Latent Variable Modeling: Tensor and Graphical Approaches

Anima Anandkumar

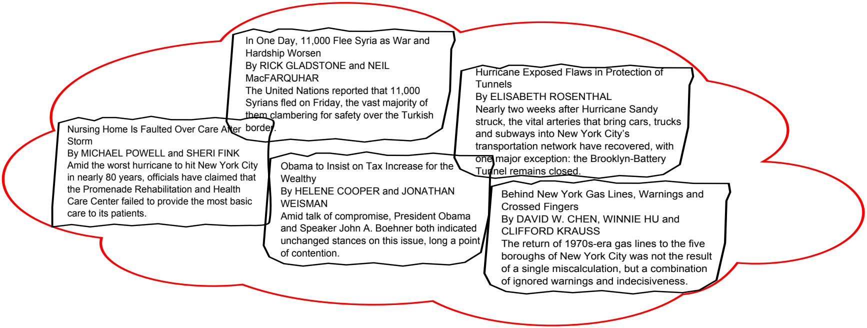
U.C. Irvine

Latent Variable Modeling

Goal: Discover hidden effects from observed measurements

Example: document modeling

- Observations: words. Hidden: topics.



Learning latent variable models: efficient methods and guarantees

Challenges and Approaches

Challenges: High-Dimensional Regime

- Sample and Computational complexities
- Identifiability: when can hidden variables be discovered?

Challenges and Approaches

Challenges: High-Dimensional Regime

- Sample and Computational complexities
- Identifiability: when can hidden variables be discovered?

Our Approach: Two Perspectives

Challenges and Approaches

Challenges: High-Dimensional Regime

- Sample and Computational complexities
- Identifiability: when can hidden variables be discovered?

Our Approach: Two Perspectives

Method of Moments

- Hidden choice variable and observed samples
- Inverse moment method: solve equations relating hidden variable to observed moments
- Low order tensor form and efficient decomposition methods

Challenges and Approaches

Challenges: High-Dimensional Regime

- Sample and Computational complexities
- Identifiability: when can hidden variables be discovered?

Our Approach: Two Perspectives

Method of Moments

- Hidden choice variable and observed samples
- Inverse moment method: solve equations relating hidden variable to observed moments
- Low order tensor form and efficient decomposition methods

Graphical Modeling

- Qualitative: graph structure. Quantitative: interaction strengths.
- Markov relationships: graphs with long cycles and hidden variables.
- Greedy graph estimation method: efficient tradeoffs.

Results from Two Approches

Learning Mixture Models through Tensor Decomposition

Topic 1	Topic 2	Topic 3
bush	company	show
president	percent	book
government	million	women
official	companies	family
campaign	market	film
political	business	school
law	stock	look
leader	billion	home
george_bush	money	children
al_gore	cost	friend

- Top 10 words for three topics from NYTimes data set.

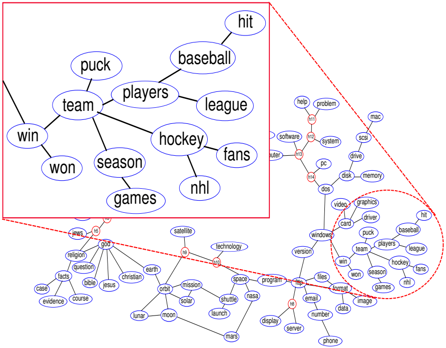
Results from Two Approches

Learning Mixture Models through Tensor Decomposition

Topic 1	Topic 2	Topic 3
bush	company	show
president	percent	book
government	million	women
official	companies	family
campaign	market	film
political	business	school
law	stock	look
leader	billion	home
george_bush	money	children
al_gore	cost	friend

- Top 10 words for three topics from NYTimes data set.

Graph Estimation Through Greedy Methods



- Graph: Topic-Word Relationships.

Other Motivating Applications

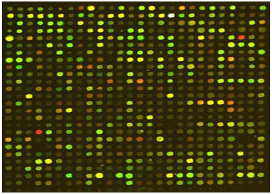
Social Network Modeling

- Community detection: Discovering hidden communities
- Dynamic network modeling: Predicting vertex co-presence

Other Motivating Applications

Social Network Modeling

- Community detection: Discovering hidden communities
- Dynamic network modeling: Predicting vertex co-presence



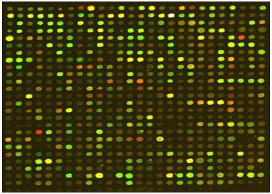
Bio-Informatics

- Modeling gene associations
- Hidden variables may be regulators that control groups of functionally similar genes

Other Motivating Applications

Social Network Modeling

- Community detection: Discovering hidden communities
- Dynamic network modeling: Predicting vertex co-presence



Bio-Informatics

- Modeling gene associations
- Hidden variables may be regulators that control groups of functionally similar genes

Computer Vision, Phylogenetics, Financial Modeling . . .

Outline

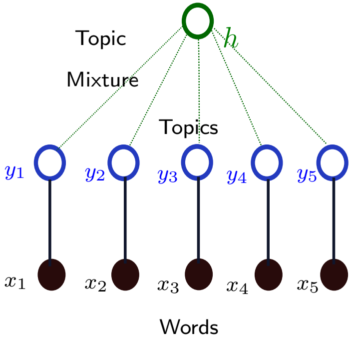
Warmup: Exchangeable Single Topic Models

Exchangeability

- Order of words does not matter
- Sufficient statistics: word counts
- DeFinetti's theorem: latent variable

Exchangeable Topic Models

- l words in a document $x_1, \ldots, x_l$.
- Document: topic mixture (draw of h).
- Word x_i generated from topic y_i.
- Exchangeability: $\boxed{x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp \ldots | h}$
- $\boxed{\Phi(i, j) := \mathbb{P}[x_m = i | y_m = j].}$



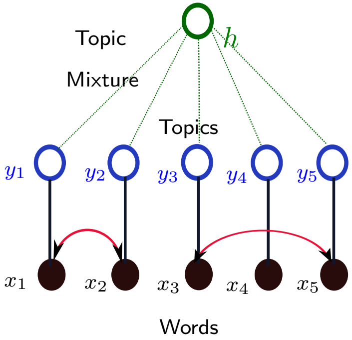
Warmup: Exchangeable Single Topic Models

Exchangeability

- Order of words does not matter
- Sufficient statistics: word counts
- DeFinetti's theorem: latent variable

Exchangeable Topic Models

- l words in a document $x_1, \ldots, x_l$.
- Document: topic mixture (draw of h).
- Word x_i generated from topic y_i.
- Exchangeability: $\boxed{x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp \ldots | h}$
- $\boxed{\Phi(i,j) := \mathbb{P}[x_m = i | y_m = j].}$



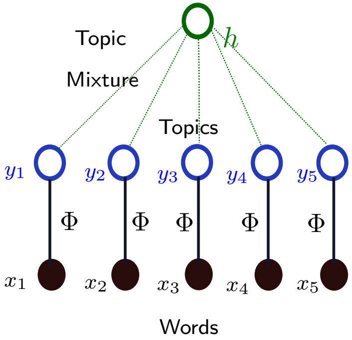
Warmup: Exchangeable Single Topic Models

Exchangeability

- Order of words does not matter
- Sufficient statistics: word counts
- DeFinetti's theorem: latent variable

Exchangeable Topic Models

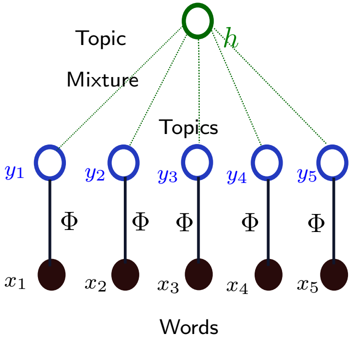
- l words in a document $x_1, \ldots, x_l$.
- Document: topic mixture (draw of h).
- Word x_i generated from topic y_i.
- Exchangeability: $\boxed{x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp \ldots | h}$
- $\boxed{\Phi(i,j) := \mathbb{P}[x_m = i | y_m = j].}$



Warmup: Exchangeable Single Topic Models

Exchangeability

- Order of words does not matter
- Sufficient statistics: word counts
- DeFinetti's theorem: latent variable



Exchangeable Topic Models

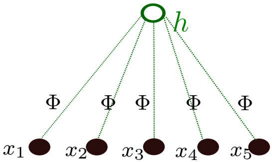
- l words in a document $x_1, \ldots, x_l$.
- Document: topic mixture (draw of h).
- Word x_i generated from topic y_i.
- Exchangeability: $\boxed{x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp \ldots | h}$
- $\boxed{\Phi(i,j) := \mathbb{P}[x_m = i | y_m = j].}$

Single topic model

- Each document has only one hidden topic: $y_i = h$.
- h is a discrete variable and let $\boxed{\lambda_i := \mathbb{P}[h = i].}$

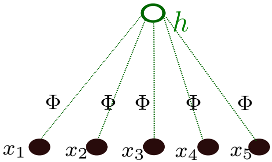
Form of Observed Moments

- $\boxed{\vec{\lambda} := [\mathbb{P}[h = i]]_i.}$
  $\boxed{\Phi(i, j) := \mathbb{P}[x_m = i | h = j].}$
- Learning: Loading matrix Φ and Vector $\vec{\lambda}$



Form of Observed Moments

- $\boxed{\vec{\lambda} := [\mathbb{P}[h = i]]_i.}$
  $\boxed{\Phi(i, j) := \mathbb{P}[x_m = i | h = j].}$
- Learning: Loading matrix Φ and Vector $\vec{\lambda}$

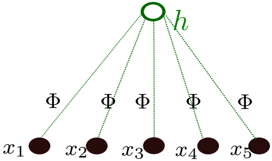


Pairwise Probability Matrix M_2

$$M_2(a, b) := \mathbb{P}(x_1 = a, x_2 = b) = \sum_r \lambda_r \Phi(a, r) \Phi(b, r)$$

Form of Observed Moments

- $\boxed{\vec{\lambda} := [\mathbb{P}[h = i]]_i.}$
  $\boxed{\Phi(i, j) := \mathbb{P}[x_m = i | h = j].}$
- Learning: Loading matrix Φ and Vector $\vec{\lambda}$



Pairwise Probability Matrix M_2

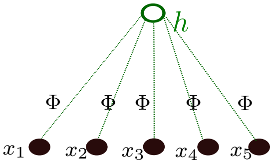
$$M_2(a, b) := \mathbb{P}(x_1 = a, x_2 = b) = \sum_r \lambda_r \Phi(a, r)\Phi(b, r)$$

