NP-hart (#P; durch Reduktion auf 3-SAT)

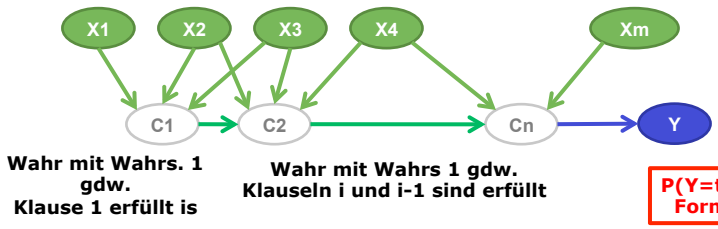
- In der Realität
  - Nutze Strukturen aus
  - Viele effektive Approximationsalgorithm (einige so gar mit Guarantien)

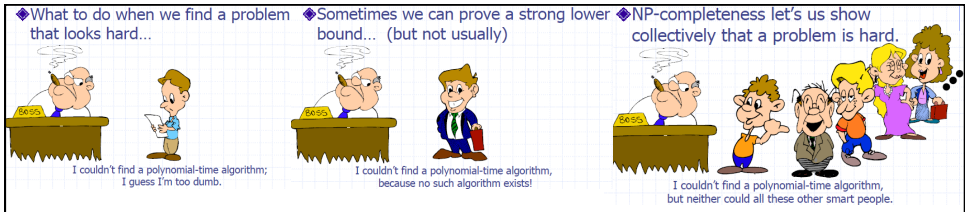


## Komeplexität

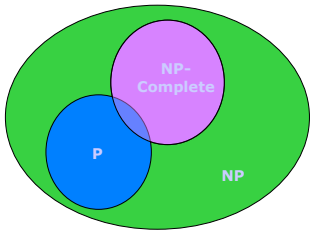
- Wie schwierig ist es, P(X|**E=e**) zu brechnen?
- Reduktion auf 3-SAT mit leerer Evidenz **E**
- Kann die folgende 3-SAT Formel erfüllt werden?

$$\underbrace{(\overline{X}_1 \vee X_2 \vee X_3)}_{C_1} \wedge \underbrace{(\overline{X}_2 \vee X_3 \vee X_4)}_{C_2} \wedge \ldots \underbrace{\quad}_{C_n}$$

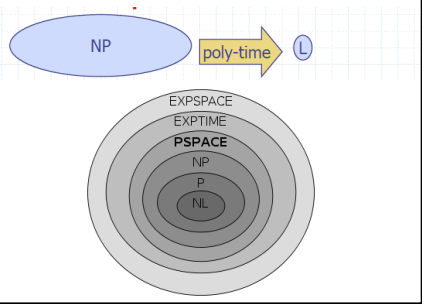





- P = { L | L wird von einer deterministischen Turing Machine in Polynomialzeit akzeptiert }
- NP = { L | L wird von einer nicht-deterministischen Turing Machine in Polynomialzeit akzeptiert }

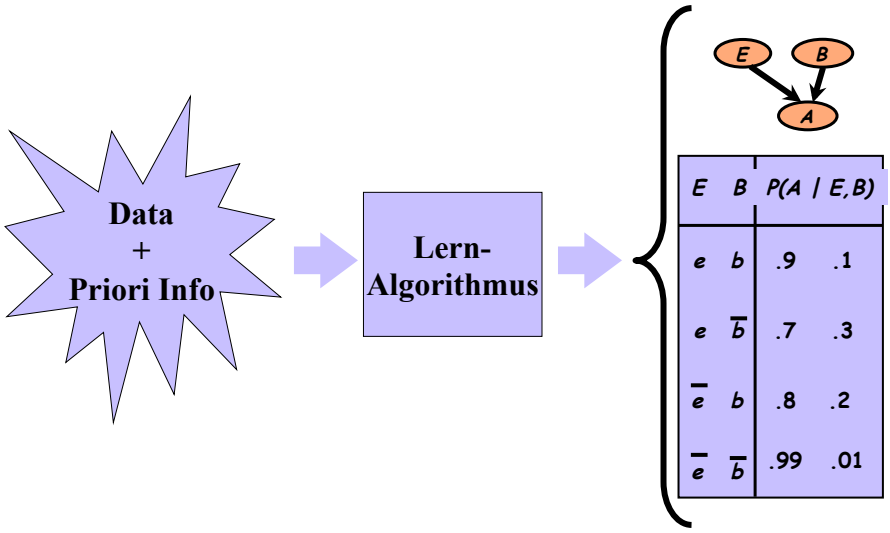



- Ein Problem L ist NP-schwer, wenn jedes Problem in NP auf L in Polynomialzeit reduziert werden kann
- L is NP-vollständig, wenn es in NP und NP-schwer ist




