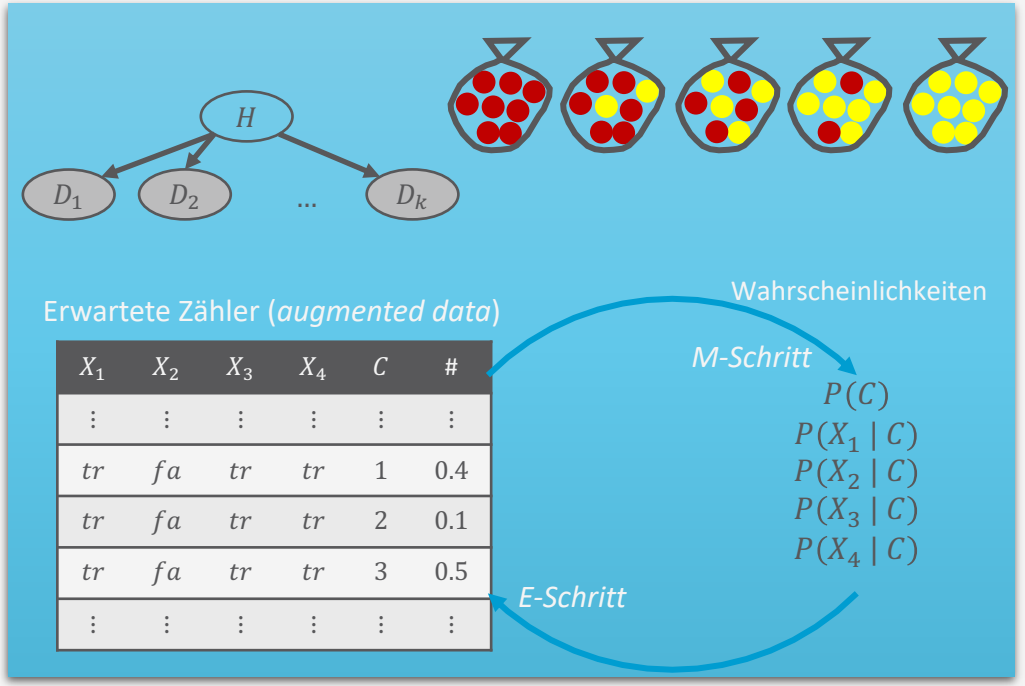
WWU
MÜNSTER

Lernalgorithmen für Episodische PGMs

Einführung in die Künstliche Intelligenz



wissen.leben

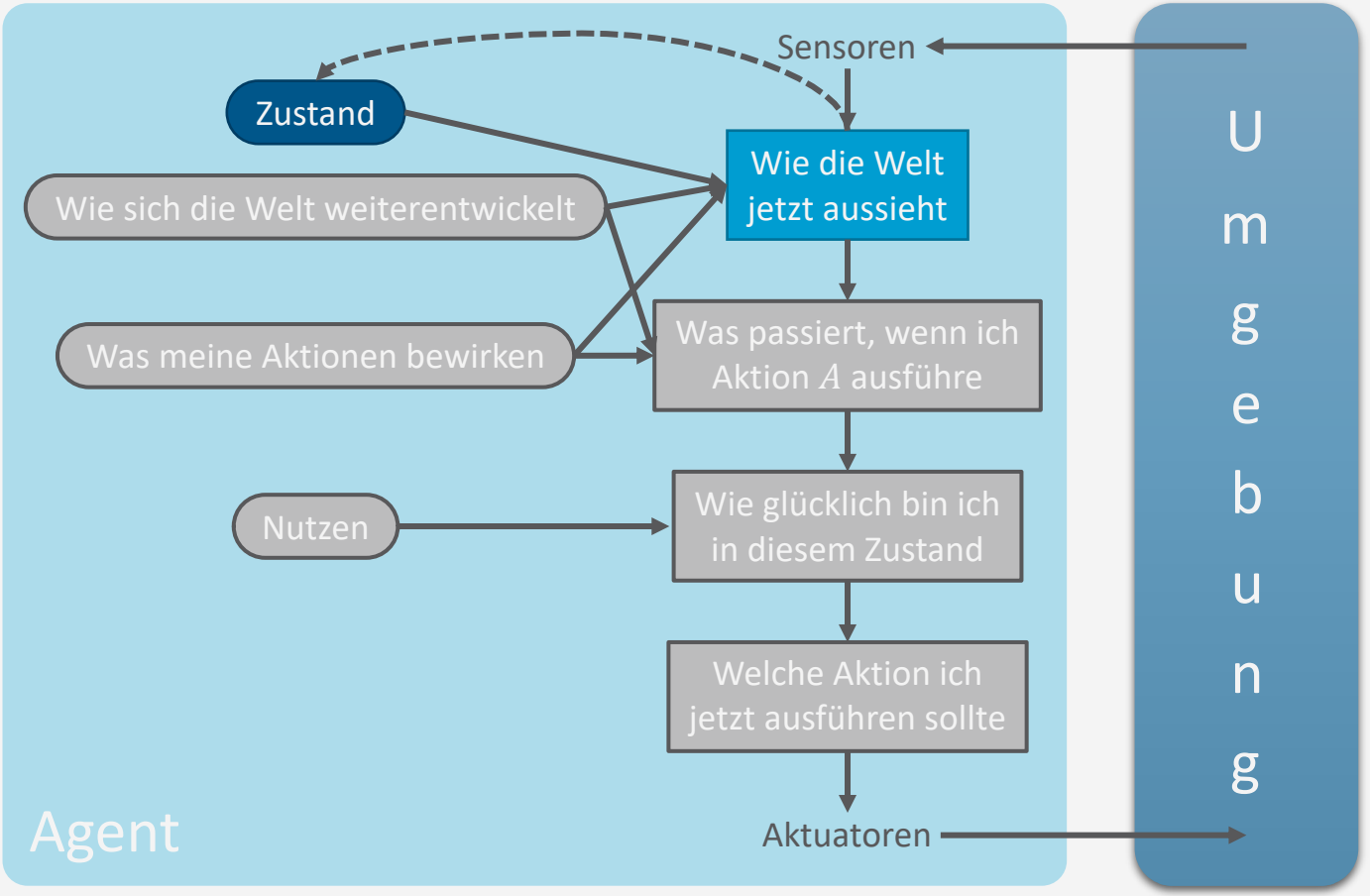
Tanya Braun
Arbeitsgruppe Data Science, Institut für Informatik

Inhalte

1. **Künstliche Intelligenz & Agenten**
   - Agentenabstraktion, Rationalität
   - Aufgabenumgebung
2. **Episodische PGMs**
   - Gerichtetes Modell: Bayes Netze (BNs)
   - Ungerichtete Modelle
3. **Exakte Inferenz in episodischen PGMs**
   - Wahrscheinlichkeits- und Zustandsanfragen
   - Direkt auf den Modellen, mittels Hilfsstrukturen
4. **Approximative Inferenz in episodischen PGMs**
   - Wahrscheinlichkeitsanfragen
   - Deterministische, stochastische Algorithmen
5. **Lernalgorithmen für episodische PGMs**
   - Bei (nicht) vollständigen Daten, (un)bekannter Struktur
6. **Sequentielle PGMs und Inferenz**
   - Dynamische BNs, Hidden-Markov-Modelle
   - filtering / prediction / hindsight Anfragen, wahrscheinlichste Zustandssequenz
   - Exakter, approximativer Algorithmus
7. **Entscheidungstheoretische PGMs**
   - Präferenzen, Nutzenprinzip
   - PGMs mit Entscheidungs- und Nutzenknoten
   - Berechnung der besten Aktion (Aktionssequenz)
8. **Abschlussbetrachtungen**

Einordnung der Vorlesung: *Modell- und nutzenbasierter Agent*

- Nachfolgende Themen der Vorlesung
  2. Episodische PGMs
  3. Exakte Inferenz in episodischen PGMs
  4. Approximative Inferenz in episodischen PGMs
  5. Lernalgorithmen für episodische PGMs
  6. Sequentielle PGMs und Inferenz
  7. Entscheidungstheoretische PGMs

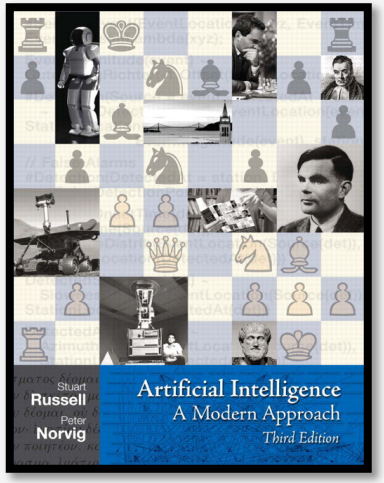


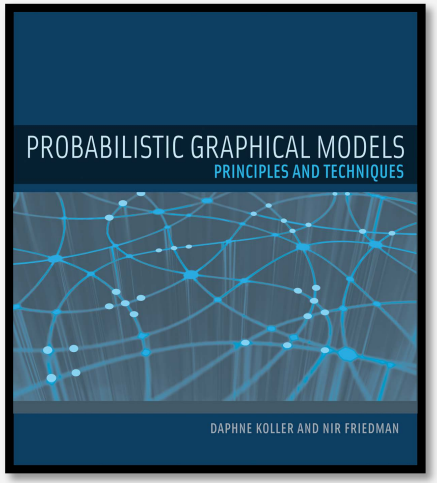
Literaturhinweise

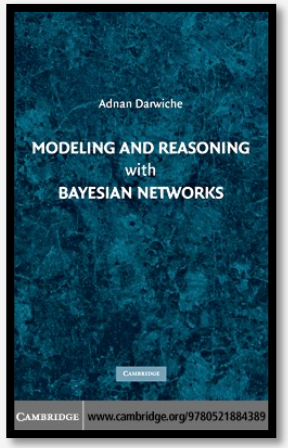
Inhalte dieses Themenblocks werden in den folgenden Kapiteln der Vorlesungsbücher behandelt

- AIMA(de)
  - Kap. 20: Lernen probabilistischer Modelle
- PGM
  - *Teil III: Lernen*

Dieser Thementeil basiert auf AIMA(de), Kap. 20, sowie den Papieren zu LDA und IFP wie auf den Folien zitiert, mit einigen Hinweisen rechts und links über den Tellerrand.
Hinweis: Das PGM Buch deckt sehr viel mehr ab, als die hier besprochenen Inhalte.



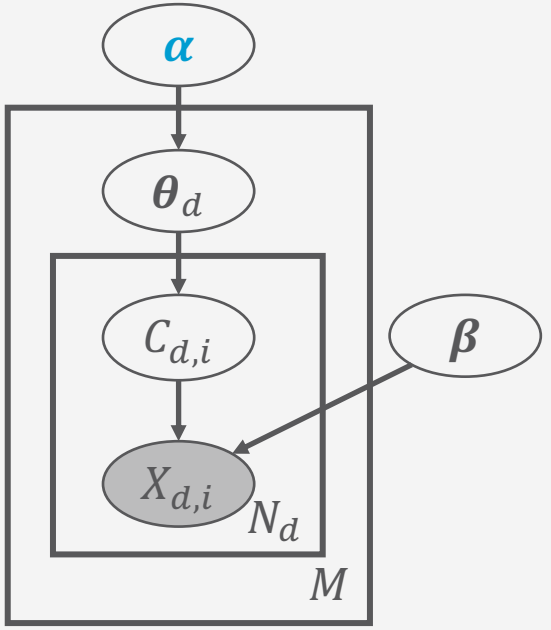




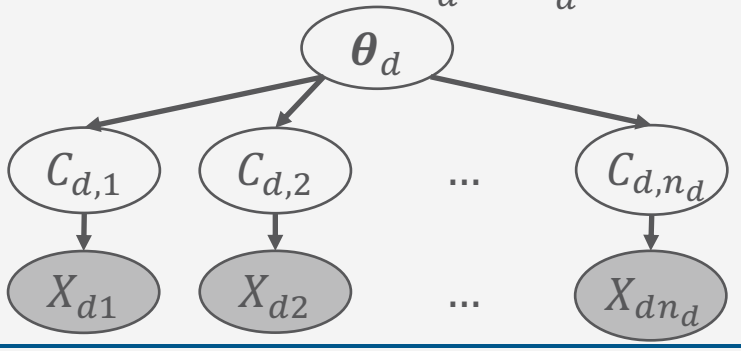
- Wer gerne ein anderes Buch ausprobieren möchte (Fokus auf BNs):
  - Adnan Darwiche, *Modelling and Reasoning with Bayesian Networks*, 2009.

Inferenzaufgaben rund um Topic Modelle

- Gegeben ein Topic Modell für ein Korpus
  - Anfrage: Gegeben ein unbekanntes Dokument, welches Topic / welche Topic-Verteilung hat es?
    - Exakte + approximative Anfragenbeantwortung in PGMs: Thema 3 + 4
  - Generierung eines neuen Dokuments: Gegeben ein gewähltes Topic, generiere die Worte eines neuen Dokuments
    - Sampling aus PGMs: Thema 4
- Gegeben ein Korpus
  - Lerne ein Topic Modell: Thema 5
    - Gegeben: Worte in den Dokumenten (nach Vorverarbeitung), Anzahl Topics K, Hyperparameter $\boldsymbol{\alpha}$
    - Lerne: Topicverteilungen pro Dokument, Wortverteilungen pro Topic (Basis-Verfahren: *Latent Dirichlet Allocation, LDA*)



Pro Dokument d mit $\boldsymbol{\theta}_d \neq \boldsymbol{\theta}_{d'}$:



David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

Überblick: 5. Lernalgorithmen für episodische PGMs

A. *Überblick*
- Full Bayesian Learning, MAP Learning, Maximum Likelihood (ML) Learning

B. *ML-basiertes Parameter Lernen*
- In BNs: ML (vollständige Daten), Expectation-Maximisation (EM; nicht-vollständige Daten)
- In Faktormodellen: Iterative Proportional Fitting (IPF), EM-IPF
- Anwendung: LDA

C. *ML-basiertes Struktur Lernen*
- Model Selection, Structural EM

D. *Abschlussbemerkungen*
- Alternativen, generative vs. diskriminative Modelle, Korrelation vs. Kausalität

Lernen

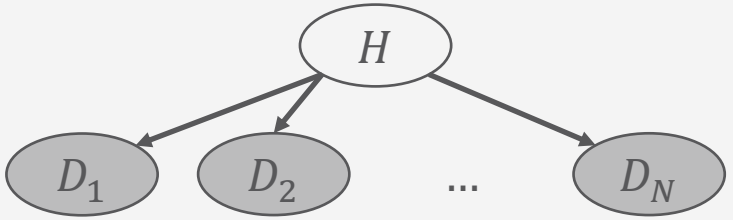
- Aufgabe: Lernen einer Verteilung
  - Gegeben Daten (als Evidenz)
    - Menge von zusammengesetzten Ereignissen
    - Annahme: aus Verteilung erzeugt
  - Parameter (Einträge in Verteilung) bestimmen, so dass die Daten am wahrscheinlichsten werden
    - Hypothese: Mögliche Parameterausprägung
      - Theorie darüber, wie Umgebung funktioniert
      - Hypothesenraum: alle möglichen Ausprägungen

Quasi Sampling rückwärts:
- Sampling: Gegeben Verteilung, Daten erzeugen
- Lernen: Gegeben Daten, Verteilung erzeugen

- Lernproblem anhand Vorhersage eines nächsten Datums anschauen
  - Welche Verteilung generiert das Datum?
- Lernansätze
  - Full Bayesian Learning
    - Unter Berücksichtigung aller möglichen Hypothesen nächstes Datum vorhersagen
  - Maximum A Posteriori (MAP) Learning
    - Unter Berücksichtigung der MAP Hypothese nächstes Datum vorhersagen
  - Maximum Likelihood (ML) Learning
    - Unter Berücksichtigung der ML Hypothese nächstes Datum vorhersagen

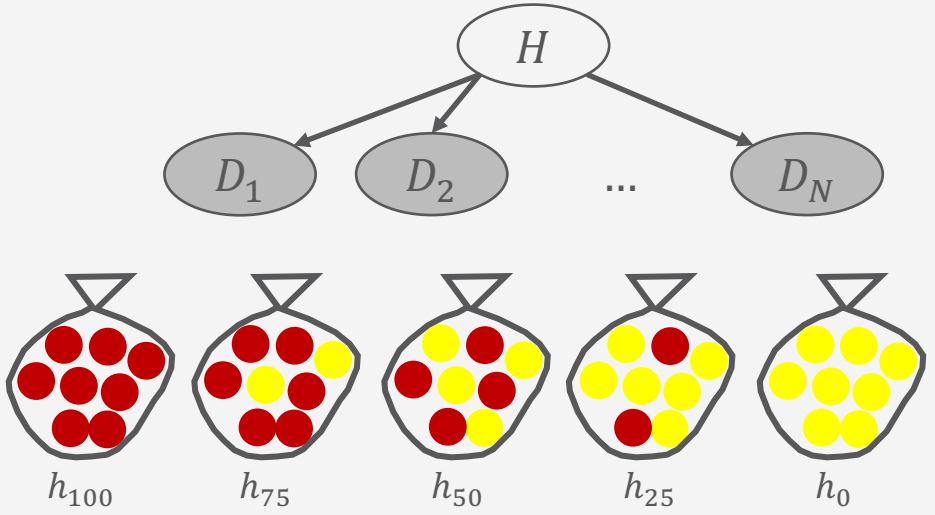
Full Bayesian Learning

- Full Bayesian Learning
  - Aktualisierung einer Wahrscheinlichkeitsverteilung über den Hypothesenraum gegeben Daten
- H ist die Hypothesen-Zufallsvariable
  - Mögliche Hypothesen (Domäne von H) $\mathrm{Val}(H) = \{h_1, \dots, h_n\}$
  - D.h. Zufallsvariable mit dem Hypothesenraum als Domäne
  - $P(H) =$ a priori Wahrscheinlichkeitsverteilung über Hypothesenraum
- Evidenzvariablen D_j für Beobachtungen über $j = 1, \dots, N$
  - Trainingsdaten $\boldsymbol{d} = d_1, \dots, d_N$
  - Bedingte Wahrscheinlichkeiten $P(\boldsymbol{d} \mid h_i)$: Gegeben h_i, Wahrscheinlichkeit der Daten $\boldsymbol{d}$
    - Bei unabhängig und identisch verteilten Daten (*independent and identically distributed*, i.i.d.): $P(\boldsymbol{d} \mid h_i) = P(d_1 \mid h_i) \cdot \dots \cdot P(d_N \mid h_i) = \prod_{j=1}^{N} P(d_j \mid h_i)$



Beispiel

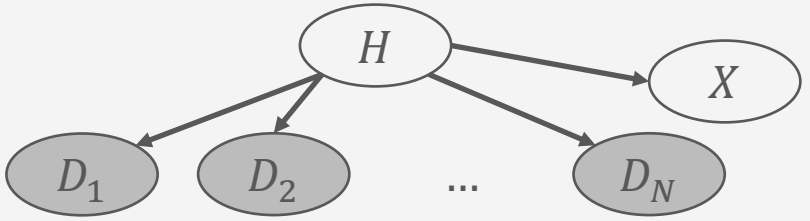
- Wir nehmen an, es gibt 5 Arten von Bonbon-Tüten
  - Bonbons gleich verpackt
  - 10% sind 100% Kirsch-Bonbons (h_{100})
  - 20% sind 75% Kirsch + 25% Zitronen-Bonbons (h_{75})
  - 40% sind 50% Kirsch + 50% Zitronen-Bonbons (h_{50})
  - 20% sind 25% Kirsch + 75% Zitronen-Bonbons (h_{25})
  - 10% sind 100% Zitronen-Bonbons (h_0)
  - Bezeichnen wir mit θ den Parameter, der den Anteil an Kirsch-Bonbons spezifiziert und h_θ die dazugehörige Hypothese → 5 Hypothesen $h_{100}, h_{75}, h_{50}, h_{25}, h_0$, $P(H) = (0.1, 0.2, 0.4, 0.2, 0.1)$



- Dann beobachten wir Bonbons aus einer Tüte: ●●●●●●●●●● ($d_1, \dots, d_{10}$ = 10 mal Zitronen-Bonbons)
- Frage: Welche der 5 Arten von Tüten hat die 10 Beobachtungen generiert? $P(h_\theta \mid \boldsymbol{d})$
  - Welchen Geschmack wird der nächste Bonbon haben? Vorhersage $P(X \mid \boldsymbol{d}) = P(D_{N+1} \mid \boldsymbol{d})$

Full Bayesian Learning

- Gegeben die Daten soweit hat jede Hypothese h_i eine a posteriori Wahrscheinlichkeit:
  - $P(h_i \mid \boldsymbol{d}) \propto P(\boldsymbol{d} \mid h_i)P(h_i)$ (Bayes Theorem)
    - $P(\boldsymbol{d} \mid h_i)$ Likelihood der Daten unter Hypothese h_i genannt



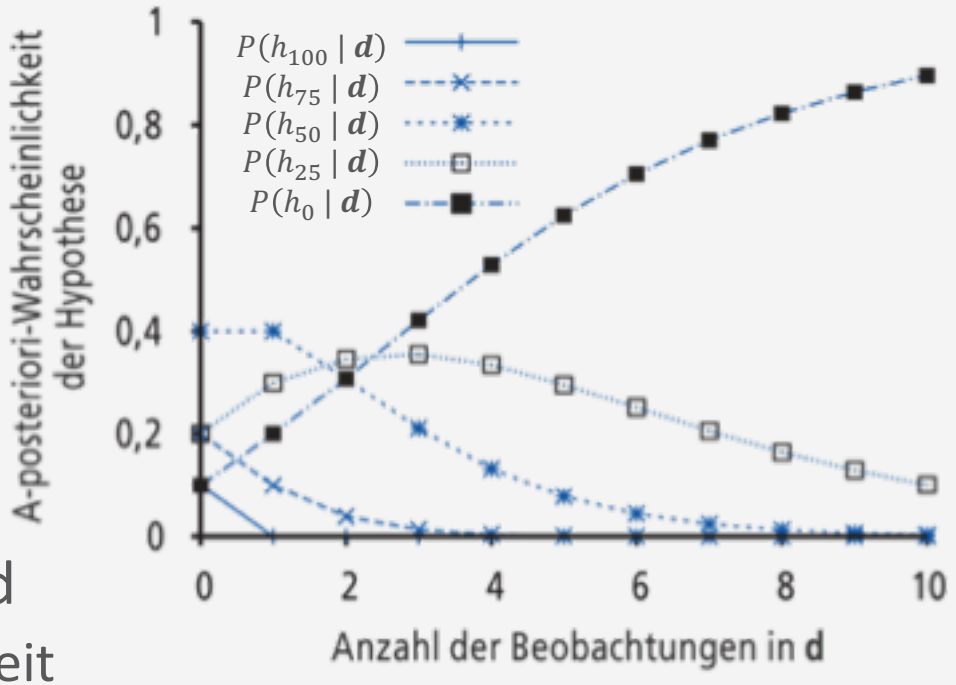
- Vorhersagen über eine neue Entität X sind der gewichtete Durchschnitt über die Vorhersagen aller Hypothesen:
  - $P(X \mid \boldsymbol{d}) = \sum_{h_i \in \mathrm{Val}(H)} P(X, h_i \mid \boldsymbol{d})$ (Nicht-Anfragevariable aussummieren)
    $= \sum_{h_i \in \mathrm{Val}(H)} P(X \mid h_i, \boldsymbol{d}) P(h_i \mid \boldsymbol{d})$ (Kettenregel)
    $= \sum_{h_i \in \mathrm{Val}(H)} P(X \mid h_i) P(h_i \mid \boldsymbol{d})$ (X unabhängig von $\boldsymbol{d}$ gegeben h_i)
    $\propto \sum_{h_i \in \mathrm{Val}(H)} P(X \mid h_i) P(\boldsymbol{d} \mid h_i) P(h_i)$ (Bayes Theorem – wie oben)
- Vorteil: Nicht eine Hypothese wählen müssen
- Nachteil: Iteration über alle Hypothesen

Beispiel: Likelihood

- Wenn unsere Tüten groß genug sind oder genauer, wir jeden Bonbon nach dem Check der Sorte wieder verpacken und in die Tüte zurücklegen, sind die zehn Beobachtungen i.i.d.:
  - $P(\boldsymbol{d} \mid h_\theta) = \prod_j P(d_j \mid h_\theta), j = 1, \dots, 10$
- Für eine gegebene Hypothese h_θ, der Wert von $P(d_j \mid h_\theta)$ ist
  - $P(d_j = cherry \mid h_\theta) = \theta, (d_j = lime \mid h_\theta) = 1 - \theta$
- Gegeben N Beobachtungen, von denen c Kirsche und $l = N - c$ Zitrone sind, gilt
  - $P(\boldsymbol{d} \mid h_\theta) = \prod_{j=1}^{c} \theta \prod_{j=1}^{l} (1-\theta) = \theta^c (1-\theta)^l$
    - Binomial-Verteilung: Wahrscheinlichkeit von # an Erfolgen in einer Reihe von N unabhängigen Versuchen mit binärem Ausgang, jeder Versuch ist erfolgreich mit Wahrscheinlichkeit θ
  - Beispiel: Nachdem man $c = 3$ mal Zitronen-Bonbons beobachtet hat:
    - $P([lime, lime, lime] \mid h_{50}) = 0.5^3$
      - Wahrscheinlichkeit Zitrone zu beobachten ist 0.5 unter der Hypothese h_{50}

