ECE 6504: Advanced Topics in Machine Learning

Probabilistic Graphical Models and Large-Scale Learning

Topics

- Bayes Nets: Inference
  - (Finish) Variable Elimination
  - Graph-view of VE: Fill-edges, induced width

Readings: KF 9.3,9.4; Barber 5.2

Dhruv Batra

Virginia Tech

Administrativia

- HW1
  - Due in 2 weeks: ~~Feb 17~~, Feb 19, 11:59pm

- Project Proposal
  - Due: ~~Mar 12~~, Mar 5, 11:59pm
  - <=2pages, NIPS format

- HW2
  - Out soon
  - Due: ~~Mar 5~~, Mar 12, 11:59pm

Project

- Individual or Groups of 2
  - we prefer teams of 2

- Deliverables:
  - 5%: Project proposal (NIPS format): <= 2 pages
  - 10%: Midway presentations (in class)
  - 10%: Final report: webpage with results

Proposal

- 2 Page (NIPS format)
  - http://nips.cc/Conferences/2013/PaperInformation/StyleFiles

- Necessary Information:
  - Project title
  - Project idea.
    - This should be approximately two paragraphs.
  - Data set details
    - Ideally existing dataset. No data-collection projects.
  - Software
    - Which libraries will you use?
    - What will you write?
  - Papers to read.
    - Include 1-3 relevant papers. You will probably want to read at least one of them before submitting your proposal.
  - Teammate
    - will you have a teammate? If so, whom? Maximum team size is two students.
  - Mid-sem Milestone
    - What will you complete by the project milestone due date? Experimental results of some kind are expected here.

Project

- Main categories
  - Application/Survey
    - Compare a bunch of existing algorithms on a new application domain of your interest
  - Formulation/Development
    - Formulate a new model or algorithm for a new or old problem
  - Theory
    - Theoretically analyze an existing algorithm
- Rules
  - Should fit in “Advanced Machine Learning”
  - Can apply ML to your own research.
    - Must be done this semester.
  - OK to combine with other class-projects
    - Must declare to both course instructors
    - Must have explicit permission from BOTH instructors
    - Must have a sufficient ML component
  - Using libraries
    - No need to implement all algorithms
    - OK to use standard MRF, BN, Structured SVM, etc libraries
    - More thought+effort => More credit

Recap of Last Time

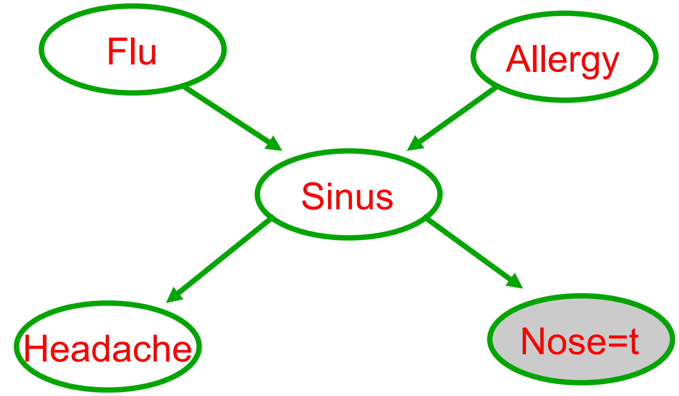
Main Issues in PGMs

- Representation
  - How do we store $P(X_1, X_2, \ldots, X_n)$
  - What does my model mean/imply/assume? (Semantics)

- Learning
  - How do we learn parameters and structure of $P(X_1, X_2, \ldots, X_n)$ from data?
  - What model is the right for my data?

- Inference
  - How do I answer questions/queries with my model? such as
  - Marginal Estimation: $P(X_5 \mid X_1, X_4)$
  - Most Probable Explanation: argmax $P(X_1, X_2, \ldots, X_n)$

Possible Queries



- Evidence: **E**=**e** (e.g. N=t)
- Query variables of interest **Y**

- Conditional Probability: P(**Y** | **E**=**e**)
  - E.g. P(F,A | N=t)
  - Special case: Marginals P(F)

- Maximum a Posteriori: argmax P(All variables | **E**=**e**)
  - argmax_{f,a,s,h} P(f,a,s,h | N = t)