Triples Probability Tensor M_3

$$M_3(a, b, c) := \mathbb{P}(x_1 = a, x_2 = b, x_3 = c) = \sum_r \lambda_r \Phi(a, r)\Phi(b, r)\Phi(c, r)$$

Form of Observed Moments

- $\boxed{\vec{\lambda} := [\mathbb{P}[h = i]]_i.}$
  $\boxed{\Phi(i,j) := \mathbb{P}[x_m = i | h = j].}$
- Learning: Loading matrix Φ and Vector $\vec{\lambda}$



Pairwise Probability Matrix M_2

$$M_2(a,b) := \mathbb{P}(x_1 = a, x_2 = b) = \sum_r \lambda_r \Phi(a,r)\Phi(b,r)$$

Triples Probability Tensor M_3

$$M_3(a,b,c) := \mathbb{P}(x_1 = a, x_2 = b, x_3 = c) = \sum_r \lambda_r \Phi(a,r)\Phi(b,r)\Phi(c,r)$$

Matrix and Tensor Forms: $\phi_r := r^{\text{th}}$ column of Φ.

$$\boxed{M_2 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r. \qquad M_3 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r \otimes \phi_r}$$

Tensor Basics: Multilinear Transformations

- For a tensor M_3, define (for matrices V_i of appropriate dimensions)

$$\boxed{[M_3(V_1, V_2, V_3)]_{i_1,i_2,i_3} := \sum_{j_1,j_2,j_3} (M_3)_{j_1,j_2,j_3} \prod_{m\in[3]} V_1(j_m, i_m)}$$

- For a matrix M_2, $\boxed{M(V_1, V_2) := V_1^\top M_2 V_2}$.

$$\boxed{M_3 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r \otimes \phi_r}$$

$$\boxed{\begin{aligned} M_3(W, W, W) &= \sum_{r\in[k]} \lambda_r (W^\top \phi_r)^{\otimes 3} \\ M_3(I, v, v) &= \sum_{r\in[k]} \lambda_r \langle v, \phi_r\rangle^2 \phi_r. \\ M_3(I, I, v) &= \sum_{r\in[k]} \lambda_r \langle v, \phi_r\rangle \phi_r \phi_r^\top. \end{aligned}}$$

Inverse Moment Methods for Learning

$$\boxed{M_2 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r, \quad M_3 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r \otimes \phi_r}$$

Identifiability Using 2^{nd} and 3^{rd} Order Moments

Matrix Φ has linearly independent columns and $\vec{\lambda} > 0$.

Inverse Moment Methods for Learning

$$\boxed{M_2 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r, \quad M_3 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r \otimes \phi_r}$$

Identifiability Using 2^{nd} and 3^{rd} Order Moments

Matrix Φ has linearly independent columns and $\vec{\lambda} > 0$.

Special Case: Orthogonality

- If Φ is an orthogonal matrix $\boxed{M_3(I, \phi_r, \phi_r) = \lambda_r \phi_r}$.
- Loading vectors $\{\phi_r\}$ are eigenvectors of the tensor M_3

Inverse Moment Methods for Learning

$$\boxed{M_2 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r, \quad M_3 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r \otimes \phi_r}$$

Identifiability Using 2^{nd} and 3^{rd} Order Moments

Matrix Φ has linearly independent columns and $\vec{\lambda} > 0$.

Special Case: Orthogonality

- If Φ is an orthogonal matrix $\boxed{M_3(I, \phi_r, \phi_r) = \lambda_r \phi_r}$.
- Loading vectors $\{\phi_r\}$ are eigenvectors of the tensor M_3

How to obtain an orthogonal tensor form?

Orthogonal Tensor Decomposition

$$\boxed{M_2 = \sum_{r \in [k]} \lambda_r \phi_r \otimes \phi_r, \quad M_3 = \sum_{r \in [k]} \lambda_r \phi_r \otimes \phi_r \otimes \phi_r}$$

- Define $W = UD^{-1}$, where $M_2 = UDU^\top$.
- Let $\widetilde{\phi}_i := \sqrt{\lambda_i}\ W^\top \phi_i$. They are orthonormal.

$$\boxed{M_2(W, W) \ = \ \sum_{i \in [k]} W^\top (\sqrt{\lambda_i} \phi_i)(\sqrt{\lambda_i} \phi_i)^\top W \ = \ \sum_{i \in [k]} \widetilde{\phi}_i \widetilde{\phi}_i^\top \ = \ I,}$$

- Now define $\widetilde{M}_3$, so that

$$\boxed{\widetilde{M}_3 = M_3(W, W, W) = \sum_{i \in [k]} \lambda_i \, (W^\top \phi_i)^{\otimes 3} = \sum_{i \in [k]} \frac{1}{\sqrt{\lambda_i}} \ \widetilde{\phi}_i^{\otimes 3}.}$$

Learning: Tensor Decomposition of $\widetilde{M}_3$

Orthogonal Tensor Eigen Analysis

- Consider orthogonal symmetric tensor $T = \sum_i w_i \mu_i^{\otimes 3}$

$$\boxed{T = \sum_{i=1}^{k} w_i \mu_i^{\otimes 3}. \quad T(I, \mu_i, \mu_i) = w_i \mu_i}$$

Orthogonal Tensor Eigen Analysis

- Consider orthogonal symmetric tensor $T = \sum_i w_i \mu_i^{\otimes 3}$

$$\boxed{T = \sum_{i=1}^{k} w_i \mu_i^{\otimes 3}. \quad T(I, \mu_i, \mu_i) = w_i \mu_i}$$

Obtaining eigenvectors through power iterations

$$\boxed{u \mapsto \frac{T(I, u, u)}{\|T(I, u, u)\|}}$$

Orthogonal Tensor Eigen Analysis

- Consider orthogonal symmetric tensor $T = \sum_i w_i \mu_i^{\otimes 3}$

$$\boxed{T = \sum_{i=1}^{k} w_i \mu_i^{\otimes 3}. \quad T(I, \mu_i, \mu_i) = w_i \mu_i}$$

Obtaining eigenvectors through power iterations

$$\boxed{u \mapsto \frac{T(I,u,u)}{\|T(I,u,u)\|}}$$

Challenges and Solution

- Challenge: Other eigenvectors present

  Solution: Only stable vectors are basis vectors $\{\mu_i\}$

Orthogonal Tensor Eigen Analysis

- Consider orthogonal symmetric tensor $T = \sum_i w_i \mu_i^{\otimes 3}$

$$\boxed{T = \sum_{i=1}^{k} w_i \mu_i^{\otimes 3}. \quad T(I, \mu_i, \mu_i) = w_i \mu_i}$$

Obtaining eigenvectors through power iterations

$$\boxed{u \mapsto \frac{T(I,u,u)}{\|T(I,u,u)\|}}$$

Challenges and Solution

- Challenge: Other eigenvectors present

  Solution: Only stable vectors are basis vectors $\{\mu_i\}$
- Challenge: empirical moments

  Solution: robust tensor decomposition methods

Optimization Viewpoint for Tensor Eigen Analysis

Consider Norm Optimization Problem for Tensor T

- $\boxed{\max_u \; T(u,u,u) \qquad s.t. \; u^\top u = I}$
- Constrained stationary fixed points $T(I,u,u) = \lambda u$ and $u^\top u = I$.
- u is a local isolated maximizer if $w^\top (T(I,I,u) - \lambda I) w < 0$ for all w such that $w^\top w = I$ and w is orthogonal to u.

Review for Symmetric Matrices $M = \sum_i w_i \mu_i^{\otimes 2}$

- Constrained stationary points are the eigenvectors
- Only top eigenvector is a maximizer and stable under power iterations

Orthogonal Symmetric Tensors $T = \sum_i w_i \mu_i^{\otimes 3}$

- Stationary points are the eigenvectors (up to scaling)
- All basis vectors $\{\mu_i\}$ are local maximizers and stable under power iterations

Tensor Decomposition: Perturbation Analysis

- Observed tensor $\widetilde{T} = T + E$, where $T = \sum_{i \in k} w_i \mu_i^{\otimes 3}$ is orthogonal tensor and perturbation E, and $\|E\| \leq \epsilon$.
- Recall power iterations $\boxed{u \mapsto \dfrac{\widetilde{T}(I,u,u)}{\|\widetilde{T}(I,u,u)\|}}$

Tensor Decomposition: Perturbation Analysis

- Observed tensor $\widetilde{T} = T + E$, where $T = \sum_{i \in k} w_i \mu_i^{\otimes 3}$ is orthogonal tensor and perturbation E, and $\|E\| \leq \epsilon$.
- Recall power iterations $\boxed{u \mapsto \dfrac{\widetilde{T}(I,u,u)}{\|\widetilde{T}(I,u,u)\|}}$
- "Good" initialization vector $\boxed{\langle u^{(0)}, \mu_i \rangle = \Omega\left(\dfrac{\epsilon}{w_{\min}}\right)}$

Tensor Decomposition: Perturbation Analysis

- Observed tensor $\widetilde{T} = T + E$, where $T = \sum_{i \in k} w_i \mu_i^{\otimes 3}$ is orthogonal tensor and perturbation E, and $\|E\| \leq \epsilon$.
- Recall power iterations $\boxed{u \mapsto \frac{\widetilde{T}(I,u,u)}{\|\widetilde{T}(I,u,u)\|}}$
- "Good" initialization vector $\boxed{\langle u^{(0)}, \mu_i \rangle = \Omega\left(\frac{\epsilon}{w_{\min}}\right)}$

Perturbation Analysis

After N iterations, eigen pair (w_i, μ_i) is estimated up to $O(\epsilon)$ error, where

$$\boxed{N = O\left(\log k + \log\log \frac{w_{\max}}{\epsilon}\right).}$$

A. Anandkumar, R. Ge, D. Hsu, S.M. Kakade and M. Telgarsky "Tensor Decompositions for Learning Latent Variable Models," Preprint, October 2012.

Robust Tensor Power Method

$$\boxed{\widetilde{T} = \sum_i w_i \mu_i^{\otimes 3} + E}$$

Basic Algorithm

- Pick random initialization vectors
- Run power iterations $\boxed{u \mapsto \dfrac{\widetilde{T}(I,u,u)}{\|\widetilde{T}(I,u,u)\|}}$
- Go with the winner, deflate and repeat

Robust Tensor Power Method

$$\boxed{\widetilde{T} = \sum_i w_i \mu_i^{\otimes 3} + E}$$

Basic Algorithm

- Pick random initialization vectors
- Run power iterations $\boxed{u \mapsto \frac{\widetilde{T}(I,u,u)}{\|\widetilde{T}(I,u,u)\|}}$
- Go with the winner, deflate and repeat

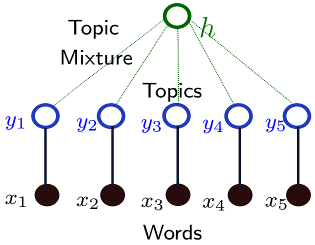
Further Improvements

- Initialization: Use long document vectors for initialization
- Stabilization: $\boxed{u^{(t)} \mapsto \alpha \frac{\widetilde{T}(I,u^{(t-1)},u^{(t-1)})}{\|\widetilde{T}(I,u^{(t-1)},u^{(t-1)})\|} + (1-\alpha)u^{(t-1)}}$

Extensions...