100% Zitrone: A Posteriori Wahrscheinlichkeitsverteilung von H

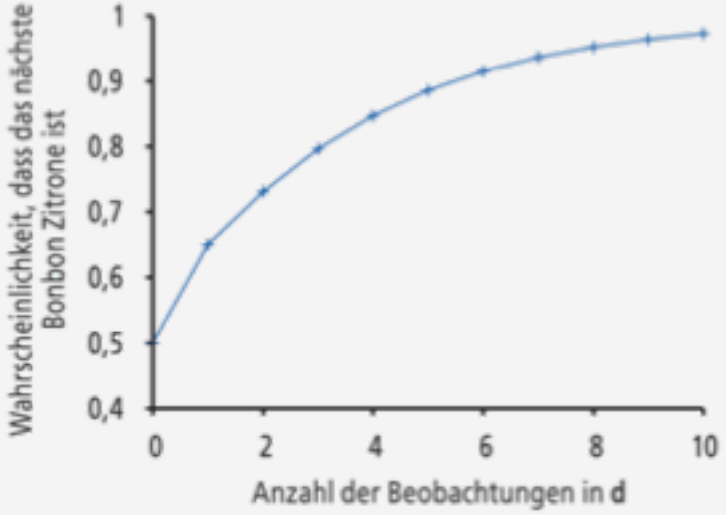
- Abbildung zeigt $P(h_i \mid \boldsymbol{d}) \propto P(\boldsymbol{d} \mid h_i)P(h_i)$ bei Beobachtungen rein von Zitrone
  - $\boldsymbol{d} = [lime, lime, \dots]$
- Initial dominieren die Hypothesen h_i mit größeren Apriori-Wahrscheinlichkeiten
  - h_{50} mit Apriori-Wahrscheinlichkeit 0.4
- Während Daten reinkommen, fängt die am Ende beste Hypothese (h_0) an zu dominieren, da die Wahrscheinlichkeit diese Daten zu beobachten gegeben die anderen Hypothesen immer kleiner wird
  - Nach drei Mal Zitrone in Reihe, nimmt Wahrscheinlichkeit extrem zu, dass wir eine 100% Zitrone Tüte haben

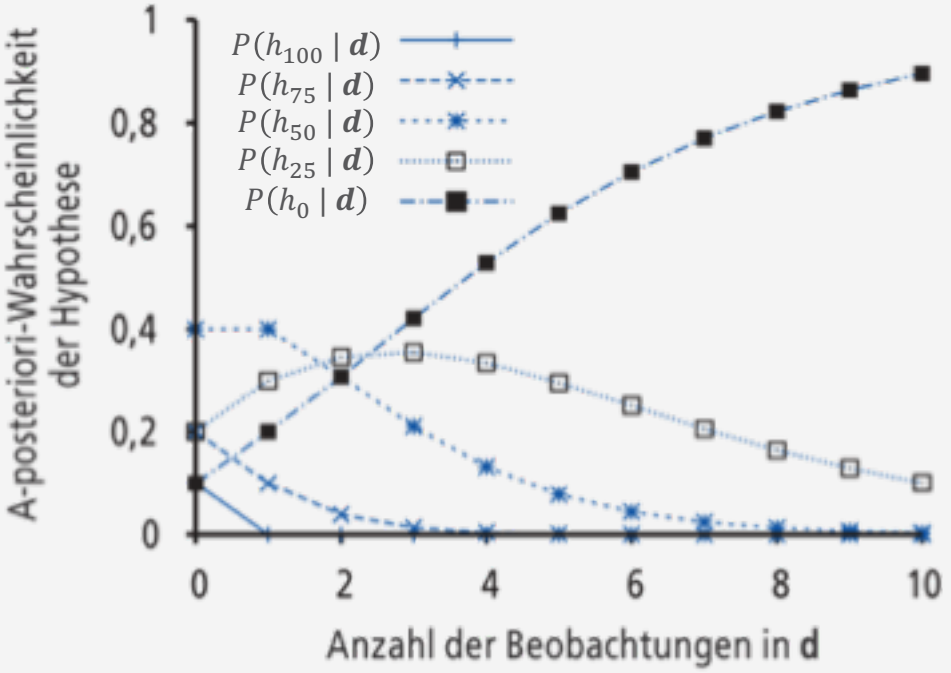


$P(H) = (0.1, 0.2, 0.4, 0.2, 0.1)$

Vorhersagewahrscheinlichkeiten

- Abbildung unten zeigt $P(D_{N+1} = lime \mid \boldsymbol{d})$ $\propto \sum_{h_i \in \mathrm{Val}(H)} P(D_{N+1} = lime \mid h_i) P(\boldsymbol{d} \mid h_i) P(h_i)$





$P(H) = (0.1, 0.2, 0.4, 0.2, 0.1)$

- Wahrscheinlichkeit, dass nächster Bonbon Zitrone ist steigt mit der Wahrscheinlichkeit, dass die Tüte eine 100%-Zitrone-Tüte ist

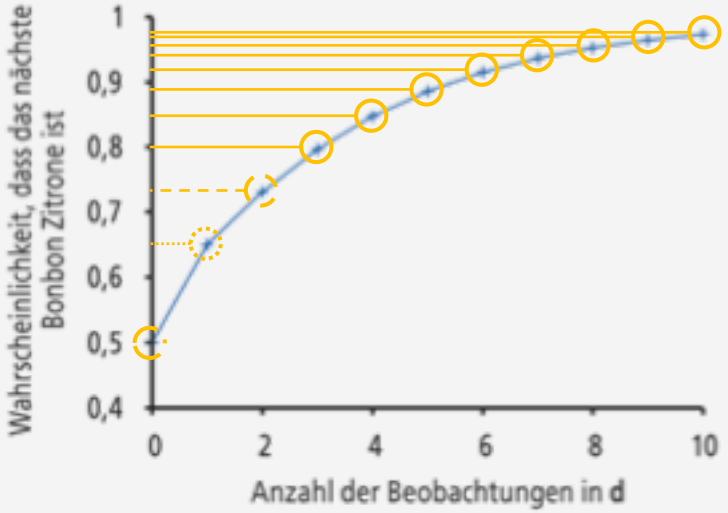
Abbildungen: AIMAde, S. 930

MAP Approximation

- Full Bayesian Learning erscheint wie eine sehr sichere Wette, aber funktioniert nicht gut in der Praxis
  - Über den Hypothesen-Raum zu summieren (oder integrieren) ist meist intraktabel
    - $2^{2^6} = 18{,}446{,}744{,}073{,}709{,}551{,}616$ Boolesche Funktionen für sechs Features
- Übliche Annäherung: Maximum-a-posteriori (MAP) Learning
  - Anstatt Vorhersagen zu treffen, indem man alle möglichen Hypothesen berücksichtigt, wie in
    - $P(X \mid \boldsymbol{d}) = \sum_{h_i \in \mathrm{Val}(H)} P(X \mid h_i) P(h_i \mid \boldsymbol{d}) \propto \sum_{h_i \in \mathrm{Val}(H)} P(X \mid h_i) P(\boldsymbol{d} \mid h_i) P(h_i)$
  - Vorhersagen treffen basierend auf h_{MAP}, welche $P(h_i \mid \boldsymbol{d})$ maximiert
    - I.e., maximiert $P(\boldsymbol{d} \mid h_i) P(h_i) \rightarrow h_{MAP} = \arg\max_{h_i \in \mathrm{Val}(H)} P(\boldsymbol{d} \mid h_i) P(h_i)$
    - $P(X \mid \boldsymbol{d}) \approx P(X \mid h_{MAP})$

MAP Approximation

- MAP ist eine Schätzung, wenn $P(X \mid \boldsymbol{d}) \approx P(X \mid h_{MAP})$
  - Im Beispiel ist h_{MAP} die 100%-Zitrone-Tüte nach nur 3 Bonbons, mit der Vorhersage, dass der nächste Bonbon Zitrone ist mit Wahrscheinlichkeit 1
  - Full Bayesian Learner gab da noch die Vorhersage von Zitrone mit einer Wahrscheinlichkeit von 0.8 an



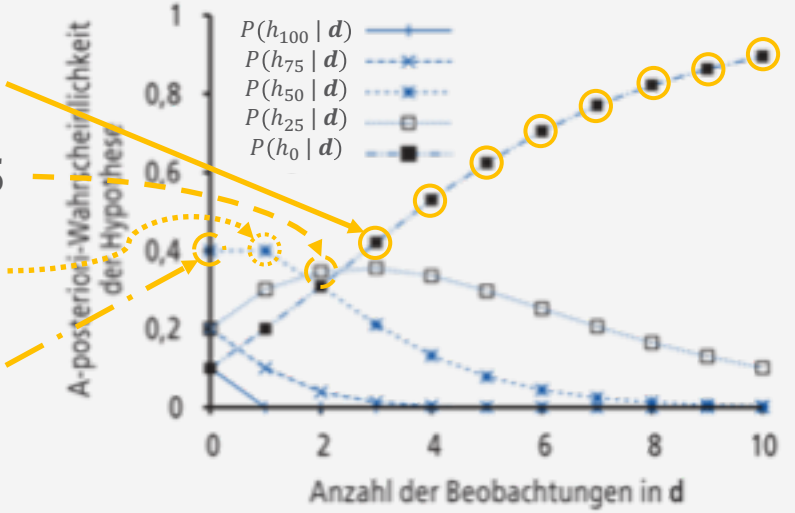
$$\vdots$$

$$h^3_{MAP} = h_0 \rightarrow P(X \mid h_0) = 1.0$$

$$h^2_{MAP} = h_{25} \rightarrow P(X \mid h_{25}) = 0.75$$

$$h^1_{MAP} = h_{50} \rightarrow P(X \mid h_{50}) = 0.5$$

$$h^0_{MAP} = h_{50} \rightarrow P(X \mid h_{50}) = 0.5$$

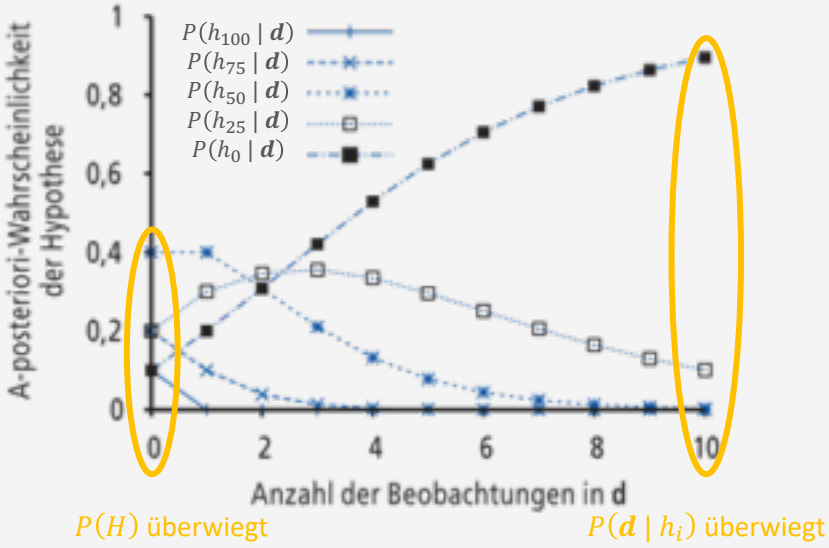


Bias

- Je mehr Daten eintreffen, desto näher beieinander liegen die MAP und Full Bayesian Vorhersagen, da die Nicht-MAP-Hypothesen weniger wahrscheinlich werden
- Meistens einfacher MAP Hypothesen zu finden, da Optimierungsproblem (statt Summierungsproblem)
- $P(H)$ spielt wichtige Rolle in MAP und Full Bayesian Learning
  - Gibt den Bias an: bestimmte Hypothesen werden a priori bevorzugt
  - Trade-off zwischen Modellkomplexität und Fähigkeit, Daten möglichst gut zu reproduzieren
    - Komplexere Modelle können die Daten meist besser erklären → höhere $P(\boldsymbol{d} \mid h_i)$
      - Gefahr der Überanpassung (*overfitting*)
    - Komplexere Modelle weniger wahrscheinlich a priori, da es mehr komplexere Modelle gibt als einfachere → geringere $P(h_i)$
  - I.e., üblicher Bias beim Lernen: Komplexität bestrafen

Maximum Likelihood (ML) Learning

- Weitere Vereinfachung gegenüber Full Bayesian und MAP Learning
  - A-priori-Verteilung über den Hypothesenraum uniform
  - MAP Learning vereinfacht sich zu ML Learning, da $P(h_i)$ immer gleich
    - MAP: Maximiere $P(\boldsymbol{d} \mid h_i)P(h_i)$ → ML: Maximiere $P(\boldsymbol{d} \mid h_i)$
      - $h_{MAP} = \arg\max_{h_i \in \mathrm{Val}(H)} P(\boldsymbol{d} \mid h_i)P(h_i) \rightarrow h_{ML} = \arg\max_{h_i \in \mathrm{Val}(H)} P(\boldsymbol{d} \mid h_i)$
- Wann ist ML angemessen?
  - Wenn die Hypothesen tatsächlich gleich wahrscheinlich sind (z.B., gleiche Komplexität → keine Hypothese zu bevorzugen)
  - In der Statistik die Standard (nicht-Bayes'sche) statistische Lernmethode, verwendet vor allem bei Misstrauen gegenüber subjektiven Natur der a priori Verteilung über die Hypothesen
  - Bei sehr großen Datenmengen neigt $P(\boldsymbol{d} \mid h_i)$ den Einfluss von $P(h_i)$ zu überwinden



Ronald A. Howard: Decision Analysis: Perspectives on Inference, Decision, and Experimentation. In: *Proceedings of the IEEE 58(5)*, 1970

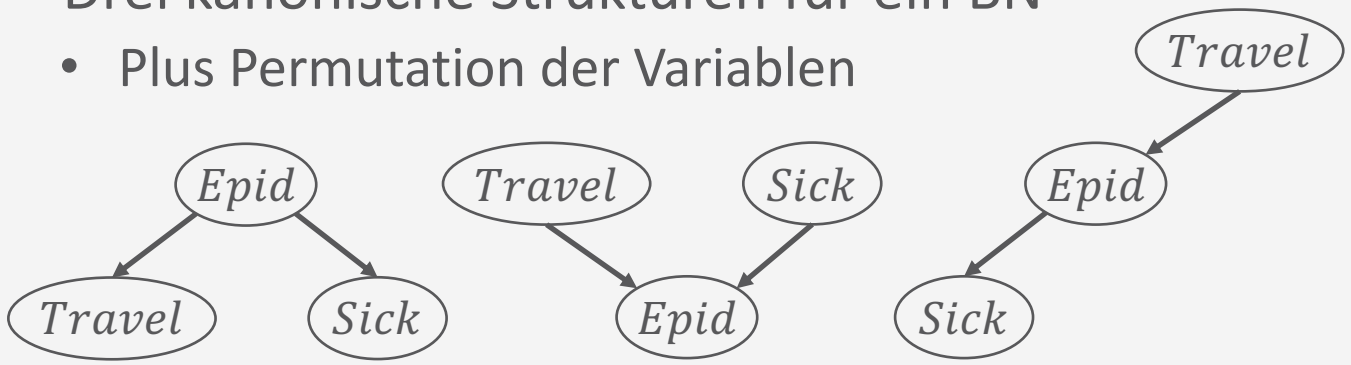
Lernen in PGMs: Dimensionen

- Annahme: Faktorisierung der vollständigen gemeinsamen Verteilung in irgendeiner Form gewünscht
- Hypothesenraum
  - Mögliche Strukturen
    - Welche Variablen?
    - Welche Kanten?
  - Mögliche Parameter
    - Welche Einträge in den CPDs / Faktoren?
  - Beispiel: Wie die folgenden Variablen kombinieren?

- Daten
  - Vollständig vorhanden
    - Für jede existierende Zufallsvariable gibt es Beobachtungen
      - Dann Variablen in möglichen Strukturen bekannt
  - Nicht vollständig vorhanden
    - Zufallsvariable bekannt, aber keine Daten vorhanden
    - Nicht bekannt, ob es nur weitere Zufallsvariablen geben sollte (somit auch keine Daten dafür)
  - Beispiel: Ereignisse für alle drei Variablen oder nur für eine Untermenge von ihnen

Beispiel

- Drei kanonische Strukturen für ein BN
  - Plus Permutation der Variablen



- Struktur links: Verteilungen mit $\theta_0, \dots, \theta_4 \in [0,1]^5$ für $|\mathrm{Val}(R)| = 2$

Epid	*P*
false	$1-\theta_0$
true	θ_0

Epid	*Travel*	*P*
false	*false*	$1-\theta_1$
false	*true*	θ_1
true	*false*	$1-\theta_2$
true	*true*	θ_2

Epid	*Sick*	*P*
false	*false*	$1-\theta_3$
false	*true*	θ_3
true	*false*	$1-\theta_4$
true	*true*	θ_4

Travel	*Epid*	*Sick*
false	*false*	*false*
false	*false*	*true*
true	*false*	*false*
false	*true*	*true*
true	*false*	*false*
true	*false*	*true*
false	*false*	*false*
false	*false*	*false*

Menge von zusammengesetzten Ereignissen als Evidenz für Ausprägungen von $\theta_0, \dots, \theta_4 \in [0,1]^5$ bei *vollständigen* Daten

Bei unvollständigen Daten z.B. nur Beobachtungen zu *Travel* und *Sick*

In der Regel geht man davon aus, dass man eine Struktur vorgeben kann und nur noch die Parameter (bei vollständigen oder nicht vollständigen Daten) bestimmen muss

Lernen in PGMs: Lernansätze

- Im Unterschied zu vorher keine Datenvorhersage, sondern Daten im Vorhinein vorhanden
- Frage: Welche Modellausprägung ist die „richtige“?
  - Ausprägung der Modellstruktur und Parameter, welche die Daten am besten modelliert
    - Gegebene Daten am wahrscheinlichsten wieder reproduzieren kann (durch *Sampling*)
  - Full Bayesian Learning: alle Ausprägungen berücksichtigen
    - Pro Verteilung kann der Parameter bei Boole‘schen Zufallsvariablen in $[0,1]$ liegen → kontinuierlich!
    - Erschwert zudem die Anfragebeantwortung, da nicht eine Anfrage an ein Modell gestellt wird, sondern die gleiche Anfrage an alle möglichen Modelle, deren Antworten gemittelt werden
  - MAP Learning: die MAP Ausprägung wählen
  - ML Learning: ML Ausprägung wählen
    - MAP + ML: ein Modell als Ausgabe → Anfragebeantwortung wie bisher

Zwischenzusammenfassung

- Lernproblem: Parameter einer Verteilung lernen
  - Parameter: Eintrag in CPD / Faktor
- Hypothesenraum
  - Hypothese: für jeden Parameter eine mögliche Ausprägung
  - Menge der Hypothesen: Menge der Kombinationen aller möglichen Ausprägungen
- Bestimmen der passenden Hypothese
  - Hypothese, die Daten am wahrscheinlichsten macht
- Methoden
  - Full Bayesian Learning: alle Hypothesen betrachten
  - MAP Learning: maximiert $P(\boldsymbol{d} \mid h_i)P(h_i)$
  - ML Learning: A priori Verteilung über Hypothesenraum uniform; maximiert $P(\boldsymbol{d} \mid h_i)$

Überblick: 5. Lernalgorithmen für episodische PGMs

A. *Überblick*
- Full Bayesian Learning, MAP Learning, Maximum Likelihood (ML) Learning

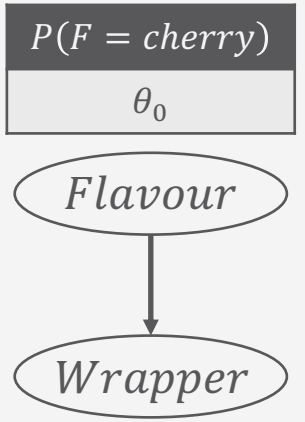
B. *ML-basiertes Parameter Lernen*
- In BNs: ML (vollständige Daten), Expectation-Maximisation (EM; nicht-vollständige Daten)
- In Faktormodellen: Iterative Proportional Fitting (IPF), EM-IPF
- Anwendung: LDA

C. *ML-basiertes Struktur Lernen*
- Model Selection, Structural EM

D. *Abschlussbemerkungen*
- Alternativen, generative vs. diskriminative Modelle, Korrelation vs. Kausalität

Gerichtete Modelle: BNs

ML-basiertes Parameter lernen

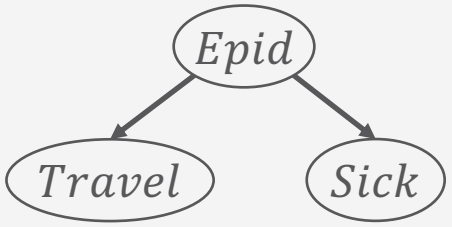


F	$P(W = red \mid F)$
$cherry$	θ_1
$lime$	θ_2

Lernen in BNs: Data Science mit vollständigen Daten

- Wir starten mit dem einfachsten Problem in BN Lernen:
  - Bekannte Struktur
  - Daten beinhalten Beobachtungen für alle Variablen
    - Alle Variablen beobachtbar, keine fehlenden Daten
- Es müssen nur die BN Parameter in den CPDs gelernt werden

Modell



Daten

Travel	Epid	Sick
false	false	false
false	false	true
true	false	false
false	true	true
⋮	⋮	⋮

→ Wahrscheinlichkeitsverteilungen zu lernen

$P(Epid), P(Travel \mid Epid), P(Sick \mid Epid)$

Epid	P
false	$1-\theta_0$
true	θ_0

Epid	Travel	P
false	false	$1-\theta_1$
false	true	θ_1
true	false	$1-\theta_2$
true	true	θ_2

Epid	Sick	P
false	false	$1-\theta_3$
false	true	θ_3
true	false	$1-\theta_4$
true	true	θ_4

Maximum-Likelihood-Parameterschätzung

- Nehme an, die Struktur eines BNs sei bekannt
- Ziel: Schätze BN-Parameter $\boldsymbol{\theta}$
  - Einträge in CPDs $P\big(R \mid \mathrm{Pa}(R)\big)$
- Parametrisierung mit $\boldsymbol{\theta}$ ist gut, falls hierdurch die beobachteten Daten wahrscheinlich generiert werden:

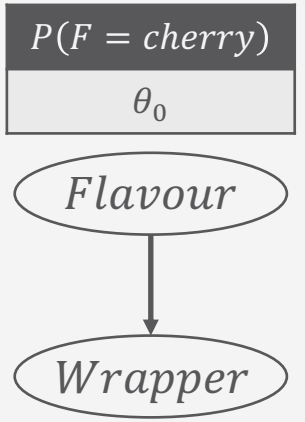
$$P(\boldsymbol{d} \mid \boldsymbol{\theta}) = \prod_{j=1}^{N} P\big(d_j \mid \boldsymbol{\theta}\big)$$

i.i.d. Datenpunkte

- Maximum Likelihood Estimation (MLE) Prinzip: Wähle $\boldsymbol{\theta}$ so, dass $P(\boldsymbol{d}|\boldsymbol{\theta})$ maximiert wird

Beispiel

- Bonbonfabrik:
  - Farbe des Bonbonpapiers mit einer bestimmten Wahrscheinlichkeit je nach Geschmack, wobei die entsprechende Verteilung nicht bekannt sei
    - Wenn $Flavour = cherry$, wähle $Wrapper = red$ mit Wahrscheinlichkeit θ_1
    - Wenn $Flavour = lime$, wähle $Wrapper = red$ mit Wahrscheinlichkeit θ_2
  - Entsprechendes BN dazu enthält drei zu lernenden Parameter
    - $\theta_0, \theta_1, \theta_2$



F	$P(W = red \mid F)$
$cherry$	θ_1
$lime$	θ_2

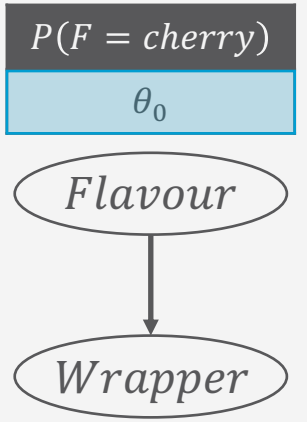
Beispiel

- Wahrscheinlichkeit für z.B. $W = yellow, F = cherry$

$$P(W = yellow, F = cherry \mid h_{\theta_0\theta_1\theta_2})$$
$$= P(W = yellow \mid F = cherry, h_{\theta_0\theta_1\theta_2}) P(F = cherry \mid h_{\theta_0\theta_1\theta_2})$$
$$= (1 - \theta_1)\theta_0$$

- Wir packen N Bonbons aus
  - Jeder Versuch liefert eine Kombination aus Papier und Geschmack wie oben
  - c sind Kirsche, l sind Zitrone ($c + l = N$)
  - c_r sind Kirsche mit rotem Papier, c_y sind Kirsche mit gelbem Papier ($c_r + c_y = c$)
  - l_r sind Zitrone mit rotem Papier, l_y sind Zitrone mit gelbem Papier ($l_r + l_y = l$)
- Wahrscheinlichkeit der Daten:

$$P(\boldsymbol{d} \mid h_{\theta_0\theta_1\theta_2}) = \prod_{j=1}^{N} P(d_j \mid h_{\theta_0\theta_1\theta_2}) = \theta_0^c (1 - \theta_0)^l \theta_1^{c_r} (1 - \theta_1)^{c_y} \theta_2^{l_r} (1 - \theta_2)^{l_y}$$