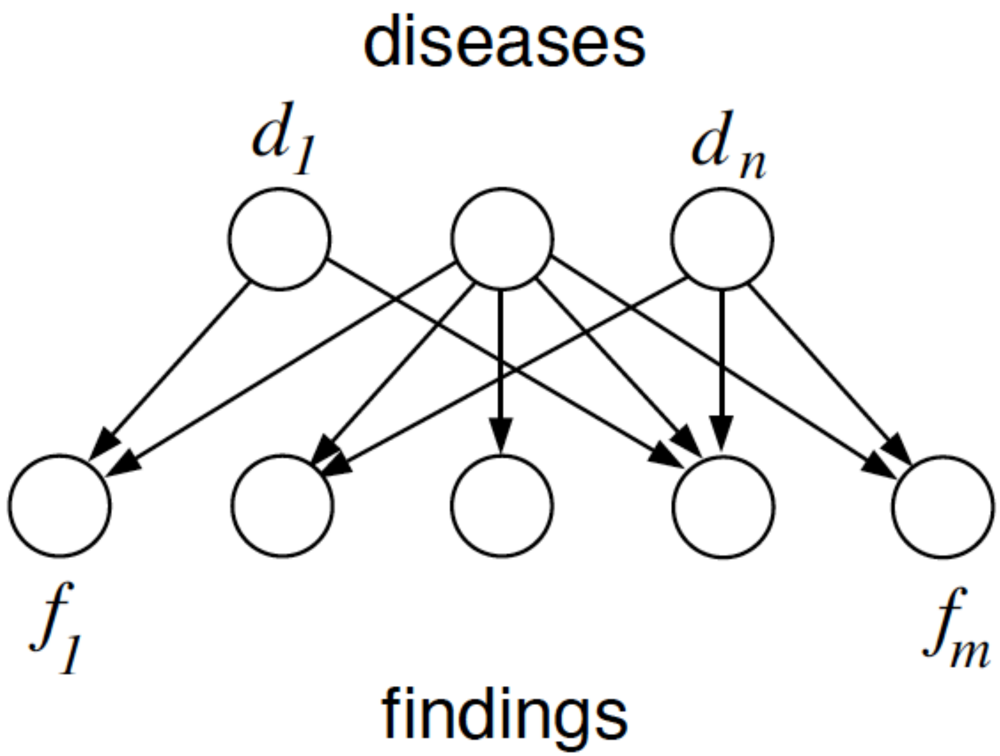
Old-school terminology: MPE

- Marginal-MAP: argmax_y P(**Y** | **E**=**e**)

Old-school terminology: MAP

  - = argmax_{y} Σ_o P(**Y**=**y**, **O**=**o** | **E**=**e**)

Application: Medical Diagnosis



Are MAP and Max of Marginals Consistent?

P(S=f)=0.6
P(S=t)=0.4

P(N|S)

Hardness

- Find P(All variables) — Easy for BN: O(n)
- MAP
  - Find argmax P(All variables | **E**=**e**) — NP-hard
  - Find any assignment P(All variables | **E**=**e**) > p — NP-hard
- Conditional Probability / Marginals
  - Is P(Y=y | **E**=**e**) > 0 — NP-hard
  - Find P(Y=y | **E**=**e**) — #P-hard
  - Find |P(Y=y | **E**=**e**) – p| <= ε — NP-hard for any ε<0.5
- Marginal-MAP
  - Find argmax_{y} Σ_o P(**Y**=**y**, **O**=**o** | **E**=**e**) — NP^{PP}-hard

Inference in BNs hopeless?

- In general, yes!
  - Even approximate!

- In practice
  - Exploit structure
  - Many effective approximation algorithms
    - some with guarantees

- Plan
  - Exact Inference
  - Transition to Undirected Graphical Models (MRFs)
  - Approximate inference in the unified setting

Algorithms

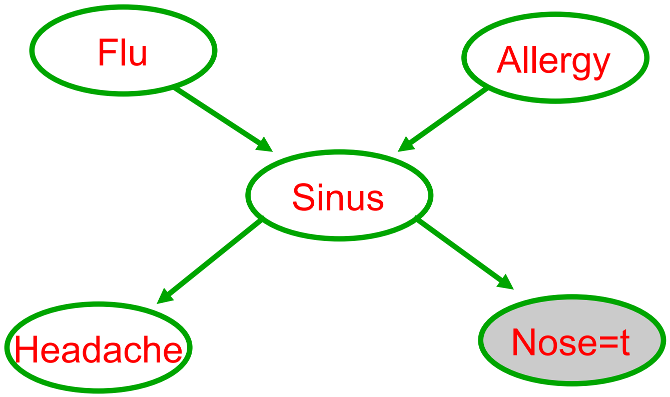
- Conditional Probability / Marginals
  - Variable Elimination
  - Sum-Product Belief Propagation
  - Sampling: MCMC