Latent Dirichlet Allocation

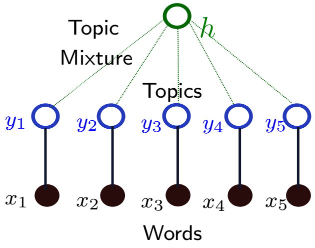
- Each document a topic mixture rather than a single topic
- Modified second and third order moments reduce to symmetric tensor.



Extensions...

Latent Dirichlet Allocation

- Each document a topic mixture rather than a single topic
- Modified second and third order moments reduce to symmetric tensor.

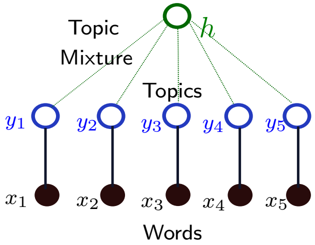


Spherical Gaussian Mixtures, Hidden Markov Models, Independent Component Analysis (ICA) . . .

Extensions...

Latent Dirichlet Allocation

- Each document a topic mixture rather than a single topic
- Modified second and third order moments reduce to symmetric tensor.



Spherical Gaussian Mixtures, Hidden Markov Models, Independent Component Analysis (ICA) ...

Community Modeling and Detection in Social Networks

- Mixed membership model (Airoldi et. al): overlapping communities
- Edge counts and 3-star counts: tensor decomposition

A. Anandkumar, R. Ge, D. Hsu, S. Kakade, " Learning Mixed Membership Block Models."

Preliminary Experiments

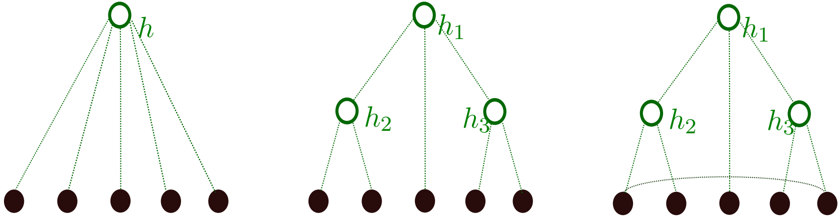
Top 10 words for 5 topics (NYTimes data)

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
bush	company	show	official	team
president	percent	book	water	game
government	million	women	attack	season
official	companies	family	u_s	player
campaign	market	film	food	play
political	business	school	united_states	games
law	stock	look	afghanistan	point
leader	billion	home	taliban	run
george_bush	money	children	air	win
al_gore	cost	friend	military	won

Outline

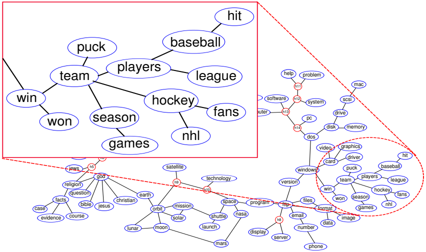
Hierarchical Latent Variable Models

So far... | Latent Tree | Loopy Model

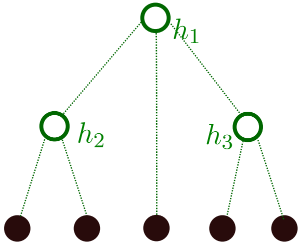


Graph Estimation with Latent Variables

- # and location of hidden variables unknown
- Estimate graph over all variables
- Trees and girth-constrained graphs



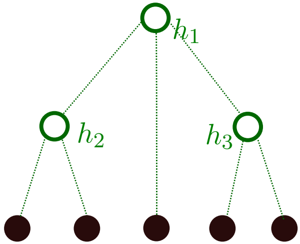
Learning Latent Tree Models



Learning Latent Tree Models

Information Distances $\{d_{ij}\}$

- Gaussian: $d_{ij} := -\log|\rho_{ij}|$.
- Discrete: $d_{ij} := -\log|\operatorname{Det}(P_{i,j})|$.



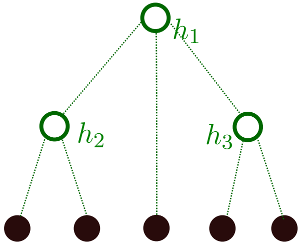
Learning Latent Tree Models

Information Distances $\{d_{ij}\}$

- Gaussian: $d_{ij} := -\log|\rho_{ij}|$.
- Discrete: $d_{ij} := -\log|\mathrm{Det}(P_{i,j})|$.

$[d_{i,j}]$ is an additive tree metric:

$$\boxed{d_{k,l} = \sum_{(i,j)\in \mathrm{Path}(k,l;E)} d_{i,j}.}$$



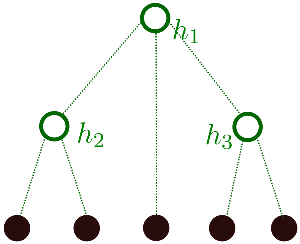
Learning Latent Tree Models

Information Distances $\{d_{ij}\}$

- Gaussian: $d_{ij} := -\log|\rho_{ij}|$.
- Discrete: $d_{ij} := -\log|\operatorname{Det}(P_{i,j})|$.

$[d_{i,j}]$ is an additive tree metric:

$$\boxed{d_{k,l} = \sum_{(i,j)\in \mathrm{Path}(k,l;E)} d_{i,j}.}$$



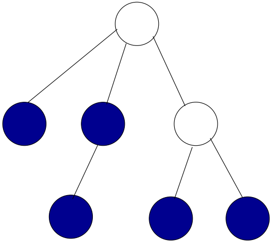
Learning latent tree using $[\hat{d}_{i,j}]$

Siblings Test Based on Information Distances

Exact Statistics: Distances $[d_{i,j}]$

Let $\Phi_{ijk} := d_{i,k} - d_{j,k}$.

- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j} \; \forall\, k, k' \neq i, j,$ $\iff$ i, j leaves with common parent
- $\Phi_{ijk} = d_{i,j}, \; \forall\, k \neq i, j,$ $\iff$ i is a leaf and j is its parent.



Sample Statistics: ML Estimates $[\hat{d}_{i,j}]$

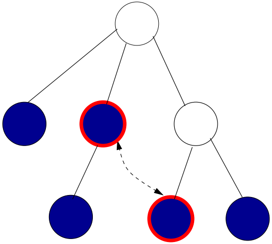
Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships

Siblings Test Based on Information Distances

Exact Statistics: Distances $[d_{i,j}]$

Let $\Phi_{ijk} := d_{i,k} - d_{j,k}$.

- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j} \; \forall \, k, k' \neq i, j,$ $\iff$ i, j leaves with common parent
- $\Phi_{ijk} = d_{i,j}, \; \forall \, k \neq i, j,$ $\iff$ i is a leaf and j is its parent.



Sample Statistics: ML Estimates $[\hat{d}_{i,j}]$

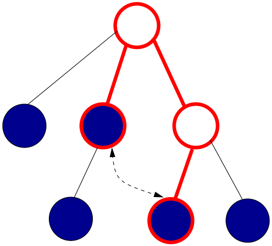
Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships

Siblings Test Based on Information Distances

Exact Statistics: Distances $[d_{i,j}]$

Let $\Phi_{ijk} := d_{i,k} - d_{j,k}$.

- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j} \;\forall\, k, k' \neq i, j,$ $\iff$ i, j leaves with common parent
- $\Phi_{ijk} = d_{i,j}, \;\forall\, k \neq i, j,$ $\iff$ i is a leaf and j is its parent.



Sample Statistics: ML Estimates $[\hat{d}_{i,j}]$

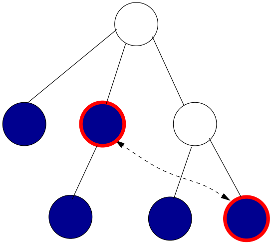
Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships

Siblings Test Based on Information Distances

Exact Statistics: Distances $[d_{i,j}]$

Let $\Phi_{ijk} := d_{i,k} - d_{j,k}$.

- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j} \ \forall\, k, k' \neq i, j,$ $\iff$ i, j leaves with common parent
- $\Phi_{ijk} = d_{i,j},\ \forall\, k \neq i, j,$ $\iff$ i is a leaf and j is its parent.



Sample Statistics: ML Estimates $[\hat{d}_{i,j}]$

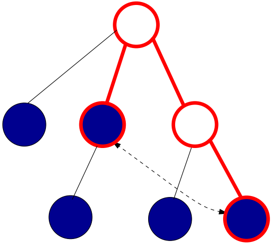
Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships

Siblings Test Based on Information Distances

Exact Statistics: Distances $[d_{i,j}]$

Let $\Phi_{ijk} := d_{i,k} - d_{j,k}$.

- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j} \ \forall\, k, k' \neq i, j,$ $\iff$ i, j leaves with common parent
- $\Phi_{ijk} = d_{i,j}, \ \forall\, k \neq i, j,$ $\iff$ i is a leaf and j is its parent.



Sample Statistics: ML Estimates $[\hat{d}_{i,j}]$

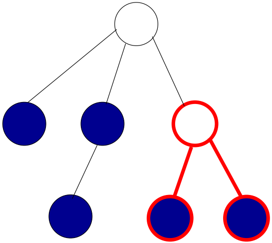
Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships

Siblings Test Based on Information Distances

Exact Statistics: Distances $[d_{i,j}]$

Let $\Phi_{ijk} := d_{i,k} - d_{j,k}$.

- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j} \; \forall \, k, k' \neq i, j,$ $\iff$ i, j leaves with common parent
- $\Phi_{ijk} = d_{i,j}, \; \forall \, k \neq i, j,$ $\iff$ i is a leaf and j is its parent.