F	$P(W = red \mid F)$
$cherry$	θ_1
$lime$	θ_2

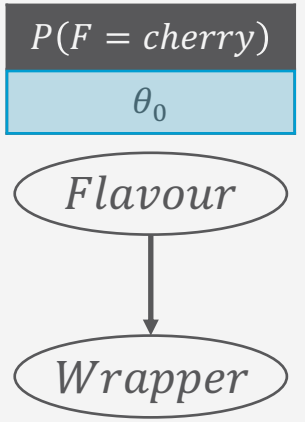
Beispiel

- ML-Hypothese: Werte für $\theta_0\theta_1\theta_2$, so dass $P(\boldsymbol{d} \mid h_{\theta_0\theta_1\theta_2}) = \prod_{j=1}^{N} P(d_j \mid h_{\theta_0\theta_1\theta_2})$ maximal
- Erhält man auch durch Maximierung der Log-Wahrscheinlichkeit:

$$L(d \mid h_{\theta_0\theta_1\theta_2}) = \log P(\boldsymbol{d} \mid h_{\theta_0\theta_1\theta_2}) = \sum_{j=1}^{N} \log P(d_j \mid h_{\theta_0\theta_1\theta_2})$$

  - Vorteil: Aus dem Produkt wird eine Summe
    - In der Regel leichter zu maximieren
  - Für das Beispiel heißt das:

$$\begin{aligned} & L(d \mid h_{\theta_0\theta_1\theta_2}) \\ &= \log\left(\theta_0^{c}(1-\theta_0)^{l}\theta_1^{c_r}(1-\theta_1)^{c_y}\theta_2^{l_r}(1-\theta_2)^{l_y}\right) \\ &= \log\theta_0^{c} + \log(1-\theta_0)^{l} + \log\theta_1^{c_r} + \log(1-\theta_1)^{c_y} + \log\theta_2^{l_r} + \log(1-\theta_2)^{l_y} \\ &= c\log\theta_0 + l\log(1-\theta_0) + c_r\log\theta_1 + c_y\log(1-\theta_1) + l_r\log\theta_2 + l_y\log(1-\theta_2) \end{aligned}$$



F	$P(W = red \mid F)$
$cherry$	θ_1
$lime$	θ_2

Beispiel

- Bestimmung der Ableitungen bzgl. $\boldsymbol{\theta}$ und Maximalwerte bei Ableitung gleich 0 finden
  - Im Beispiel
    - $L(d \mid h_{\theta_0\theta_1\theta_2}) = c \log \theta_0 + l \log(1-\theta_0) + c_r \log \theta_1 + c_y \log(1-\theta_1) + l_r \log \theta_2 + l_y \log(1-\theta_2)$
    - Ableitung bzgl. θ_0

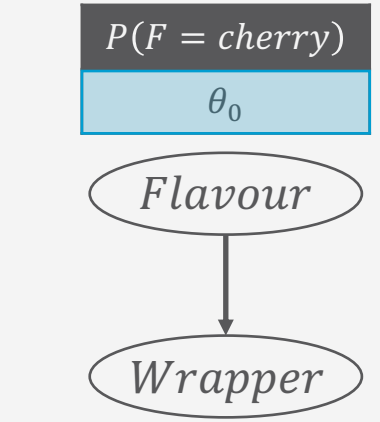
$$\frac{\partial L}{\partial \theta_0} = \frac{c}{\theta_0} - \frac{l}{1-\theta_0} = 0 \quad \Rightarrow \quad \theta_0 = \frac{c}{c+l}$$

    - Ableitung bzgl. θ_1

$$\frac{\partial L}{\partial \theta_1} = \frac{c_r}{\theta_1} - \frac{c_y}{1-\theta_1} = 0 \quad \Rightarrow \quad \theta_1 = \frac{c_r}{c_r+c_y}$$

    - Ableitung bzgl. θ_2

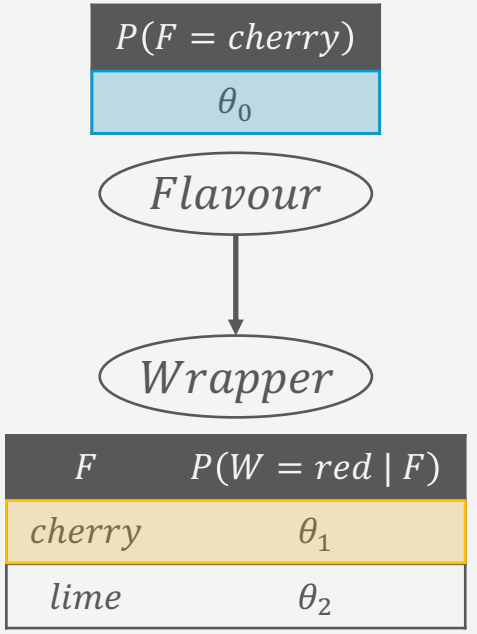
$$\frac{\partial L}{\partial \theta_2} = \frac{l_r}{\theta_2} - \frac{l_y}{1-\theta_2} = 0 \quad \Rightarrow \quad \theta_2 = \frac{l_r}{l_r+l_y}$$



F	$P(W = red \mid F)$
$cherry$	θ_1
$lime$	θ_2

Maximum-Likelihood-Parameterschätzung

- Schätzung durch Bildung relativer Häufigkeiten
- Prozess auf jedes voll beobachtbare BN anwendbar
- Mit vollständigen Daten und MLE:
  - Parameter lernen zerfällt in separate Lernprobleme für jeden Parameter (CPD) durch Logarithmierung
  - Jeder Parameter wird durch die relative Häufigkeit eines Knotenwertes bei gegebenen Werten der Elternknoten bestimmt



Maximum-Likelihood-Parameterschätzung: Allgemeines Vorgehen

- Viel mathematischer Aufwand, um das Offensichtliche zu entdecken
- Skizziert aber ein generelles Vorgehen für das ML-Parameterlernen:
  - Aufgabe: Maximiere $P(\boldsymbol{d} \mid \boldsymbol{\theta})$, i.e., $h_{ML} = \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{d} \mid \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{j=1}^{N} P(d_j \mid \boldsymbol{\theta})$
  - Vorgehen
    1. Aufschreiben eines Ausdrucks für die Wahrscheinlichkeit der Daten als Funktion der Parameter ($P(\boldsymbol{d} \mid \boldsymbol{\theta})$)
    2. Ableitung der Log-Wahrscheinlichkeit für jeden Parameter
    3. Ermittlung von Parameterwerten, für die die Ableitungen null sind
    - Danach in den Daten die benötigten relativen Häufigkeiten bestimmen (i.e., zählen)
- Nebenbemerkung: Sampling lässt sich auf diese Art interpretieren
  - Lernen der wahren Verteilung aus Samples mittels Bestimmen von relativen Häufigkeiten (ML)

Erweiterung auf MAP

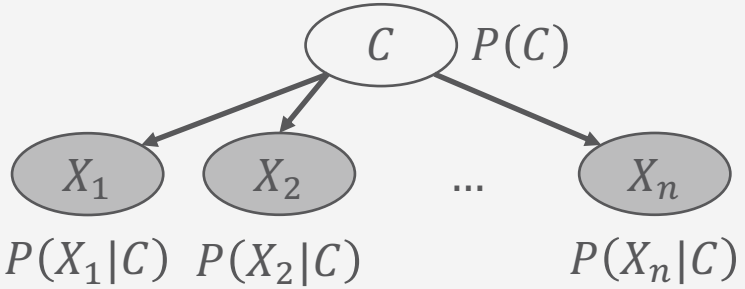
- Für die MAP-Hypothese erhalten wir

$$
\begin{aligned}
h_{MAP} &= \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{d} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \log P\left(d_j \mid \boldsymbol{\theta}\right) P(\boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \log P\left(d_j \mid \boldsymbol{\theta}\right) + \log P(\boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \log \prod_{j=1}^{N} P\left(d_j \mid \boldsymbol{\theta}\right) + \log P(\boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \sum_{j=1}^{N} \log P\left(d_j \mid \boldsymbol{\theta}\right) + \log P(\boldsymbol{\theta})
\end{aligned}
$$

- Gleichungen identisch bis auf Einbindung von $P(\boldsymbol{\theta})$
  - Likelihood $P\left(d_j \mid \boldsymbol{\theta}\right)$ nun gewichtet mit der Apriori-Wahrscheinlichkeit $P(\boldsymbol{\theta})$

Anwendung: Naives Bayes-Klassifikator

- Klassifikation:
  - Menge von Features $X_1, \dots, X_n$
  - Zuweisung eines Klassennamens $C = c$ auf Basis der Featurewerte $x_1, \dots, x_n$
  - Annahme: bedingte Unabhängigkeit gegeben C

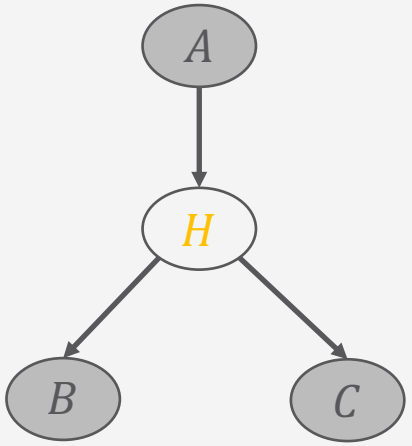


- Vorteile: Relativ wenige Parameter (Einträge in den CPDs) zu lernen
  - $2n + 1$ Parameter bei Booleschen Variablen benötigt (allgemein: $r(r-1)n + r - 1$)
    - $r = |\mathrm{Val}(C)|$, $r - 1$ Einträge in $P(C)$, je $r(r-1)$ Einträge in $P(X_i|C)$ (pro c, $(r-1)$ Einträge lernen)
- Überwachtes Lernen (*supervised learning*; d.h., gelabelte Daten vorhanden):
  - Menge von Featurewerten mit dazugehörigem Klassennamen $\boldsymbol{D} = \left\{\left(x_1^j, \dots, x_n^j, c^j\right)\right\}_{j=1}^{N}$
  - Zählen der relativen Häufigkeiten von jedem c in $\boldsymbol{D}$ für $P(C)$, jedem x_i, c in $\boldsymbol{D}$ für $P(X_i \mid C)$

Parameterlernen mit versteckten Variablen

- Bisherige Annahme: Daten für jede Variable des BNs vorhanden
- Nächst schwierigeres Problem
  - Versteckte Variablen im Netz
    - Keine Beobachtungen vorhanden
  - Ansatz mit relativen Häufigkeiten so nicht übertragbar, da keine Zähler für diese Variablen vorhanden
    - Beispiel: Kein Zähler für H

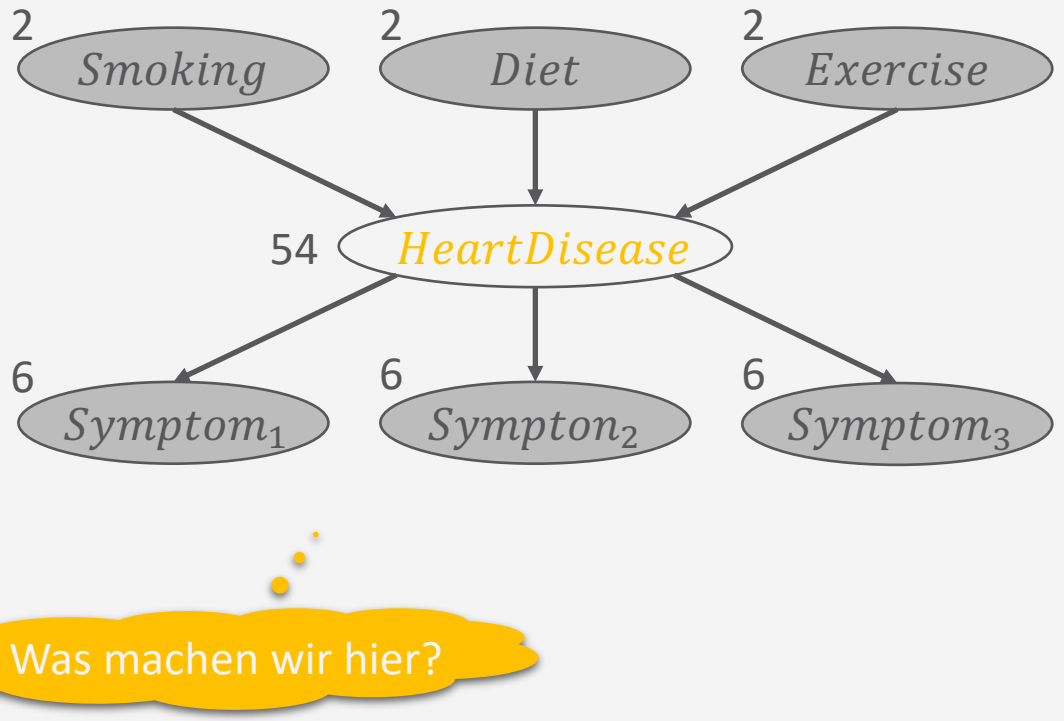
A	B	C
$true$	$false$	$true$
$false$	$true$	$true$
$true$	$true$	$false$
	$\vdots$	



Vermeintliche Lösung

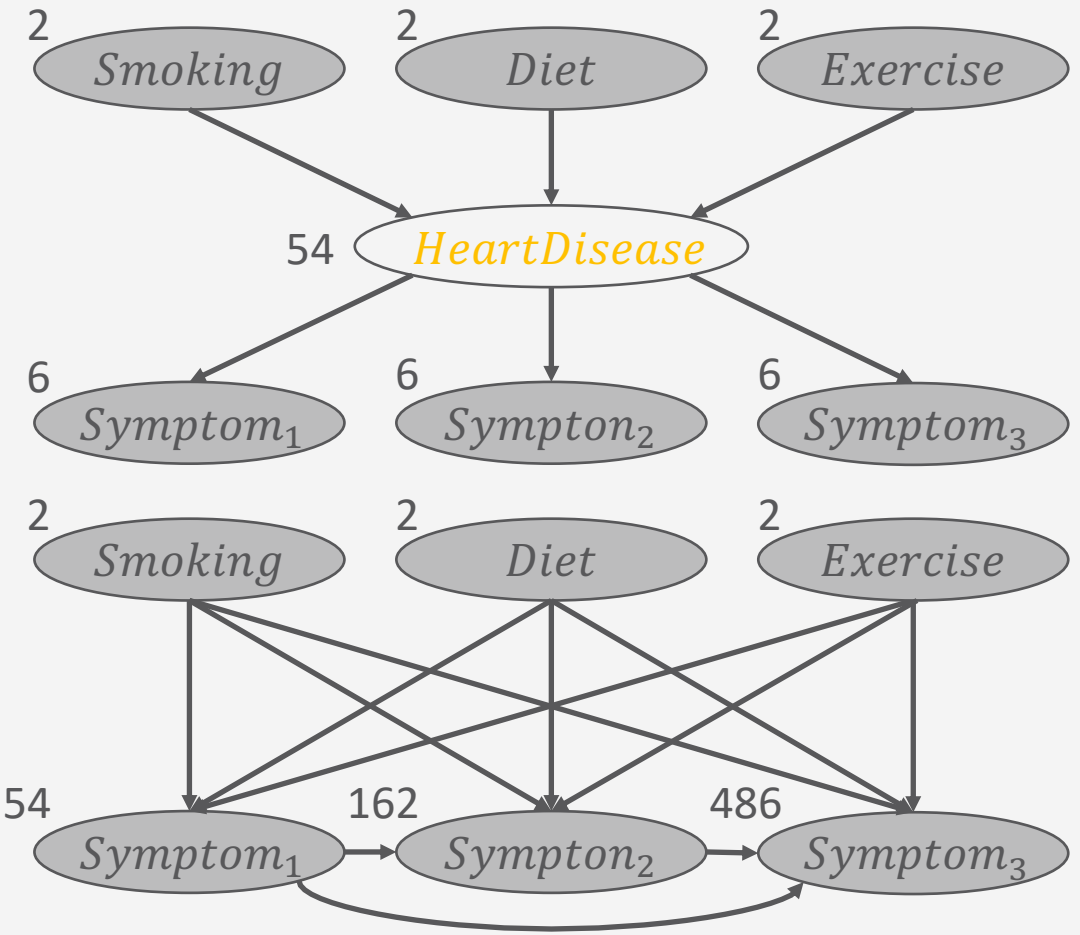
- Vermeide versteckte Variablen
- Könnte klappen bei kleinen Netzwerken

- Beispiel:
  - Jede Variable habe 3 Werte $(low, moderate, high)$
  - Zahlen an den Knoten repräsentieren, wie viele Parameter für die CPDs des jeweiligen Knotens bestimmt werden müssen
    - 78 Wahrscheinlichkeitswerte insgesamt



HeartDisease wegzulassen ist gar keine gute Lösung!

- Ohne versteckten Knoten müssen die indirekten Abhängigkeiten durch Kanten direkt abgebildet werden
  - Symptome nicht länger bedingt unabhängig gegeben die Elternknotenwerte
  - Sehr viel mehr Kanten, sehr viel mehr Wahrscheinlichkeiten zu spezifizieren: 708 insgesamt
  - Brauchen sehr viel mehr Daten, um diese ganzen Werte vernünftig zu lernen



Expectation-Maximisation (EM)

- Wenn wir versteckte Variablen beibehalten und die Parameter aus Daten lernen wollen, haben wir eine Form des unüberwachten Lernens (*unsupervised learning*)
  - Daten geben nicht über alle Aspekte von Datenpunkten Auskunft

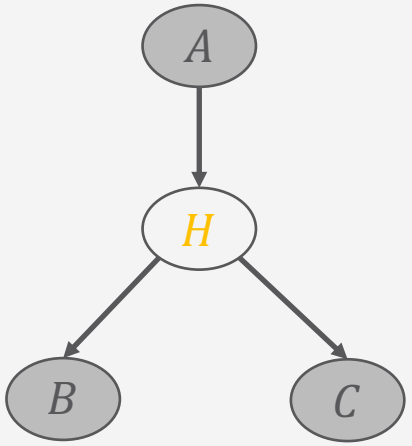
- Expectation-Maximisation (EM)
  - Genereller Algorithmus zum Lernen von Parameters bei unvollständigen Daten
  - Hier für diskrete Datenwerte im BN-Kontext gezeigt

EM: Generelle Idee

- Falls wir Daten für alle Variablen im BN hätten, könnten wir die Parameter mit ML- (oder MAP-) Ansätzen lernen
  - Relative Häufigkeiten bestimmen, wie vorher besprochen

- Falls wir die Parameter hätten, könnten wir die Wahrscheinlichkeiten jedes Ereignisses schätzen
  - $P(H \mid A, B, C)$
  - Könnten also Werte für H schätzen

A	B	C
$true$	$false$	$true$
$false$	$true$	$true$
$true$	$true$	$false$
	$\vdots$	



Dempster, A.P., Laird. N.M., Rubin, D.B.: Maximum-Likelihood from Incomplete Data via the EM Algorithm. In: *Journal of the Royal Statistical Society*, **1977**

EM: Generelle Idee

- Algorithmus startet mit „erfundenen“ (zufällig generierten) Informationen, um das Lernproblem zu lösen
  - Erfundene Parameterwerte (virtuell notiert in den CPDs)

- Dann werden die Initialwerte in zwei Schritten verfeinert
  - Expectation (E):
    Aktualisiere die Daten (virtuell) mit aus dem aktuellen Modell hergeleiteten Erwartungen
  - Maximisation (M):
    Gegeben die aktualisierten Daten, aktualisiere die Parameter mittels ML Ansatz
    - Der gleiche Schritt wie im voll beobachtbaren Fall

EM: Naive Bayes-Modelle als Anwendung

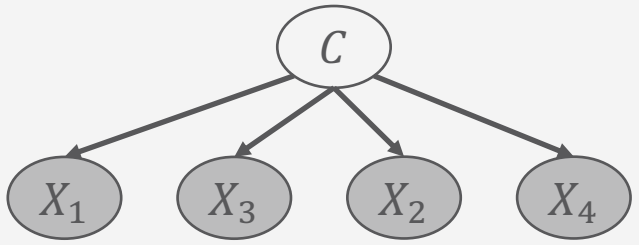
- Betrachte die folgenden Daten
  - N Beispiele mit Booleschen Attributen $X_1, \ldots, X_4$
  - … die wir kategorisieren wollen mit möglichen Werten einer Klasse $C, \mathrm{Val}(C) = \{1,2,3\}$

- Naive Bayes Klassifikator mit versteckter Variable C

Daten

X_1	X_2	X_3	X_4
$true$	$false$	$true$	$true$
$false$	$true$	$true$	$false$
$true$	$false$	$true$	$true$
⋮			

Modell



→ Wahrscheinlichkeitsverteilungen zu lernen

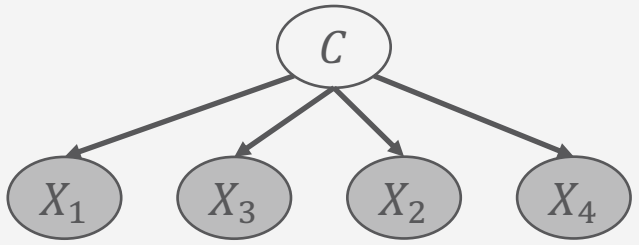
$P(C)$?
$P(X_1 \mid C)$?
$P(X_2 \mid C)$?
$P(X_3 \mid C)$?
$P(X_4 \mid C)$?

X_1	X_2	X_3	X_4	Zähler
$false$	$false$	$false$	$false$	19
$false$	$false$	$false$	$true$	7
$false$	$false$	$true$	$false$	13
⋮				

EM: Initialisierung

- Algorithmus startet mit „erfundenen“ (zufällig generierten) Informationen, um das Lernproblem zu lösen
  - Erfundene Parameterwerte (virtuell notiert in den CPDs)

Modell



→ Wahrscheinlichkeitsverteilungen zu lernen

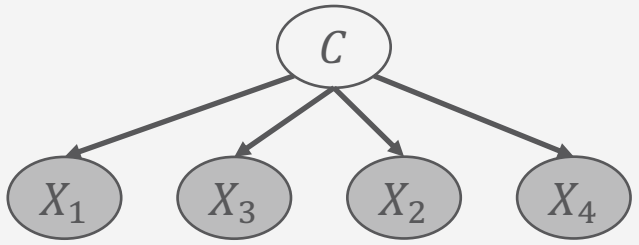
$P(C)$?
$P(X_1 \mid C)$?
$P(X_2 \mid C)$?
$P(X_3 \mid C)$?
$P(X_4 \mid C)$?

Definiere zufällige Parameter

EM: Expectation-Schritt (Bestimmung der erwarteter Zahlen)

- Was benötigen wir, um die Netzwerkparameter mit dem ML-Ansatz zu lernen?
  - $P(C = k) = \frac{Anzahl(Datenpunkte\ mit\ C{=}k)}{Anzahl(alle\ Datenpunkte)}$ für jedes $k = 1,2,3$
  - $P(X_i = x \mid C = k) = \frac{Anzahl(Datenpunkte\ mit\ X_i{=}x\ und\ C{=}k)}{Anzahl(Datenpunkte\ mit\ C{=}k)}$ für jedes $x \in \mathrm{Val}(X_i), i = 1, \dots, 4, k = 1,2,3$
- Bislang haben wir nur: $N = Anzahl(alle\ Datenpunkte)$
  - Wir approximieren alle weiteren Zähler mit erwarteten Zählerwerten, die aus dem Modell mit den „erfundenen“ Parametern abgeleitet werden

Modell



Daten

X_1	X_2	X_3	X_4	Zähler
$false$	$false$	$false$	$false$	19
$false$	$false$	$false$	$true$	7
$false$	$false$	$true$	$false$	13
		⋮		

→ Wahrscheinlichkeitsverteilungen zu lernen

$P(C)$?
$P(X_1 \mid C)$?
$P(X_2 \mid C)$?
$P(X_3 \mid C)$?
$P(X_4 \mid C)$?