- MAP
  - Variable Elimination
  - Max-Product Belief Propagation
  - Sampling MCMC

  - Integer Programming
    - Linear Programming Relaxation
  - Combinatorial Optimization (Graph-cuts)

Marginal Inference Example

- Evidence: **E**=**e** (e.g. N=t)
- Query variables of interest **Y**



- Conditional Probability: P(**Y** | **E**=**e**)
  - P(F | N=t)
  - Derivation on board

Variable Elimination algorithm

- Given a BN and a query P(**Y**|**e**) ≈ P(**Y**,**e**)
  - "Instantiate Evidence"

IMPORTANT!!!

- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- For i = 1 to n, If $X_i \notin \{\mathbf{Y},\mathbf{E}\}$
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

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

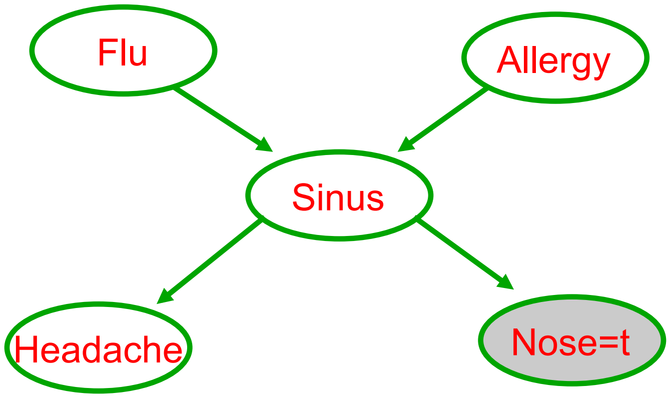
  - Variable X_i has been eliminated!
- Normalize P(Y,e) to obtain P(Y|e)

Plan for today

- BN Inference
  - (Finish) Variable Elimination
    - VE for MAP Inference
  - Graph-view of VE
    - Moralization
    - Fill edges
    - Induced Width
    - Tree width
  - (Start) Undirected Graphical Models

VE for MAP Inference

- Evidence: **E**=**e** (e.g. N=t)
- Query variables of interest **Y**



- Conditional Probability: P(**Y** | **E**=**e**)
  – P(F | N=t)

- Maximum a Posteriori: argmax P(All variables | **E**=**e**)
  – argmax_{f,a,s,h} P(f,a,s,h | N = t)
  – Derivation on board

- VE or Dynamic Programming extends to arbitrary commutative semi-rings!

VE for MAP – Forward Pass

- Given a BN and a MAP query $\max_{x_1,\ldots,x_n} P(x_1,\ldots,x_n,\mathbf{e})$
  - "Instantiate Evidence"
- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- For i = 1 to n, If $X_i \notin \mathbf{E}$
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

$$g = \max_{x_i} \prod_{j=1}^{k} f_j$$

  - Variable X_i has been eliminated!

