Text Models 1

Internet Analytics (COM-308)

Prof. Matthias Grossglauser
School of Computer and Communication Sciences

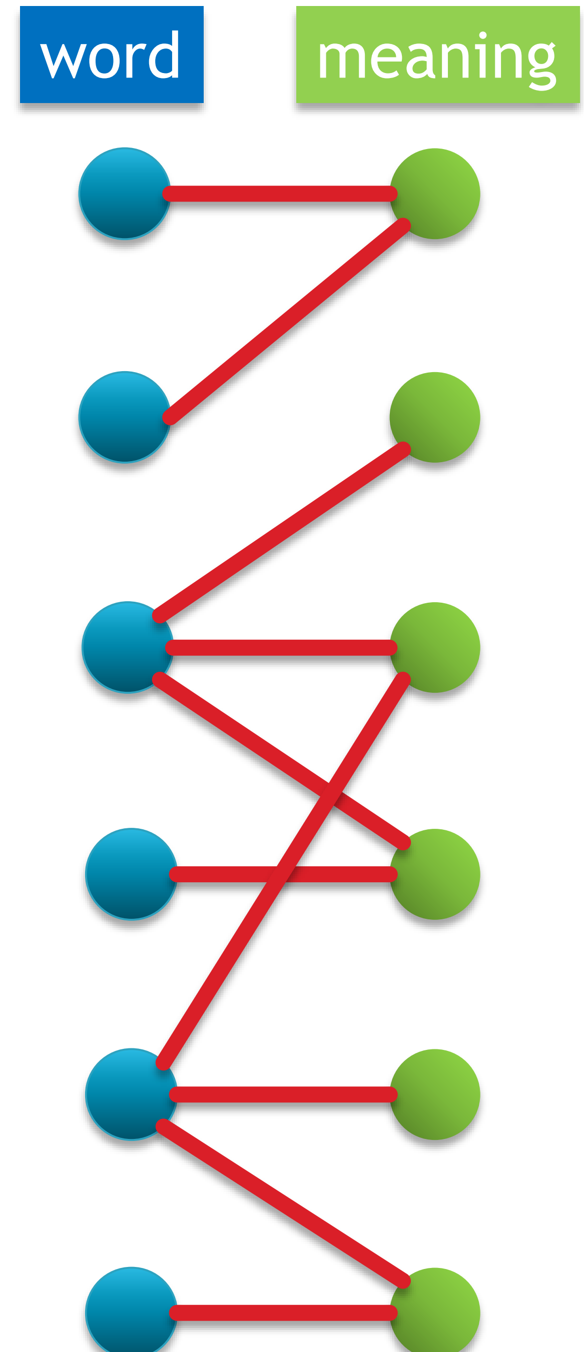
EPFL

Overview

- Probabilistic models for text
  - Recall naïve Bayes: words i.i.d. conditional on a class
- Word embeddings:
  - Find a compact representation (vector) that captures semantics of a word
  - Applications: sentiment analysis; machine translation,…
- Topic models:
  - Find a generative model for a set of documents in a corpus
  - Applications: summarization; information retrieval; dimensionality reduction; …
  - Detour: introduction to graphical models

Synonymy and polysemy

- Synonymy:
  - Different words with the same meaning
  - “car” and “automobile”
- Polysemy/homonymy:
  - One word with different meanings
  - “jaguar”: animal, brand of car
- For many applications, the meaning is more useful than the symbol
  - Information retrieval
  - Sentiment analysis
  - Dialog systems



Word embeddings

- Recall: vector space model for words

$$f(w) = [0,0, \dots, 1, \dots, 0]$$

One-hot at w's position in the dictionary

- Problem: this high-dimensional vector space has no relationship to meaning
  - Distance is the same between any two words
- Question: can we find a lower-dimensional representations $v(w)$ such that
  - Words with similar meanings are close (“coffee” and “tea”)
  - Relationships among words reflected in the space

Similar context → similar meaning

- "The box with the baits was under the stern of the skiff along with the club that was used to subdue the big fish when they were brought alongside."
- In a parallel universe, Hemingway might have (less elegantly) written "boat" or "craft", without fundamentally changing the sentence
- But the same sentence with "toaster" or "porosity" are unlikely to be observed in a corpus
- The context (set of nearby words) suggest meaning; exchangeable words in a given context share meaning

[Ernest Hemingway, The Old Man and the Sea (1952)]

Word2vec

- "The wooden hull of the ship crashed through the waves…" → lemming & stemming →

wood	hull	ship	crash	wave

- Word2vec key idea: learn a model of words and their context
- Two forms: for some window around word, **predict**
  - Continuous bag-of-words → predict center from context

wood	hull	?	crash	wave

  - Skip-gram → predict context from center

?	?	ship	?	?

Word2vec skip-gram model

$C(w)$	w	$C(w)$

- Data: sliding window → $X = \{(w, C(w))\}$

box	baits	under	stern	skiff
baits	under	stern	skiff	along
under	stern	skiff	along	club

- Skip-gram model: conditional probability of context $C(w)$ given word w, parametrized in some way (θ)
- We want to learn θ such that corpus probability

$$\prod_{w \in X} \prod_{c \in C(w)} p(c|w;\theta) = \prod_{(w,c) \in X} p(c|w;\theta)$$

is maximized

Parameterization of skip-gram

- Each word $w \in V$ and each context word $c \in V$ is represented by a (relatively) low-dimensional vector in $\mathbb{R}^d$ (d a few hundred)
  - $u_{"cat"}$ is the vector to represent $w = "cat"$
  - $v_{"cat"}$ is the vector to represent $c = "cat"$
  - θ is the collection of all these vectors
- Conditional probability of context given word:

$$p(c|w) = \frac{e^{u_w \cdot v_c}}{\sum_{c' \in V} e^{u_w \cdot v_{c'}}}$$

  - Note: this is the **soft-max** of $u_w \cdot v_c$ over all $c \in V$
- Q: why different vectors for middle words and context words?

Negative sampling

- Note: denominator $\sum_{c' \in V} e^{u_w \cdot v_{c'}}$ very costly to evaluate
  - $|V|$ in a large corpus 10-100s k
  - Need to compute for every (w, c) in a corpus
- Negative sampling: modified objective, cheaper to compute
  - Classification: given a pair (w, c), is it from the corpus?
  - Def: $p(D = 1|w, c) = \frac{1}{1+e^{-u_w \cdot v_c}}$: prob. “plausible text”
  - Maximize probability that all $(w, c) \in X$ are plausible

$$\max_{\theta} \sum_{(w,c)\in X} \log p(D = 1|w, c)$$

Negative sampling

- Problem: trivial solution: all vectors equal and large → $e^{-u_w \cdot v_c}$ extremely small
- Need penalty for false positives, ie, $D = 1$ for implausible text
- Approach: let X'= all pairs $(w, c) \notin X$
- New loss with negative sampling:

$$\max_{\theta} \sum_{(w,c) \in X} \log p(D = 1 | w, c) + \sum_{(w,c) \in X'} \log p(D = 0 | w, c)$$

$$\max_{\theta} \sum_{(w,c) \in X} \log \sigma(u_w \cdot v_c) + \sum_{(w,c) \in X'} \log \sigma(-u_w \cdot v_c)$$