EM: Expectation-Schritt (Bestimmung der erwarteter Zahlen)

- Der erwartete Zähler $\widehat{N}(C = k)$ ist die Summe, über alle N Beispieldaten, der Wahrscheinlichkeiten, dass jedes Beispiel in Kategorie k fällt

$$\widehat{N}(C = k) = \sum_{j=1}^{N} P\left(C = k \mid x_1^j, x_2^j, x_3^j, x_4^j\right)$$

Featurewerte des j'ten Beispiels

- Schätzung dann

$$P(C = k) = \frac{erwarteteAnzahl(Datenpunkte\ mit\ C = k)}{Anzahl(alle\ Datenpunkte)} = \frac{\widehat{N}(C = k)}{N}$$

- Wie erhalten wir die Werte?
  - Im Naive Bayes Modell einfach berechnen:

$$P\left(C = k \mid x_1^j, x_2^j, x_3^j, x_4^j\right) = \frac{P\left(C = k, x_1^j, x_2^j, x_3^j, x_4^j\right)}{P\left(x_1^j, x_2^j, x_3^j, x_4^j\right)} = \frac{P\left(x_1^j \mid C = k\right) \dots P\left(x_4^j \mid C = k\right) P(C = k)}{P\left(x_1^j, x_2^j, x_3^j, x_4^j\right)}$$

Erfundene Parameter für das Netzwerk nutzen

Auch über erfundene Parametern errechenbar

EM: Expectation-Schritt (Bestimmung der erwarteter Zahlen)

- Durch einen ähnlichen Schritt erhalten wir die erwarteten Zähler für Beispiele mit Featurewert $X_i = x$ und Kategorie k

Auch alles über erfundene Parametern errechenbar

$$\widehat{N}(X_i = x, C = k) = \sum_{j=1}^{N} P\left(C = k \mid x_i^j = x, x_{i'}^j, x_{i''}^j, x_{i'''}^j\right), i, i', i'', i''' \in \{1,2,3,4\}, i \neq i' \neq i'' \neq i'''$$

  - Zum Beispiel:

$$\widehat{N}(X_1 = true, C = 1) = \sum_{\boldsymbol{e}_j = (x_1, x_2, x_3, x_4), x_1 = true} P\left(C = 1 \mid x_1^j = true, x_2^j, x_3^j, x_4^j\right)$$

- Schätzung von $P(X_i \mid C)$

$$P(X_i = x \mid C = k) = \frac{erwarteteAnzahl(Datenpunkte\ mit\ X_i = x, C = k)}{erwarteteAnzahl(Datenpunkte\ mit\ C = k)} = \frac{\widehat{N}(X_i = x, C = k)}{\widehat{N}(C = k)}$$

  - für jedes $x \in \mathrm{Val}(X_i), i = 1, \ldots, 4, k = 1,2,3$

EM: Generelle Idee

- Algorithmus startet mit „erfundenen“ (zufällig generierten) Informationen, um das Lernproblem zu lösen
  - Erfundene Parameterwerte (virtuell notiert in den CPDs)

- Dann werden die Initialwerte in zwei Schritten verfeinert
  - Expectation (E):
    Aktualisiere die Daten (virtuell) mit aus dem aktuellen Modell hergeleiteten Erwartungen

- Maximisation (M):
  Gegeben die aktualisierten Daten, aktualisiere die Parameter mittels ML Ansatz
  - Der gleiche Schritt wie im voll beobachtbaren Fall

Maximisation-Schritt: (Verfeinerung der Parameter)

- Nun verfeinern wir die Parameter mittels ML-Lernen auf den erwarteten Zählern

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

$$P(X_i = x \mid C = k) = \frac{\hat{N}(X_i = x, C = k)}{\hat{N}(C = k)}$$

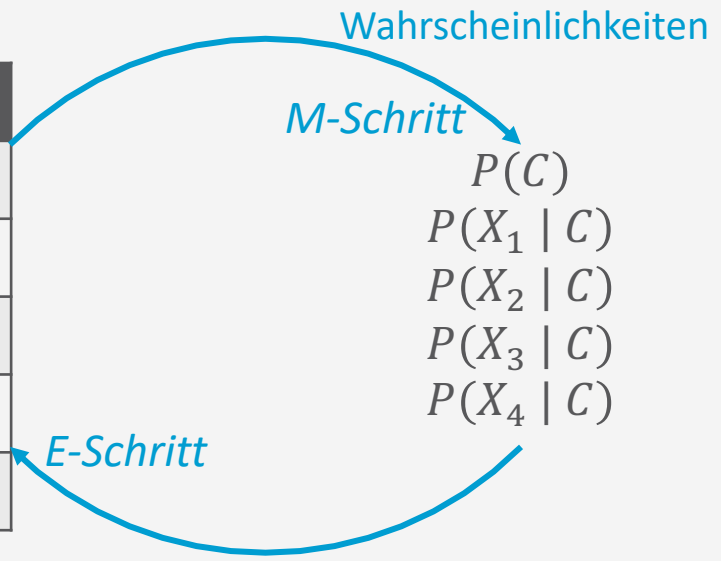
  - für jedes $x \in \mathrm{Val}(X_i), i = 1, \ldots, 4, k = 1,2,3$

EM-Zyklus

- Nun kann der E-Schritt wiederholt werden
  - Neue erwartete Zähler berechnen gegeben der gerade aktualisierten Parameter
- Dann kann der M-Schritt wiederholt werden
  - Parameter aktualisieren gegebene der gerade neu berechneten erwarteten Zähler

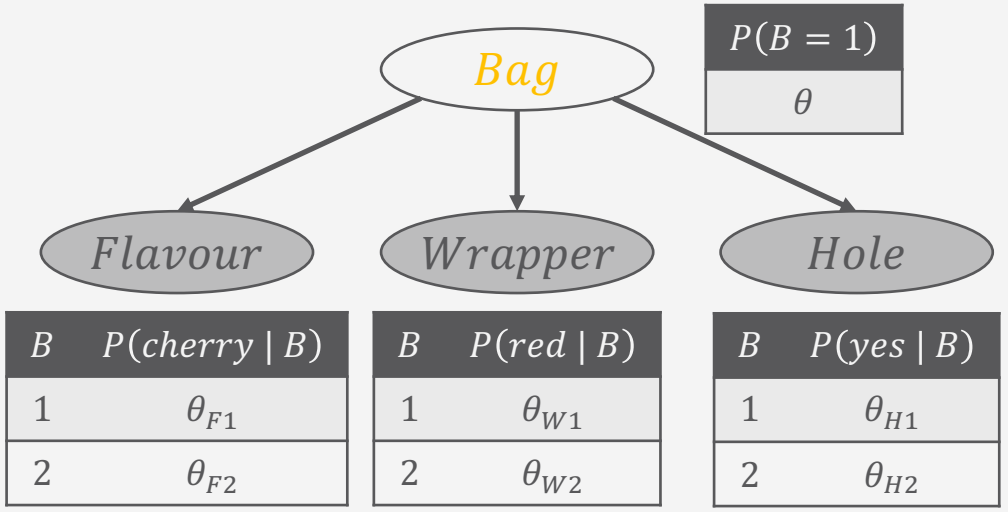
Erwartete Zähler (*augmented data*)

X_1	X_2	X_3	X_4	C	Zähler
⋮	⋮	⋮	⋮	⋮	⋮
$true$	$false$	$true$	$true$	1	0.4
$true$	$false$	$true$	$true$	2	0.1
$true$	$false$	$true$	$true$	3	0.5
⋮	⋮	⋮	⋮	⋮	⋮



Beispiel: Bonbonfabrik wieder einmal

- Zwei Bonbontüten (1 und 2) wurden vermischt
- Bonbons werden durch Eigenschaften beschrieben
  - $Flavour$
  - $Wrapper$
  - $Hole$ (ob ein Loch in der Mitte ist)
  - Merkmale von Bonbon hängen mit bestimmten Wahrscheinlichkeiten von der Tüte ab, aus der sie kommen
- Wir wollen für jedes Bonbon vorhersagen, aus welcher Tüte es kam, je nach vorgefundenen Eigenschaft: Naive Bayes Klassifikator



B	$P(cherry \mid B)$
1	θ_{F1}
2	θ_{F2}

B	$P(red \mid B)$
1	θ_{W1}
2	θ_{W2}

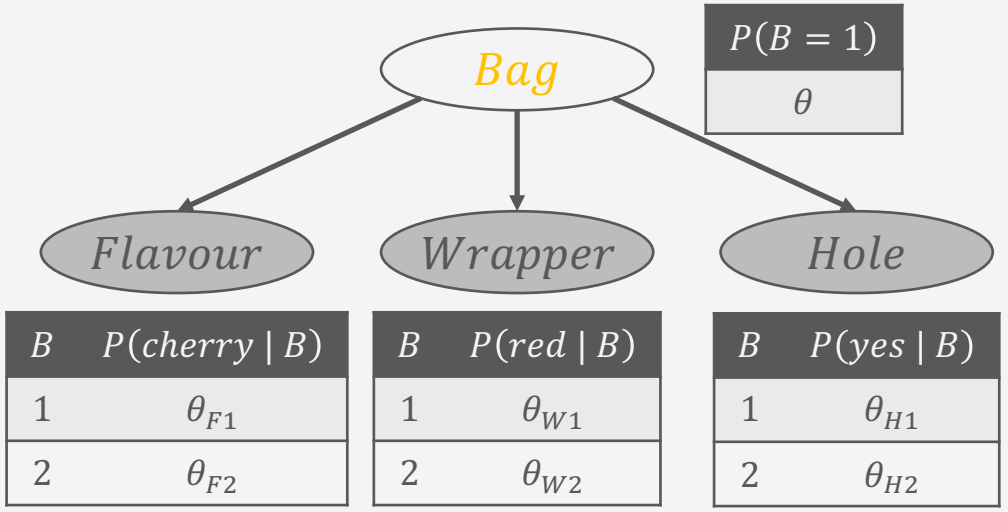
B	$P(yes \mid B)$
1	θ_{H1}
2	θ_{H2}

Daten

- Nehmen wir an, die wahren Parameter seien
  - $\theta = 0.5$
  - $\theta_{F1} = \theta_{W1} = \theta_{H1} = 0.8$
  - $\theta_{F2} = \theta_{W2} = \theta_{H2} = 0.3$
- Annahme: Die folgenden Zähler werden „gesampelt“ aus $P(B, F, W, H)$, $N = 1000$

Datensatz	$W = red$		$W = yellow$	
	$H = yes$	$H = no$	$H = yes$	$H = no$
$F = cherry$	273	93	104	90
$F = lime$	79	100	94	167

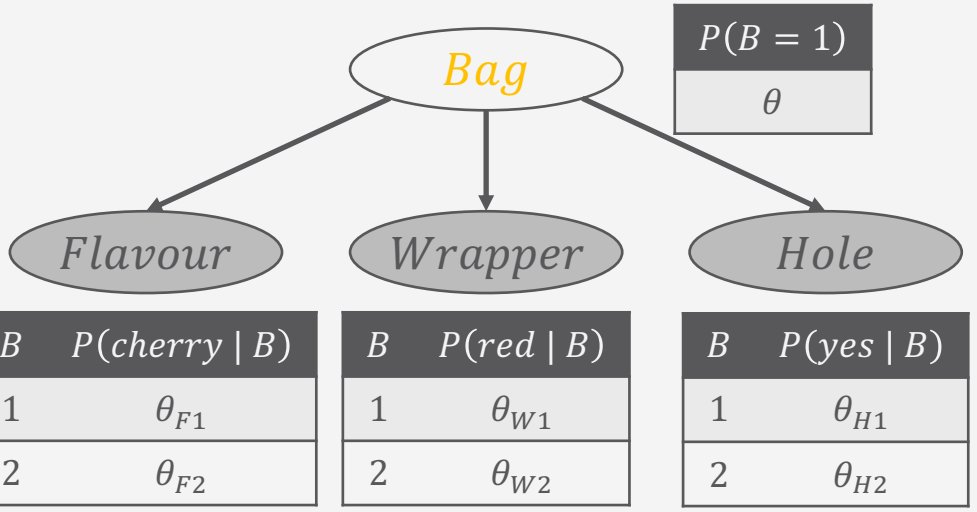
- Wir wollen nun die wahren Parameter aus diesen Daten mittels EM rekonstruieren



EM: Initialisierung

- Weise Parametern zufällige Initialwerte zu
  - Zur Vereinfachung der Darstellung verwenden wir:
  - $\theta^{(0)} = 0.6$
  - $\theta_{F1}^{(0)} = \theta_{W1}^{(0)} = \theta_{H1}^{(0)} = 0.6$
  - $\theta_{F2}^{(0)} = \theta_{W2}^{(0)} = \theta_{H2}^{(0)} = 0.4$
- Nun durchlaufen wir einen EM-Zyklus zur Berechnung von $\theta^{(1)}$
  - Beispielhaft werden wir hier uns zuerst E- und M-Schritt für $\theta^{(1)}$ anschauen, dann für $\theta_{F1}^{(1)}$
  - Konzeptionell würde man aber eher erst E für alle und dann M für alle machen

Datensatz	$W = red$		$W = yellow$	
	$H = yes$	$H = no$	$H = yes$	$H = no$
$F = cherry$	273	93	104	90
$F = lime$	79	100	94	167

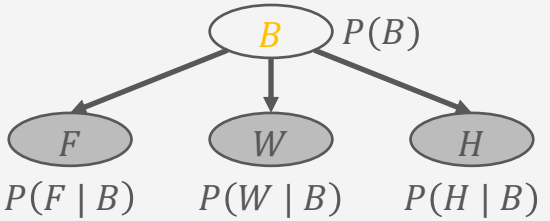


B	$P(cherry \mid B)$
1	θ_{F1}
2	θ_{F2}

B	$P(red \mid B)$
1	θ_{W1}
2	θ_{W2}

B	$P(yes \mid B)$
1	θ_{H1}
2	θ_{H2}

E-Schritt



- Zuerst bauen wir die erwarteten Zähler für Bonbons von Tüte 1:
  - Addiere die Wahrscheinlichkeiten, dass jeder der N Datenpunkte aus Tüte 1 kommt
  - Seien f_j, w_j, h_j die Werte der entsprechenden Variablen für den j'ten Datenpunkt

$$\widehat{N}(Bag = 1) = \sum_{j=1}^{N} P\big(Bag = 1 \mid f_j, w_j, h_j\big)$$

$$= \sum_{j=1}^{N} \frac{P\big(f_j, w_j, h_j \mid Bag = 1\big)P(Bag = 1)}{P\big(f_j, w_j, h_j\big)}$$

$$= \sum_{j=1}^{N} \frac{P\big(f_j \mid Bag = 1\big)P\big(w_j \mid Bag = 1\big)P\big(h_j \mid Bag = 1\big)P(Bag = 1)}{\sum_{k=1}^{2} P\big(f_j \mid Bag = k\big)P\big(w_j \mid Bag = k\big)P\big(h_j \mid Bag = k\big)P(Bag = k)}$$

E-Schritt

$$\widehat{N}(Bag = 1) = \sum_{j=1}^{N} \frac{P(f_j \mid Bag = 1)P(w_j \mid Bag = 1)P(h_j \mid Bag = 1)P(Bag = 1)}{\sum_{k=1}^{2} P(f_j \mid Bag = k)P(w_j \mid Bag = k)P(h_j \mid Bag = k)P(Bag = k)}$$

- Die Summation kann in die 8 Bonbongruppen aus der Tabelle aufgebrochen werden
  - Zum Beispiel ergibt die Summe über 273 Kirsch-Bonbons mit rotem Papier und einem Loch (erster Eintrag in der Tabelle)

$$273 \frac{\theta_{F1}^{(0)}\theta_{W1}^{(0)}\theta_{H1}^{(0)}\theta^{(0)}}{\theta_{F1}^{(0)}\theta_{W1}^{(0)}\theta_{H1}^{(0)}\theta^{(0)} + \theta_{F1}^{(0)}\theta_{W1}^{(0)}\theta_{H1}^{(0)}(1-\theta^{(0)})}$$

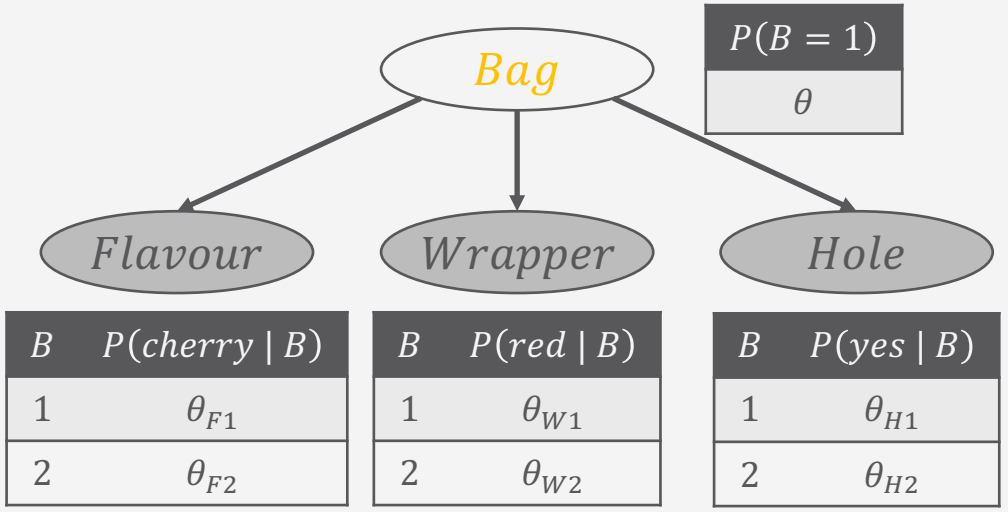
$$= 273 \frac{0.6^4}{0.6^4 + 0.4^4} = 273 \frac{0.1296}{0.1552} = 227.97$$

$$\theta^{(0)} = 0.6$$

$$\theta_{F1}^{(0)} = \theta_{W1}^{(0)} = \theta_{H1}^{(0)} = 0.6$$

$$\theta_{F2}^{(0)} = \theta_{W2}^{(0)} = \theta_{H2}^{(0)} = 0.4$$

Datensatz	$W = red$		$W = yellow$	
	$H = yes$	$H = no$	$H = yes$	$H = no$
$F = cherry$	273	93	104	90
$F = lime$	79	100	94	167



B	$P(cherry \mid B)$
1	θ_{F1}
2	θ_{F2}

B	$P(red \mid B)$
1	θ_{W1}
2	θ_{W2}

B	$P(yes \mid B)$
1	θ_{H1}
2	θ_{H2}

M-Schritt

- Mit der Berechnung der anderen 7 Bonbongruppen erhalten wir

$$\widehat{N}(Bag = 1) = 612.4$$

- Nun führen wir den M-Schritt aus, um θ zu verfeinern
- Hierzu nehmen wir den erwarteten Zähler der Datenpunkte aus Tüte 1 und teilen durch N

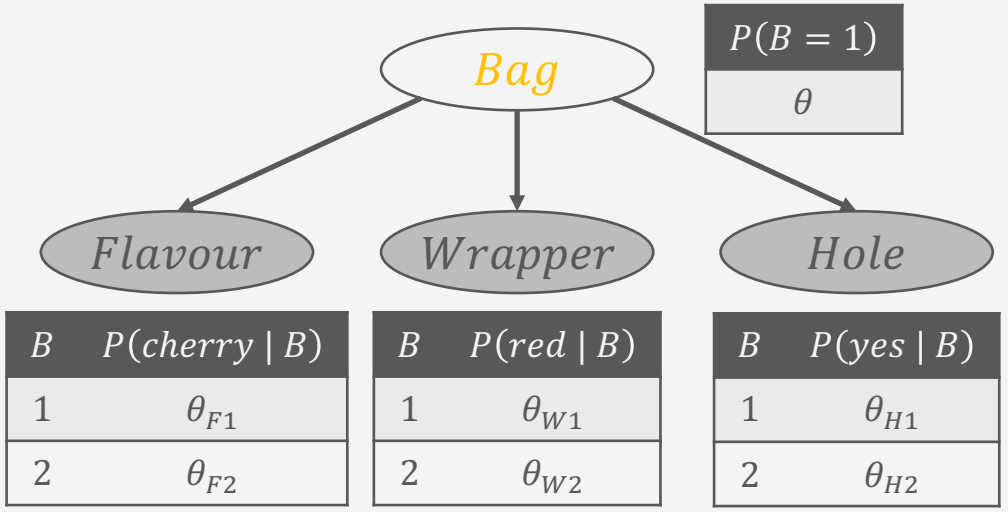
$$\theta^{(1)} = \frac{\widehat{N}(Bag = 1)}{N} = \frac{612.4}{1000} = 0.6124$$

$$\theta^{(0)} = 0.6$$

$$\theta_{F1}^{(0)} = \theta_{W1}^{(0)} = \theta_{H1}^{(0)} = 0.6$$

$$\theta_{F2}^{(0)} = \theta_{W2}^{(0)} = \theta_{H2}^{(0)} = 0.4$$

Datensatz	$W = red$		$W = yellow$	
	$H = yes$	$H = no$	$H = yes$	$H = no$
$F = cherry$	273	93	104	90
$F = lime$	79	100	94	167



B	$P(cherry \mid B)$
1	θ_{F1}
2	θ_{F2}

B	$P(red \mid B)$
1	θ_{W1}
2	θ_{W2}

B	$P(yes \mid B)$
1	θ_{H1}
2	θ_{H2}

Noch ein Parameter: θ_{F1}

$$\theta^{(0)} = 0.6$$
$$\theta_{F1}^{(0)} = \theta_{W1}^{(0)} = \theta_{H1}^{(0)} = 0.6$$
$$\theta_{F2}^{(0)} = \theta_{W2}^{(0)} = \theta_{H2}^{(0)} = 0.4$$

Datensatz	$W = red$		$W = yellow$	
	$H = yes$	$H = no$	$H = yes$	$H = no$
$F = cherry$	273	93	104	90
$F = lime$	79	100	94	167

- E-Schritt: Bestimme erwarteten Zählerwert von Kirsch-Bonbons aus Tüte 1 basierend auf $\boldsymbol{\theta}^{(0)}$

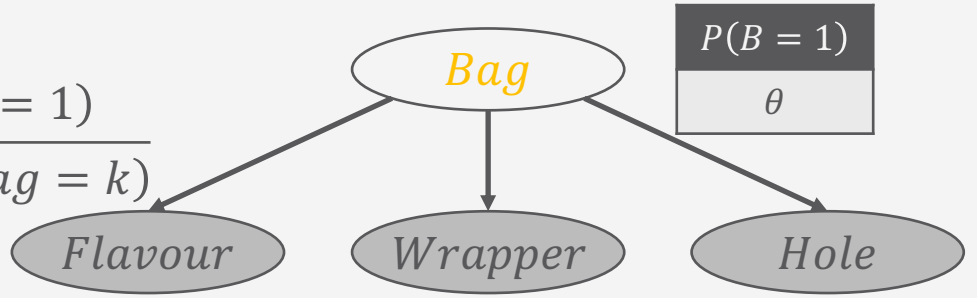
$$\hat{N}(B = 1, F = cherry) = \sum_{j:F=cherry} P(B = 1 \mid cherry_j, w_j, h_j)$$

  - Bestimmbar wie vorher besprochen

$$\hat{N}(Bag = 1, F = cherry)$$
$$= \sum_{j=1}^{N} \frac{P(cherry_j \mid Bag = 1)P(w_j \mid Bag = 1)P(h_j \mid Bag = 1)P(Bag = 1)}{\sum_{k=1}^{2} P(cherry_j \mid Bag = k)P(w_j \mid Bag = k)P(h_j \mid Bag = k)P(Bag = k)}$$

- M-Schritt: Verfeinere θ_{F1} durch Bestimmung der relativen Häufigkeiten

$$\theta_{F1}^{(1)} = \frac{\hat{N}(B = 1, F = cherry)}{\hat{N}(B = 1)}$$



B	$P(cherry \mid B)$
1	θ_{F1}
2	θ_{F2}

B	$P(red \mid B)$
1	θ_{W1}
2	θ_{W2}

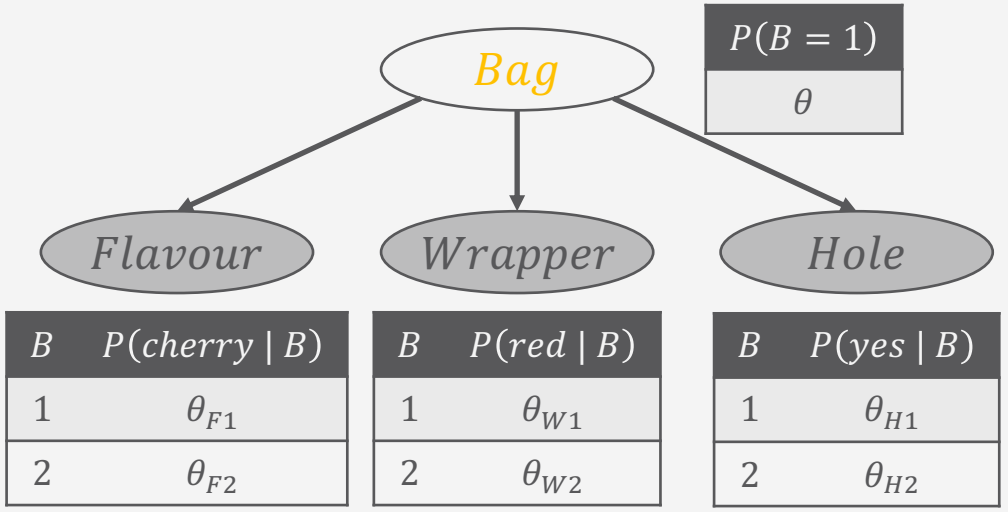
B	$P(yes \mid B)$
1	θ_{H1}
2	θ_{H2}

Lernergebnis

- Nach einem vollen Zyklus über alle Parameter
  - $\theta^{(1)} = 0.6124$
  - $\theta_{F1}^{(1)} = 0.6684, \theta_{W1}^{(1)} = 0.6483, \theta_{H1}^{(1)} = 0.658$
  - $\theta_{F2}^{(1)} = 0.3887, \theta_{W2}^{(1)} = 0.3817, \theta_{H2}^{(1)} = 0.3827$
- Für jeden Satz von Parametern Log-Wahrscheinlichkeiten bestimmbar, wie vorher

$$\theta^{(0)} = 0.6$$
$$\theta_{F1}^{(0)} = \theta_{W1}^{(0)} = \theta_{H1}^{(0)} = 0.6$$
$$\theta_{F2}^{(0)} = \theta_{W2}^{(0)} = \theta_{H2}^{(0)} = 0.4$$

Datensatz	$W = red$		$W = yellow$	
	$H = yes$	$H = no$	$H = yes$	$H = no$
$F = cherry$	273	93	104	90
$F = lime$	79	100	94	167