VE for MAP – Backward Pass

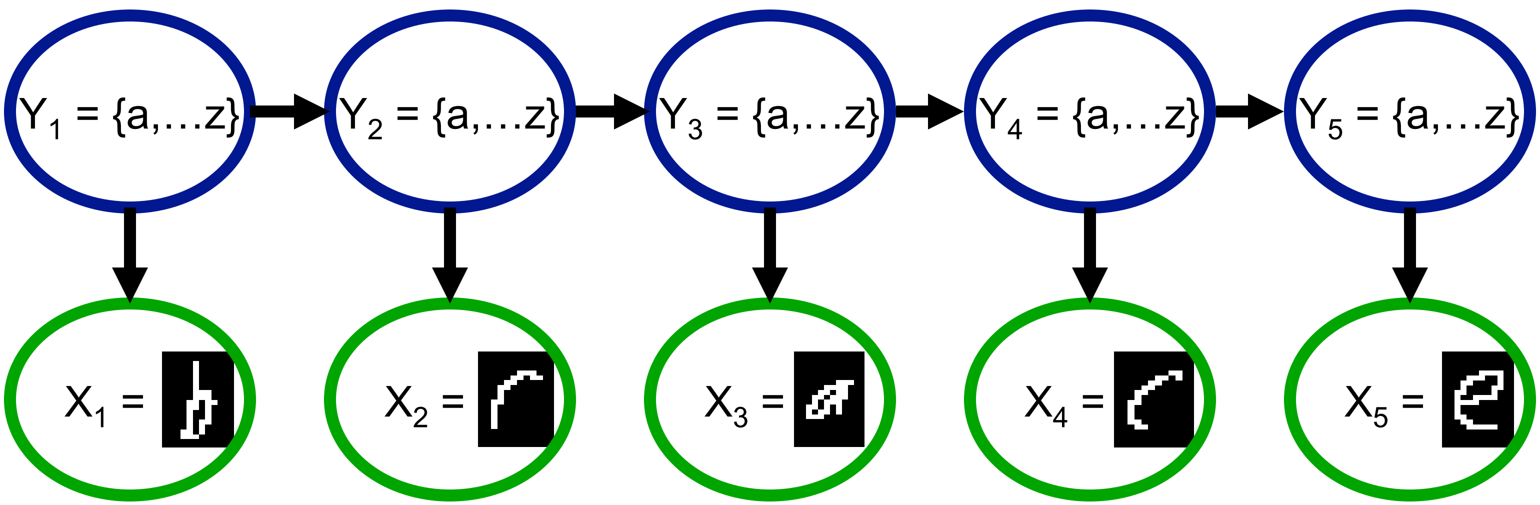
- $\{x_1^*,\ldots, x_n^*\}$ will store maximizing assignment

- For i = n to 1, If $X_i \notin \mathbf{E}$
  - Take factors $f_1,\ldots,f_k$ used when X_i was eliminated
  - Instantiate $f_1,\ldots,f_k$, with $\{x_{i+1}^*,\ldots, x_n^*\}$
    - Now each f_j depends only on X_i
  - Generate maximizing assignment for X_i:

$$x_i^* \in \operatorname*{argmax}_{x_i} \prod_{j=1}^{k} f_j$$

Instantiating Evidence

- Given a BN and a query $P(Y|e) \approx P(Y,e)$
  - This step "reduces" the size of factors



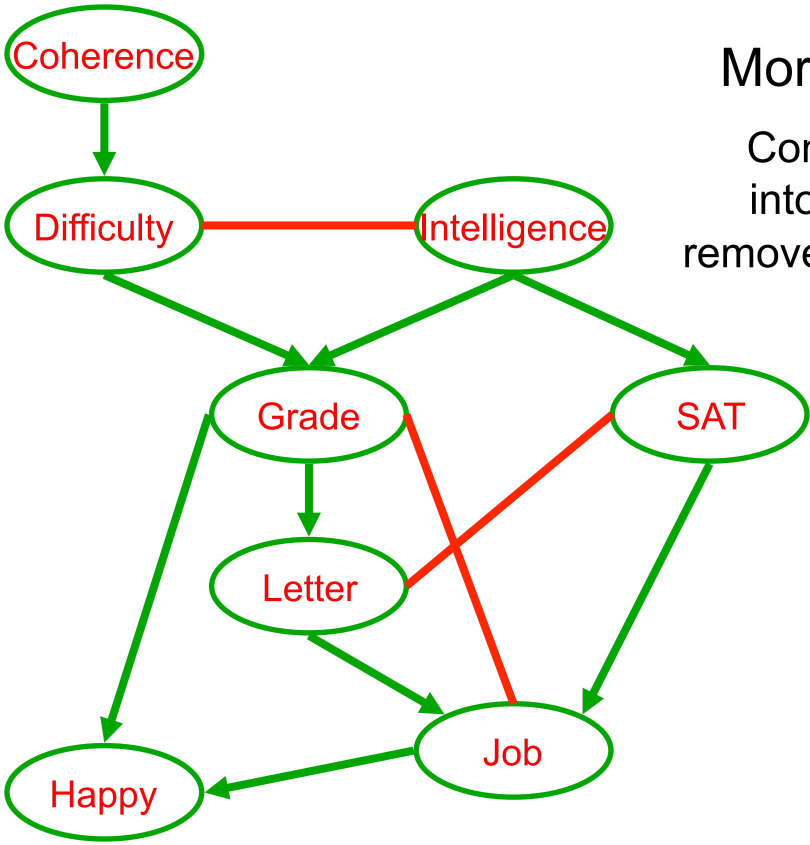
Hidden Markov Model (HMM)