  - Note: $\sigma(x) = \frac{1}{1+e^{-x}}$ is called the logistic function ("S-curve")

Negative sampling

- Problem: X' is very large!
- Idea: instead of enumerating entire X', just sample from it
- For every positive sample $(w, c) \in X$, add k random negative samples $(w, c') \notin X$ for the second term
  - This is much cheaper than to enumerate X': just generate (w, c') and check $\in X$; if yes, repeat
- Optimized with SGD
- Word2vec has some additional heuristics:
  - Biased negative sampling: favor more frequent c' in corpus
  - Adaptive window size
  - Rare word pruning

Word2vec: properties and results

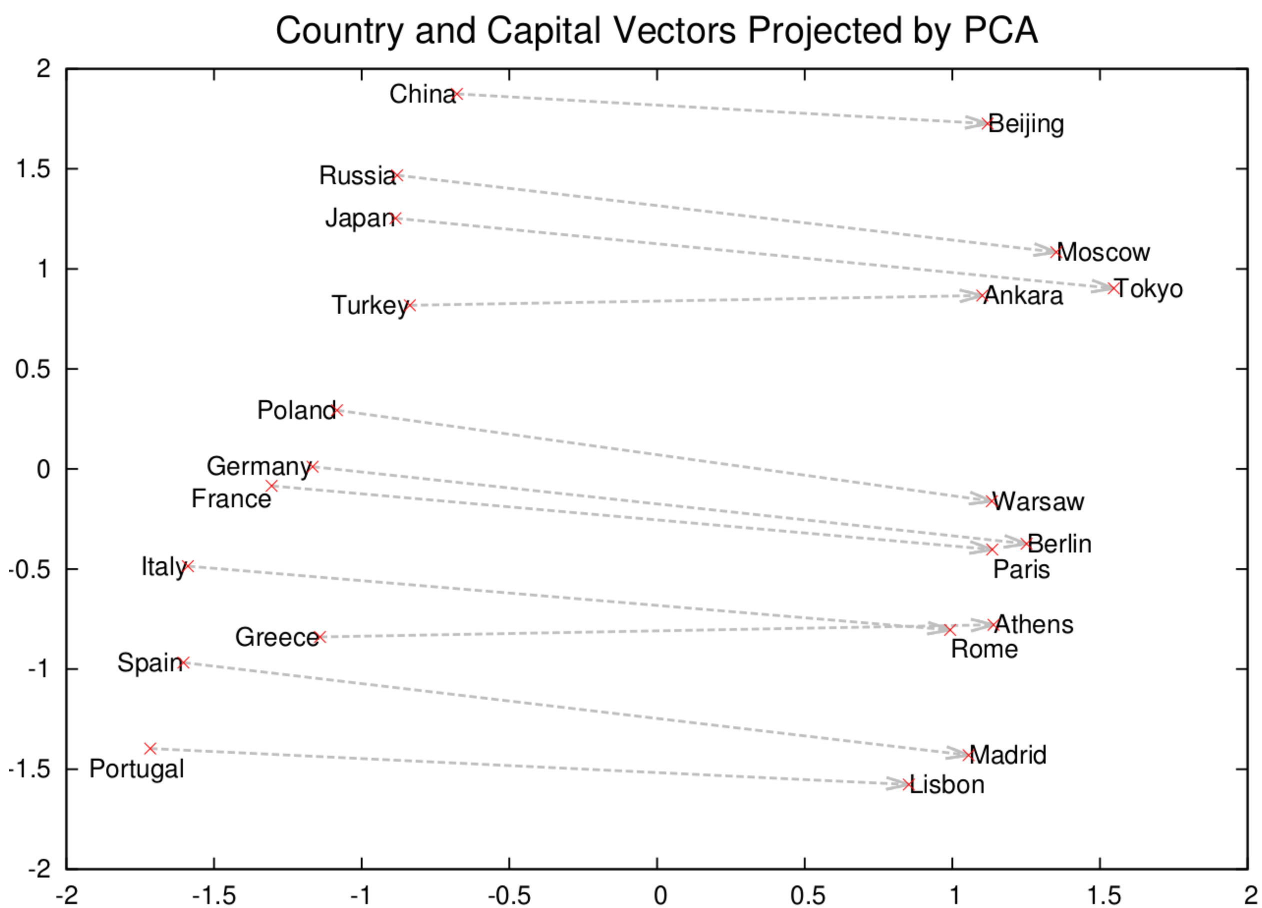
- Similarity: e.g., 8 nearest neighbors (in cosine distance $u_i \cdot u_j / \sqrt{(u_i \cdot u_i)(u_j \cdot u_j)}$) of “Sweden”:

Norway	0.76
Denmark	0.72
Finland	0.62
Switzerland	0.59
Belgium	0.59
Netherlands	0.57
Iceland	0.56
Estonia	0.55

[pathmind.com]

Word2vec: properties and results

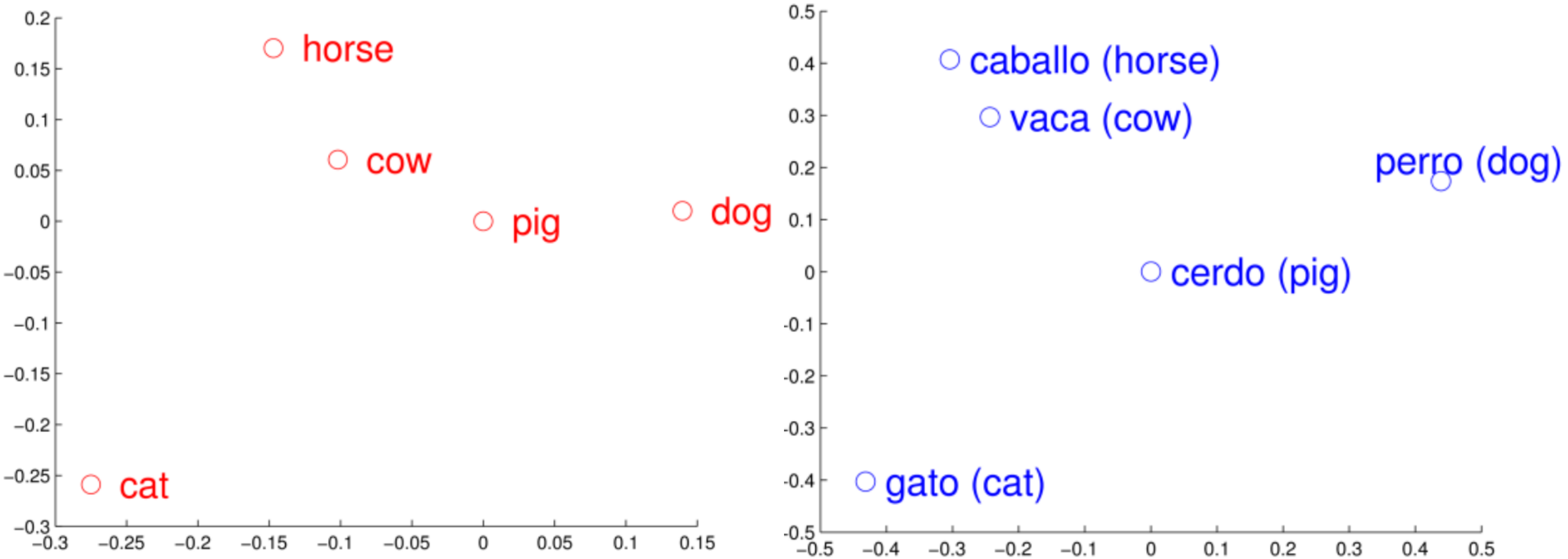
- Learning associations:
  - E.g.: NN of (**man-king+queen**) is **woman**