B	$P(cherry \mid B)$
1	θ_{F1}
2	θ_{F2}

B	$P(red \mid B)$
1	θ_{W1}
2	θ_{W2}

B	$P(yes \mid B)$
1	θ_{H1}
2	θ_{H2}

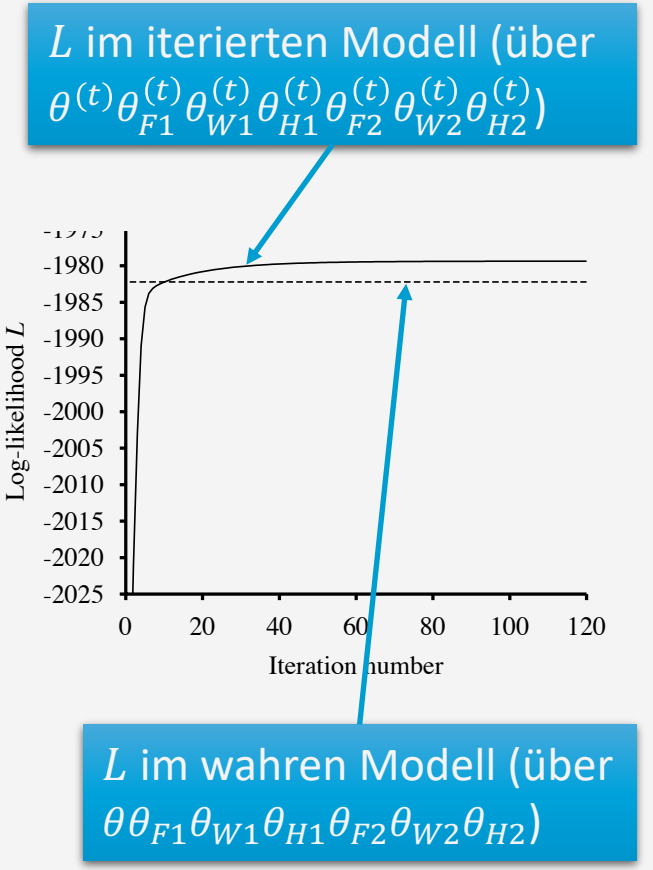
Lernergebnis

- Log-Wahrscheinlichkeiten der Parameter:

$$P\left(\boldsymbol{d} \mid h_{\theta^{(t)}\theta_{F1}^{(t)}\theta_{W1}^{(t)}\theta_{H1}^{(t)}\theta_{F2}^{(t)}\theta_{W2}^{(t)}\theta_{H2}^{(t)}}\right) = \prod_{j=1}^{N} P\left(d_j \mid h_{\theta^{(t)}\theta_{F1}^{(t)}\theta_{W1}^{(t)}\theta_{H1}^{(t)}\theta_{F2}^{(t)}\theta_{W2}^{(t)}\theta_{H2}^{(t)}}\right)$$

$$\Rightarrow \log P(\boldsymbol{d}, h_{\boldsymbol{\theta}}) = \sum_{j=1}^{N} \log P\left(d_j \mid h_{\theta^{(t)}\theta_{F1}^{(t)}\theta_{W1}^{(t)}\theta_{H1}^{(t)}\theta_{F2}^{(t)}\theta_{W2}^{(t)}\theta_{H2}^{(t)}}\right)$$

  - Man kann zeigen, dass dieser Wert mit jeder EM-Iteration steigt (Konvergenz)
    - Beispiel Bonbonfabrik: Log-Wahrscheinlichkeit $L = \log P(d, h_{\boldsymbol{\theta}})$ steigt in jeder Iteration an [Abbildung rechts]
  - EM mit ML bleibt aber bei lokalen Maxima hängen
    - Mögliche Abhilfen: mit unterschiedlichen Startwerten anfangen, MAP Lernen nehmen



EM: Diskussion

- Für komplexere BNs ist der Algorithmus der gleich
  - Im Allgemeinen müssen wir die CPD-Einträge für jede Variable R_i gegeben die Elternknotenwerte $\mathrm{Pa}(R_i)$ berechnen

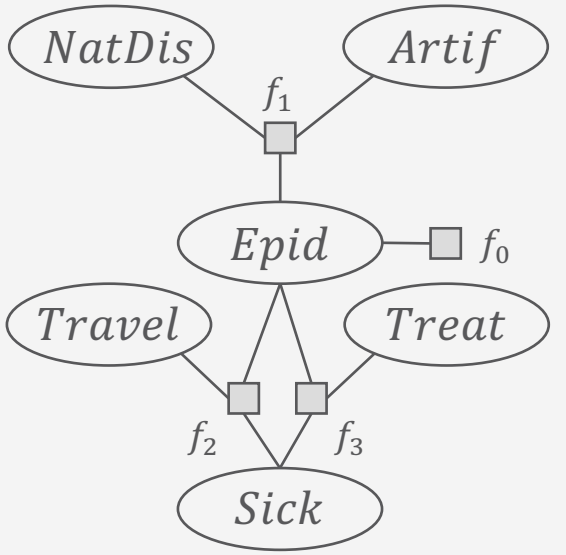
$$\theta_{ijk} = P\left(R_i = r_{ij} \mid \mathrm{Pa}(R_i) = \mathrm{pa}_k\right)$$

$$\theta_{ijk} = \frac{\hat{N}\left(R_i = r_{ij}, \mathrm{Pa}(R_i) = \mathrm{pa}_k\right)}{\hat{N}(\mathrm{Pa}(R_i) = \mathrm{pa}_k)}$$

  - Die erwarteten Zählerwerte werden durch Summierung über die Beispiele berechnet, nachdem alle notwendigen $P\left(R_i = r_{ij}, \mathrm{Pa}(R_i) = \mathrm{pa}_{ik}\right)$ mittels BN-Algorithmen bestimmt sind
- Das Verfahren kann in eine kombinatorische Explosion laufen
  - Ggf. Approximationstechniken anwenden

Zwischenzusammenfassung

- ML-basierte Parameterschätzung in BNs bei vollständigen Daten
  - Maximierung der Log-Wahrscheinlichkeit der Daten
  - Zerfällt in Abschätzung der einzelnen Parameter
  - Vorkommen der Belegungen in den Daten zählen
- ML-basierte Parameterschätzung in BNs bei nicht vollständigen Daten: EM
  - Maximierung der Log-Wahrscheinlichkeit der Daten
  - Iterativer Algorithmus
    - Erwartete Zähler berechnen
    - Parameter optimieren
  - Läuft auf Fixpunkt (lokales Optimum)



Ungerichtete Modelle

ML-basiertes Parameter lernen

Ungerichtete Modelle

- BNs haben den Vorteil einer Normalisierungskonstante von $Z = 1$ zu haben
  - Parameter lernen reduziert sich auf Schätzung der Parameter der lokalen CPTs
  - Ungerichtete Modelle haben diesen Vorteil im Allgemeinen nicht:

$$P_F = \frac{1}{Z}\prod_{i=1}^{n}\phi_i(R_1, \dots, R_k)$$

$$Z = \sum_{r_1 \in \mathrm{Val}(R_1)} \dots \sum_{r_n \in \mathrm{Val}(R_n)} \prod_{i=1}^{k}\phi_i(r_1, \dots, r_k)$$

  - Z verbindet alle Variablen im Modell in einer Funktion
  - $Z \neq 1$ in den meisten Fällen

ML Vorgehen für ungerichtete Modelle

- Gegeben ein Faktormodel $F = \{f_i\}_{i=1}^{n}$ über Zufallsvariablen $\boldsymbol{R} = \mathrm{rv}(F)$
- Lass $\boldsymbol{r}_\alpha$ die Projektion von $\boldsymbol{r}$ auf die Zufallsvariablen einer Entität α bezeichnen
  - Beispiel:
    - $\boldsymbol{r}_f = \pi_{\mathrm{rv}(f)}(\boldsymbol{r})$: wählt aus $\boldsymbol{r}$ die Werte aus, die sich auf die Zufallsvariablen in f beziehen
- Lass ϕ_f sich auf die Potentialfunktion von f beziehen

- Gegeben ein Datensatz $\boldsymbol{d}$ mit N zusammengesetzten Ereignissen für $\boldsymbol{R}$
  - I.e., vollständige Daten
  - Lass $\#(\boldsymbol{r})$ bezeichnen, wie häufig $\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})$ in $\boldsymbol{d}$ vorkommt
  - $\boldsymbol{d}$ als Multi-Menge: $\boldsymbol{d} = \{(\boldsymbol{r}, \#(\boldsymbol{r}))\}_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})}$
    - $N = \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} \#(\boldsymbol{r})$

ML Vorgehen für ungerichtete Modelle

1. Drücke die Wahrscheinlichkeit (Likelihood) $P(\boldsymbol{d}|\boldsymbol{\theta})$ der Daten $\boldsymbol{d}$ als Funktion der Parameter $\boldsymbol{\theta}$ aus, die zu lernen sind
   - $\boldsymbol{\theta}$ bezieht sich hier auf die Potentiale in den Faktoren von F
     - $\boldsymbol{d} = \{\boldsymbol{r}_j\}_{j=1}^{N} = \{(\boldsymbol{r}, \#(\boldsymbol{r}))\}_{\boldsymbol{r}\in \mathrm{Val}(\boldsymbol{R})}$

$$P(\boldsymbol{d}|\boldsymbol{\theta}) = \prod_{j=1}^{N} P(d_j|\boldsymbol{\theta}) = \prod_{\boldsymbol{r}\in \mathrm{Val}(\boldsymbol{R})} P(\boldsymbol{r}|\boldsymbol{\theta})^{\#(\boldsymbol{r})}$$

   - In der Zählerdarstellung ($\#(\boldsymbol{r})$) kann $P(\boldsymbol{d}|\boldsymbol{\theta})$ null werden, wenn ein $\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})$ nicht beobachtet wurde
     - Abhilfe: initialisiere alle Zähler mit 1

   und nehme den Logarithmus

$$\log P(\boldsymbol{d}|\boldsymbol{\theta}) = \log \prod_{\boldsymbol{r}\in \mathrm{Val}(R)} P(\boldsymbol{r}|\boldsymbol{\theta})^{\#(\boldsymbol{r})}$$

$\phi_f(\boldsymbol{r}_f)$ ist ein Parameter $\in \boldsymbol{\theta}$

$$\log P(D|\theta) = \log \prod_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} P(\boldsymbol{r}|\theta)^{\#(\boldsymbol{r})}$$

$$= \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} \#(\boldsymbol{r}) \log P(\boldsymbol{r}|\theta) = \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} \#(\boldsymbol{r}) \log \left(\frac{1}{Z} \prod_{f \in F} \phi_f(\boldsymbol{r}_f) \right)$$

$$= \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} \#(\boldsymbol{r}) \left(\log \left(\frac{1}{Z} \right) + \log \prod_{f \in F} \phi_f(\boldsymbol{r}_f) \right)$$

$$= \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} \#(\boldsymbol{r}) \left(-\log(Z) + \sum_{f \in F} \log \phi_f(\boldsymbol{r}_f) \right)$$

$$= \left(\sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} \#(\boldsymbol{r}) \sum_{f \in F} \log \phi_f(\boldsymbol{r}_f) \right) - \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} \#(\boldsymbol{r}) \log(Z)$$

$$= \left(\sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} \#(\boldsymbol{r}) \sum_{f \in F} \log \phi_f(\boldsymbol{r}_f) \right) - N \log(Z)$$

$$= \left(\sum_{f \in F} \sum_{\boldsymbol{r}_f \in \mathrm{Val}(\mathrm{rv}(f))} \#(\boldsymbol{r}_f) \log \phi_f(\boldsymbol{r}_f) \right) - N \log(Z)$$

ML Vorgehen für ungerichtete Modelle

2. Leite die Log-Wahrscheinlichkeiten nach jedem Parameter ab, i.e., $\phi_f(\boldsymbol{r}_f)$

$$\log P(\boldsymbol{d}|\boldsymbol{\theta}) = \left(\sum_{f \in F} \sum_{\boldsymbol{r}_f \in \mathrm{Val}(\mathrm{rv}(f))} \underbrace{\#(\boldsymbol{r}_f) \log \phi_f(\boldsymbol{r}_f)}_{l_1}\right) - \underbrace{N \log(Z)}_{l_2}$$

- Ableitung von l_1

$$\frac{\partial l_1}{\partial \phi_f(\boldsymbol{r}_f)} = \#(\boldsymbol{r}_f) \cdot \frac{1}{\phi_f(\boldsymbol{r}_f)}$$

konstant bzgl. $\phi_f(\boldsymbol{r}_f)$

Ableiten von log:

$$\frac{\partial \log f(x)}{\partial x} = \frac{1}{f(x)} \cdot \frac{\partial f(x)}{\partial x}$$

Z.B., wenn $f(x) = x$:

$$\frac{\partial \log x}{\partial x} = \frac{1}{x} \cdot \frac{\partial x}{\partial x} = \frac{1}{x} \cdot 1 = \frac{1}{x}$$

- Ableitung von $l_2 = N \log(Z)$

$$\frac{\partial l_2}{\partial \phi_f(\boldsymbol{r}_f)} = N \cdot \frac{1}{Z} \cdot \frac{\partial Z}{\partial \phi_f(\boldsymbol{r}_f)} = N \cdot \frac{1}{Z} \cdot \underbrace{\frac{\partial}{\partial \phi_f(\boldsymbol{r}_f)} \cdot \sum_{\boldsymbol{s} \in \mathrm{Val}(\mathrm{rv}(F))} \prod_{e \in F} \phi_e(\boldsymbol{s}_e)}$$

Nur Summanden, die $\phi_f(\boldsymbol{r}_f)$ enthalten, bleiben; Ableitung der anderen ergibt 0; Indikatorfunktion stellt das dar:

$$\mathbf{1}(\boldsymbol{r}, \boldsymbol{s}) = \begin{cases} 1 & \text{if } \boldsymbol{r} = \boldsymbol{s} \\ 0 & \text{otherwise} \end{cases}$$

$$= N \cdot \frac{1}{Z} \cdot \sum_{\boldsymbol{s} \in \mathrm{Val}(\mathrm{rv}(F))} \mathbf{1}(\boldsymbol{r}_f, \boldsymbol{s}_f) \underbrace{\frac{\partial}{\partial \phi_f(\boldsymbol{r}_f)} \cdot \prod_{e \in F} \phi_e(\boldsymbol{s}_e)}$$

Eins der e's ist f, i.e., $\phi_f(\boldsymbol{r}_f) = \phi_e(\boldsymbol{s}_e)$ für ein e; der Rest ist konstant bzgl. $\phi_f(\boldsymbol{r}_f)$

$$= \frac{\partial}{\partial \phi_f(\boldsymbol{r}_f)} \cdot \phi_f(\boldsymbol{s}_f) \cdot \prod_{\substack{e \in F \\ e \neq f}} \phi_e(\boldsymbol{s}_e) = \prod_{\substack{e \in F \\ e \neq f}} \phi_e(\boldsymbol{s}_e)$$

$$= \frac{\phi_f(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)} \cdot \prod_{\substack{e \in F \\ e \neq f}} \phi_e(\boldsymbol{s}_e) = \frac{1}{\phi_f(\boldsymbol{r}_f)} \cdot \prod_{e \in F} \phi_e(\boldsymbol{s}_e)$$

- Ableitung von $l_2 = k \log(Z)$

$$\frac{\partial l_2}{\partial \phi_f(\boldsymbol{r}_f)} = N \cdot \frac{1}{Z} \cdot \frac{\partial Z}{\partial \phi_f(\boldsymbol{r}_f)} = N \cdot \frac{1}{Z} \cdot \frac{\partial}{\partial \phi_f(\boldsymbol{r}_f)} \cdot \sum_{\boldsymbol{s} \in \mathrm{Val}(\mathrm{rv}(F))} \prod_{e \in F} \phi_e(\boldsymbol{s}_e)$$

$$= N \cdot \frac{1}{Z} \cdot \sum_{\boldsymbol{s} \in \mathrm{Val}(\mathrm{rv}(F))} \mathbf{1}(\boldsymbol{r}_f, \boldsymbol{s}_f) \frac{\partial}{\partial \phi_f(\boldsymbol{r}_f)} \cdot \prod_{e \in F} \phi_e(\boldsymbol{s}_e)$$

$$= N \cdot \frac{1}{Z} \cdot \sum_{\boldsymbol{s} \in \mathrm{Val}(\mathrm{rv}(F))} \mathbf{1}(\boldsymbol{r}_f, \boldsymbol{s}_f) \frac{1}{\phi_f(\boldsymbol{r}_f)} \cdot \prod_{e \in F} \phi_e(\boldsymbol{s}_e)$$

$$= N \cdot \frac{1}{\phi_f(\boldsymbol{r}_f)} \cdot \sum_{\boldsymbol{s} \in \mathrm{Val}(\mathrm{rv}(F))} \mathbf{1}(\boldsymbol{r}_f, \boldsymbol{s}_f) \cdot \frac{1}{Z} \cdot \prod_{e \in F} \phi_e(\boldsymbol{s}_e)$$

$$= N \cdot \frac{1}{\phi_f(\boldsymbol{r}_f)} \cdot \sum_{\boldsymbol{s} \in \mathrm{Val}(\mathrm{rv}(F))} \mathbf{1}(\boldsymbol{r}_f, \boldsymbol{s}_f) \cdot P(\boldsymbol{s})$$

$$= N \cdot \frac{1}{\phi_f(\boldsymbol{r}_f)} \cdot P(\boldsymbol{r}_f)$$

ML Vorgehen für ungerichtete Modelle

2. Leite die Log-Wahrscheinlichkeiten nach jedem Parameter ab, i.e., $\phi_f(\boldsymbol{r}_f)$

$$\log P(\boldsymbol{d}|\boldsymbol{\theta}) = \left(\sum_{f \in F} \sum_{\boldsymbol{r}_f \in \mathrm{Val}(\mathrm{rv}(f))} \underbrace{\#(\boldsymbol{r}_f) \log \phi_f(\boldsymbol{r}_f)}_{l_1}\right) - \underbrace{N \log(Z)}_{l_2}$$

- Ableitung für jeden Parameter $\phi_f(\boldsymbol{r}_f)$

$$\frac{\partial l_1}{\partial \phi_f(\boldsymbol{r}_f)} = \frac{\#(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)}$$

$$\frac{\partial l_2}{\partial \phi_f(\boldsymbol{r}_f)} = N \cdot \frac{P(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)}$$

$$\frac{\partial \log P(\boldsymbol{d}|\boldsymbol{\theta})}{\partial \phi_f(\boldsymbol{r}_f)} = \frac{\#(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)} - N \cdot \frac{P(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)}$$

ML Vorgehen für ungerichtete Modelle

3. Ermittle die Parameterwerte, für die die Ableitungen null ist

$$\frac{\partial \log P(\boldsymbol{d}|\boldsymbol{\theta})}{\partial \phi_f(\boldsymbol{r}_f)} = \frac{\#(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)} - N \cdot \frac{P(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)} = 0 \quad \rightarrow \quad \frac{\#(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)} = N \cdot \frac{P(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)}$$

$$\#(\boldsymbol{r}_f) = N \cdot P(\boldsymbol{r}_f)$$

$$\frac{\#(\boldsymbol{r}_f)}{N} = P(\boldsymbol{r}_f)$$

- Gibt an, dass die ML Schätzungen so sein sollten, dass die Modellmarginalverteilungen $P(\boldsymbol{r}_f)$ gleich den normalisierten empirischen Zählern sind:

$$P^{\#}(\boldsymbol{r}_f) \overset{\text{def}}{=} \frac{\#(\boldsymbol{r}_f)}{N} \overset{!}{=} P(\boldsymbol{r}_f)$$

- Gibt nicht an, wie man die Schätzungen erhält

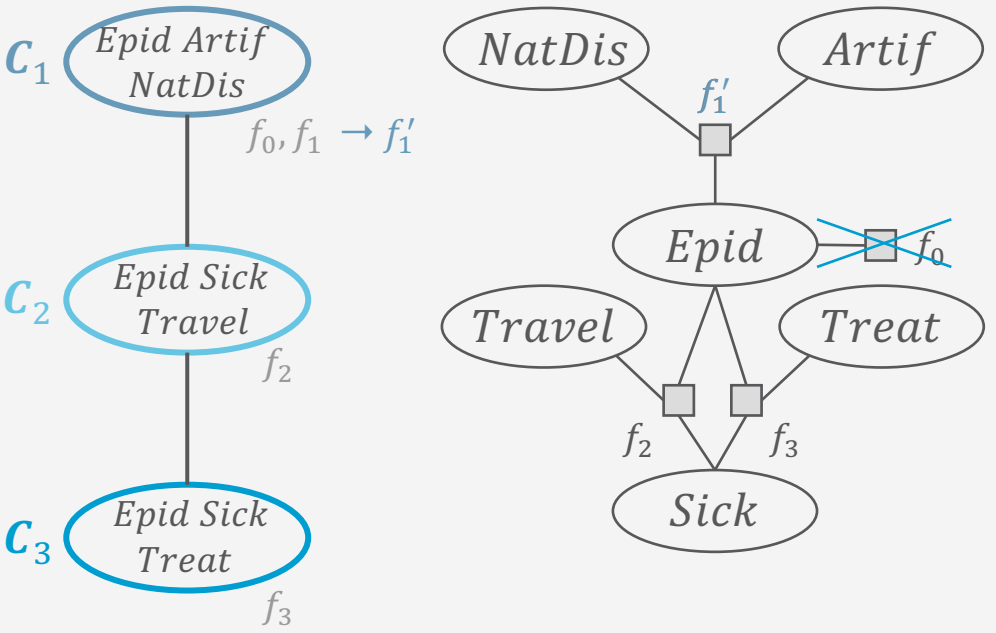
Lernen von Faktoren über maximale Cliquen

- Gegeben ein Faktormodell mit Faktoren über maximale Cliquen
  - Im entsprechenden Jtree (V, E) haben die lokalen Modelle einen Faktor f_C
- Vollständige gemeinsame Verteilung P_F:

$$P_F = \frac{1}{Z}\prod_{f\in F} f = \frac{1}{Z}\prod_{C\in V} f_C = \frac{\prod_{C\in V} P(\boldsymbol{C})}{\prod_{\{i,j\}\in E} P(\boldsymbol{S}_{ij})}$$

  - Für eine Welt $\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})$,

$$P(\boldsymbol{r}) = \frac{\prod_{C\in V} P(\boldsymbol{r}_C)}{\prod_{\{i,j\}\in E} P\left(\boldsymbol{r}_{S_{ij}}\right)} = \frac{\prod_{f\in F} P(\boldsymbol{r}_f)}{\prod_{\{i,j\}\in E} P\left(\boldsymbol{r}_{S_{ij}}\right)}$$



Wenn das Modell Faktoren f über nicht-maximale Cliquen vorsieht, dann werden diese f pro maximale Clique in einen Faktor f_C zusammengefasst.

Lernen von Faktoren über maximale Cliquen

$$P^{\#}(\boldsymbol{r}_f) \overset{\text{def}}{=} \frac{\#(\boldsymbol{r}_f)}{N} \overset{!}{=} P(\boldsymbol{r}_f)$$

- Mit $P(\boldsymbol{r}) = \frac{\prod_{\boldsymbol{C} \in V} P(\boldsymbol{r}_C)}{\prod_{\{i,j\} \in E} P(\boldsymbol{r}_{S_{ij}})} = \frac{\prod_{f \in F} P(\boldsymbol{r}_f)}{\prod_{\{i,j\} \in E} P(\boldsymbol{r}_{S_{ij}})}$, setze die Faktoren auf die normalisierten empirischen Zähler $P^{\#}(\boldsymbol{r}_f)$ und, für jeden Separator, wähle einen Nachbar und teile den Faktor durch den normalisierten empirischen Zähler des Separators $P^{\#}(\boldsymbol{r}_{S_{ij}})$, i.e.,
  - Für jedes $f \in F$,
    - Setze $\phi_f(\boldsymbol{r}_f) \leftarrow P^{\#}(\boldsymbol{r}_f)$
  - Für jedes $\{i,j\} \in E$, i.e., Separator $\boldsymbol{S}_{ij}$,
    - Wähle ein $h \in \{i,j\}$ zufällig
    - Setze $\phi_h(\boldsymbol{r}_h) \leftarrow \frac{\phi_h(\boldsymbol{r}_h)}{P^{\#}(\boldsymbol{r}_{S_{ij}})}$

  ML-Lernen mit Max Cliquen
- Sorgt für (bedingte) Wahrscheinlichkeitsverteilungen in den Faktoren → Erzwingt $Z = 1$
  - Vergleich: Importance Sampling für Faktormodelle

Lernen von Faktoren über maximale Cliquen: Beispiel

- Max-Cliquen-Faktoren auf normalisierte Zähler setzen

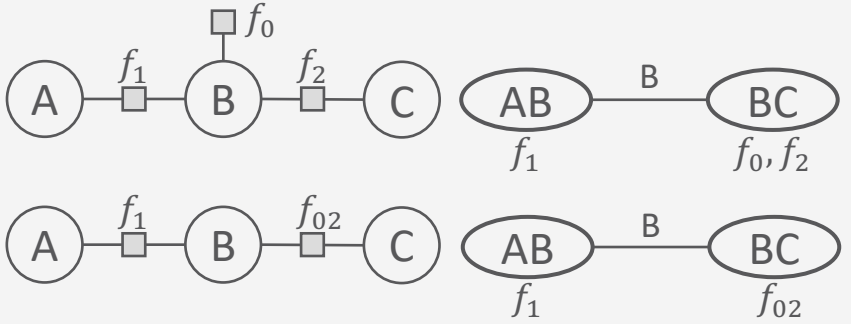
A	B	#	$P^{\#}(A,B) = f_1$
$false$	$false$	10	$10/70 = 0.14$
$false$	$true$	10	$10/70 = 0.14$
$true$	$false$	30	$30/70 = 0.43$
$true$	$true$	20	$20/70 = 0.29$

B	C	#	$P^{\#}(B,C)$
$false$	$false$	24	$24/70 = 0.34$
$false$	$true$	16	$16/70 = 0.23$
$true$	$false$	24	$24/70 = 0.34$
$true$	$true$	6	$6/70 = 0.09$

- Für jede Kante zwischen zwei maximalen Cliquen, einen der beiden Faktoren durch normalisierten Zähler des Separators teilen

B	#	$P^{\#}(B)$
$false$	40	$40/70 = 0.57$
$true$	30	$30/70 = 0.43$

B	C	$P^{\#}(B,C)/P^{\#}(B) = f_{02}$
$false$	$false$	$0.34/0.57 = 0.60$
$false$	$true$	$0.23/0.57 = 0.40$
$true$	$false$	$0.34/0.43 = 0.79$
$true$	$true$	$0.09/0.43 = 0.21$



A	B	C	#
$false$	$false$	$false$	6
$false$	$false$	$true$	4
$false$	$true$	$false$	8
$false$	$true$	$true$	2
$true$	$false$	$false$	18
$true$	$false$	$true$	12
$true$	$true$	$false$	16
$true$	$true$	$true$	4

Lernen mit allgemeinen Faktoren

- Mit Faktoren über nicht-maximale Cliquen, geschlossene Lösung nicht möglich
- Fixpunkt-Iteration: $\frac{\partial \log P(D|\theta)}{\partial \phi_f(\boldsymbol{r}_f)} = \frac{\#(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)} - N \cdot \frac{P(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)} = 0 \quad \rightarrow \quad \frac{\#(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)} = N \cdot \frac{P(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)}$

$$\frac{\#(\boldsymbol{r}_f)}{N \cdot \phi_f(\boldsymbol{r}_f)} = \frac{P(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)}$$

$$\frac{P^{\#}(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)} = \frac{P(\boldsymbol{r}_f)}{\phi_f(\boldsymbol{r}_f)}$$

$$\frac{\phi_f(\boldsymbol{r}_f)}{P^{\#}(\boldsymbol{r}_f)} = \frac{\phi_f(\boldsymbol{r}_f)}{P(\boldsymbol{r}_f)}$$

$$\phi_f(\boldsymbol{r}_f) = \phi_f(\boldsymbol{r}_f) \frac{P^{\#}(\boldsymbol{r}_f)}{P(\boldsymbol{r}_f)}$$

  - Update-Regel

$$\phi_f^{(t+1)}(\boldsymbol{r}_f) \leftarrow \phi_f^t(\boldsymbol{r}_f) \frac{P^{\#}(\boldsymbol{r}_f)}{P^t(\boldsymbol{r}_f)}$$

    - $\phi_f^t(\boldsymbol{r}_f)$ momentane Potentiale
    - $P^{\#}(\boldsymbol{r}_f)$ fest
    - $P^t(\boldsymbol{r}_f)$ Anfrage zu berechnen auf Basis des momentanen $\phi_f^t(\boldsymbol{r}_f)$

Iterative Proportional Fitting: Iteratives Proportionales Anpassen

- Iterative proportional fitting (IPF) Prozedur
  - Initialisiere alle Faktoren mit $t = 0$ uniform, z.B., alle Potentiale auf 1
  - **for** $t = 1,2, \ldots$ **do**
    - **if** Konvergenzkriterium gilt nicht **then**
      - **for** alle $\boldsymbol{r}_f \in \mathrm{Val}(\mathrm{rv}(f)), f \in F$ **do**

$$\phi_f^{(t+1)}(\boldsymbol{r}_f) \leftarrow \phi_f^t(\boldsymbol{r}_f)\frac{P^{\#}(\boldsymbol{r}_f)}{P^t(\boldsymbol{r}_f)}$$

    - **else break**

IPF

- Konvergenzkriterium, z.B., Fehlergrenze ε
  - $\forall f \in F : \phi_f^{(t+1)}(\boldsymbol{r}_f) - \phi_f^t(\boldsymbol{r}_f) < \varepsilon$
- Viele $P^t(\boldsymbol{r}_f)$ Anfragen zu berechnen für alle Faktoren!
  - Effizientere Ausführung mittels Jtree

Radim Jiroušek: Solution of the Marginal Problem and Decomposable Distributions. In: *Kybernetika 27*, 1991.
Radim Jiroušek and Stanislav Přeučil: On the Effective Implementation of the Iterative Proportional Fitting Procedure. In: *Computational Statistics & Data Analysis 19*, 1995.