Graph-view of VE

- So far: Algorithmic / Algebriac view of VE

- Next: Graph-based view of VE
  - Modifications to graph-structure as VE is running

Moralization – “Marry” Parents

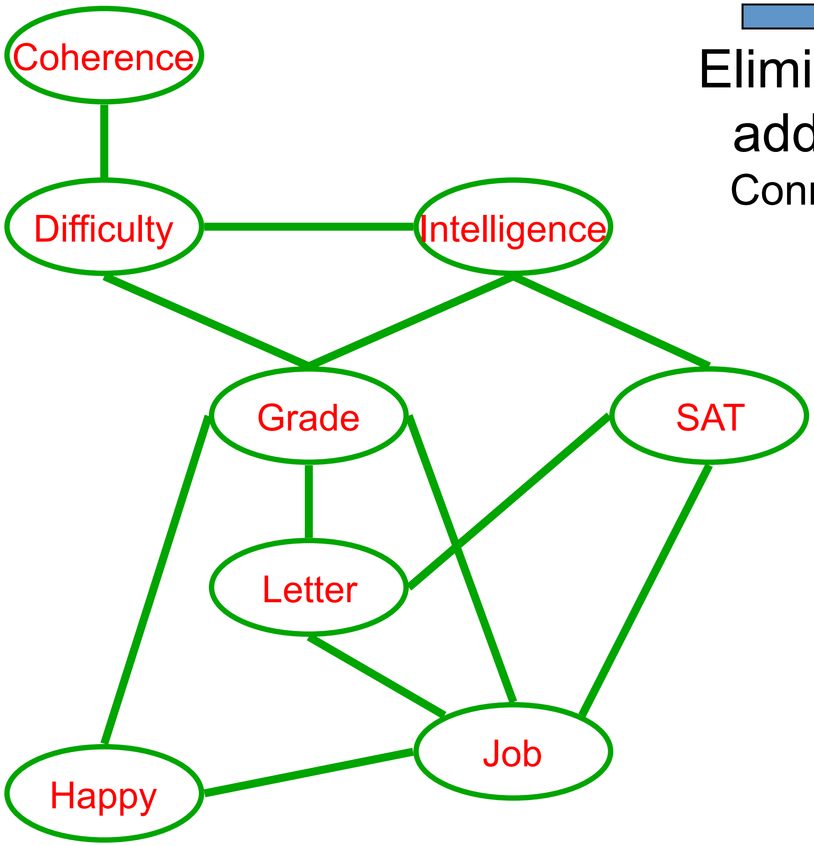


Moralize graph:

Connect parents
into a clique and
remove edge directions

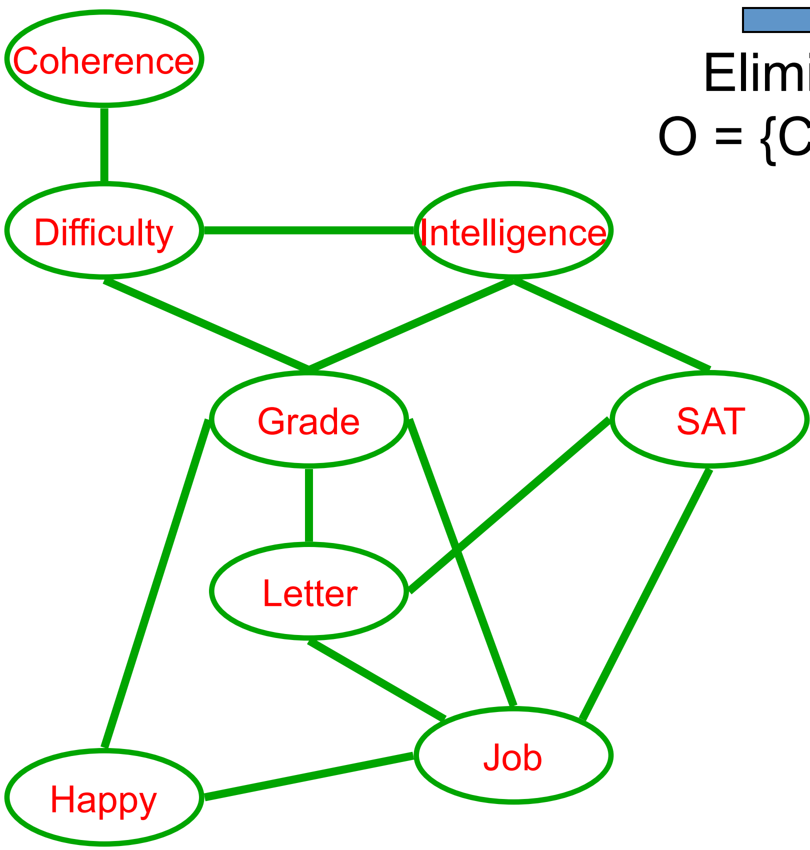
Connect nodes that appear together in an initial factor