# ABER WO KOMMEN DEN DIE ZAHLEN UND DIE GRAPHISCHE STRUKTUR HER?

## Lernen von Graphischen Modellen






## Wie sehen die Daten aus?

**Unvollständige Datensatz**

**latent**

| A1 | A2 | A3 | A4 | A5 | A6 |
|---|---|---|---|---|---|
| true | true | ? | true | false | false |
| ? | true | ? | ? | false | false |
| ... | ... | ... | ... | ... | ... |
| true | false | ? | false | true | ? |

- **Oftmals fehlen die Informationen einiger Zufallsvariablen**
- **Nicht jeder medizinischer Test wird bei jedem Patienten aus**

**fehlender Wert**

## Lernen von graphischen Modellen

| | | Graphstuktur ist gegeben (A → B) | Bekannte Variablen (A ? B) | Latente Variablen (A ? B ? H) |
|---|---|---|---|---|
| beobachtet | vollständig | Das einfachste Problem<br>Zählen | Die Kanten müssen gewählt werden<br>Data Mining | |
| | teilweise | Numerische, nicht-lineare Optimierung<br>Inferenz<br>Schwierig für große Netzwerke | Beinhalted schon ein schwieriges Problem<br>Structural EM | Wissenschaftliche Entdeckungen |



## ML Parameterschätzung (vollständige Daten)

$$\mathcal{LL}(\Theta|\mathcal{X}) = \log P(X_1, X_2, \ldots, X_n|\Theta)$$

(iid) $$= \log \prod_{i=1}^{n} P(X_i|\Theta)$$

log ∏ = ∑ log

$$= \sum_{i=1}^{n} \log P(X_i|\Theta) = \sum_{i=1}^{n} \log P(x_i^1, x_i^2, \ldots, x_i^m|\Theta)$$

| A1 | A2 | A3 | A4 | A5 | A6 |
|---|---|---|---|---|---|
| true | true | false | true | false | false |
| false | true | true | true | false | false |
| ... | ... | ... | ... | ... | ... |
| true | false | false | false | true | true |

$$= \sum_{i=1}^{n} \log \left( \prod_{j=1}^{m} P(x_i^j | \text{pa}(x_i^j), \Theta) \right)$$ (Semantik)

log ∏ = ∑ log

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} \log P(x_i^j | \text{pa}(x_i^j), \Theta)$$

$$= \sum_{j=1}^{m} \sum_{i=1}^{n} \log P(x_i^j | \text{pa}(x_i^j), \Theta_j)$$

$$= \sum_{j=1}^{m} \mathcal{LL}(\Theta_j|\mathcal{X})$$

**Nur lokale Parameter der Familie von Aj benötigt**

**Jeder Faktor kann separat betrachtet werden!!**

## Likelihood für multinomiale Zufallsvariablen

- Zufallsvariable V mit 1,...,K als Zustänfe

$$P(V = k) = \theta_k \qquad \sum_{k=1}^{K} \theta_k = 1$$

Die Wahl für $\theta_i$ beinflußt die Wahl der anderen $\theta_j$ (i<>j)

$$\mathcal{LL}(\Theta_v|\mathcal{X}) = \sum_{k=1}^{K} \log \theta_k^{N_k} = \sum_{k=1}^{K} N_k \cdot \log \theta_k$$

wobei Nk die Häufigkeit bezeichnet, mit der wir Zustand k in den Daten beobachten



## Maximierung des Likelihoods für binomial-verteilte Zufallsvariable (2 Zustände)

- Berechne die partielle Ableitung

$$\frac{\partial}{\partial \theta_i}\mathcal{LL}(\Theta_v|\mathcal{X}) = \frac{\partial}{\partial \theta_i}(N_1 \log \theta_1 + N_2 \log(1-\theta_1))$$
$$= \frac{N_1}{\theta_1} + \frac{N_2}{1-\theta_1}$$