[Mikolov, Sutskever, Chen, Corrado, Dean: Distributed Representations of Words and Phrases and their Compositionality, NIPS 2013]

Word2vec: properties and results

- Vectors learned in different languages share similar relationships for same concepts:

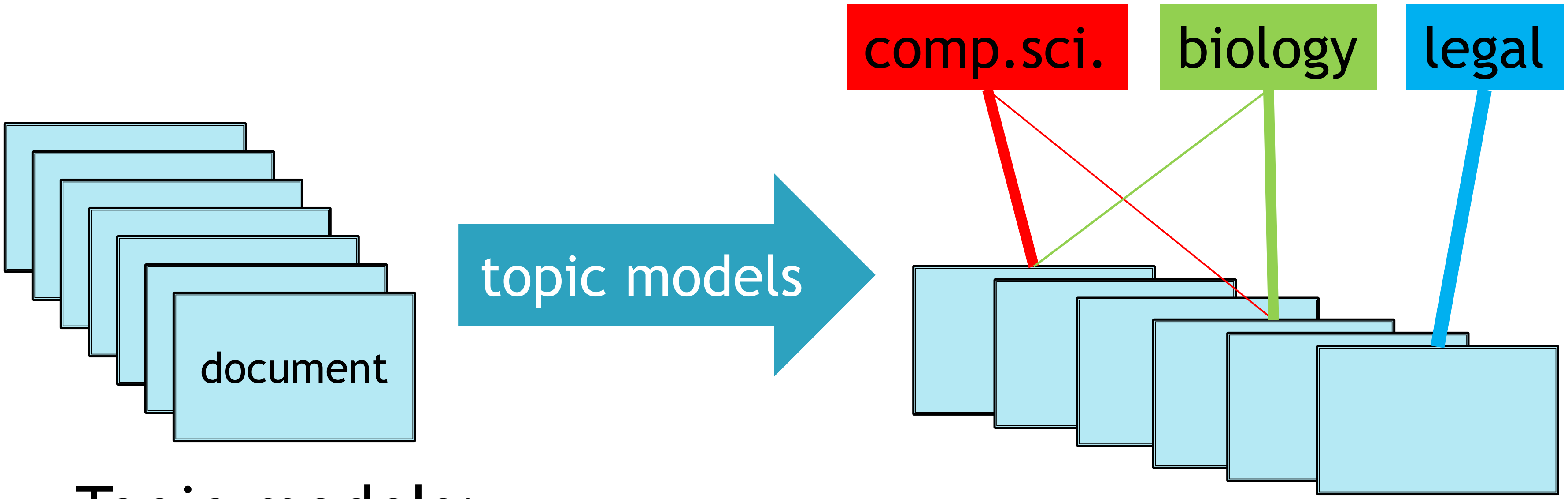


- Important in machine translation

[Mikolov, Lee, Sutskever: Exploiting Similarities among Languages for Machine Translation, 2013]

Topic models

- Without a query, how to describe a corpus?



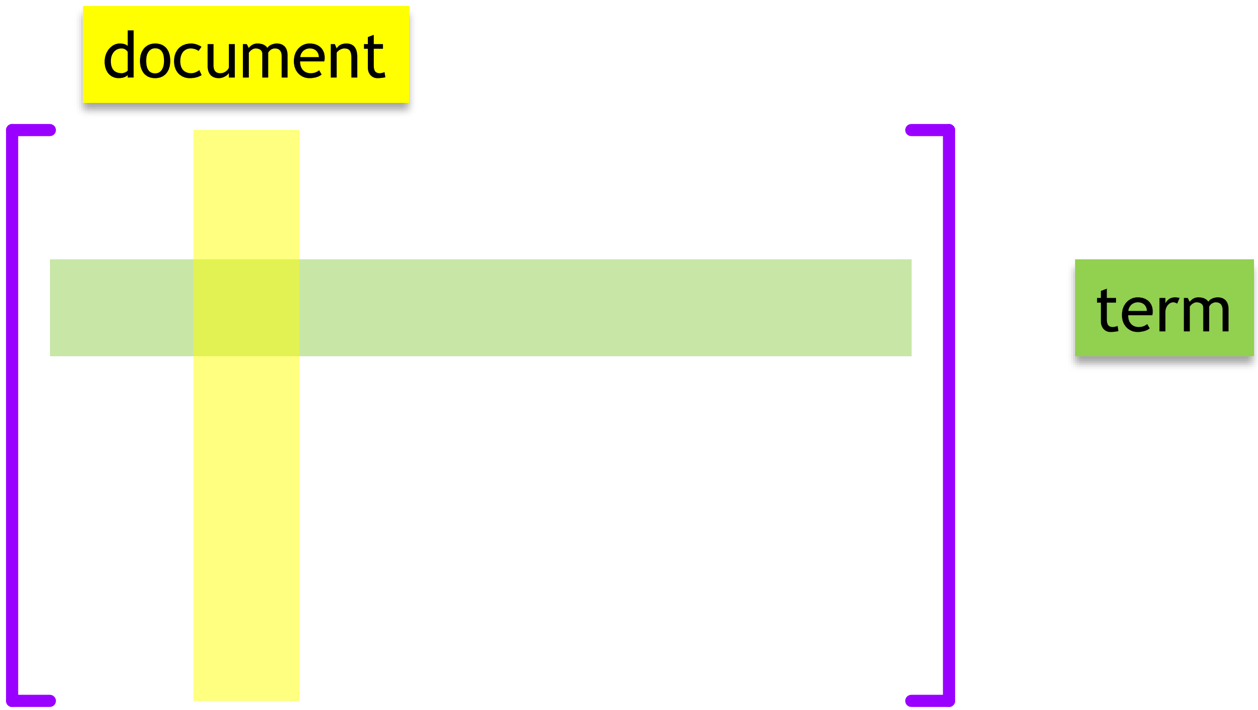
- Topic models:
  - We see the words of docs, but we want to classify the meanings of docs
  - Ambiguity of individual words – but many words per doc helps!
  - Generalization of naïve Bayes text model

Topic models

- Document classification
- Supervised: training set with known classes
  - Generalization of binary classification (spam/not spam)
- Unsupervised: need to identify sensible topic classes by comparing documents
- Assumptions:
  - Number of words per document $\gg 1$
  - Number of topics $\ll$ number of documents
- Examples:
  - News articles: topics = {countries, business, politics, celebrity, …}
  - Scientific literature: {physics, mathematics, engineering, chemistry, life sciences,…}

Approach 1: Latent Semantic Indexing (LSI)

- Synonymous: Latent Semantic Analysis (LSA)
- Starting point: TF-IDF matrix of corpus
- $X =$



- Remember: high TF-IDF means "term that is rare overall, but prominent in this doc"