IPF mit Jtrees

- Jtree nutzen um $P^t(\boldsymbol{r}_f)$ effizient zu berechnen
  - Jtree-Konstruktion unabhängig von Parametern
- IPF-JT: IPF mit Jtrees
  - Baue einen Jtree
  - Initialisiere alle Faktoren und Nachrichten uniform für $t = 0$
  - Wähle zufällig ein Cluster $\boldsymbol{C}_i$ als momentanes Cluster
  - **for** $t = 1,2, \ldots$ **do**
    - **if** Konvergenzkriterium gilt nicht **then**
      - **for** alle $\boldsymbol{r}_f \in \mathrm{Val}(\mathrm{rv}(f)), f \in F_i^t$ **do**

$$\phi_f^{(t+1)}(\boldsymbol{r}_f) \leftarrow \phi_f^t(\boldsymbol{r}_f)\frac{P^{\#}(\boldsymbol{r}_f)}{P^t(\boldsymbol{r}_f)}$$

    - **else break**
    - Wähle zufällig ein Nachbar $\boldsymbol{C}_j \in \mathrm{Nb}(\boldsymbol{C}_i)$ als neues momentanes Cluster
    - Berechne und sende Nachricht m_{ij}^t zu $\boldsymbol{C}_j$

Berechne $P^t(\boldsymbol{r}_f)$ auf $\{m_{ji}^t\}_{\boldsymbol{C}_j \in \mathrm{Nb}(\boldsymbol{C}_i)}$ und F_i^t

- Organisiere Berechnung der unterschiedlichen $P^t(\boldsymbol{r}_f)$ in sinnvoller Reihenfolge

Keine Reihenfolge vorgeschrieben; implizite Anforderung, dass alle $\boldsymbol{C}_i$ häufig genug besucht werden

IPF-JT

Yee Whye Teh and Max Welling: On Improving the Efficiency of the Iterative Proportional Fitting Procedure. In: *AISTATS-03 Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

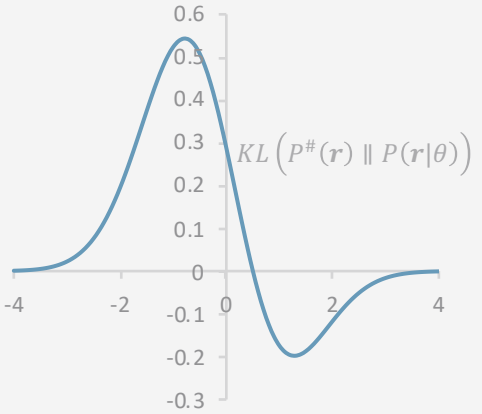
Eigenschaften von IPF Updates

- Ein Update pro Cluster bei Faktoren über maximale Cliquen
- Koordinatenanstieg-Algorithmus
  - Koordinaten = Parameter $\boldsymbol{\theta}$ in Clustern
  - In jedem Schritt erhöht das Update die Log-Wahrscheinlichkeit der Daten, $\log P(\boldsymbol{d}|\boldsymbol{\theta})$, und es konvergiert zu einem *globalen Maximum*
- Maximieren der Log-Wahrscheinlichkeit ist äquivalent zum Minimieren der KL Divergenz (Kreuzentropie, *cross entropy*)

$$\max \log P(\boldsymbol{d}|\boldsymbol{\theta}) \Leftrightarrow \min KL\left(P^{\#}(\boldsymbol{r}) \parallel P(\boldsymbol{r}|\boldsymbol{\theta})\right)$$

$$KL\left(P^{\#}(\boldsymbol{r}) \parallel P(\boldsymbol{r}|\boldsymbol{\theta})\right) = \sum_{\boldsymbol{r}} P^{\#}(\boldsymbol{r}) \log \frac{P^{\#}(\boldsymbol{r})}{P(\boldsymbol{r}|\boldsymbol{\theta})}$$

  - Max-Entropie-Prinzip für Parametrisierung: *duale Perspektive zu MLE*



IPF mit nicht vollständigen Daten

- Problem mit nicht vollständigen Daten:
  - Update-Regel

$$\phi_f^{(t+1)}(\boldsymbol{r}_f) \leftarrow \phi_f^t(\boldsymbol{r}_f)\frac{P^{\#}(\boldsymbol{r}_f)}{P^t(\boldsymbol{r}_f)}$$

  - Versteckte Variablen haben kein $P^{\#}(\boldsymbol{r}_f)$

- EM-IPF: IPF Prozedur mit dem EM-Algorithmus verbunden
  - E-Schritt: Berechne *erwartete* Zähler für versteckte Variablen
  - M-Schritt: Aktualisiere Parameter $\phi_f^{(t+1)}(\boldsymbol{r}_f)$
  - Auch mit dem Jtree für Effizienz vereinbar

Zwischenzusammenfassung

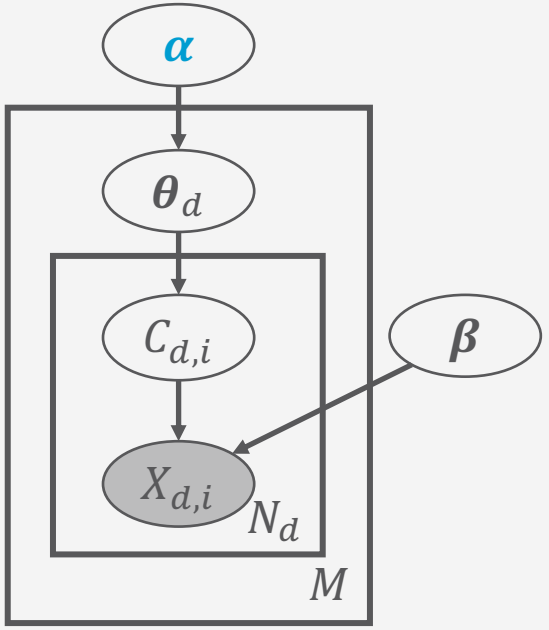
- Parameter lernen in ungerichteten Modellen schwerer, weil $Z \neq 1$
- IPF
  - Fixpunkt-Iteration
    - Reduziert sich auf ein Update pro Faktor, wenn Faktoren über maximal Cliquen gehen
      - Dann $Z = 1$ erzwingen
  - Jtrees für Effizienz
    - Update auf einem Cluster, dann aktualisierte Information an einen Nachbarn schicken
  - EM Variante für nicht vollständige Daten

Latent Dirichlet Allocation (LDA) zur Topic Modellierung

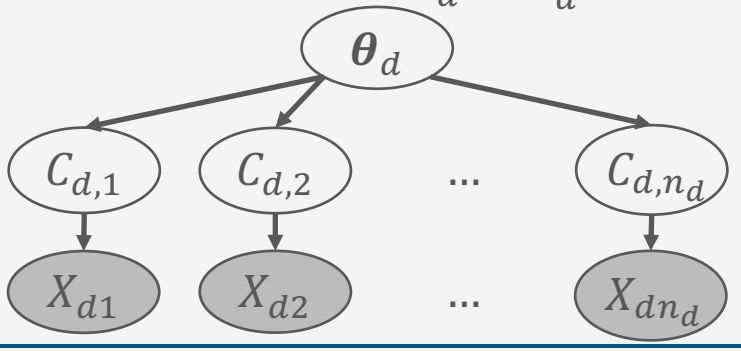
Umsetzung von Lernverfahren in komplexen Modellen

Topic Modellierung: Mixture of Topics

- Annahme: Dokumente bestehen aus einem Mix von Topics
- Pro Dokument $d \in \mathcal{C}$: Zufallsvariable X_i bezeichnet, welches Wort w aus einem Lexikon $\mathcal{D}$ an Position i in d vorkommt
  - I.e., $\text{Val}(X_i) = \{w\}_{w \in \mathcal{D}}$
    - Bzw. $\text{Val}(X_i) = \mathcal{D}$ mit $\mathcal{D}$ als Menge der Wörter im Lexikon
  - X_i unabhängig von X_j gegeben k:
$$P(X_i, X_j | C) = P(X_i | C) P(X_j | C)$$
  - Annahme: Vorkommen von w unabhängig von Position:
$$P(X_i = w) = P(X_j = w)$$
- Verteilung über mögliche A-priori-Topic-Verteilung $\boldsymbol{\theta}_d$ für jedes Dokument: Hyperparameter $\boldsymbol{\alpha}$
  - Vorwissen kodieren, welche Verteilungen wahrscheinlicher sind



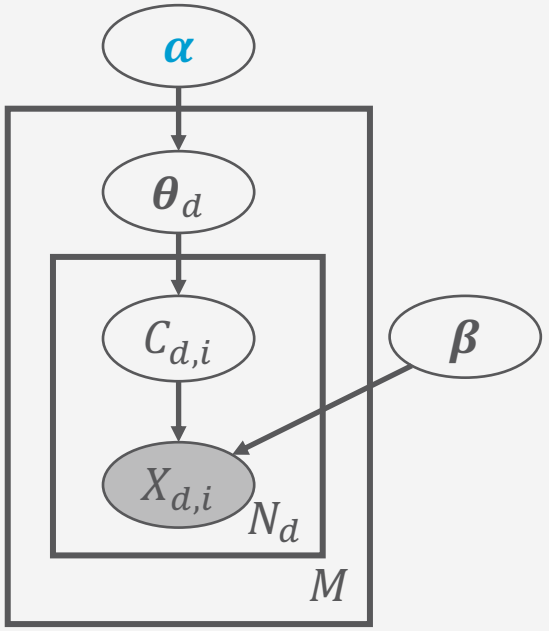
Pro Dokument d mit $\boldsymbol{\theta}_d \neq \boldsymbol{\theta}_{d'}$:



David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

Topic Modellierung: Mixture of Topics

- Dokumente bestehen aus einem Mix von Topics
  - M Dokumente, N_d Worte im Dokument d
  - K mögliche Topics, pro Dokument d Topic-Verteilung $\boldsymbol{\theta}_d$
- Annahme: Dokumente des Corpus sind Ergebnis des folgenden Generierungsprozesses
  - Sample eine Topic-*Verteilung* $\boldsymbol{\theta}_d$ über K Topics aus $Dir(\boldsymbol{\alpha})$
  - Wähle N_d (z.B. sample aus Poisson-Verteilung mit Hyperparameter ξ)
  - Wiederhole N_d mal
    - Sample Topic $c_{d,i}$ aus $P(C_{d,i}) = \boldsymbol{\theta}_d$
    - Sample Wort $x_{d,i}$ aus $P(X_{d,i}|c_{d,i}) = \boldsymbol{\beta}[c_{d,i}]$
      - Häufige Schreibweise in der Forschungsgemeinde: $c_{d,i} \sim Mul(\cdot \mid \boldsymbol{\theta}_d)$ und $x_{d,i} \sim Mul(\cdot \mid \boldsymbol{\beta}_c)$



$$\boldsymbol{\theta}_d = (0.32, 0.34, 0.25, 0.09)$$
$$N_d = 68$$

David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

Topic Modellierung: Mixture of Topics

- LDA-Sicht auf ein Dokument
  - Worte des Dokuments kommen aus 4 Topics

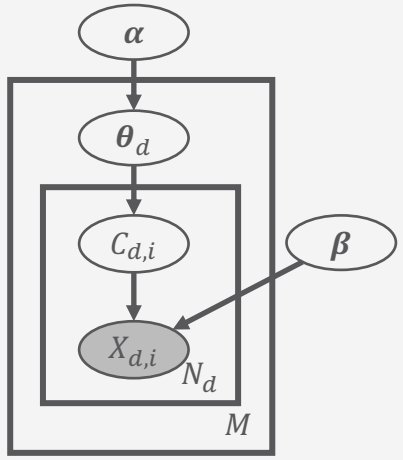
The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropoli-tan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants and act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200.000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400.000 each. The Juilliard School, where music and the performing arts are taught, will get $250.000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100.000 donation, too.

1 (Arts)	2 (Budgets)	3 (Children)	4 (Education)
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

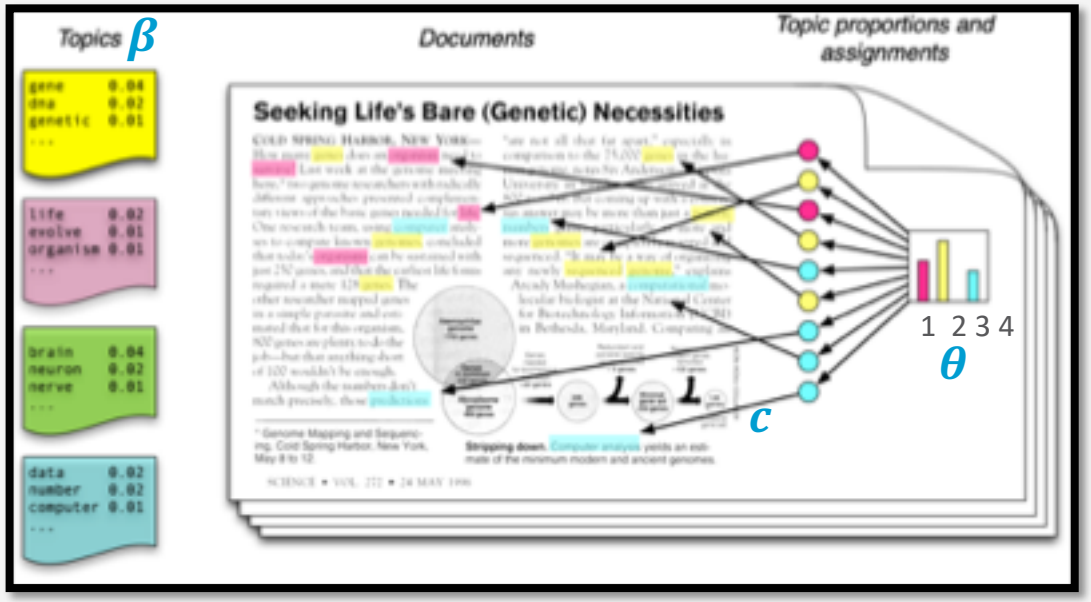
David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

Unbekannte Parameter in LDA

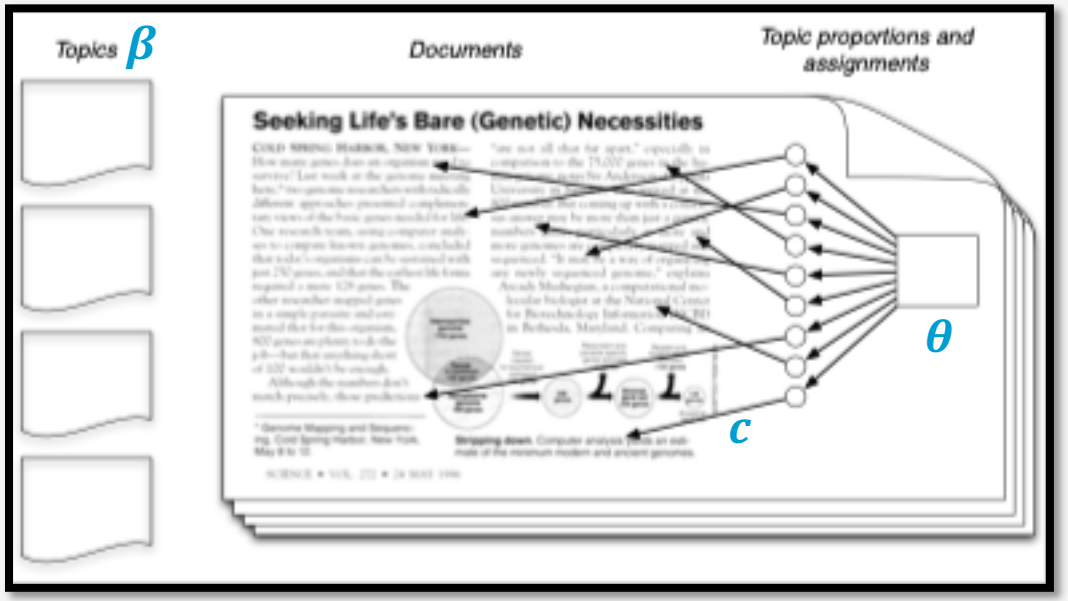
- Was sind die *Topics der Worte*, wie lauten die *Topicverteilungen der Dokumente* und wie lauten die *Wortverteilungen der Topics*?
  - $P(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{c} | \boldsymbol{x}, \boldsymbol{\alpha})$



Ziel



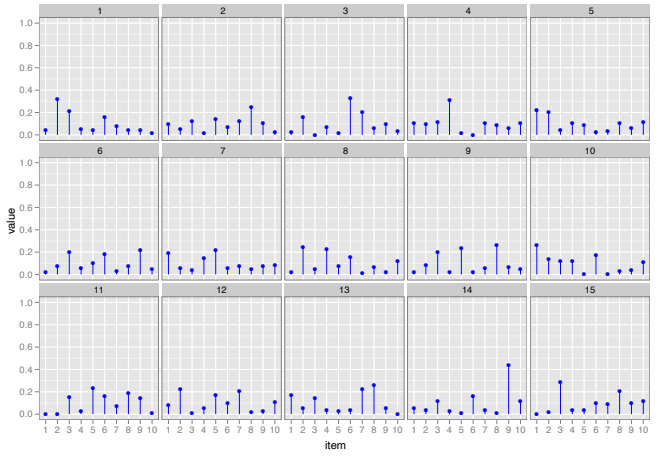
Vorhanden



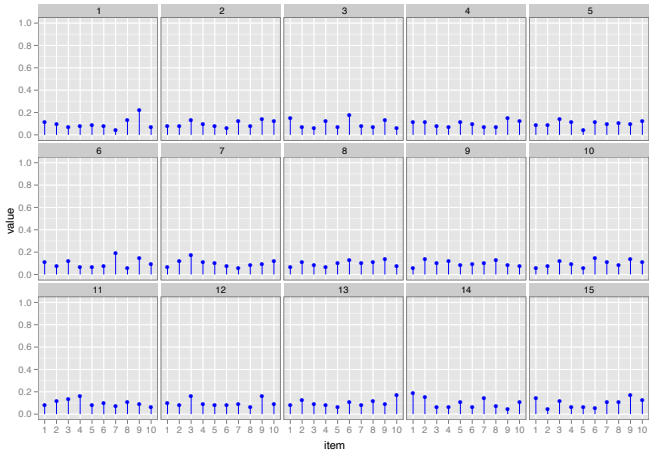
David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

Hyperparameter α: Einfluss auf das Lernergebnis

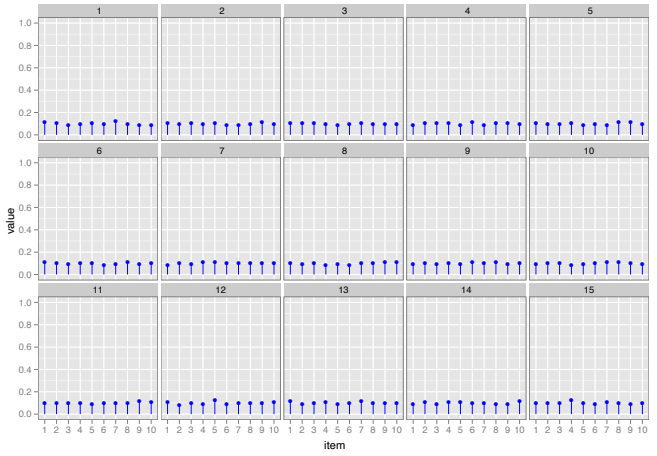
$\alpha = 1$



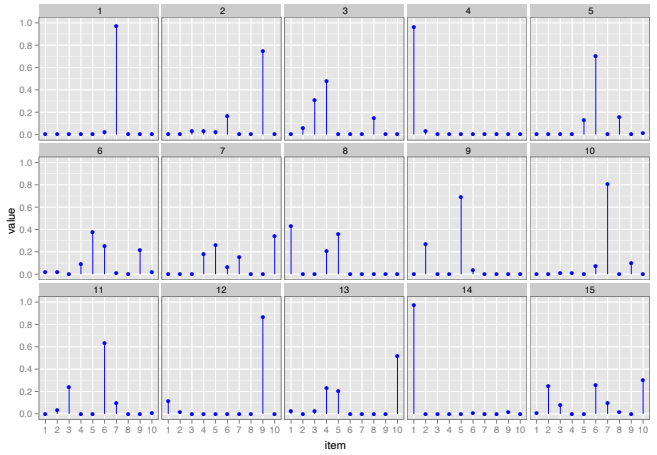
$\alpha = 10$



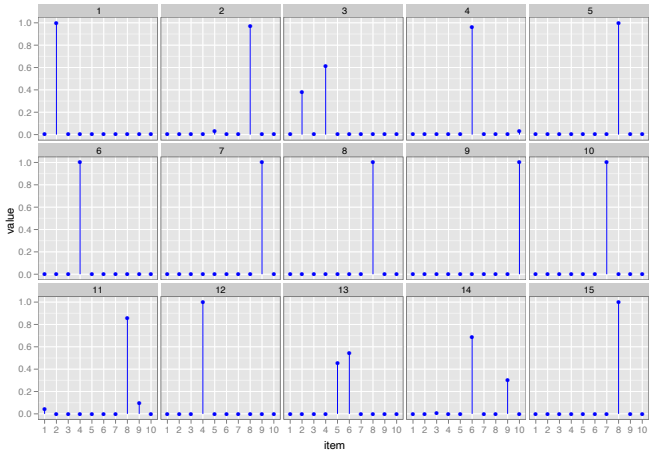
$\alpha = 100$



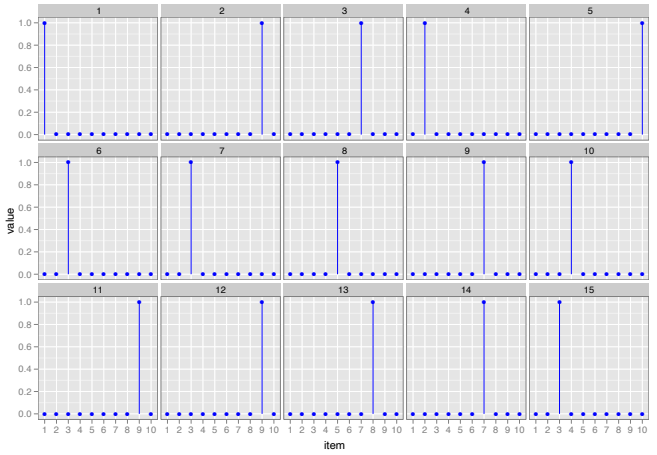
$\alpha = 0.1$



$\alpha = 0.01$



$\alpha = 0.001$



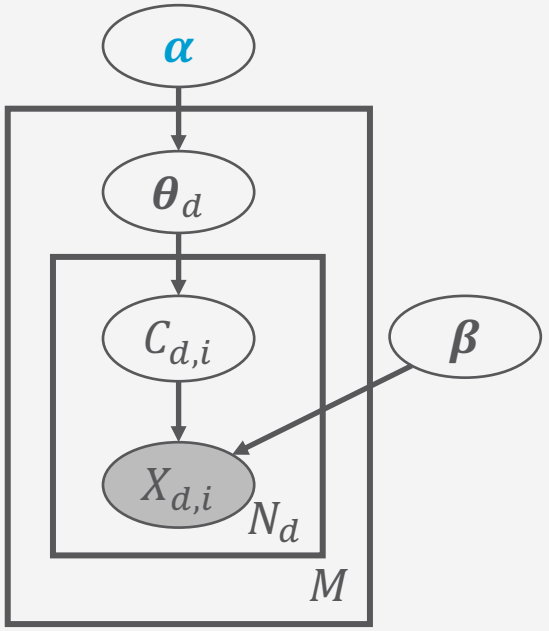
David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