$\theta_1 + \theta_2 = 1$

- Wir setzten die partielle Ableitung auf 0

$$\frac{\partial}{\partial \theta_i}\mathcal{LL}(\Theta_v|\mathcal{X}) = 0 \Leftrightarrow \frac{N_1}{\theta_1} + \frac{N_2}{1-\theta_1} = 0$$

=> MLE sind $\theta_1^* = \frac{N_1}{N_1 + N_2}$

## Maximierung des Likelihoods für multinomial - verteilte Zufallsvariable (>2 Zustände)

- Mittels Lagrange Multiplikatoren kann gezeigt werden, dass im Allgemeinen:

**Der MLE ist**

$$\theta_i^* = \frac{N_i}{\sum_j N_j}$$



## Maximierung des Likelihoods für bedingte multinomial -verteilte Zufallsvariable

$$P(V = k | \text{pa}(V) = \text{pa})$$

- Für jede Konfiguration (Verbundzustand) der Eltern eines Knotens ist das gerade eine multinomiale Verteilung

$$P(k|1,1), P(k|1,2), P(k|2,1), P(k|2,2)$$

$$\mathcal{LL}(\Theta_v|\mathcal{X})$$

$$= \sum_{\text{pa}} \sum_{k=1}^{K} \log \theta_{k|\text{pa}}^{N_{k,\text{pa}}} = \sum_{\text{pa}} \sum_{k=1}^{K} N_{k,\text{pa}} \cdot \theta_{k|\text{pa}}$$

**Der MLE ist**

$$\theta_{k|\text{pa}}^* = \frac{N_{k,\text{pa}}}{N_{\text{pa}}}$$

## ML Parameterschätzung für vollständige Daten

- Für jede Zufallsvariable (Spalte der Datenmatrix) gehen wir durch die Datenpunkte (Zeilen der Datenmatrix) und zählen wie häufig ein Zustand für jede Elternkonfiguration vorkommt
- Dann normalisieren wir entsprechend

- Da Datenbankanfragen teuer sein können, ist es besser, einmal über die Daten zu gehen und entsprechende Zähler hochzusetzen

Wie funktioniert das, wenn die Daten unvollständig sind?



## Idee des Expectation-Maximization Ansatzes

- Wenn die Daten vollständig sind, ist das Schätzen der ML Parameter einfach:
  - **Einfaches Zählen** (1 Iteration über die Daten)

- Aber unsere Daten sind nicht vollständig!!!!!!!!

1. **Vervollständige die Daten** (Imputation)
   - Wahrscheinlichste? Mittelwert? ...
2. **Zähle**
3. **Wiederholungen**

## Idee des Expectation-Maximization Ansatzes


$\theta_{A=\text{true}} = \frac{1}{2}$
$\theta_{B=\text{true}|A=\text{true}} = \frac{1}{2}$
$\theta_{B=\text{true}|A=\text{false}} = \frac{1}{2}$

**Unvollständige Daten**

vervollständigen

| A | B |
|---|---|
| true | true |
| true | ? |
| false | true |
| true | false |
| false | ? |

$P(B = \text{true}|A = \text{true}) = 0.5$
$P(B = \text{true}|A = \text{false}) = 0.5$

**Vollständige Daten**

| A | B | N |
|---|---|---|
| true | true | 1.5 |
| true | false | 1.5 |
| false | true | 1.5 |
| false | false | 0.5 |

Erwartete Häufigkeiten

maximiere

wiederhole

$\theta_{A=\text{true}} = \frac{1.5+1.5}{1.5+1.5+1.5+0.5} = 0.6$
$\theta_{B=\text{true}|A=\text{true}} = \frac{1.5}{1.5+1.5} = 0.5$
$\theta_{B=\text{true}|A=\text{false}} = \frac{1.5}{1.5+0.5} = 0.75$



## Idee des Expectation-Maximization Ansatzes


$\theta_{A=\text{true}} = 0.6$
$\theta_{B=\text{true}|A=\text{true}} = 0.5$
$\theta_{B=\text{true}|A=\text{false}} = 0.875$

**Unvollständige Daten**

vervollständigen

| A | B |
|---|---|
| true | true |
| true | ? |
| false | true |
| true | false |
| false | ? |

$P(B = \text{true}|A = \text{true}) = 0.5$
$P(B = \text{true}|A = \text{false}) = 0.875$

**Vollständige Daten**

| A | B | N |
|---|---|---|
| true | true | 1.5 |
| true | false | 1.5 |
| false | true | 1.875 |
| false | false | 0.125 |