SVD of TF-IDF matrix

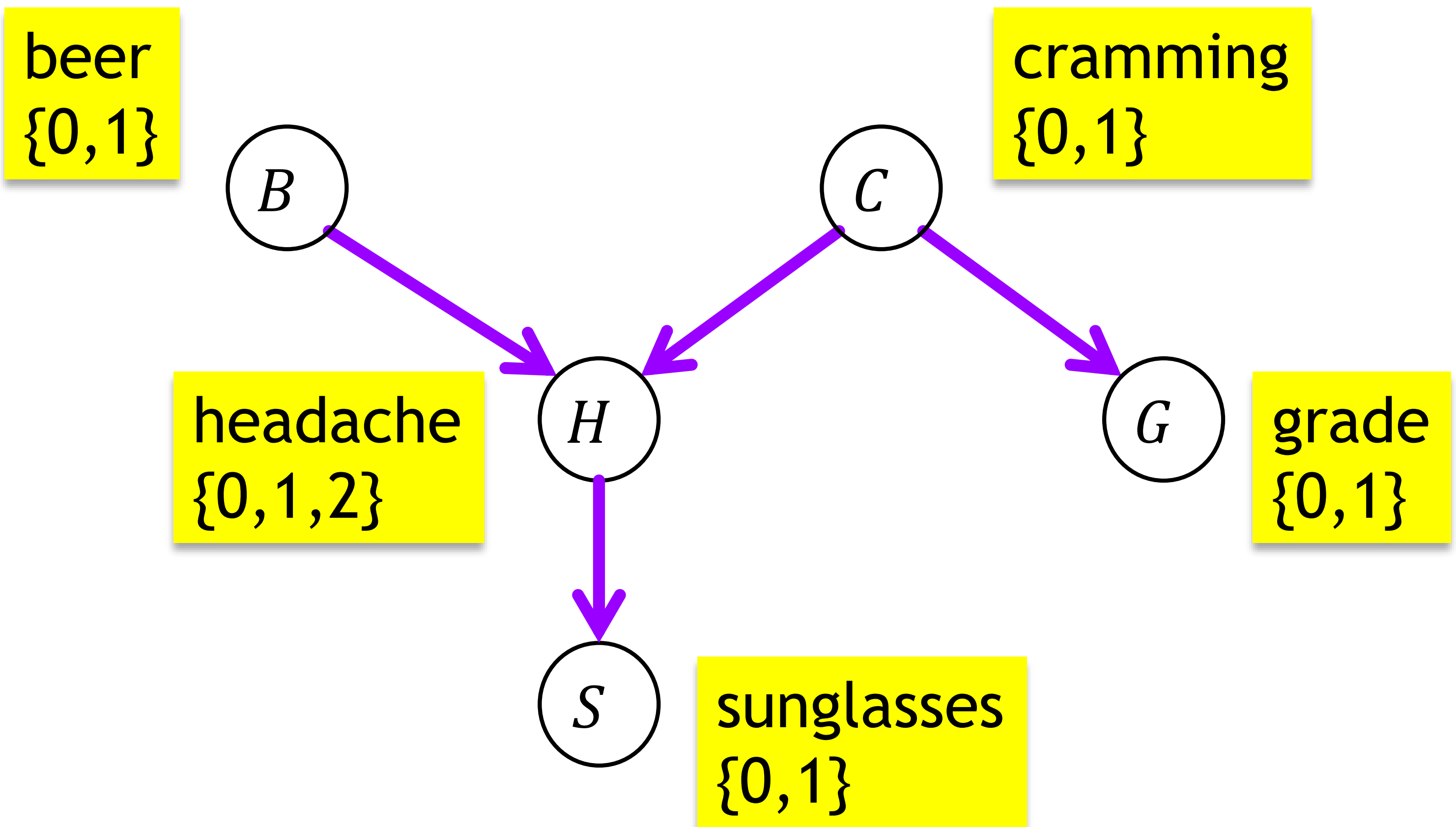
- Latent factors: “topics”
- Typically 100-300
- Should bunch together synonyms
- Should separate homonyms
- Critique:
  - Heuristic, no clean statistical foundation
  - Sometimes difficult to interpret results
  - Modern approaches based on probabilistic models:
    - better performance
    - better interpretability
    - generative

Gentle introduction to graphical models

- Modeling a multivariate distribution
- Example: insights from an expert:
  - “Drinking too much beer can result in headaches”
  - “Studying too much can cause headaches as well”
  - “To get a good grade, one must study”
  - “Wearing sunglasses tempers the pain of a headache”
- How to translate this into a probabilistic model?
  - Random variables
  - Dependencies?
    - Option: define/learn full joint distribution → many parameters, memory-intensive, hard to learn
    - Option: encode «causal structure» into model

Bayesian Network

- Edges = “direct” influence



Bayesian Network

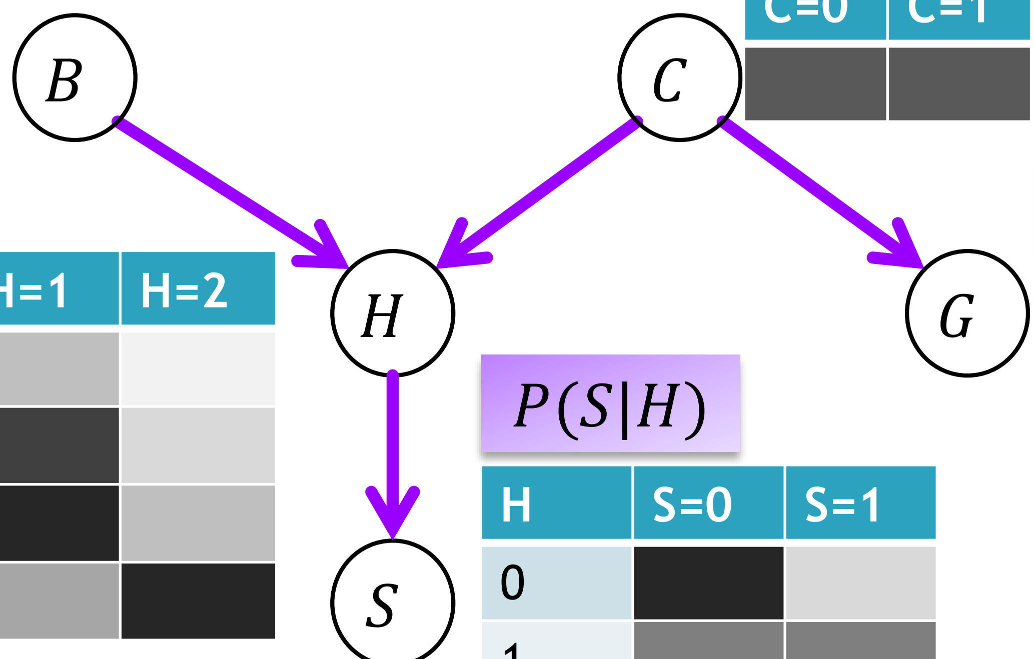
- One conditional distribution per node → full joint distribution

$P(B)$

B=0	B=1

$P(C)$

C=0	C=1



$P(H|B,C)$

(B,C)	H=0	H=1	H=2
(0,0)			
(0,1)			
(1,0)			
(1,1)			