Eliminating a node – Fill edges



Eliminate variable
add *Fill Edges*:
Connect neighbors

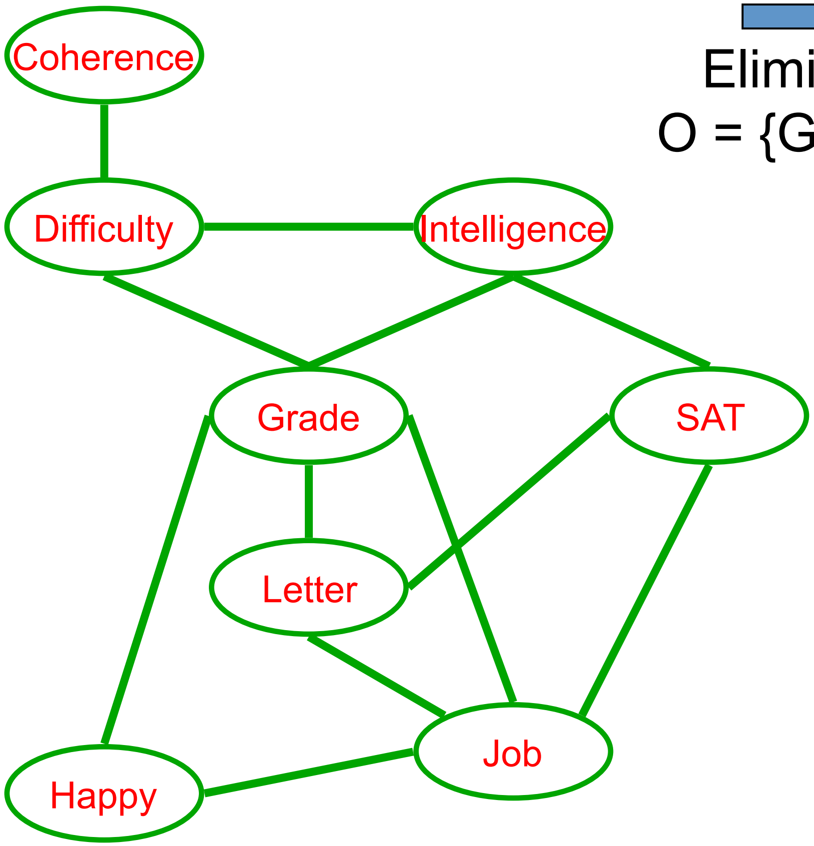
Induced graph



Elimination order:
O = {C,D,S,I,L,H,J,G}

The induced graph I_{FO} for elimination order O has an edge $X_i - X_j$ if X_i and X_j appear together in a factor generated by VE for elimination order O on factors F