Erwartete Häufigkeiten

maximiere

wiederhole

$\theta_{A=\text{true}} = \frac{1.5+1.5}{1.5+1.5+1.875+0.125} = 0.6$
$\theta_{B=\text{true}|A=\text{true}} = \frac{1.5}{1.5+1.5} = 0.5$
$\theta_{B=\text{true}|A=\text{false}} = \frac{1.875}{1.875+0.125} = 0.9375$

## EM für multinomiale Zufallsvariablen

- Man kann zeigen, dass alles wie bisher funktioniert, so lange man Häufigkeiten durch „erwartete Häufigkeiten“ ersetzt:

$$EN_k = \sum_{i=1}^{m} P(k|X_i)$$

**MLE ist** $\dfrac{EN_i}{\sum_k EN_k}$ $\theta^*_{k|\mathrm{pa}} = \dfrac{EN_{k,\mathrm{pa}}}{EN_{\mathrm{pa}}}$



## Schätzen der ML Parameter bei unvollständigen Daten (für multinomiale Zufallsvariablen)

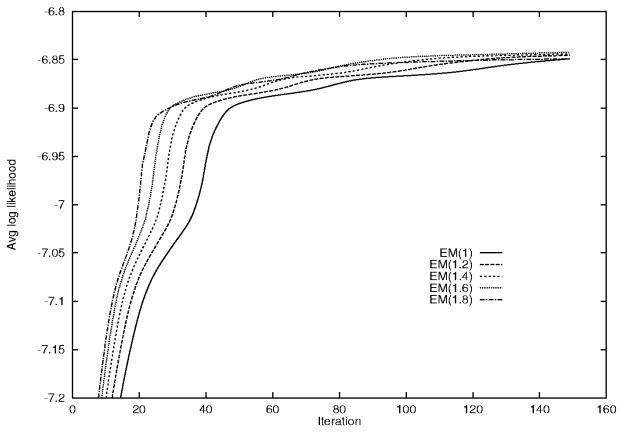
1. Initialisiere die Parameter
2. Bestimme die **erwarteten Häufigkeiten** für jedes Zufallsvariablen

$$\theta^*_{k|\mathrm{pa}} = \frac{\sum_{i=1}^{m} P(k, \mathrm{pa}|X_i)}{\sum_{i=1}^{m} P(\mathrm{pa}|X_i)}$$

**Schlussfolgern Unter Unsicherheit**

3. **Setze die Parameter entsprechend neu**
4. Falls nicht konvergiert, gehe zu 2 zurück

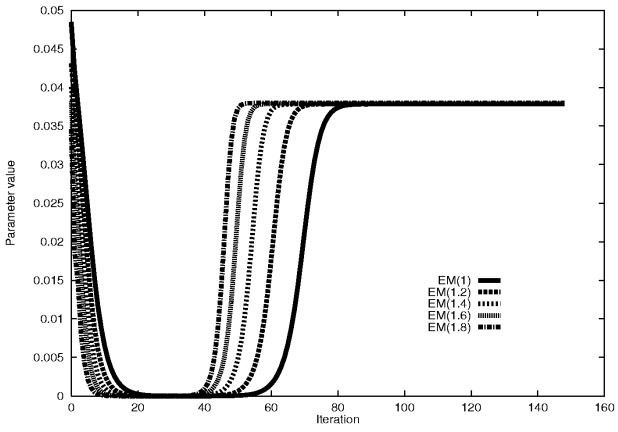
## Log-Likelihod auf der Trainingmenge (Alarm)




Experiment von Bauer, Koller and Singer [UAI97]



## Werte der Parameter (Alarm)




Experiment von Bauer, Koller and Singer [UAI97]