$P(G|C)$

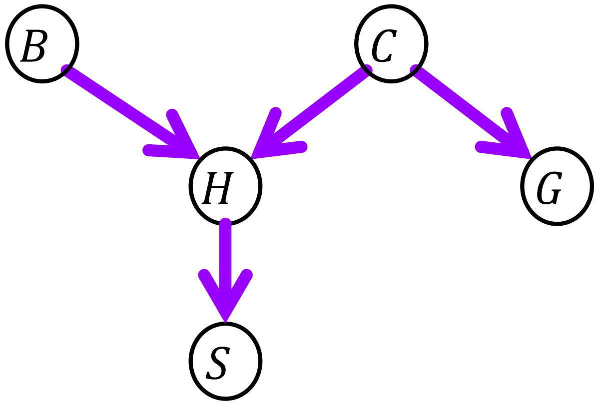
C	G=0	G=1
0		
1		

$P(S|H)$

H	S=0	S=1
0		
1		
2		

Full joint dist.: 47 params
BN: 1+1+8+3+2=15 params

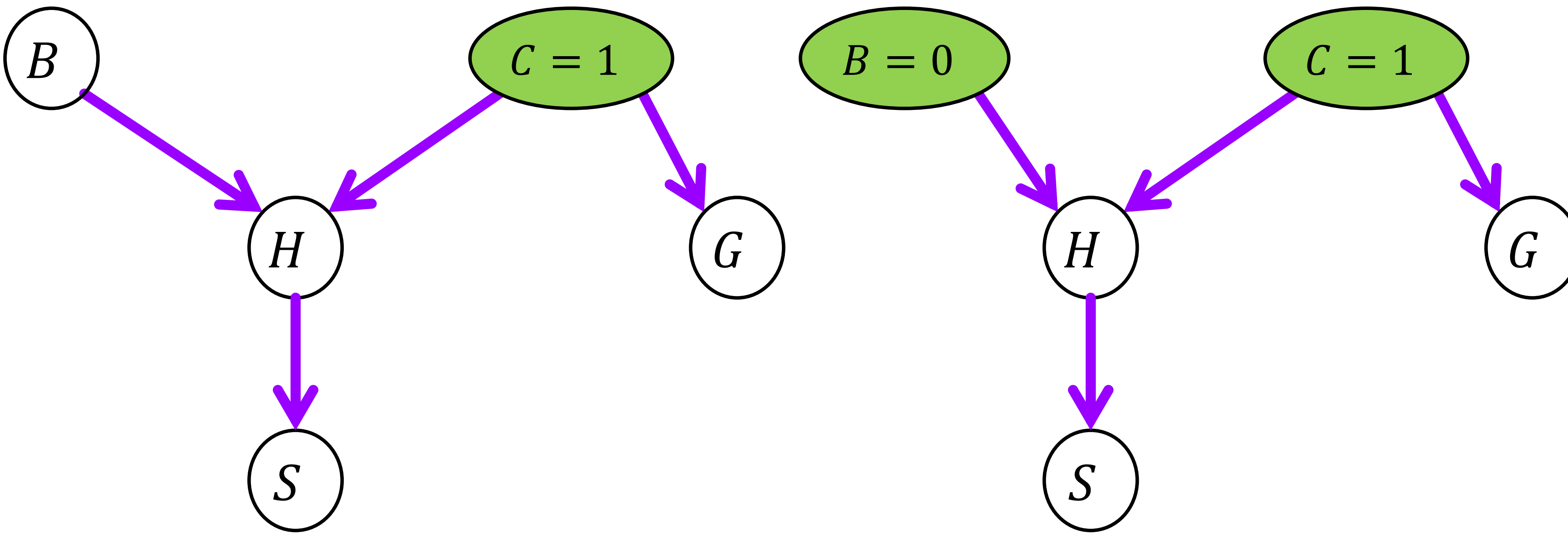
Joint distribution from CPDs



- Joint distribution from chain rule
- $P(b, c, h, g, s) =$
- $= P(c, h, g, s|b)P(b) =$
- $= P(h, g, s|b, c)P(c|b)P(b) =$
- $= P(h, s|b, c)P(g|b, c)P(c)P(b) =$
- $= P(s|b, c, h)P(h|b, c)P(g|c)P(c)P(b) =$
- $= P(s|h)P(h|b, c)P(g|c)P(c)P(b)$
- Joint distribution = product of all individual per-node factors
  - With the joint distribution, everything else follows: all marginal and conditional distributions we could want

Types of reasoning

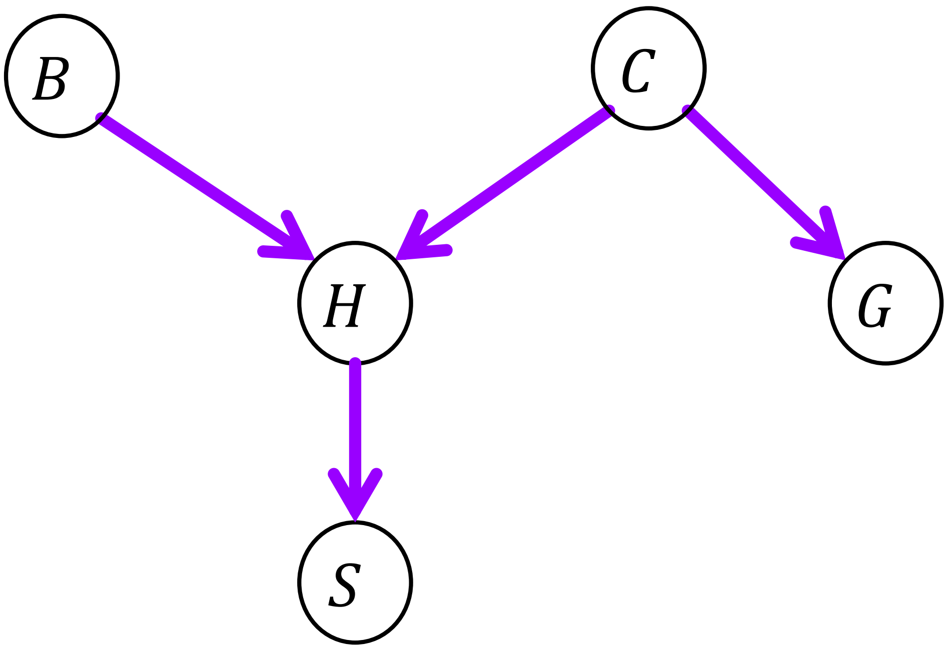
- Causal reasoning / prediction: downstream flow of influence