LDA: Parameter Lernen

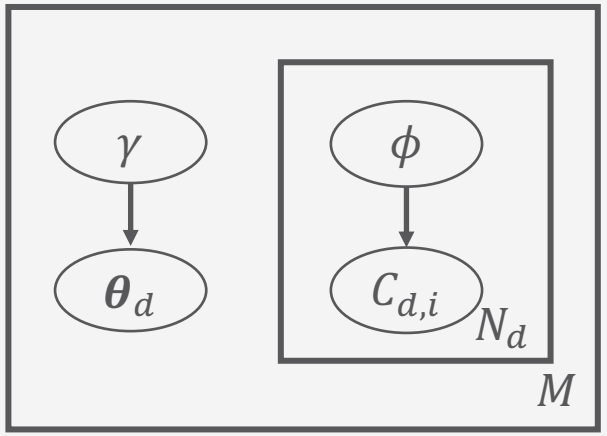
- Zwei Methoden
  - Variational Inference / EM
    - Numerische Approximation über untere Schranken
    - Bias in der Lösung
    - Konvergenz mit numerischen Garantien
  - Gibbs Sampling
    - Stochastische Simulation
    - Kein Bias in der Lösung
    - Stochastische Konvergenz

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. In: *Journal of Machine Learning Research*, 3:993-1022, January 2003

LDA: Variational Inference – Skizze



- Ersetze LDA Modell P mit einem einfacheren Modell Q:



- Minimiere die KL Divergenz iterativ im EM-Stil:

$$(\gamma^*, \phi^*) = \arg\min_{(\gamma,\phi)} KL(Q(\theta, z|\gamma, \phi) \parallel P(\theta, z|w, \alpha, \beta))$$

David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

LDA: Gibbs Sampling – Skizze

- MCMC Algorithmus
  - Halte alle momentanen Werte außer einem fest und sample den Wert, z.B., für $C_{i,d}$

$$P(C_{i,d} = k | \boldsymbol{c}_{-i,d}, \boldsymbol{w})$$

    - $\boldsymbol{c}_{-i,d}$ die Topic-Zuweisungen der Worte $X_{j \neq i}$ (ohne i)
  - Notation
    - c entspricht z, x entspricht w
    - Alle Dokument gleiche Länge N
    - $n_{d,k}$: Häufigkeit von Topic k in Dokument d → über alle k normiert ergeben die Zähler $\boldsymbol{\theta}_d$
    - $n_{k,w}$: Häufigkeit von Wort w in Topic $k \rightarrow \boldsymbol{\beta}_k$
    - n_k: Häufigkeit von Topic $k \rightarrow \boldsymbol{\alpha}$ (angepasst)
  - Konvergiert *irgendwann* zur wahren Verteilung

Input: words $\mathbf{w} \in$ documents $\mathbf{d}$
Output: topic assignments $\mathbf{z}$ and counts $n_{d,k}, n_{k,w}$, and n_k
begin
  randomly initialize $\mathbf{z}$ and increment counters
  foreach *iteration* **do**
    for $i = 0 \rightarrow N - 1$ **do**
      $word \leftarrow w[i]$
      $topic \leftarrow z[i]$
      $n_{d,topic}$-=1; $n_{word,topic}$-=1; n_{topic}-=1
      for $k = 0 \rightarrow K - 1$ **do**
        $p(z = k|\cdot) = (n_{d,k} + \alpha_k)\frac{n_{k,w} + \beta_w}{n_k + \beta \times W}$
      end
      $topic \leftarrow$ sample from $p(z|\cdot)$
      $z[i] \leftarrow topic$
      $n_{d,topic}$+=1; $n_{word,topic}$+=1; n_{topic}+=1
    end
  end
  return $\mathbf{z}$, $n_{d,k}, n_{k,w}, n_k$
end

Algorithm 1: LDA Gibbs Sampling

Algorithmus https://u.cs.biu.ac.il/~89-680/darling-lda.pdf

LDA: Parameter lernen

- Zwei Methoden
  - Variational Inference / EM
    - Numerische Approximation über untere Schranken
    - Bias in der Lösung
    - Konvergenz mit numerischen Garantien
  - Gibbs Sampling
    - Stochastische Simulation
    - Kein Bias in der Lösung
    - Stochastische Konvergenz
- Vergleich
  - Gibbs Sampling ist langsamer
    - Läuft Tage für einen Datensatz mittlerer Größe
    - Variational Inference braucht nur ein paar Stunden dafür
  - Gibbs Sampling ist genauer
    - Gibbs Sampling Konvergenz schwer zu testen

David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

Warum LDA funktioniert?

- Trade-off zwischen zwei Zielen:
  1. Für jedes Dokument: Worte so wenig Topics wie möglich zuschreiben
  2. Für jedes Topic: Hohe Wahrscheinlichkeiten so wenigen Worten wie möglich zuweisen
- Diese Ziele stehen im Konflikt
  - Worte eines Dokuments d einem einzigen Topic k zuschreiben macht Ziel 2 schwer erfüllbar: Alle Worte in d müssen eine nicht vernachlässigbare Wahrscheinlichkeit in k haben
  - Wenige Worte einem Topic k zuschreiben, so dass (nur) diese eine hohe Wahrscheinlichkeit in k haben, macht Ziel 1 schwer erfüllbar:
    Um die Worte eines Dokuments d mit hoher Wahrscheinlichkeit zu erzeugen, müssen die Worte vielen Topics zugeschrieben sein
- Diese Ziele gegeneinander abwägen ermöglicht Gruppen von häufig miteinander auftretenden Worten zu finden

David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

LDA: Anwendung zur Analyse von sozialen Netzwerken

- “follow”-Beziehung zwischen Nutzer*innen sieht erstmal chaotisch aus
  - Netzwerk willkürlich entstanden und nicht zentral kontrolliert

- Strukturen in dieser Art von Beziehung finden, indem man
  - Nutzer*innen basierend auf ihren „Topic“-Interessen gruppiert und
  - Jede Beziehung mit der identifizierten „Topic“-Gruppe labelt

- Nutzt Gibbs Sampling

- Getestet auf Twitter Daten

Youngchul Cha and Junghoo Cho, Social-Network Analysis Using Topic Models. In: *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '12), 2012

LDA: Anwendung zur Analyse von sozialen Netzwerken

- Spezifische Topic-Gruppe
  - g: Nutzer*in, |e'(g)|: Follower-Zahl, Bio: Twitter Bio

g	|e'(g)|	Bio
lancearmstrong	2370	7-time Tour de France winner, full time cancer fighter - LIVESTRONG!
levileipheimer	285	Pro Cyclist
ghincapie	234	*[No Bio] Professional Cyclist*
johanbruyneel	203	9X winning Tour De France Sports Director
barackobama	7410	44th President of the United States
teamradioshack	141	*[No Bio] Pro Cycle Team RadioShack*
dzabriskie	137	Professional Bike Racer
phillliggett	126	Pro-cycling commentator.
christianvdv	114	Professional Cyclist
stephenfry	3384	British Actor, Writer, Lord of Dance, Prince of Swimwear & Blogger

- Topic-Gruppe „mobile gadget blog" aus dem „non-popular" Datensatz
  - Nur Twitter Nutzer*innen mit |e'(g)|≤100

g	|e'(g)|	Bio
googleimages	94	News, tips, and tricks direct from the Google Images team
androidandme	84	Meet your new Android friend.
androidguys	73	Your trusted source for Android news and opinion since November 5, 2007.
prethinking	66	PreThinking.com - your source for all things Palm Pre
engadgetmobile	97	Official Twitter account of Engadget Mobile!
skyfire	88	Skyfire 3.0 for Android and Skyfire 2.0 for iPhone are changing the way people browse on thier mobile devices! www.get.skyfire.com
precentral	75	All about Palm webOS devices
boygenius	95	President, General Manager, Editor-in-chief of BGR.com. Professional smack talker.
theandroidsite	71	News and reviews of Google's Android OS
nokconv	58	A team of Nokia tweeters. Follow us for various updates, info, and the occasional banal banter.

LDA: Performanz bewerten mittels Perplexität

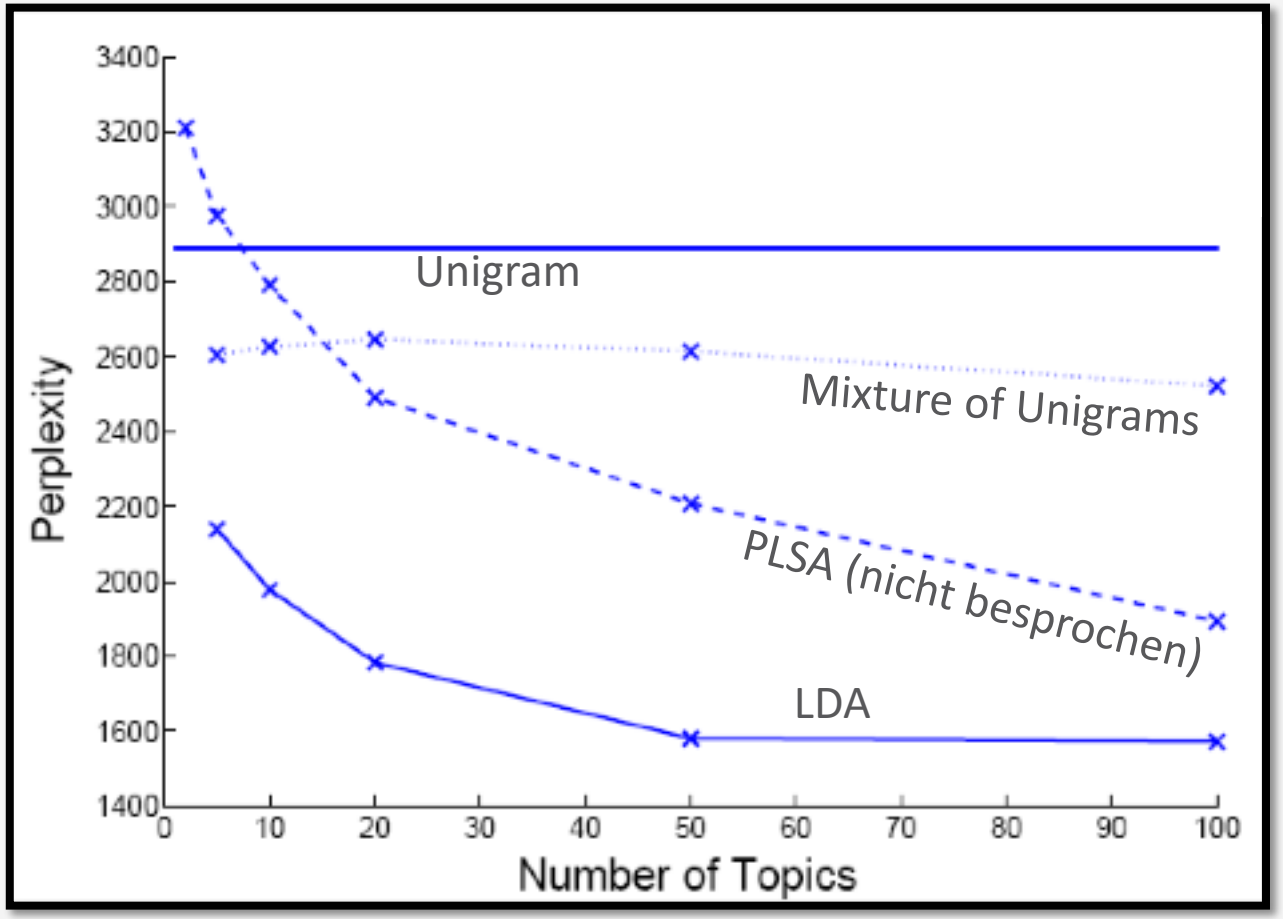
- Wie gut sagt eine Wahrscheinlichkeitsverteilung oder probabilistisches Modell ein Sample vorher?
  - Q: Modell einer unbekannten Wahrscheinlichkeitsverteilung P gelernt aus einem Datensatz bestehend aus Datenpunkten gesampelt aus P
  - Werte Q aus, indem man fragt, wie gut Q ein gesondertes Test-Sample $x_1, \dots, x_n$ (auch aus P gesampelt) vorhersagt
  - Perplexität (*perplexity*) von Q definiert als

$$b^{-\frac{1}{n}\sum_{i=1}^{n}\log_b Q(x_i)}$$

    - b üblicherweise 2
  - Ein besseres Q tendiert dazu höhere Wahrscheinlichkeiten $Q(x_i)$ zuzuweisen → geringere Perplexität („weniger überrascht – perplex – vom Sample“)

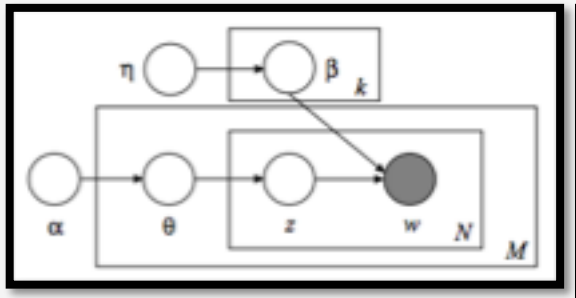
Perplexität unterschiedlicher Modelle

Weniger ist besser →

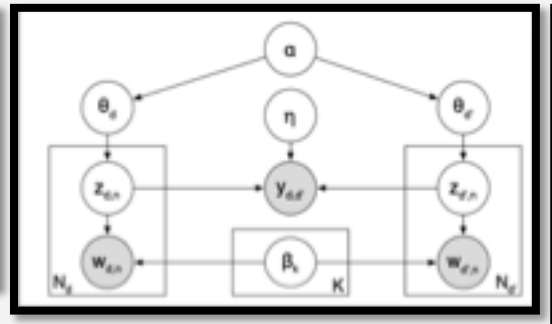


LDA Erweiterungen

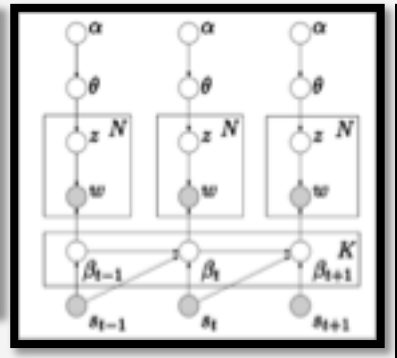
Smoothed LDA (Hyperparameter wie α zur Beeinflussung der Wortverteilungen pro Topic)



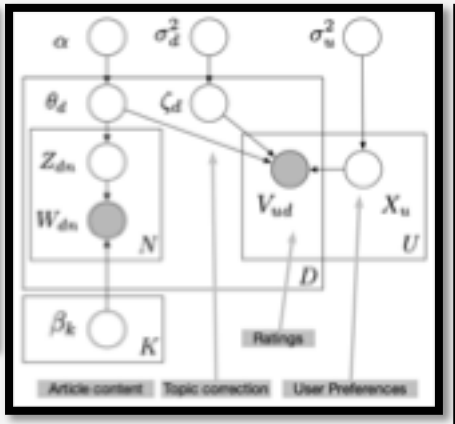
Relational Topic Model (Links zwischen Papieren)



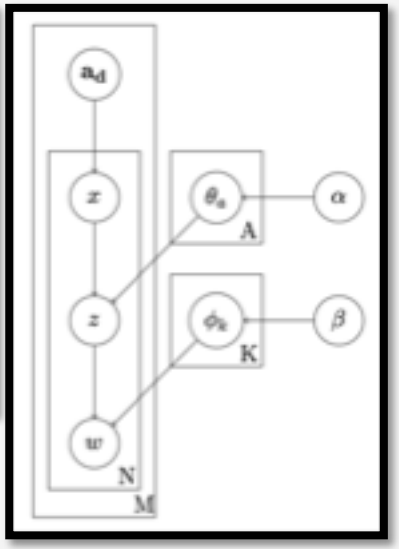
Dynamic Topic Model (über die Zeit)



Topic Model für Empfehlungen



Autor-Topic Modell (Übung 2)



- Viele Erweiterungen sind im Laufe der Zeit vorgeschlagen worden
- Implementierungen in Python vorhanden (mindestens LDA, aber auch andere Varianten)

Abbildung: David Blei, Vortrag. ICML, 2012.

Zwischenzusammenfassung

- Komplexes Modell, in dem viele versteckte Parameter zu lernen sind
- Annahme: Dokumente sind durch einen generativen Prozess entstanden
  - Worte als Evidenz dieses Prozesses
- Parameter lernen mittels
  - Variational Inference + EM
    - Einfacheres Modell nehmen, KL Divergenz minimieren mit einem EM-Vorgehen
  - Gibbs Sampling
    - Relative Häufigkeiten durch Samples abschätzen

Überblick: 5. Lernalgorithmen für episodische PGMs

A. Überblick
- Full Bayesian Learning, MAP Learning, Maximum Likelihood (ML) Learning

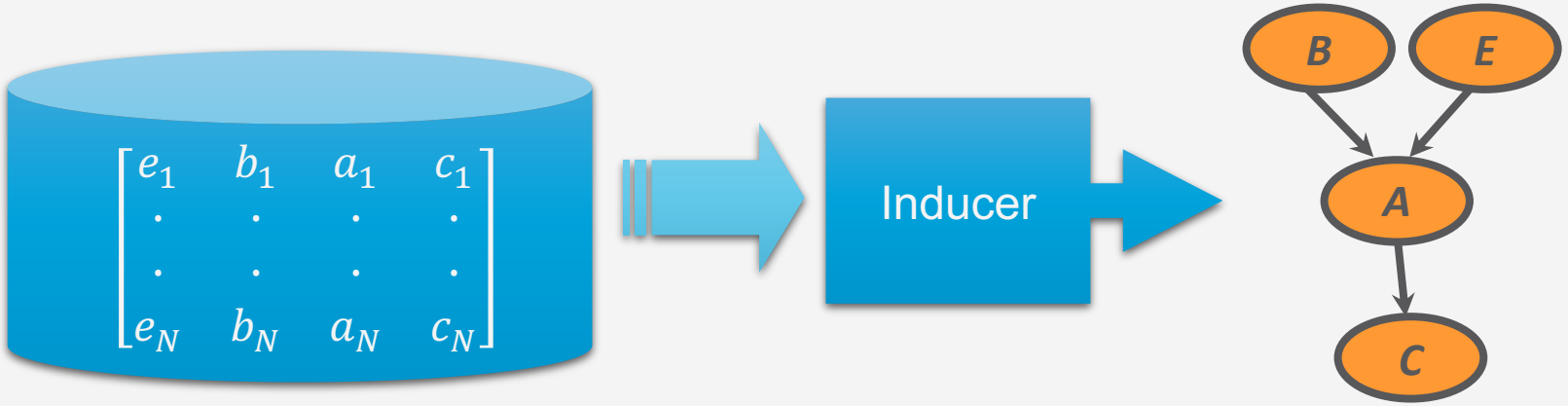
B. ML-basiertes Parameter Lernen
- In BNs: ML (vollständige Daten), Expectation-Maximisation (EM; nicht-vollständige Daten)
- In Faktormodellen: Iterative Proportional Fitting (IPF), EM-IPF
- Anwendung: LDA

C. ML-basiertes Struktur Lernen
- Model Selection, Structural EM

D. Abschlussbemerkungen
- Alternativen, generative vs. diskriminative Modelle, Korrelation vs. Kausalität

Struktur lernen

- Unbekannte Graphstruktur
- Gegeben Datensatz $\boldsymbol{d}$
- Finde Modell, das am besten zu $\boldsymbol{d}$ passt; beinhaltet:
  - *Model Selection*: Modellwahl
  - Parameter Schätzung (wie auf den vorherigen Folien)

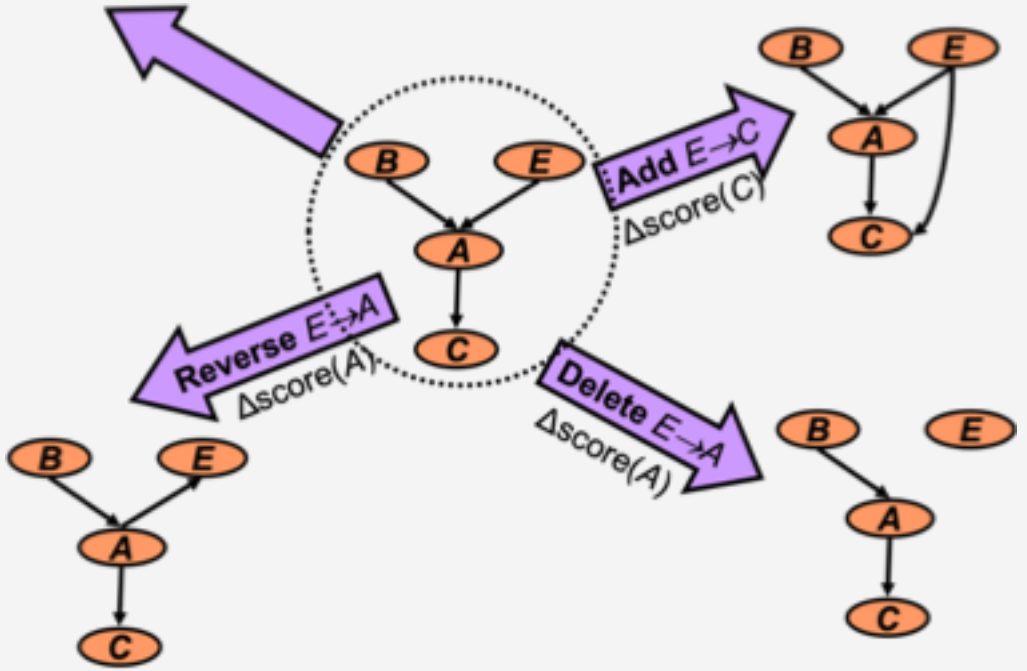


Data $\boldsymbol{d}$

Model Selection: Modellauswahl

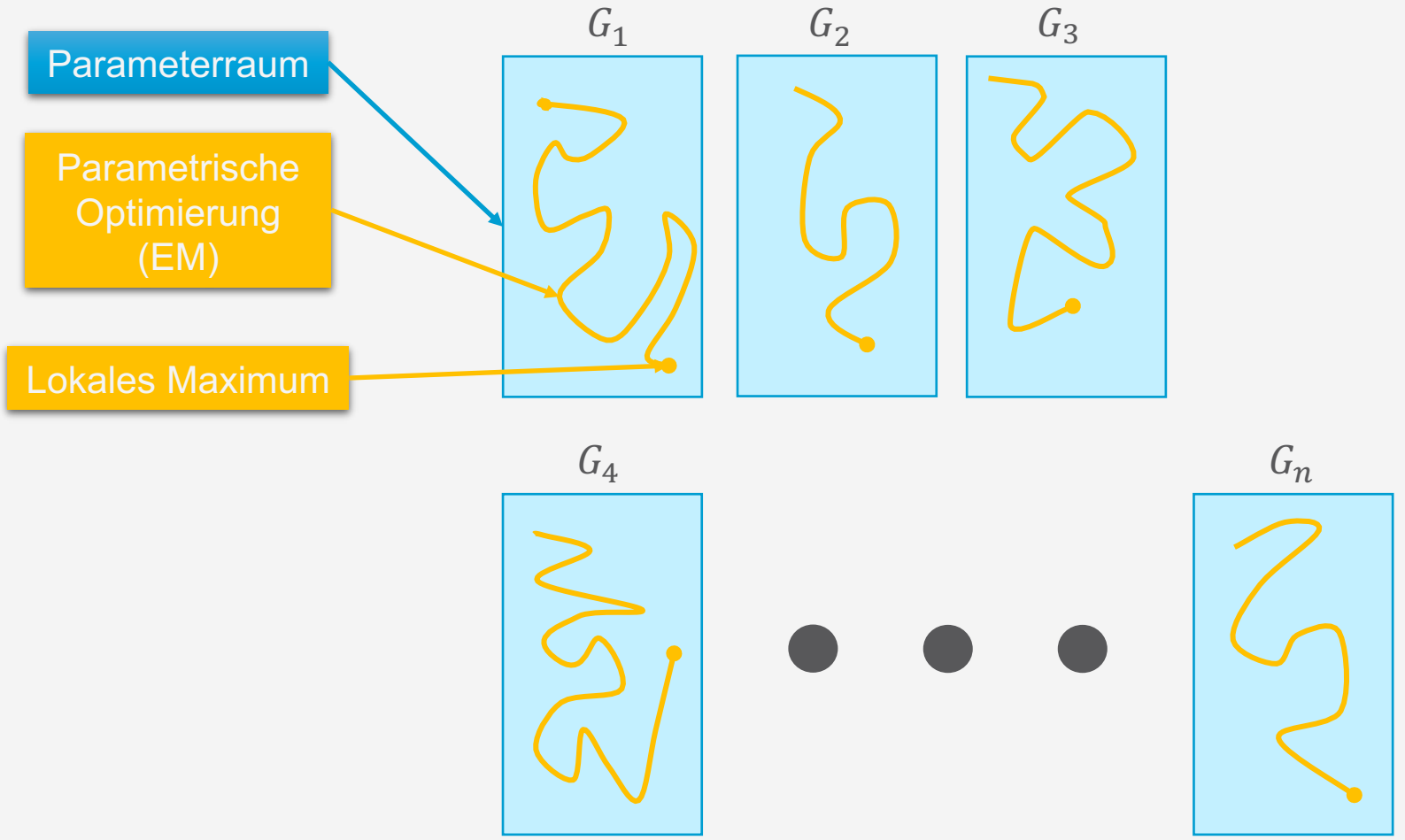
- Ziel: Beste Graphstruktur wählen
- Eingabe: Daten, Bewertungsfunktion (*scoring function*)
- Ausgabe: Graphstruktur mit maximaler Bewertung