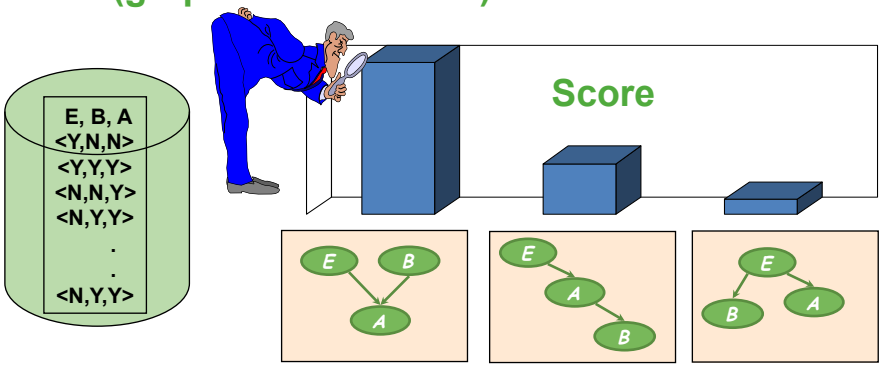
## Probleme

- EM kann oft “steckenbleiben”
  - **Lokale Maxima, Plateaux , …**
- Übliche “Workarounds”
  - Zufällige Neustarts und Perturbationen
  - TABU Suche
  - Simulated annealing (Abkühlenden Lernen)



## Modelselektion mittels Bewertungsfunktion

**Die Bewertungsfunktion sagt, wie gut ein Model (graphische Struktur) die Daten erklärt**




**Finde eine Struktur, die die Score maximiert**

## Likelihood als Bewertungsfunktion?

$$\mathrm{l}(G:D) = \log L(G:D) = M \sum_i \left( I(X_i; Pa_i^G) - H(X_i) \right)$$

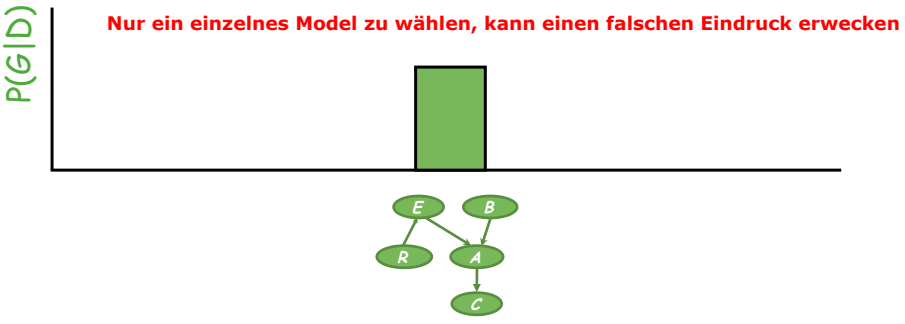
**Gegenseitige Information von $X_i$ und seinen Eltern**

**Entropie $X_i$**

- Je stärker $X_i$ von $Pa_i$ (statistisch) abhängt desto höher die Score
- Hinzufügen von Kanten kann die Score nur verbessern
  - $I(X; Y) \leq I(X; \{Y,Z\})$
  - Maximale Score wird durch das "vollständige" Netzerk erreicht. Aber das führt zu Overfitting. Eine schlechte Idee



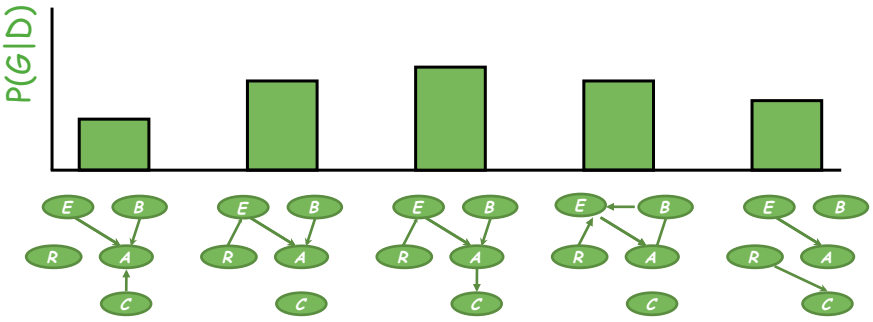
## Eine Bayes'sche Bewertungsfunktion




### Probleme

- Bei nur wenigen Daten gibt es viele gute Netzwerke
- Antworten basierend auf einem einzigen Model of nicht korrekt

## Eine Bayse‘sche Bewertungsfunktion




**Besser**

- Ziehe mehrere Modelle in Betracht!