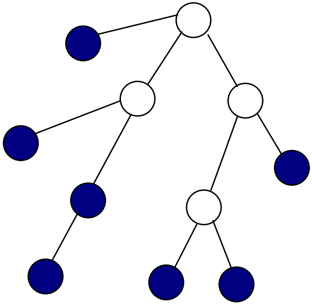
Sample Statistics: ML Estimates $[\hat{d}_{i,j}]$

Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships

Recursive Grouping

Recursive Grouping Algorithm (Choi, Tan, **A.**, Willsky)

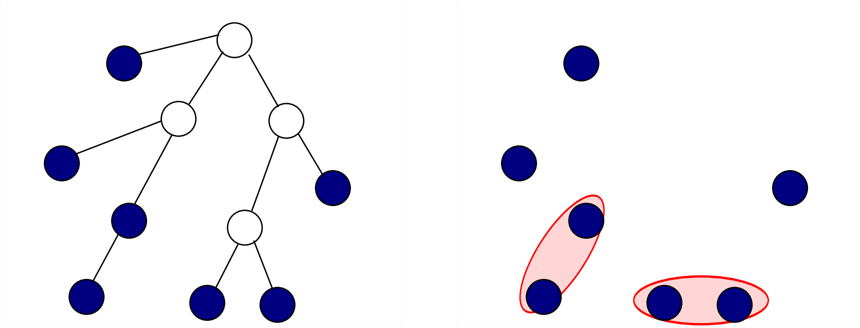
- Sibling test and remove leaves
- Build tree from bottom up



Recursive Grouping

Recursive Grouping Algorithm (Choi, Tan, **A.**, Willsky)

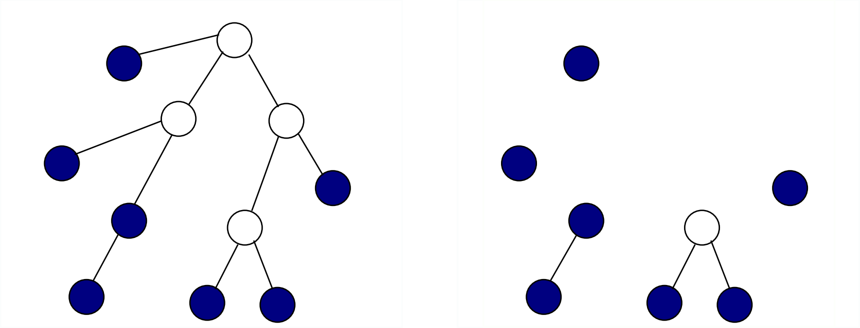
- Sibling test and remove leaves
- Build tree from bottom up



Recursive Grouping

Recursive Grouping Algorithm (Choi, Tan, **A.**, Willsky)

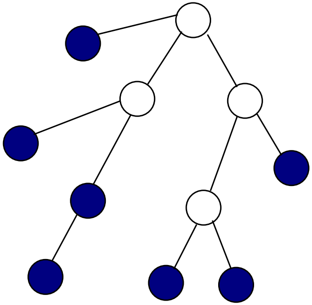
- Sibling test and remove leaves
- Build tree from bottom up

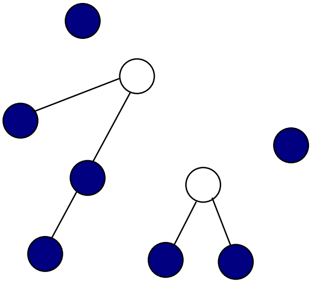


Recursive Grouping

Recursive Grouping Algorithm (Choi, Tan, **A.**, Willsky)

- Sibling test and remove leaves
- Build tree from bottom up

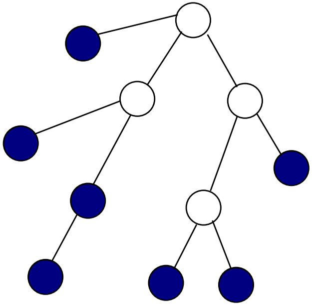


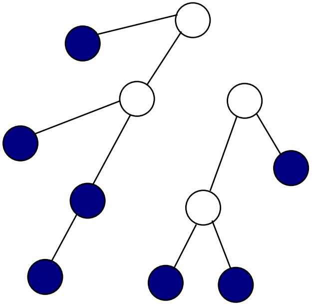


Recursive Grouping

Recursive Grouping Algorithm (Choi, Tan, **A.**, Willsky)

- Sibling test and remove leaves
- Build tree from bottom up

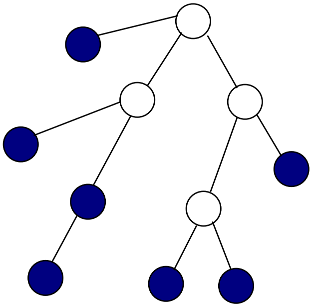


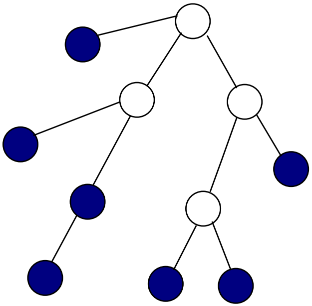


Recursive Grouping

Recursive Grouping Algorithm (Choi, Tan, **A.**, Willsky)

- Sibling test and remove leaves
- Build tree from bottom up

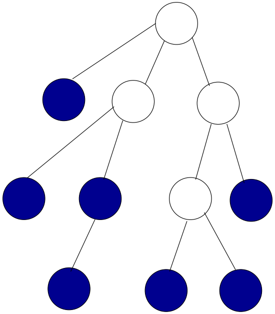


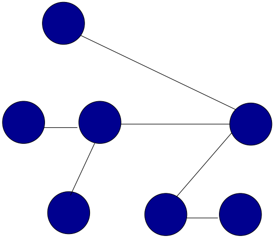


Chow-Liu Based Grouping Algorithm

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.



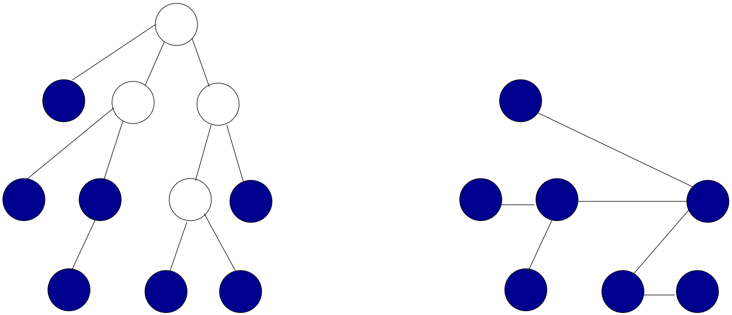


Chow-Liu Based Grouping Algorithm

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)

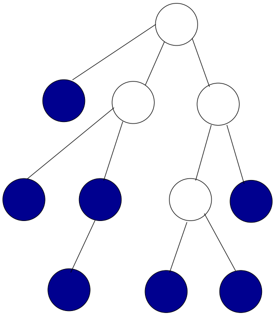


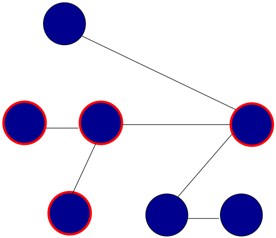
Chow-Liu Based Grouping Algorithm

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)



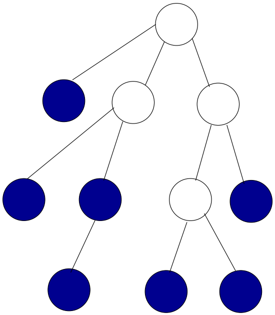


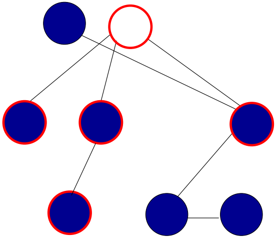
Chow-Liu Based Grouping Algorithm

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)



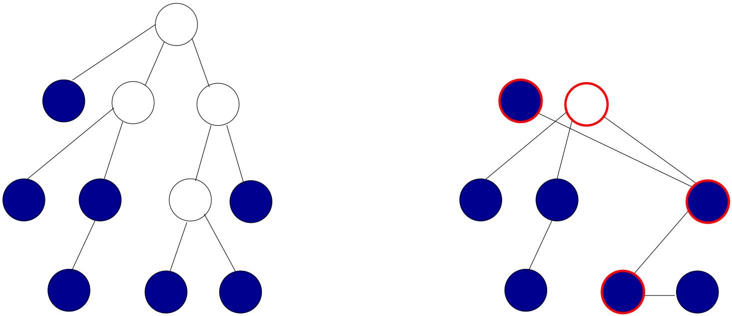


Chow-Liu Based Grouping Algorithm

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)

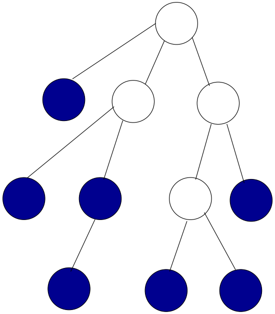


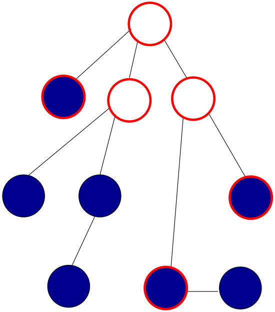
Chow-Liu Based Grouping Algorithm

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)



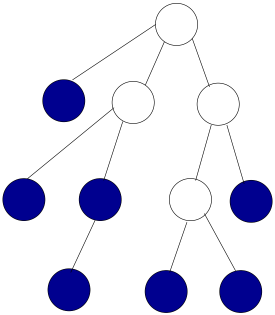


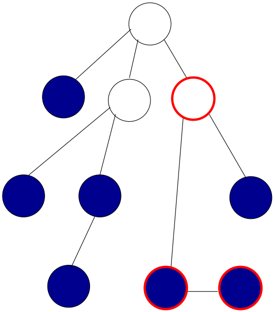
Chow-Liu Based Grouping Algorithm

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)



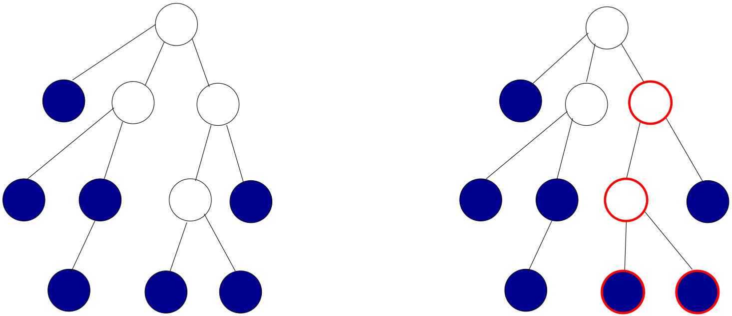


Chow-Liu Based Grouping Algorithm

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)



Proof Ideas

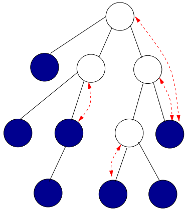
Relating Chow-Liu Tree with Latent Tree

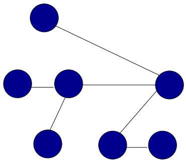
- Surrogate $\mathrm{Sg}(i)$ for node i: observed node with strongest correlation

$$\boxed{\mathrm{Sg}(i) := \operatorname*{argmin}_{j \in V} d_{i,j}}$$

- Neighborhood preservation

$$\boxed{(i,j) \in T \Rightarrow (\mathrm{Sg}(i), \mathrm{Sg}(j)) \in T_{\mathrm{ML}}.}$$





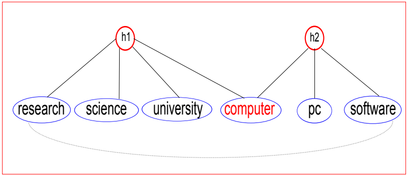
Chow-Liu grouping reverses edge contractions

Proof by induction

Loopy Graphical Models with Latent Nodes

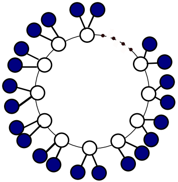
Motivation: Topic Models

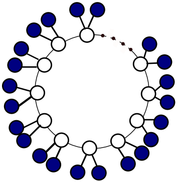
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.



Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods

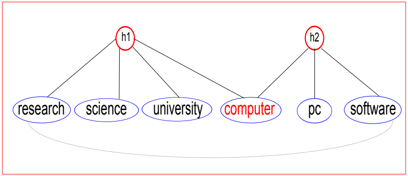




Loopy Graphical Models with Latent Nodes

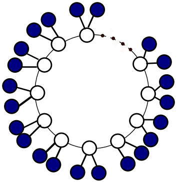
Motivation: Topic Models

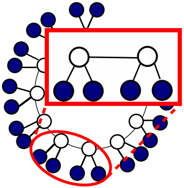
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.



Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods

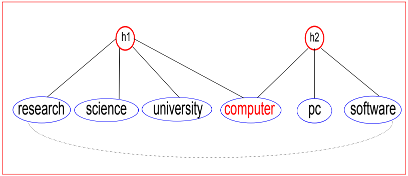




Loopy Graphical Models with Latent Nodes

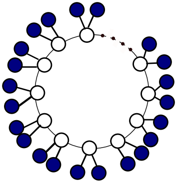
Motivation: Topic Models

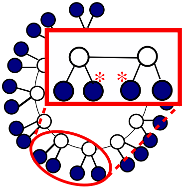
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.



Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods

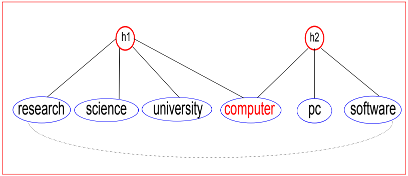




Loopy Graphical Models with Latent Nodes

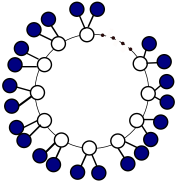
Motivation: Topic Models

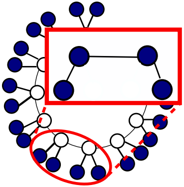
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.



Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods

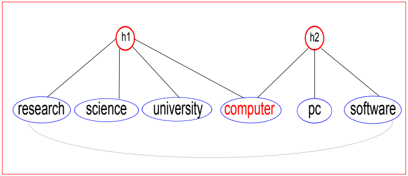




Loopy Graphical Models with Latent Nodes

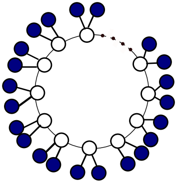
Motivation: Topic Models

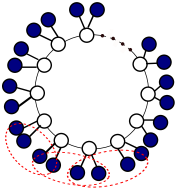
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.



Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods

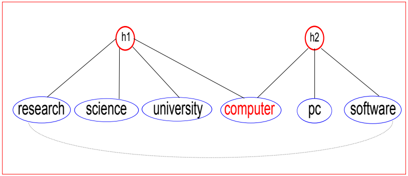




Loopy Graphical Models with Latent Nodes

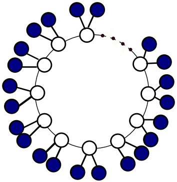
Motivation: Topic Models

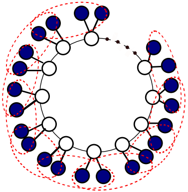
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.



Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods

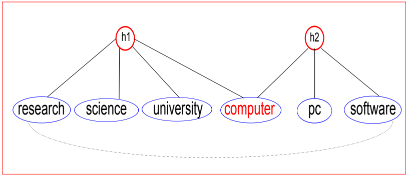




Loopy Graphical Models with Latent Nodes

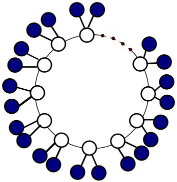
Motivation: Topic Models

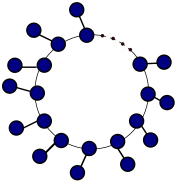
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.



Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods

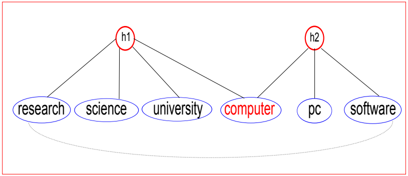




Loopy Graphical Models with Latent Nodes

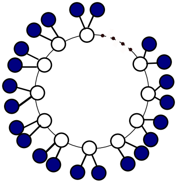
Motivation: Topic Models

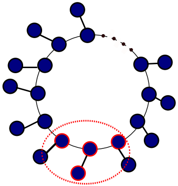
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.



Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods

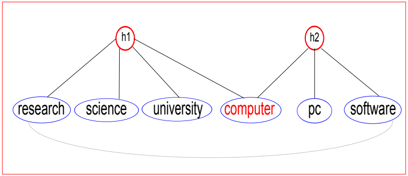




Loopy Graphical Models with Latent Nodes

Motivation: Topic Models

- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.



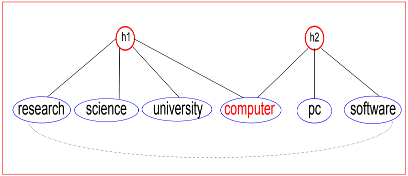
Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods

Loopy Graphical Models with Latent Nodes

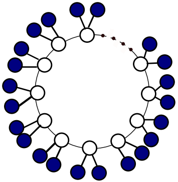
Motivation: Topic Models

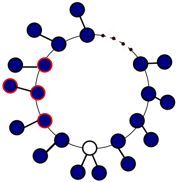
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.



Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods

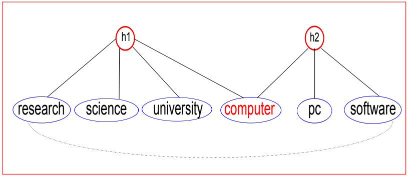




Loopy Graphical Models with Latent Nodes

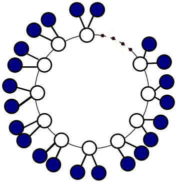
Motivation: Topic Models

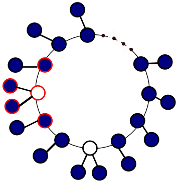
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.



Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods

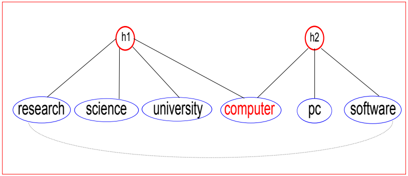




Loopy Graphical Models with Latent Nodes

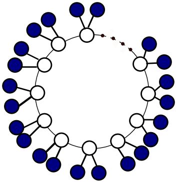
Motivation: Topic Models

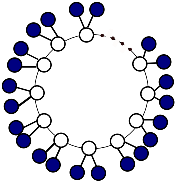
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.



Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods





Guarantees for Latent Structure Learning

- Ising model with minimum edge potential $J_{\min}$.

$$\boxed{p(x) \propto \exp\left[\sum_{(i,j)\in G} J_{i,j} x_i x_j + \sum_{i \in V} h_i x_i\right]}$$

- Depth δ: worst-case distance between hidden and observed nodes.
- Parameter β: depends on min. and max. node and edge potentials
  - $\beta = 1$ for homogeneous models.

Guarantees for Latent Structure Learning

- Ising model with minimum edge potential $J_{\min}$.

$$\boxed{p(x) \propto \exp\left[\sum_{(i,j)\in G} J_{i,j} x_i x_j + \sum_{i\in V} h_i x_i\right]}$$

- Depth δ: worst-case distance between hidden and observed nodes.
- Parameter β: depends on min. and max. node and edge potentials
  - $\beta = 1$ for homogeneous models.

Theorem (A. , Valluvan '12)

Proposed method correctly recovers graph structure w.h.p. on p observed nodes and n samples when

$$\boxed{\frac{J_{\min}^{-2\delta\beta(\beta+1)-2} \log p}{n} = O(1).}$$

A. Anandkumar and R. Valluvan "Learning Loopy Graphical Models with Latent Variables: Efficient Methods and Guarantees" Under revision, Annals of Statistics, June 2012.

Insights and Implications

Tradeoff between depth δ and girth g

Roughly require: $\delta < g/4$.

Tradeoff between max. edge strength $J_{\max}$ and degree Δ

Require $J_{\max} < \operatorname{atanh}(\Delta^{-1})$.

Insights and Implications

Tradeoff between depth δ and girth g

Roughly require: $\delta < g/4$.

Tradeoff between max. edge strength $J_{\max}$ and degree Δ

Require $J_{\max} < \operatorname{atanh}(\Delta^{-1})$.

Sample complexity for uniform node sampling

Given ρ fraction of nodes as observed nodes,

$$\boxed{n = \Omega\left(\Delta^2 \rho^{-4} (\log p)^5\right).}$$

Necessary conditions for structure recovery

For any deterministic algorithm, the number of samples n needs to be

$$\boxed{n = \Omega\left(\frac{\Delta_{\min}}{\rho} \log p\right)}$$

Insights and Implications

Tradeoff between depth δ and girth g

Roughly require: $\delta < g/4$.

Tradeoff between max. edge strength $J_{\max}$ and degree Δ

Require $J_{\max} < \operatorname{atanh}(\Delta^{-1})$.