1. Führe eine heuristische Suche nach Modellkandidaten durch
2. Schätze in jedem Modell die Parameter mit EM
   - Bei vollständigen Daten → z.B. MLE
3. Bewerte jedes Modell
4. Wähle das Modell mit der höchsten Bewertung



Lokale Suche in der Praxis

- Schätze in jedem Modell die Parameter mit EM
- Berechnungstechnisch teuer
  - Parameter optimieren mit EM – nicht trivial
  - EM für alle Kandidaten ausführen
  - Verbringt Zeit auch auf schwachen Kandidaten
  - Daten können unvollständig sein (nicht repräsentativ)



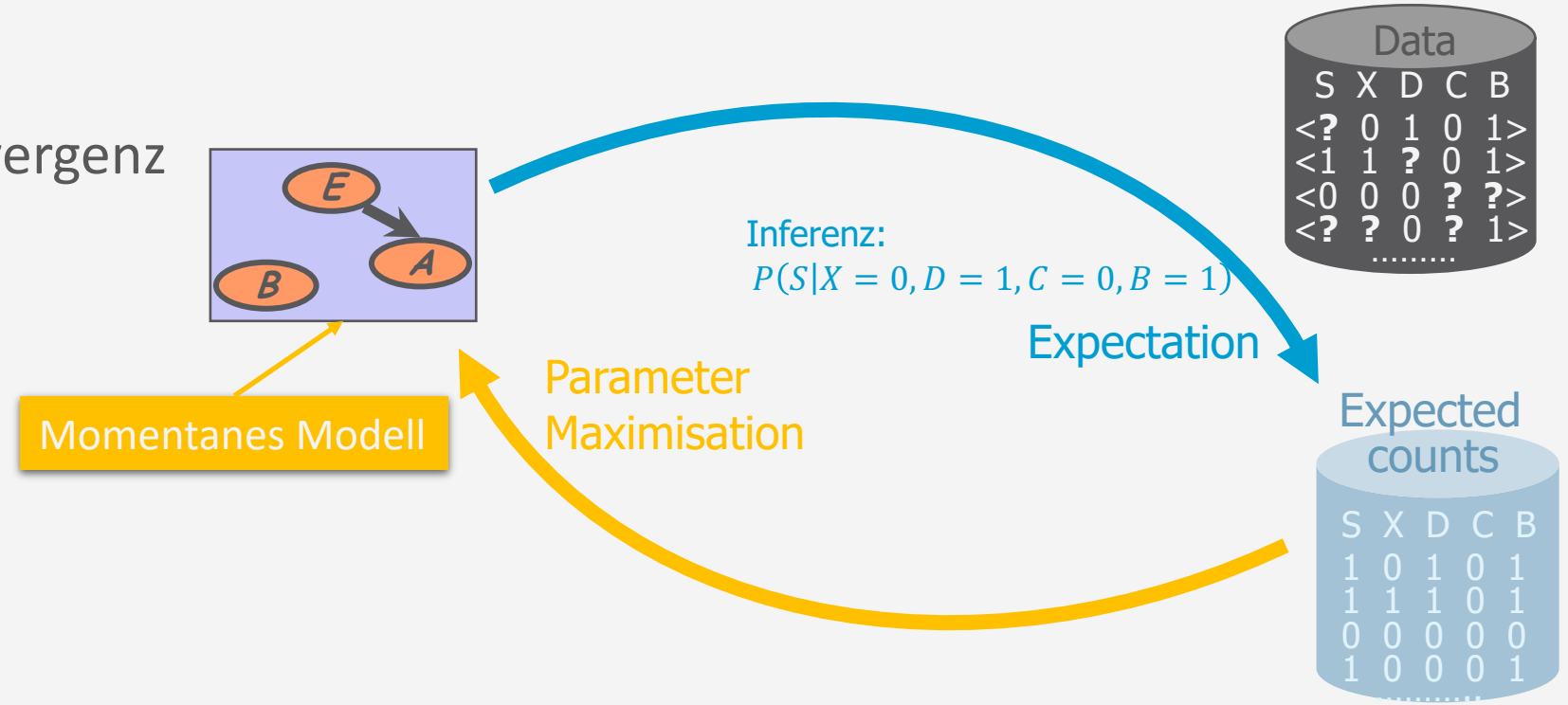
Struktur lernen: Nicht vollständige Daten

- Problem: versteckte, unbekannte Variablen
  - Keine Daten für die versteckten Variablen
  - Suchraum sehr groß
- Idee:
  - Nutze das momentane Modell um neue Strukturen zu bewerten
- Übersicht:
  - Suche im (Struktur, Parameter)-Raum durchführen
  - In jeder Iteration, das momentane Modell nutzen zum Finden von:
    - Besser bewertete Parameter: „parametrischer“ EM-Schritt

    oder
    - Besser bewertete Struktur: „struktureller“ EM-Schritt

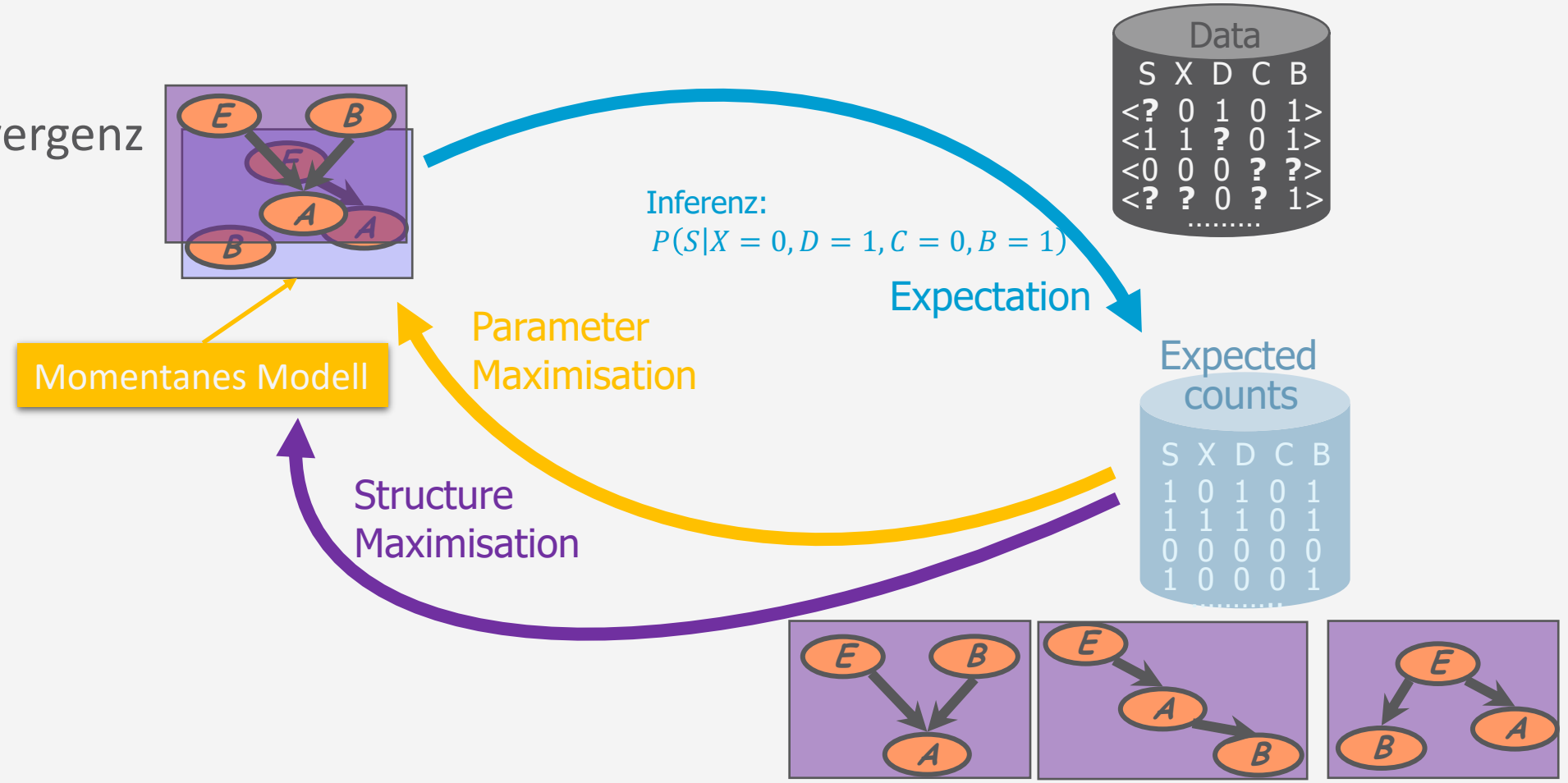
EM

EM-Algorithmus:
Iteriere bis zur Konvergenz



Structural EM

SEM-Algorithmus:
Iteriere bis zur Konvergenz



Zwischenzusammenfassung

- Heuristische Suche nach geeigneten Graphstrukturen
- Model Selection
  - Suche nach Kandidaten
  - EM für jeden Kandidaten, um beste Parameter zu bestimmen
  - Den Kandidaten mit höchster Bewertung nehmen
- Structural EM
  - Immer nur ein momentanes Modell, welches man entweder bezogen auf die Parameter aktualisiert oder bezogen auf die Struktur anpasst
  - Abwechselnd Parameter und Struktur optimieren

Überblick: 5. Lernalgorithmen für episodische PGMs

A. Überblick
- Full Bayesian Learning, MAP Learning, Maximum Likelihood (ML) Learning

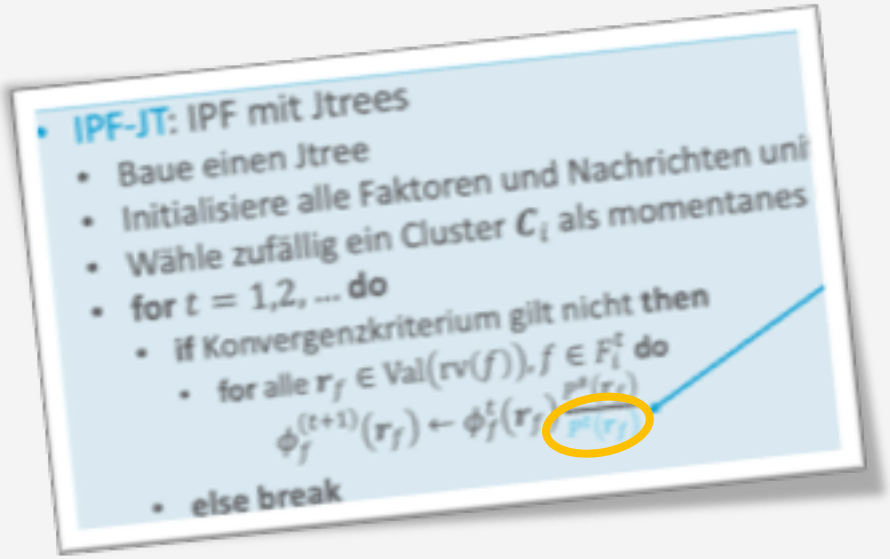
B. ML-basiertes Parameter Lernen
- In BNs: ML (vollständige Daten), Expectation-Maximisation (EM; nicht-vollständige Daten)
- In Faktormodellen: Iterative Proportional Fitting (IPF), EM-IPF
- Anwendung: LDA

C. ML-basiertes Struktur Lernen
- Model Selection, Structural EM

D. Abschlussbemerkungen
- Alternativen, generative vs. diskriminative Modelle, Korrelation vs. Kausalität

Weitere Möglichkeiten

- Approximative Inferenz um Anfragen in Lern-Prozeduren zu beantworten
  - Z.B. Sampling: Beispiel Gibbs-LDA
- Andere Optimierungskriterien nutzen
  - Pseudo-Likelihood
  - Kontrastive Divergenz
- Struktur lernen
  - Ähnliche Ideen für ungerichtete Modelle verfolgen
    - Model Selection + Parameter Estimation verschränken wie in structural EM
  - Unabhängigkeitstests nutzen
    - (Bedingt) unabhängige Variablen finden, Kanten löschen
      - In Vorlesungsteil 2 Episodische PGMs skizziert
- Mehr Informationen in PGM, Kapitel 20



Generative & Diskriminative Modelle

- Methoden hier sind dafür da, Modelle zu lernen, die eine vollständige gemeinsame Wahrscheinlichkeitsverteilung $P(\boldsymbol{R})$ kodieren
  - Jede Art von Inferenzaufgabe lösen: Vorhersage, Klassifizierung, Wahrscheinlichkeiten von Ereignissen, wahrscheinlichste Zustände, neue Daten *generieren* (Sampling)

→ Generative Modelle

- Im Gegensatz dazu stehen: Diskriminative Modelle (wie neuronale Netze)
  - Spezifisch dafür gebaut und trainiert um Klassifikationsleistung zu maximieren: $P(C|\boldsymbol{X})$
    - Keine allgemeinen Inferenzaufgaben damit lösbar
  - Bei Klassifikationsanfragen im Allgemeinen bessere Performanz dieser Modelle im Vergleich zu generativen Modellen, wenn eine angemessene Menge an Trainingsdaten zur Verfügung steht
    - Fokus auf Abbildung einer bedingten Verteilung (muss nicht $P(C, \boldsymbol{X})$ darstellen, nur $P(C|\boldsymbol{X})$)

In beiden Fällen gilt: Modelle sind eine Abstraktion / Generalisierung der Daten, welche die Daten nicht mit 100% Genauigkeit repräsentieren kann. Nur die Daten selbst können das.

Vorsicht: Korrelation vs. Kausalität

- Korrelative Zusammenhänge beim Lernen ausgenutzt
  - Aussage: Wenn A vorkommt, kommt auch B häufig vor
    - Keine Aussage darüber, ob A B verursacht oder B A oder ob es eine dritte Variable C gibt, die Verursacher ist, oder ob gar kein Zusammenhang besteht (*spurious correlation*)
    - ML-basiertes Parameter lernen: Wir haben beim Zählen für bedingte Wahrscheinlichkeiten nur gezählt, wie häufig $R = r$ vorkam gegeben jeweils die Werte von $\mathrm{Pa}(R)$
      - Reine Korrelation, keine Aussage über Kausalität
- Kausaler Zusammenhang unbekannt!

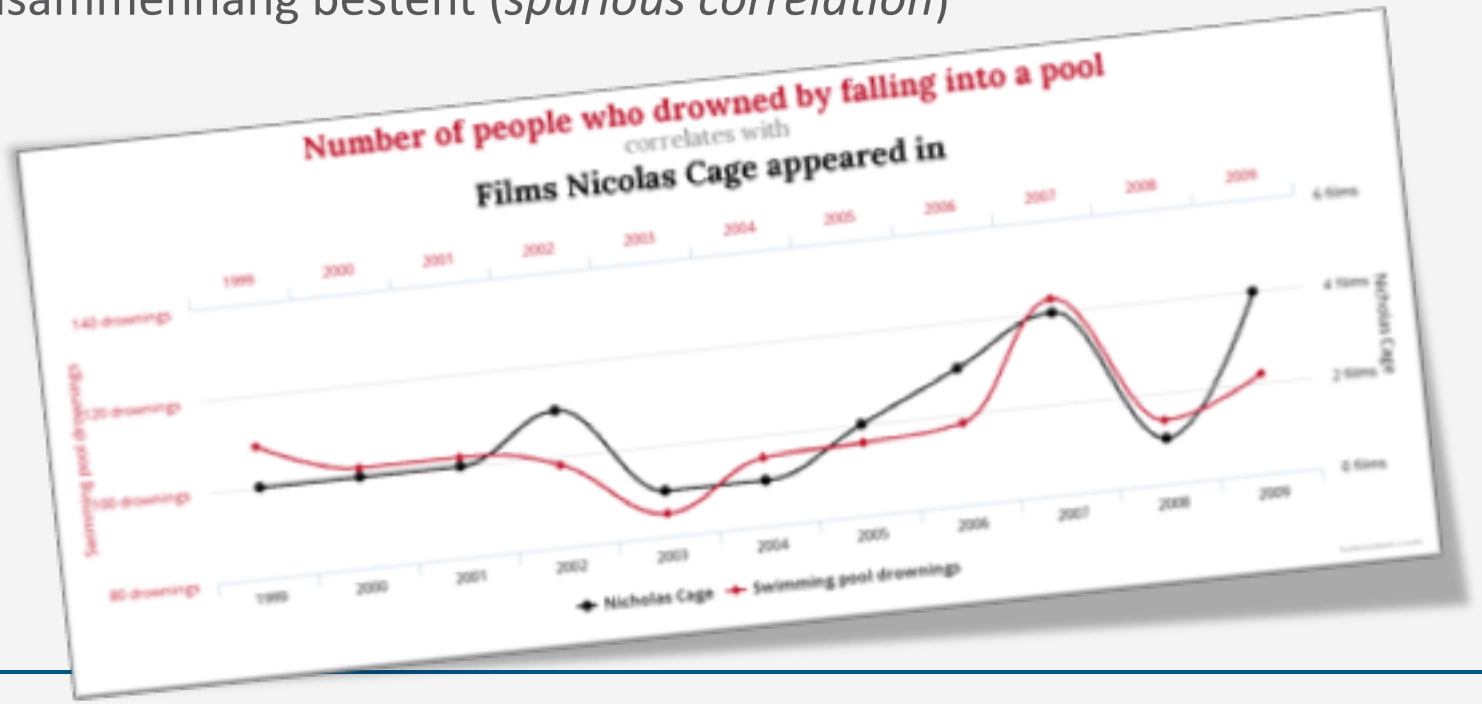


Abbildung: Korrelation: 66.6% (r=0.666004), https://www.tylervigen.com/spurious-correlations

Vorsicht: Korrelation vs. Kausalität

- Kausaler Zusammenhang unbekannt!
  - Kann vorhanden sein und bei der Interpretation der Daten helfen
  - Kann einen aber auch zu falschen Schlüssen verleiten
  - Beispiele
    - Zwei Variablen: Cholesterinwerte und Herzerkrankung
      - Positiv korreliert: Cholesterinwerte höher, häufiger Herzerkrankung vorhanden
      - Verbindung zwischen Essgewohnheiten und Herzerkrankungen komplex, aber wahrscheinlich
    - Zwei Variablen: Verkäufe von Sonnenbrillen und Hautkrebsvorkommen
      - Positiv korreliert, aber erscheint albern, dass Sonnenbrillenkäufe Hautkrebs verursachen
      - Dritter Einflussfaktor: Klima, welches dafür sorgt, dass Menschen sich der Sonne aussetzen
  - Expertenwissen kann helfen, kausale Zusammenhänge zu bestimmen

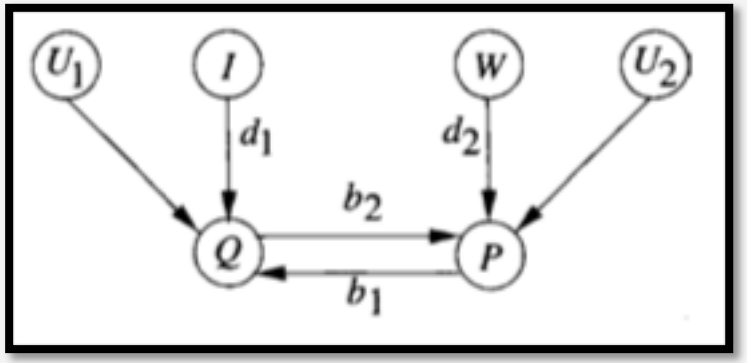
Abbildung: Korrelation: 66.6% (r=0.666004), https://www.tylervigen.com/spurious-correlations

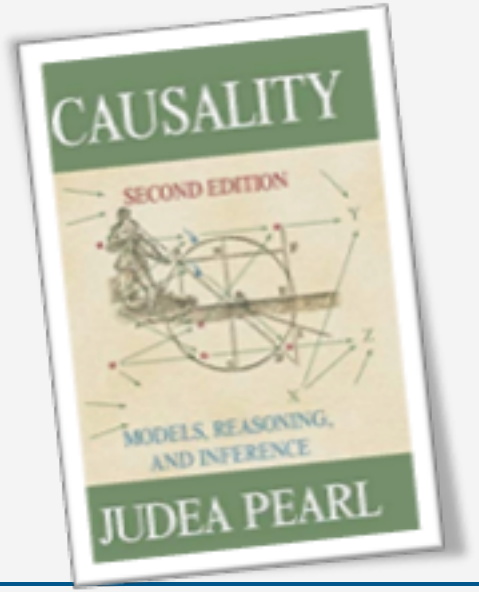
Vorsicht: Korrelation vs. Kausalität

- Problem: Auf Kausalität schwer zu prüfen
  - Wenn A B verursacht, kann B nicht ohne A auftreten
    - Krater (B) treten nur auf, wenn ein Asteroid (A) ihn verursacht hat durch eine Crash-Landung an der Stelle
  - Wenn wir nur A und B zusammen beobachten, kann das ein Indiz dafür sein
  - Es ist aber ungewiss, ob das Auftreten Ausdruck des kausalen Zusammenhangs ist oder ob B (ohne A) einfach noch nicht beobachtet wurde
    - Bzw. ob B durch etwas anderes, nicht beobachtetes hervorgerufen werden kann
- Möglichkeit: Hypothesentests einsetzen, um kausale Zusammenhänge zu prüfen
  - Quantifizieren, ob es genug Evidenz zur Unterstützung einer Hypothese gibt
- Negativbeispiele würden helfen, sind aber selten vorhanden

Vorsicht: Korrelation vs. Kausalität

- Forschungsgebiet: *Kausale Inferenz*
  - Untersuchung von Ursache-Effekt-Beziehungen
  - Modellierung wahrscheinlichkeitsbasiert
    - Structural equation models als mathematische Darstellung
    - Graph-basierte Darstellungen an BNs angelehnt
  - Fragestellungen
    - Was ist Ursache, was ist Wirkung? → Richtung der Pfeile
    - Gibt es einen spürbaren Effekt bei der Durchführung einer Handlung im Vergleich zur Nicht-Durchführung der Handlung? → *do-Kalkül, Interventionen*
    - Was wäre gewesen, wenn...? → *Counterfactuals*
  - Mehr zum Thema zum Beispiel von Judea Pearl
    - Tech Report: *Causal Inference in Statistics*
      https://ftp.cs.ucla.edu/pub/stat_ser/r350.pdf





Judea Pearl: Causality: Models, Reasoning, and Inference, 2009.
Abbildung ebenda (Fig. 1.5, Seite 28).

Zwischenzusammenfassung

- ML-basierte Methoden nicht der einzige Weg zum Ziel
- Generative Modelle erlauben beliebige Anfragen, diskriminative Modelle Klassifikationsanfragen
  - Diskriminative Modelle spezialisiert auf Klassifikation, deshalb meist besser bei der speziellen Aufgabe als generative Modelle, die alle Anfragetypen mit annehmbarer Genauigkeit beantworten können sollen
- Korrelation vs. Kausalität
  - Kausalität häufig Intuition hinter bedingten Wahrscheinlichkeiten, vor allem in BNs
    - Gemeinsamer Effekt, gemeinsame Ursache, etc.
  - Beim Lernen aus Daten nur Korrelation genutzt, die eine reine Scheinkorrelation sein kann
  - Zusätzlicher Aufwand oder Expertenwissen notwendig, um kausale Zusammenhänge zu bestätigen

Überblick: 5. Lernalgorithmen für episodische PGMs

A. *Überblick*
- Full Bayesian Learning, MAP Learning, Maximum Likelihood (ML) Learning

B. *ML-basiertes Parameter Lernen*
- In BNs: ML (vollständige Daten), Expectation-Maximisation (EM; nicht-vollständige Daten)
- In Faktormodellen: Iterative Proportional Fitting (IPF), EM-IPF
- Anwendung: LDA

C. *ML-basiertes Struktur Lernen*
- Model Selection, Structural EM

D. *Abschlussbemerkungen*
- Alternativen, generative vs. diskriminative Modelle, Korrelation vs. Kausalität

→ Sequentielle PGMs und Inferenz

Einordnung der Vorlesung: *Modell- und nutzenbasierter Agent*

- Nachfolgende Themen der Vorlesung
  2. Episodische PGMs
  3. Exakte Inferenz in episodischen PGMs
  4. Approximative Inferenz in episodischen PGMs
  5. Lernalgorithmen für episodische PGMs
  6. Sequentielle PGMs und Inferenz
  7. Entscheidungstheoretische PGMs