## Eine Bayes‘sche Bewertungsfunktion

**Likelihood**:

$$L(G:D) = P(D \mid G, \hat{\theta}_G)$$

Max likelihood params

**Bayes’scher Ansatz**:

- Wir weisen allen Möglichkeiten einen Wahrscheinlichkeit zu

$$P(D \mid G) = \int P(D \mid G, \theta) P(\theta \mid G) \, d\theta$$

Marginalisierte Likelihood

Likelihood

Prior über den Parametern

$$P(G \mid D) = \frac{P(D \mid G) P(G)}{P(D)}$$

## Eine Bayes'sche Bewertungsfunktion

- Im Allgemeinen schwierig zu lösende Integrale
- Wenn man sich auf die Parameter beschränkt und einen Dirichlet-Prior animmt, können wir eine einfache Approximation nehmen, "Bayes information criterion" (BIC)
- **Theorem:** Bei einem Dirichlet-Prior und einem graphischen Model mit Dim(*G*) unabhängigen Parametern, *für m→∞:*

$$\log P(D \mid G) = \log(D \mid G, \theta) - \frac{\log M}{2} \dim(G) + O(1)$$

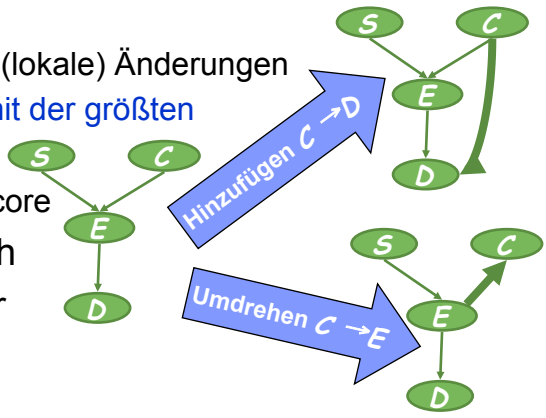
Likelihood der Parameter — Komplexitätszuschlag

- Mit wachsender Größe der Daten (M) muss die empirische Verteilung immer besser geschätzt werden.
- Der Komplexitätszuschlag vermeidet die Anpassung ans Rauschen



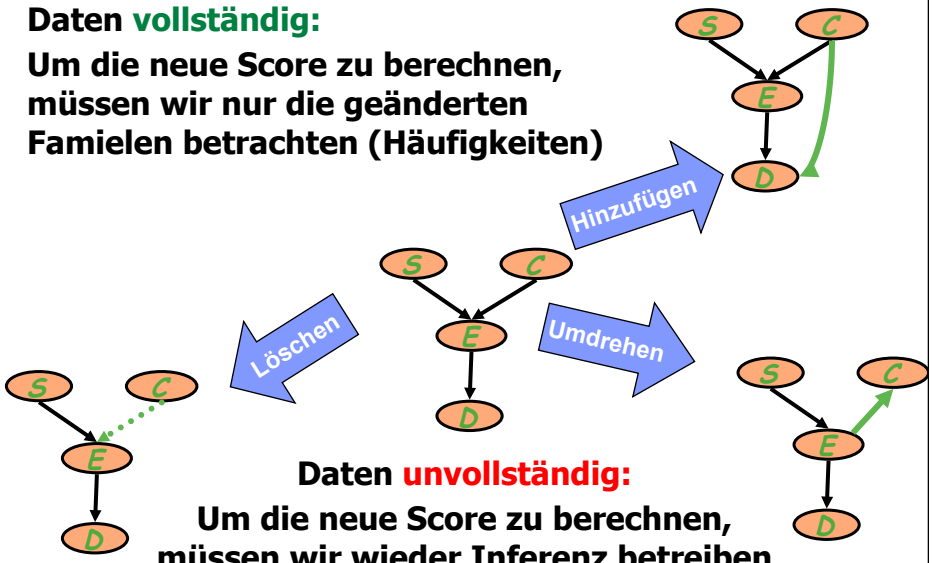
## Heuristische Suche (hier Greedy Search)

1. Fange mit einem initialen Model an
   - Leeres Netzwerk, bester Baum, zufälliges Netzwerk
2. In jeder Iteration:
   - Bewerte alle mögliche (lokale) Änderungen
   - Wähle die Änderung mit der größten Verbesserung der Bewertungsfunktion/Score
3. Höre auf, wenn sich die Score nicht mehr verbessert



## Heuristische Suche (hier Greedy Search)

**Daten vollständig:**
**Um die neue Score zu berechnen, müssen wir nur die geänderten Famielen betrachten (Häufigkeiten)**