Sample complexity for uniform node sampling

Given ρ fraction of nodes as observed nodes,

$$\boxed{n = \Omega\left(\Delta^2 \rho^{-4} (\log p)^5\right).}$$

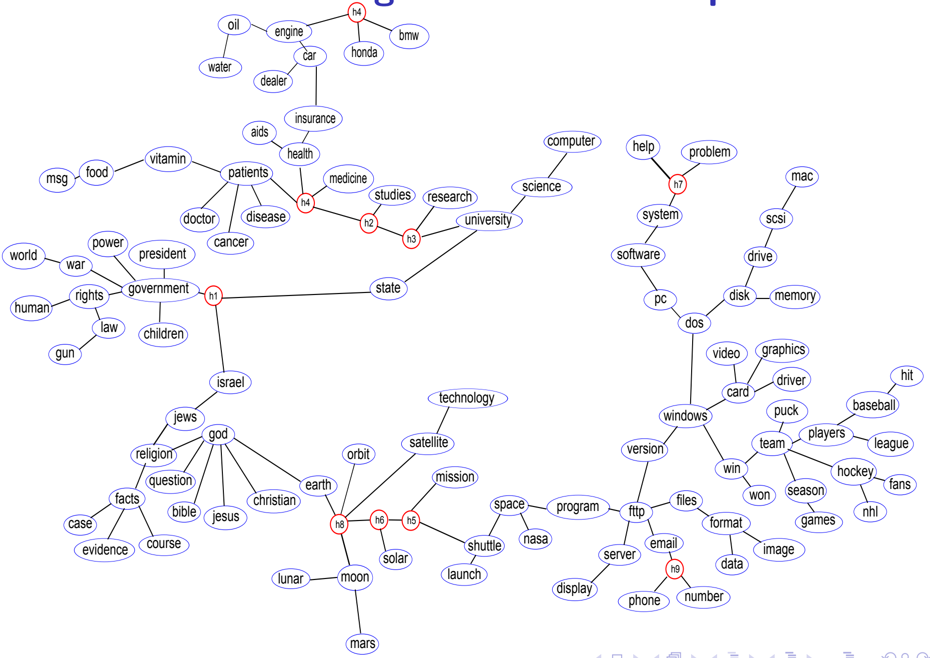
Necessary conditions for structure recovery

For any deterministic algorithm, the number of samples n needs to be

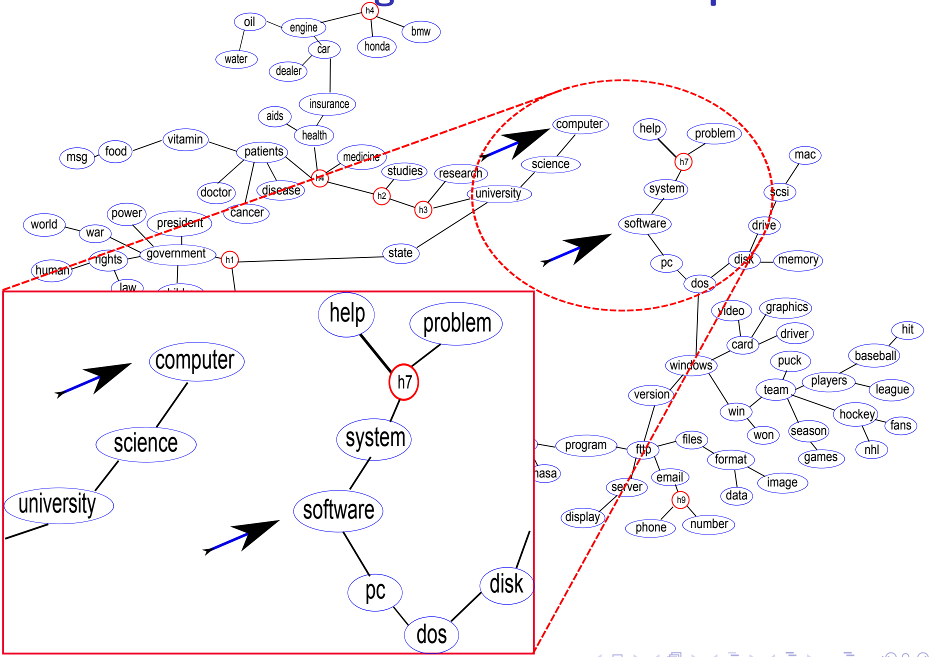
$$\boxed{n = \Omega\left(\frac{\Delta_{\min}}{\rho} \log p\right)}$$