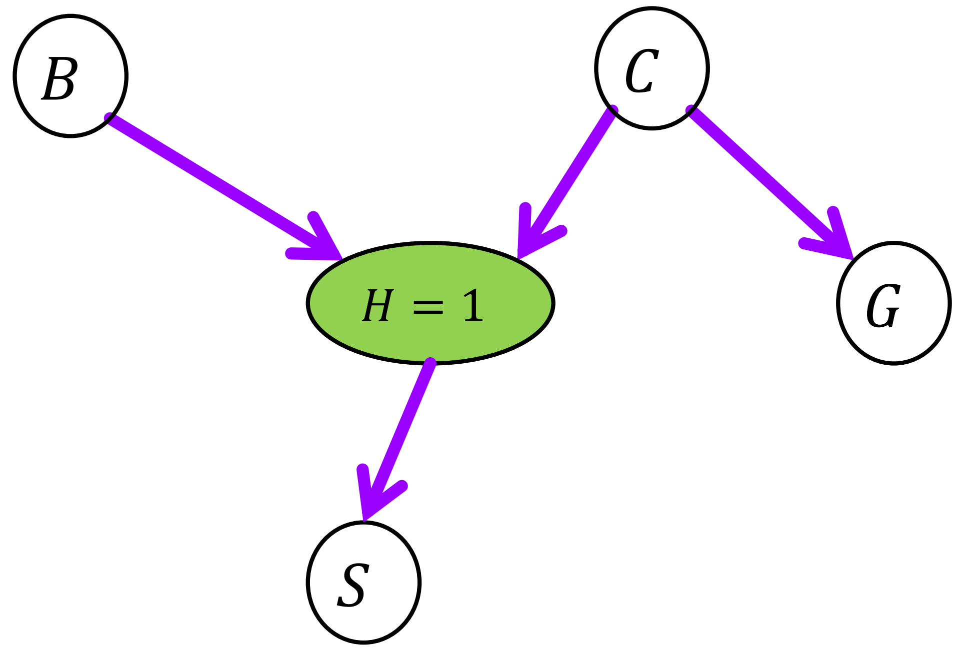


$$P(S = 1|C = 1) > P(S = 1|B = 0, C = 1)$$

Types of reasoning

- Evidential reasoning / explanation: upstream flow of influence

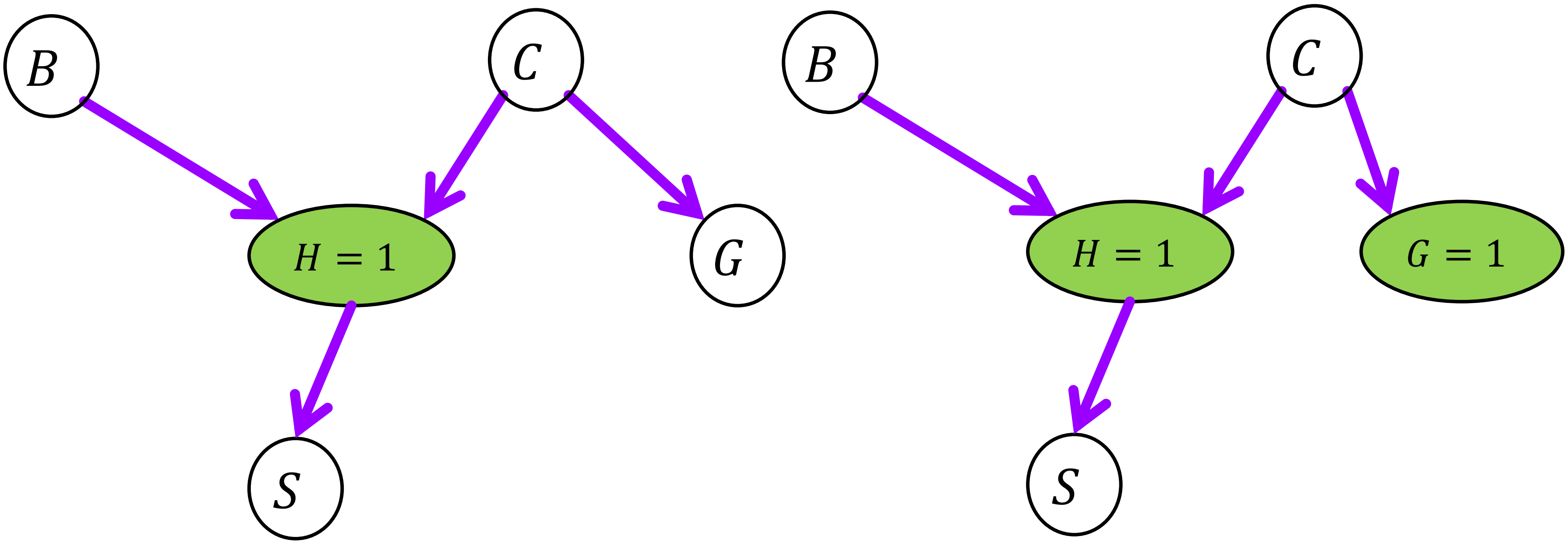




$$P(B=1|H=1) > P(B=1)$$
$$P(C=1|H=1) > P(C=1)$$

Types of reasoning

- Intercausal reasoning: combination of upstream/downstream

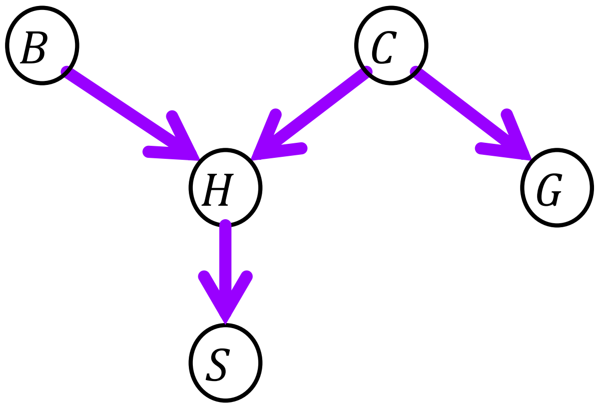


$$P(B=1|H=1) > P(B=1|H=1,G=1)$$

Explaining away: the "good grade" explains the "headache", making the possible cause "beer" less likely

Basic independencies in BNs

- Example: "the wearing of sunglasses depends only on the presence and strength of a headache"
  - Formally: $(S \perp B, C, G | H)$



- Also:
  - $(G \perp B, H, S | C)$
  - $(B \perp C)$
  - $(H \perp G | B, C)$
  - $(B \perp C, G)$
- How about $(H \perp S, G | B, C)$?
  - No! Intuition: suppose we know $B = 0$ and $C = 1$; then the guess for S changes according to $H = 0,1,2$

Basic conditional independencies in BNs

- Bayesian Network: directed acyclic graph (DAG) G
- Def: $Pa(X_i)$=parents of X_i in G
- Def: $ND(X_i)$=non-descendents of X_i in G
- Property: G has the following local independence properties:
  - For each X_i:

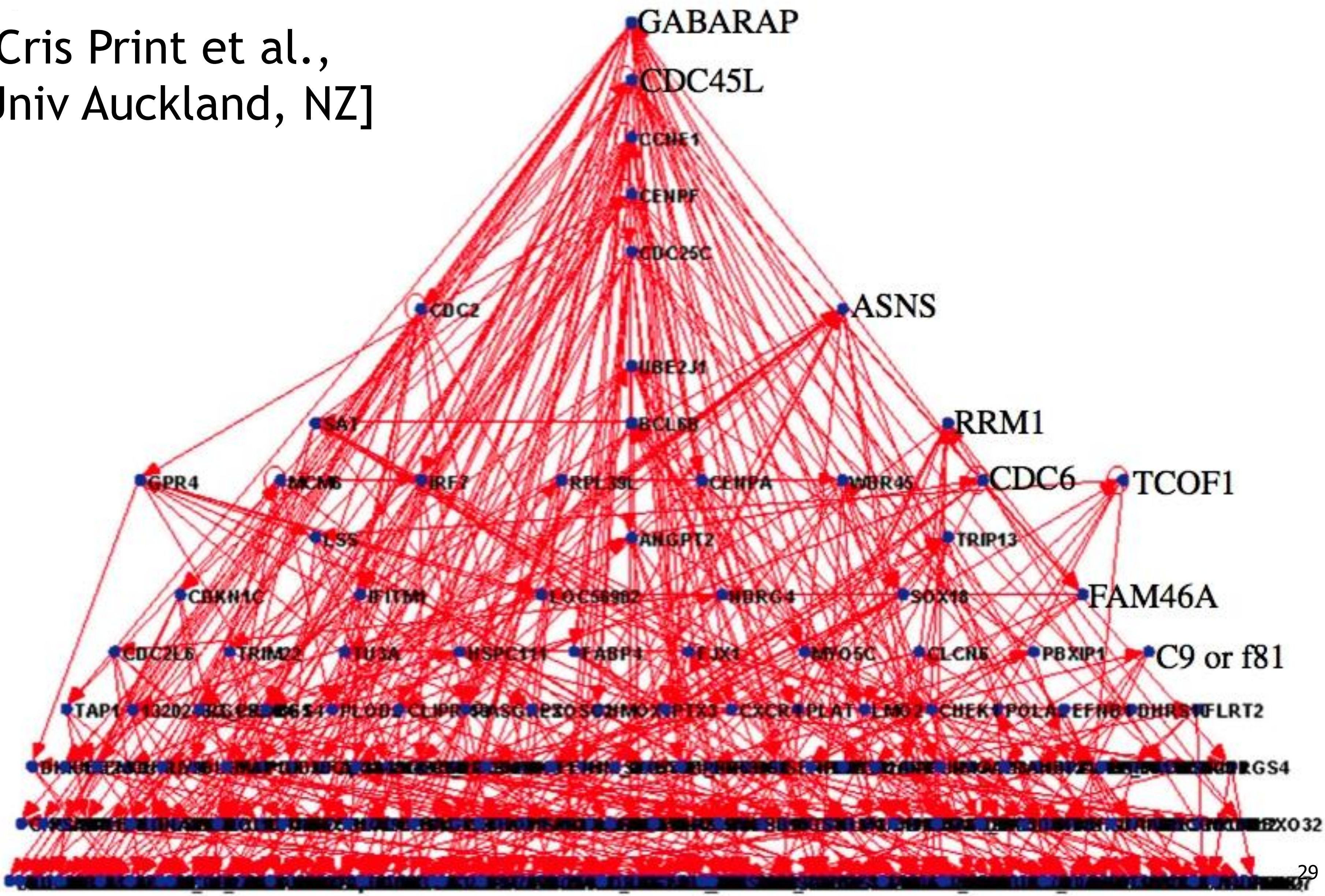
$$(X_i \perp ND(X_i) \mid Pa(X_i))$$

Bayesian Networks: recap

- Defines a multivariate probability distribution
- Models direct causal influences
  - This comes from expert knowledge, underlying mechanisms, data about the problem,…
- In practice: as sparse as possible
- Conditional independence properties as graph (path) properties
- Inference:
  - Observe some variables (observables)
  - Obtain conditional distribution of some other variables of interest → estimate
  - Some variables we do not care (latent)

Example: gene network

[Cris Print et al., Univ Auckland, NZ]



Computational challenge in large models

- Suppose G large; a few variables $Y \subset X$ are observed, $Z = X \backslash Y$ are not observed
- Want to estimate $P(Z_{573}|Y)$, where Z_{573} is e.g. one of many diseases in a medical diagnostic system
- Need to compute $P(Z_{573}|Y) =$

$$\sum_{Z_1, Z_2, \ldots, Z_{572}, Z_{574}, \ldots} P(Z_1, Z_2, \ldots, Z_{572}, Z_{573}, Z_{574}, \ldots |Y)$$

  - Very costly to marginalize out all other latent variables
- Inference methods:
  - Exact
  - Markov Chain Monte Carlo (MCMC)
  - Variational inference

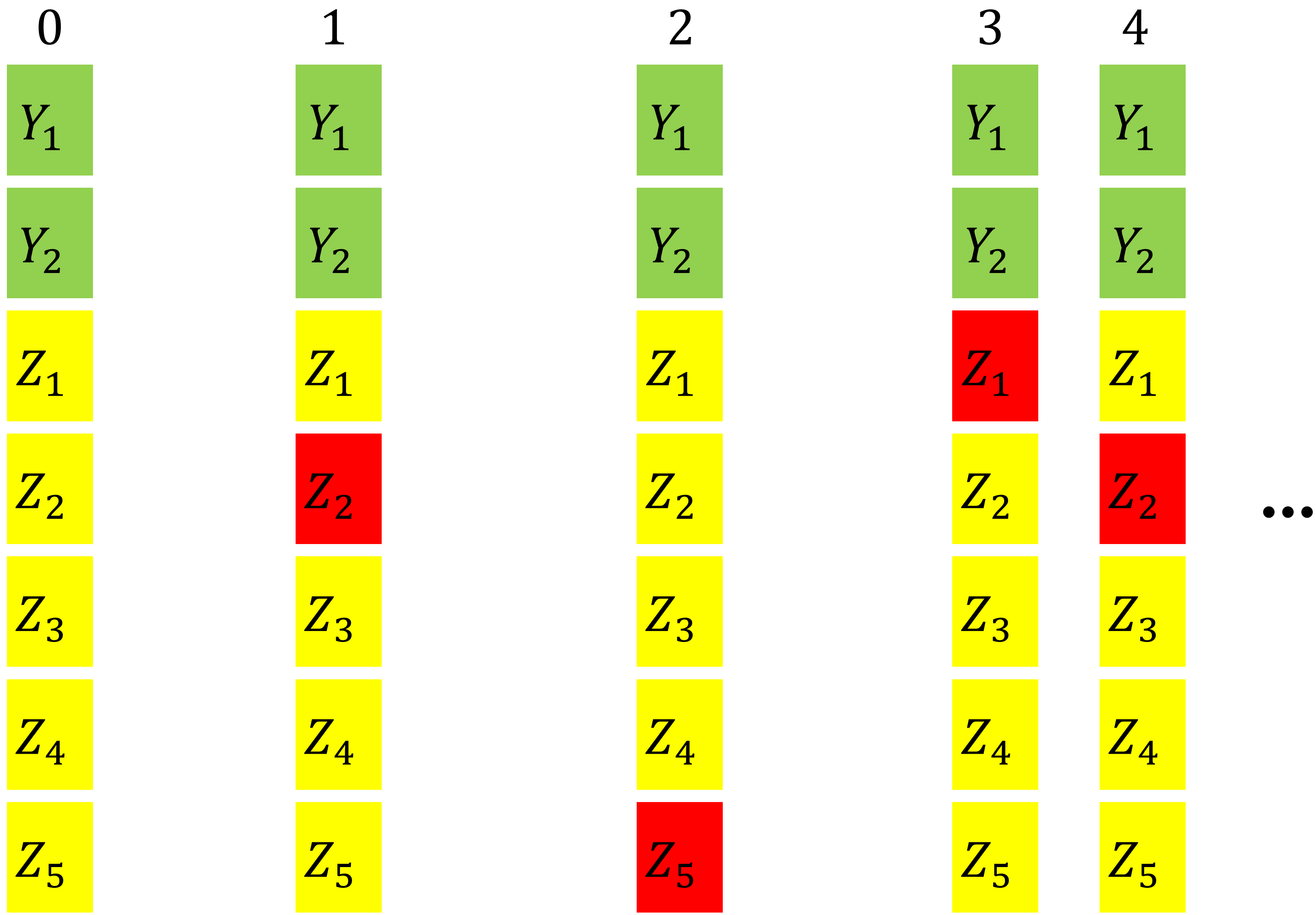
Inference: MCMC

- Probabilistic model:
  - Joint distribution $P(x)$ over $X = (X_1, X_2, \dots, X_n) = (Z, Y)$
  - $Y = (Y_1, \dots, Y_a)$: observed variables
  - $Z = (Z_1, \dots, Z_b)$: unobserved/latent variables
- Goal:
  - Obtain samples from $P(Z|Y = y)$