**Daten unvollständig:**
**Um die neue Score zu berechnen, müssen wir wieder Inferenz betreiben**

## Structural EM
[Friedman et al. 98]

**Lokale Suche bei unvollständigen Daten ist sehr zeitaufwendig**

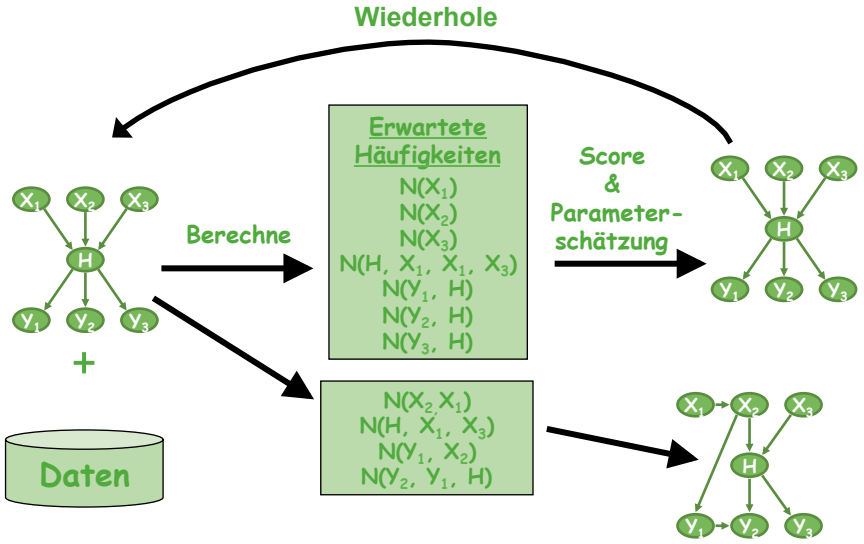
**Idee:**
- Benutzte das aktuelle Model, um die Schätzung der neuen Parameter zu beschleunigen

**Das Prinzip:**
- Such in Model und Parameterraum
- In jeder Iteration benutzen wir das aktuelle Model:
  - Bessere Paramet: "parametrischer" EM Schritt

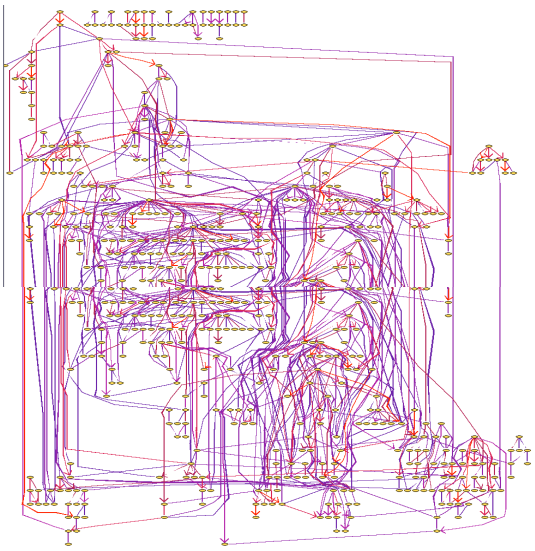
oder
  - Bessere Struktur: "struktureller" EM Schritt

# Structural EM

[Friedman et al. 98]




## Genregulationsnetzwerke (Friedman et al.)



Yeast Daten
[Hughes et al 2000]

- 600 Gene
- 300 experimente

# EINE EINFACHE ANWENDUNG: FILTERN VON SPAM MITTELS NAIVE BAYES



## Naive Bayes

**Aufgabe:** Classify a new email *D* based on a tuple of attribute values $D = \langle x_1, x_2, \ldots, x_n \rangle$ into one of the classes $c_j \in C$

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid x_1, x_2, \ldots, x_n)$$

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(x_1, x_2, \ldots, x_n \mid c) P(c)}{P(x_1, x_2, \ldots, x_n)}$$

$$= \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

Z.B. bag-of-word Merkmale wenn wir Textdokumente klassifizieren wollen

## Naïve Bayes: Annahmen

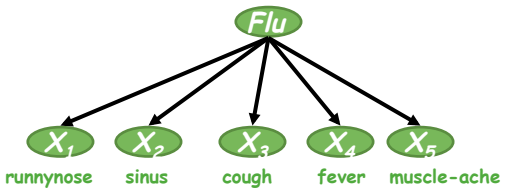
- $P(c_j)$
  - Häufigkeiten der Klassen geschätzt aus den Daten
- $P(x_1,x_2,\ldots,x_n|c_j)$
  - O(|X|$^n$•|C|) Parameter
  - Können wir nur gut schätzen, wenn wir Millionen oder gar Milliarden an Emails haben (Google kann das, ;-))