Outline

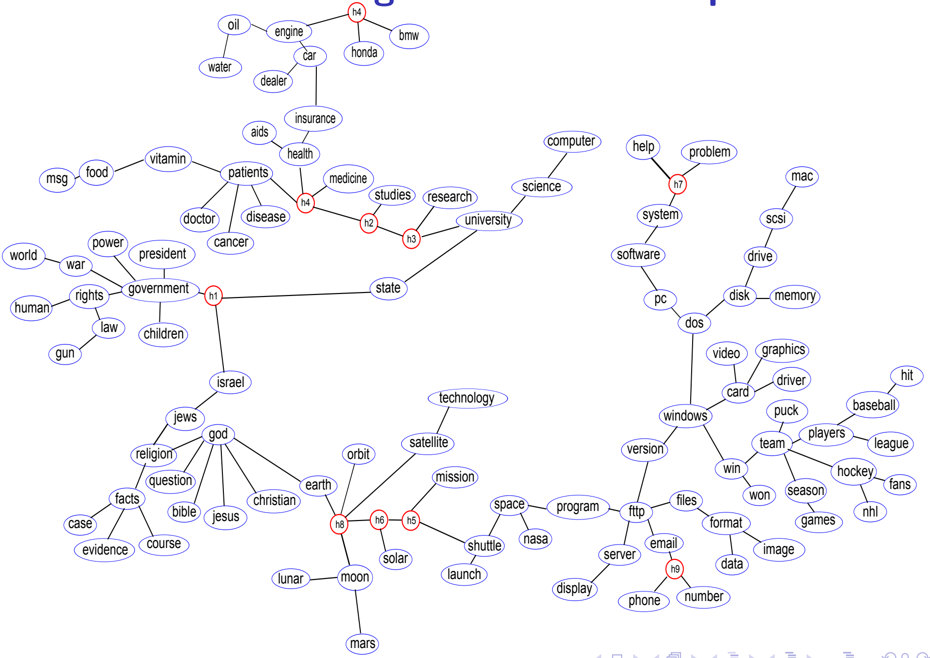
Discovering Word Relationships



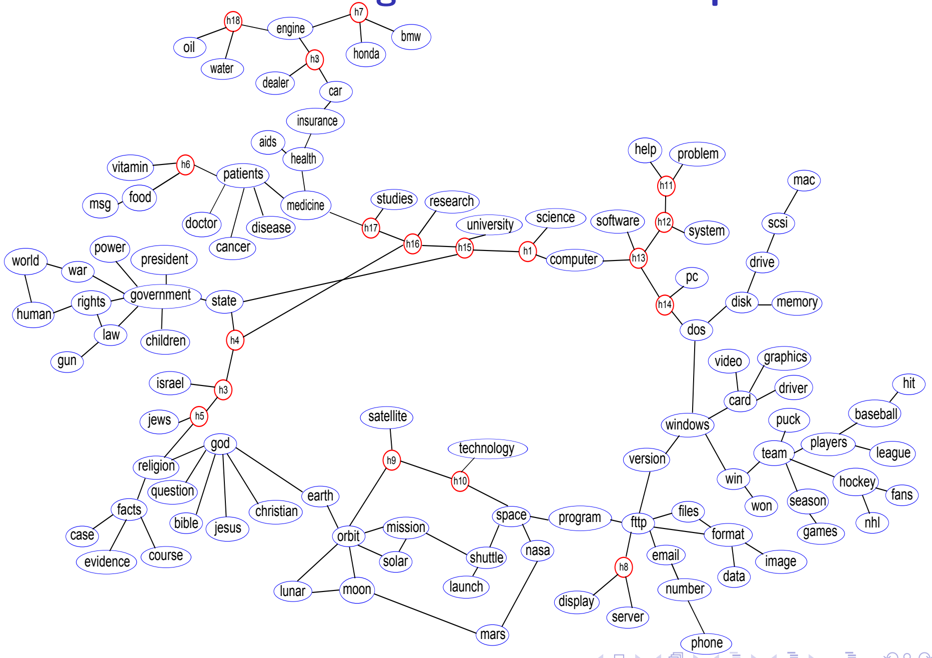
Discovering Word Relationships



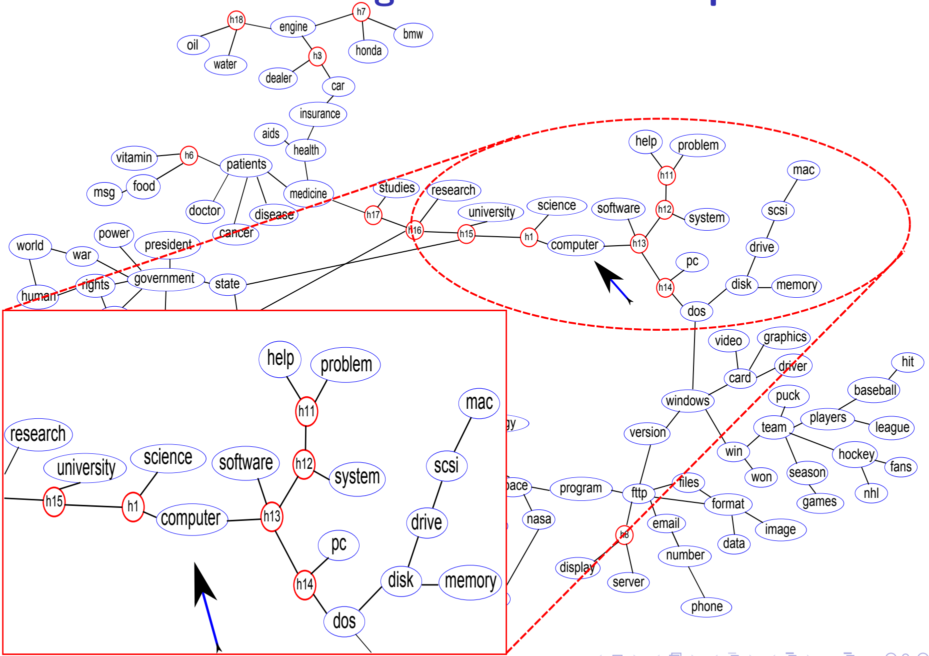
Discovering Word Relationships



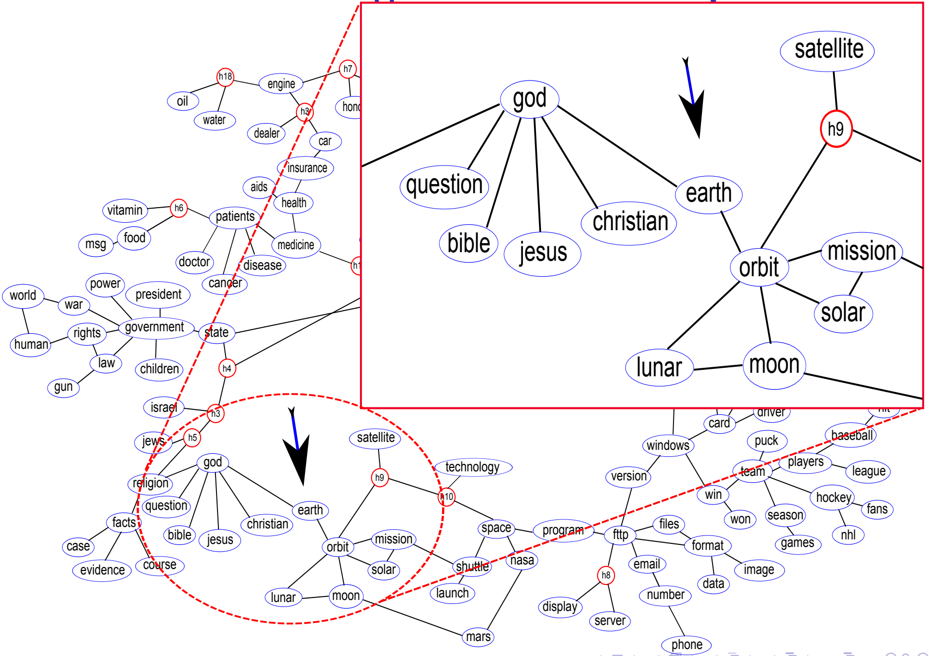
Discovering Word Relationships



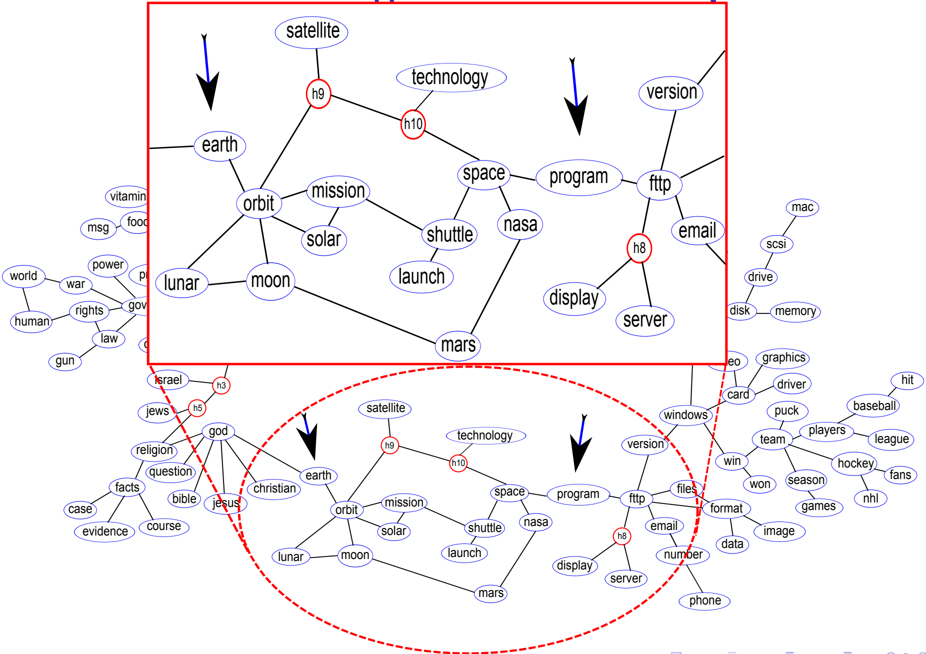
Discovering Word Relationships



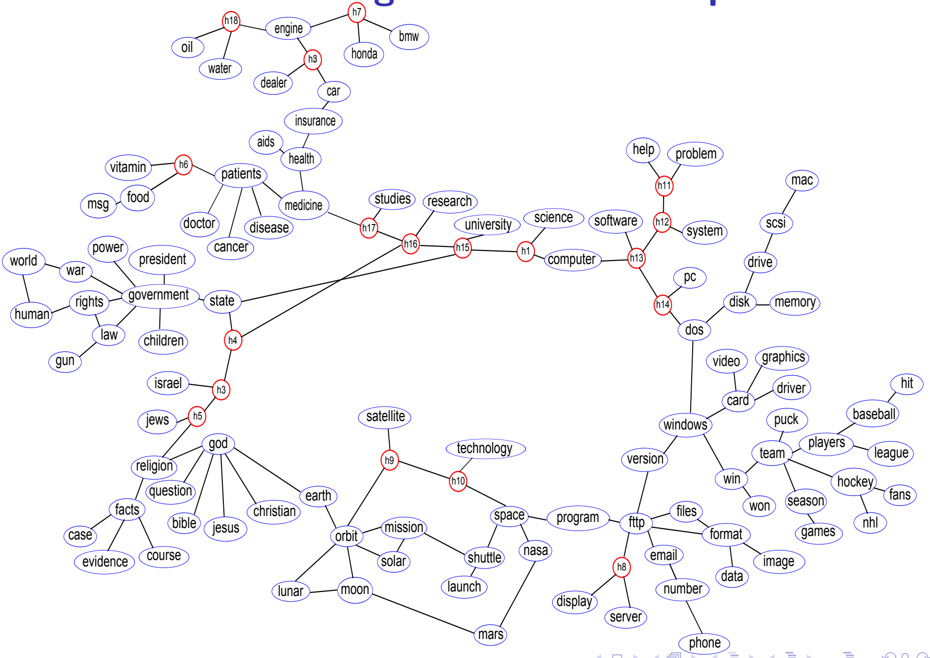
Discovering Word Relationships



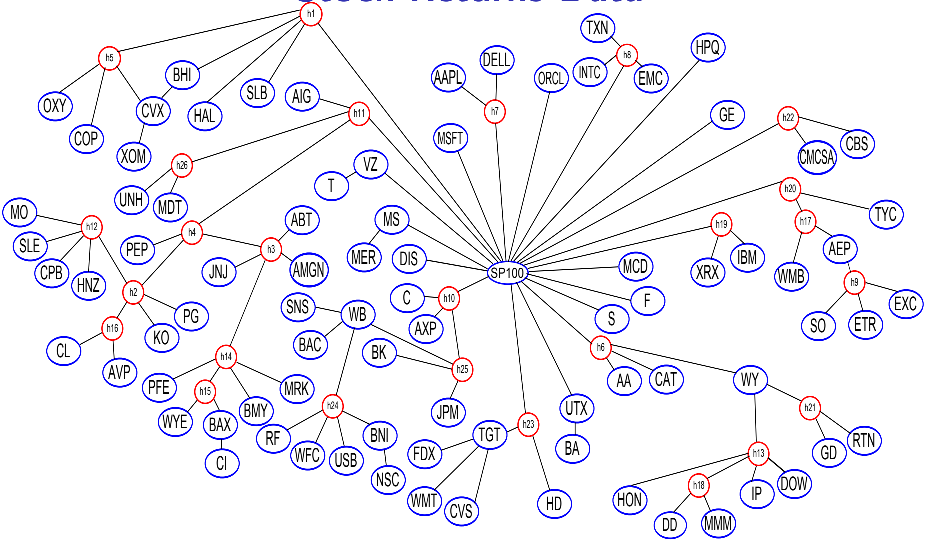
Discovering Word Relationships



Discovering Word Relationships

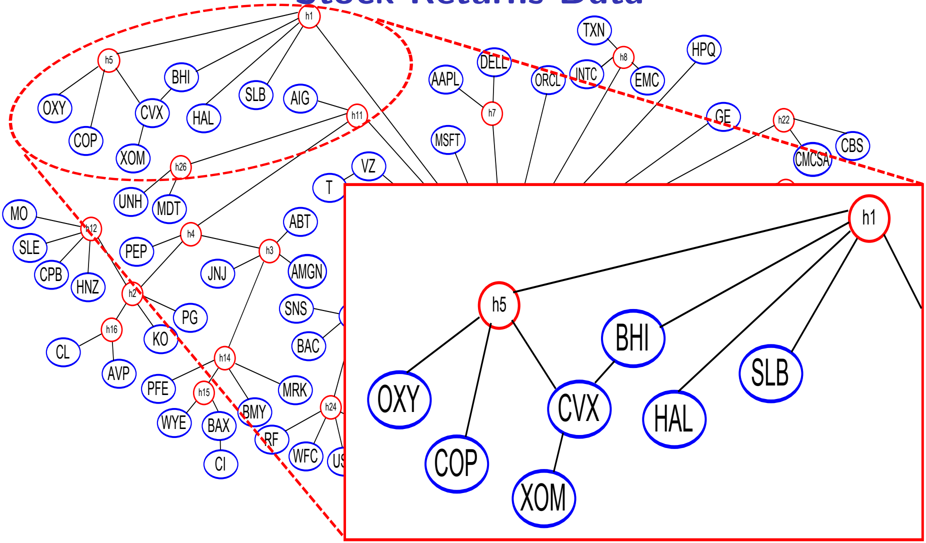


Stock Returns Data



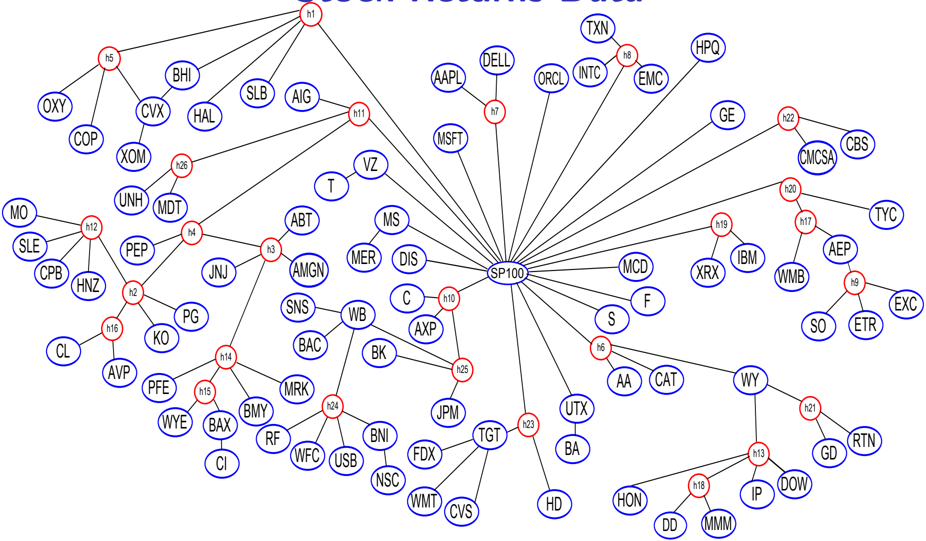
A. Anandkumar and R. Valluvan "Learning Loopy Graphical Models with Latent Variables: Efficient Methods and Guarantees" Under revision, Annals of Statistics, June 2012.