Gibbs sampling

- Markov chain Q:
  - State of Q is a variable assignment $\boldsymbol{Z}$
  - Pick K uniformly from $\{1, \dots, b\}$ (or cycle through)
  - Sample Z_K from $P(Z_K | Z_1, Z_2, \dots, Z_{K-1}, Z_{K+1}, \dots, Z_b, Y = y)$
  - Repeat
- Possible transition in Q:
  - Def: $\boldsymbol{z}' \sim_k \boldsymbol{z}$ if $\boldsymbol{z}' = (z_1, z_2, \dots, z_{K-1}, *, z_{K+1}, \dots, z_b)$,i.e., equal to $\boldsymbol{z}$ except at position k
  - Transition $\boldsymbol{z} \to \boldsymbol{z}'$ only possible for $\boldsymbol{z}' \sim_k \boldsymbol{z}$ for some k

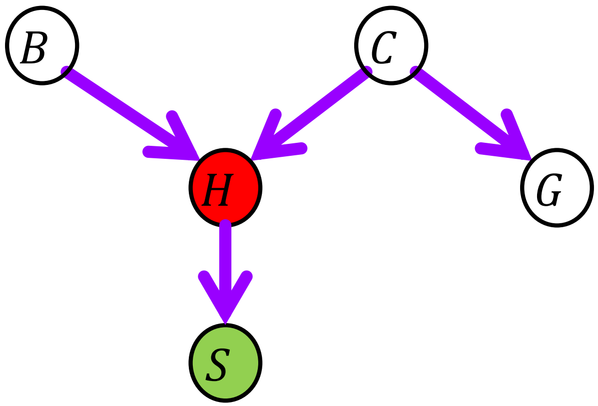
Gibbs sampling: illustration



$Z(1) \sim_2 Z(0)$ $\quad$ $Z(2) \sim_5 Z(1)$

Gibbs sampling for BNs: example

- Resampling variable H conditional on S
- $P(H|B,C,G,S) =$
- $= \dfrac{P(H,B,C,G,S)}{P(B,C,G,S)} =$
- $= \dfrac{P(H,B,C,G,S)}{\sum_H P(H,B,C,G,S)} =$
- $= \dfrac{P(B)P(C)P(H|B,C)P(G|C)P(S|H)}{\sum_{H'} P(B)P(C)P(H'|B,C)P(G|C)P(S|H')} =$
- $= \dfrac{P(H|B,C)P(S|H)}{\sum_{H'} P(H'|B,C)P(S|H')}$



Sampling from a variable only involves factors (CPDs) "touched" by this variable!

Gibbs sampling

- Claim:
  - Q is a reversible MC with stationary distribution
$$\pi(\boldsymbol{z}) = P(\boldsymbol{Z} = \boldsymbol{z} | Y = y)$$
  - Interpretation: run the MC Q and collect large # of samples of $\boldsymbol{Z}|Y = y$, then compute whatever statistic needed: mean, moments, confidence intervals, etc.
  - But: samples are correlated!
- Reminder:
  - An ergodic MC (irreducible, aperiodic, pos-recurrent) MC has a single stationary distribution π
  - Ergodic theorem: temporal averages → ensemble expectations
  - Reversible MC: if Q is ergodic and we can find a $\pi(.)$ such that for all $\boldsymbol{z}, \boldsymbol{z}'$, $\pi(\boldsymbol{z})Q(\boldsymbol{z}, \boldsymbol{z}') = \pi(\boldsymbol{z}')Q(\boldsymbol{z}', \boldsymbol{z})$, then $\pi(.)$ is the stationary distribution

Transition matrix of Q

- Write $P(Z_K|Z_1,\dots,Z_{K-1},Z_{K+1},\dots,Z_b,y)\times$
  $\times P(Z_1,\dots,Z_{K-1},Z_{K+1},\dots,Z_b|y) =$
  $= P(Z_1,\dots,Z_b|y)$

Does not depend on Z_K

- Transition matrix:

$$Q(\mathbf{z},\mathbf{z}') = \begin{cases} \dfrac{P(Z_K = z_K'|z_1,\dots,z_{K-1},z_{K+1},\dots,z_b,y)}{b} & \mathbf{z}'\sim_k\mathbf{z} \text{ for some } k \\ 0 & \text{otherwise} \end{cases}$$

$$Q(\mathbf{z},\mathbf{z}') = \begin{cases} \dfrac{P(\mathbf{Z}=\mathbf{z}'|y)}{b\,\sum_{\mathbf{z}''\sim_k\mathbf{z}} P(\mathbf{Z}=\mathbf{z}''|y)} & \mathbf{z}'\sim_k\mathbf{z} \\ 0 & \text{otherwise} \end{cases}$$

Gibbs sampling: proof

- Proof:
  - $\pi(z)Q(z,z') =$
  - $= P(Z=z|y)Q(z,z') =$
  - $= \frac{P(Z=z|y)P(Z=z'|y)}{b\sum_{z''\sim_k z} P(Z=z''|y)} =$
  - $= \frac{P(Z=z'|y)P(Z=z|y)}{b\sum_{z''\sim_k z'} P(Z=z''|y)} =$
  - $= P(Z=z'|y)Q(z',z) =$
  - $= \pi(z')Q(z',z)$

Note: z and z' only differ at position k; therefore,
$z''\sim_k z \Leftrightarrow z''\sim_k z'$

Detailed balance equations
→ global balance equations
→ $\pi(z)$ is stationary distrib. of MC Q

Bayesian Network: key ideas

- Two functions:
  - Compact representation for a set of conditional independence assumptions among RVs
  - A data structure to encode a joint distribution compactly through its factors
- Flexibility: model does not specify observables
- Example: 100 binary RVs
  - Full joint distribution: $2^{100} - 1 \sim 10^{30}$ values
  - All independent: 100 values, but very limiting
  - In practice, much closer to «everything independent» than to «full joint distribution»
    - Tradeoff: compact representation & efficient inference, but still capture main dependencies
- Next week: topic models using graphical models

References

- [D. Koller, N. Friedman: Probabilistic Graphical Models, MIT Press, 2009]
- [Ch. D. Manning, P. Raghavan, H. Schütze: Introduction to Information Retrieval, Cambridge, 2008]
- [C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006]