**Naïve Bayes Unabhängigkeitsannahme hilft uns hier!**

- ***Wir nehmen an, dass die Wahrscheinlichkeit, eine Merkmalsausprägung zu beobachten, propotional zum Produkt der Einzelwahrscheinlichkeiten $P(x_i|c_j)$ ist.***

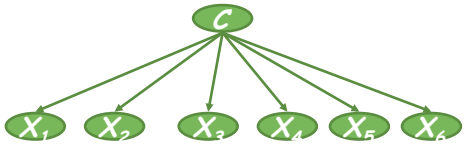


## Naïve Bayes




$$P(X_1,\ldots,X_5 \mid C) = P(X_1 \mid C) \bullet P(X_2 \mid C) \bullet \cdots \bullet P(X_5 \mid C)$$

## Learning the Model



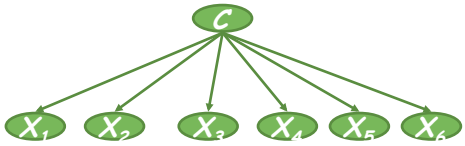

- Maximum Likelihood Schätzung:

$$\hat{P}(c_j) = \frac{N(C = c_j)}{N}$$

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}$$



## Probleme




- Was passiert, wenn wir niemals beobachten, dass ein Patient Muskelschmerzen hat bei einer Grippe?

$$P(X_1, \ldots, X_5 \mid C) = P(X_1 \mid C) \bullet P(X_2 \mid C) \bullet \cdots \bullet P(X_5 \mid C)$$

$$\hat{P}(X_5 = t \mid C = flu) = \frac{N(X_5 = t, C = flu)}{N(C = flu)} = 0$$

- Alles wird 0 !!!!!!!!!!

$$1 = \arg\max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c) = 0$$

## Glättung

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k}$$

**# der Zustände von $X_i$**

- Oder allgemeiner:

**Unsere erwartete Häufigkeit für $X_i = x_{i,k}$**

$$\hat{P}(x_{i,k} \mid c_j) = \frac{N(X_i = x_{i,k}, C = c_j) + mp_{i,k}}{N(C = c_j) + m}$$



## SPAMCOP

- Naïve Bayes Klassifikator
- M ist spam falls P(Spam|M) > P(NonSpam|M)

- Herangehensweise
  - Betrachte nur Wortstämme
  - Stopp-Wörter etc. eliminiert
  - Schätzung der Parameter mittels m-Estimate
  - **Training: 160 spams, 466 non-spams**
  - **Test: 277 spams, 346 non-spams**

- Ergerbnis: Fehlerrate von 4.33%

## Zusammenfassung

- Unsicherheit ist allgegenwärtig und können mittels Wahrscheinlichkeiten dargestellt werden
- Graphische Modelle sind eine kompakte Darstellung für Verteilungen und …
- führen zu „effizienten“ Algorithmen zum Schlussfolgern unter Unsicherheit
  - z.B. Variablen-Elimination
- Parameterschäztung: EM und Co
- Modelselektion mittels heuristischer Suche und „Structural EM“
- Naive Bayes Beispiel



## Es gäbe noch sehr viel mehr zu sagen

J. Pearl, *“Probabilistic Reasoning in Intelligent Systems“*, Morgan-Kaufmann, 1988. Das einflussreichste Buch

F.W. Jensen, “Belief Networks and Decision Graphs”, Springer 2001. Sehr verständlich

C.M. Bishop, “Pattern Recognition and Machine Learning”, Springer 2006. Graphische Modelle fürs Maschinelle Lernen

R.G. Cowell, P. Dawid, S.L. Lauritzen, D.J. Spiegelhalter, “Probabilistic Networks and Expert Systems”, Springer 2007. Alles über den Bezug zwischen Graphischen Modellen und Graphen-Theorie

A. Darwiche, “Modeling and Reasoning with Bayesian Networks”, Cambridge University Press 2009. Effiziente Datenstrukturen wie z.B. OBDDs, ACs, etc.

**D. Koller, N. Friedman „Graphical Models: Principles and Techniques“, MIT Press 2009.** **The new bible**

**Danke für die Aufmerksamkeit!**