Stock Returns Data



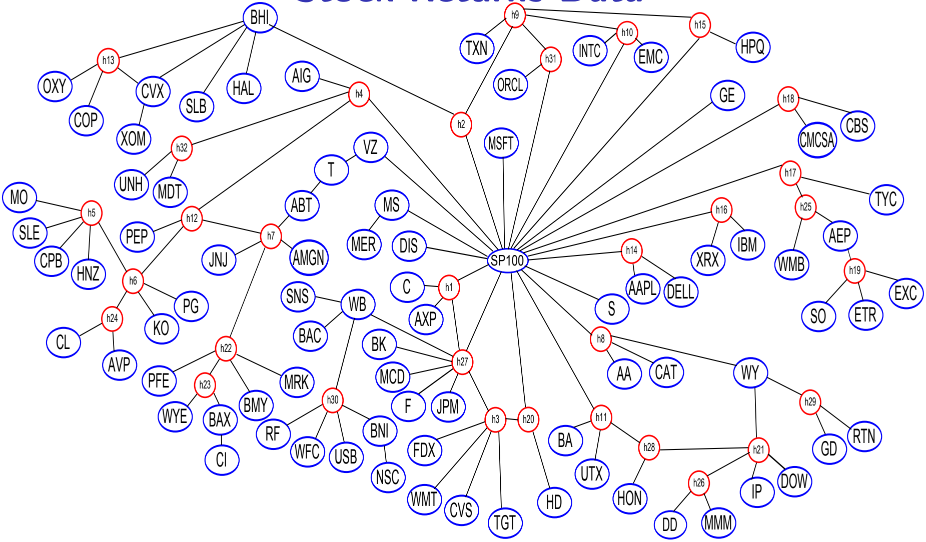
A. Anandkumar and R. Valluvan “Learning Loopy Graphical Models with Latent Variables: Efficient Methods and Guarantees” Under revision, Annals of Statistics, June 2012.

Stock Returns Data



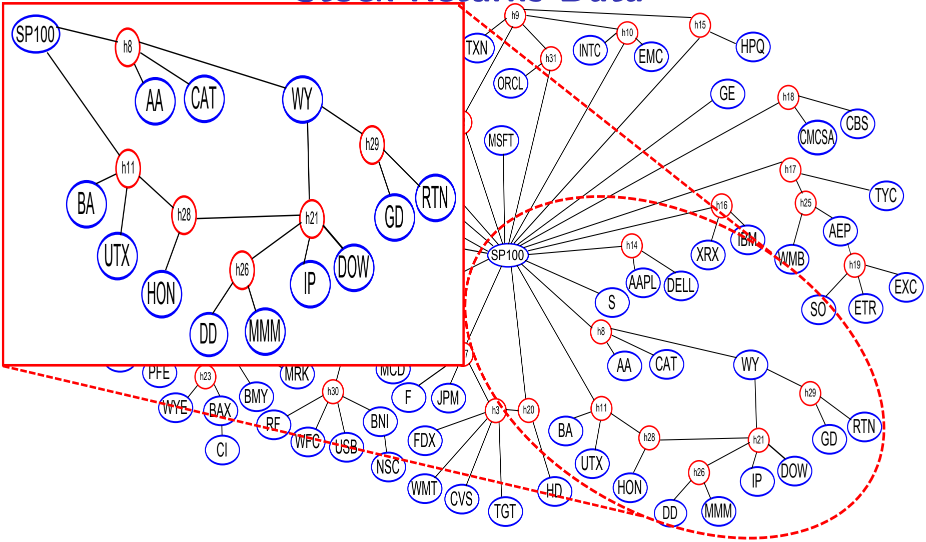
A. Anandkumar and R. Valluvan "Learning Loopy Graphical Models with Latent Variables: Efficient Methods and Guarantees" Under revision, Annals of Statistics, June 2012.

Stock Returns Data



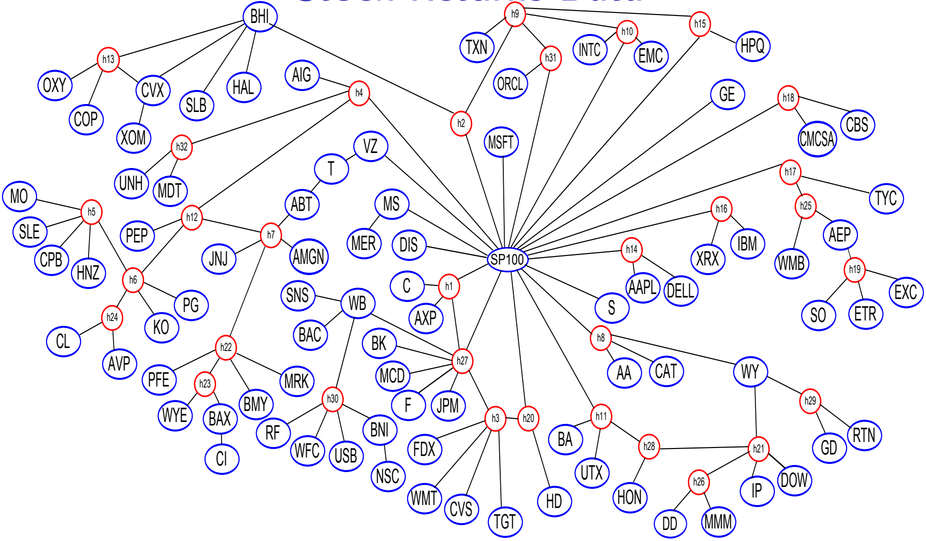
A. Anandkumar and R. Valluvan "Learning Loopy Graphical Models with Latent Variables: Efficient Methods and Guarantees" Under revision, Annals of Statistics, June 2012.

Stock Returns Data



A. Anandkumar and R. Valluvan "Learning Loopy Graphical Models with Latent Variables: Efficient Methods and Guarantees" Under revision, Annals of Statistics, June 2012.

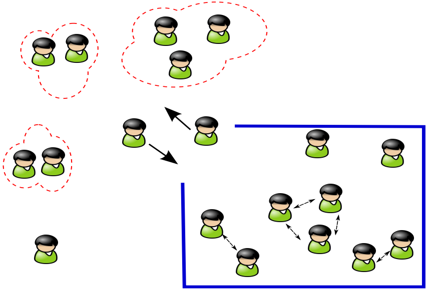
Stock Returns Data

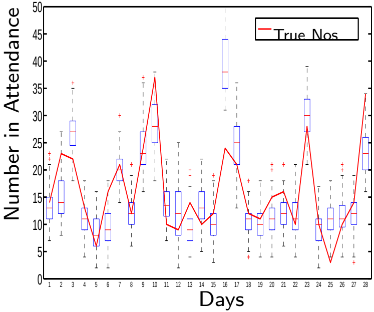


A. Anandkumar and R. Valluvan "Learning Loopy Graphical Models with Latent Variables: Efficient Methods and Guarantees" Under revision, Annals of Statistics, June 2012.

Dynamic Network Modeling

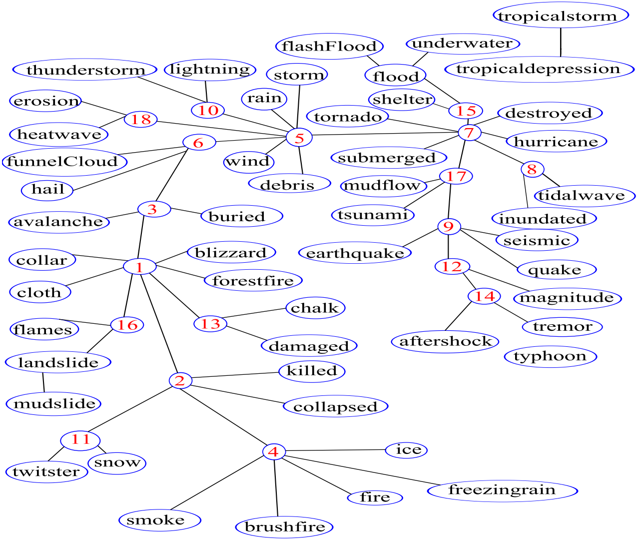
- Observations: series of graph $G_t = (V_t, E_t)$ and covariates
- Modeling vertex participation through latent graphical model
- Logistic regression for edge prediction given vertices
- Data: windsurfer interaction on a beach
- Improvement over baseline: 164% for vertices and 45% for edges.





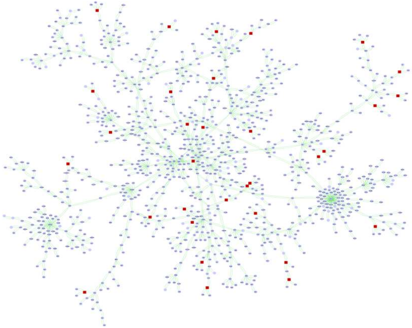
R. Valluvan, Z. W. Almquist, C. T. Butts, and A. Anandkumar. Semi-parametric vertex set prediction for dynamic networks using latent tree models. (Sunbelt 2012).

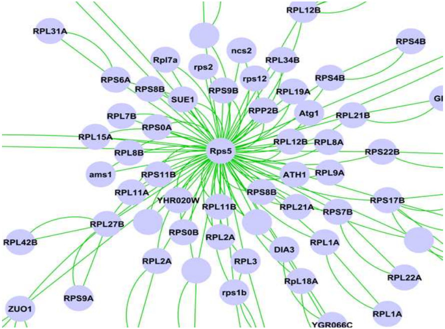
Modeling Hazard-related Tweets



Modeling Gene Associations

- Observed: gene expression levels
- Relationships between genes, e.g. genes that encode ribosomal subunits group together
- Hidden nodes: regulators that control groups of functionally similar genes, e.g. transcription factors





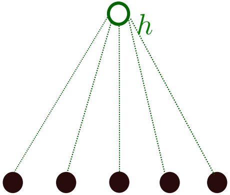
In collaboration with Anthony Gitter (Microsoft) and Ernest Fraenkel (MIT)

Outline

Summary on Learning Latent Variable Models

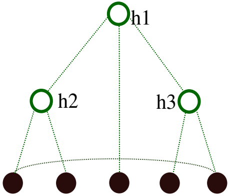
Tensor Methods

- Tensor forms for a range of models
- Efficient decomposition methods
- Perturbation analysis



Graph Estimation

- Latent modeling via graphical approaches
- Efficient methods for graph estimation
- Guarantees on sample and computational complexities



The Big Picture