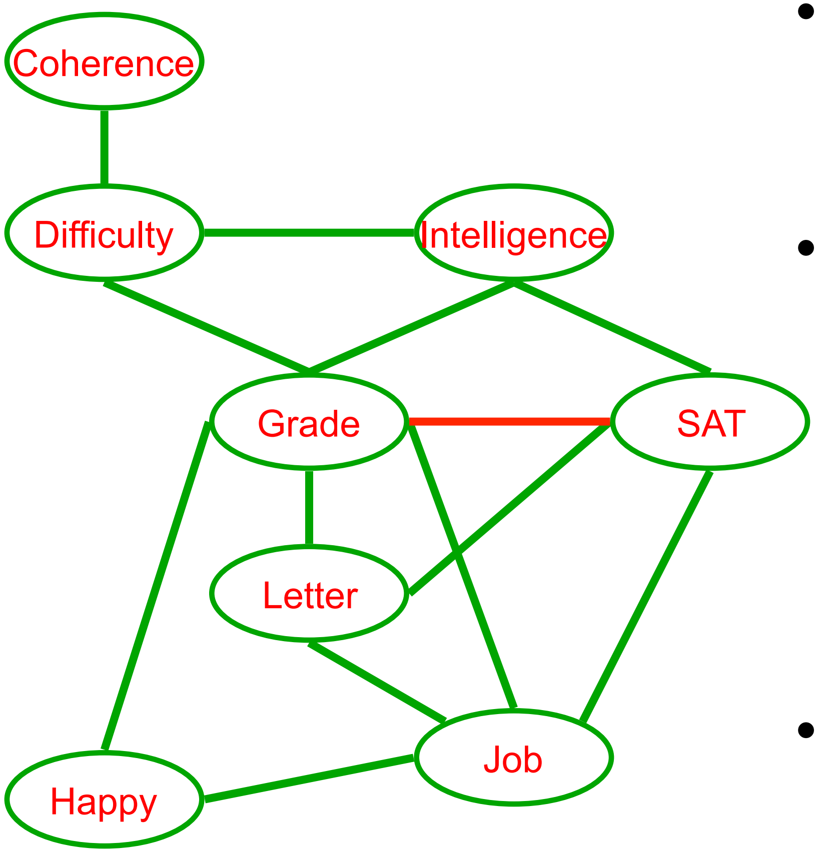
Different elimination order can lead to different induced graph



Elimination order:
O = {G,C,D,S,I,L,H,J}

Induced graph and complexity of VE

Read complexity from cliques in induced graph



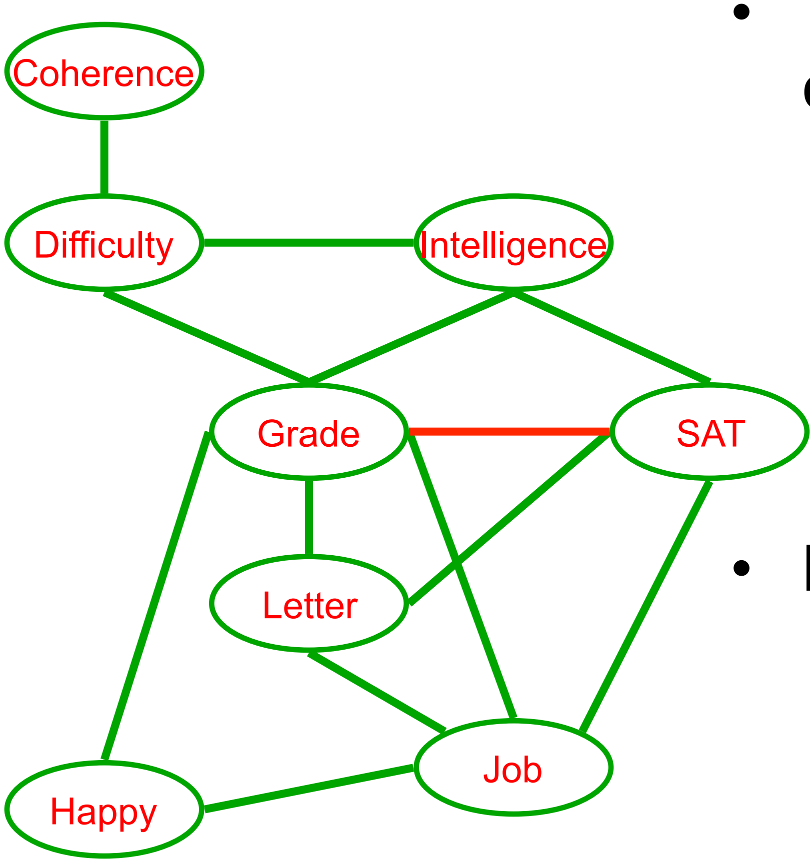
Elimination order:
O = {C,D,I,S,L,H,J,G}

- Structure of induced graph encodes complexity of VE!!!
- **Theorem**:
  – Every factor generated by VE is a clique in I_{FO}
  – Every maximal clique in I_{FO} corresponds to a factor generated by VE
- **Induced width**
  – Size of largest clique in I_{FO} minus 1
- **Treewidth**
  – induced width of best order O*

Example: Large induced-width with small number of parents

$\boxed{\text{Compact representation} \not\Rightarrow \text{Easy inference} ☹}$

Finding optimal elimination order

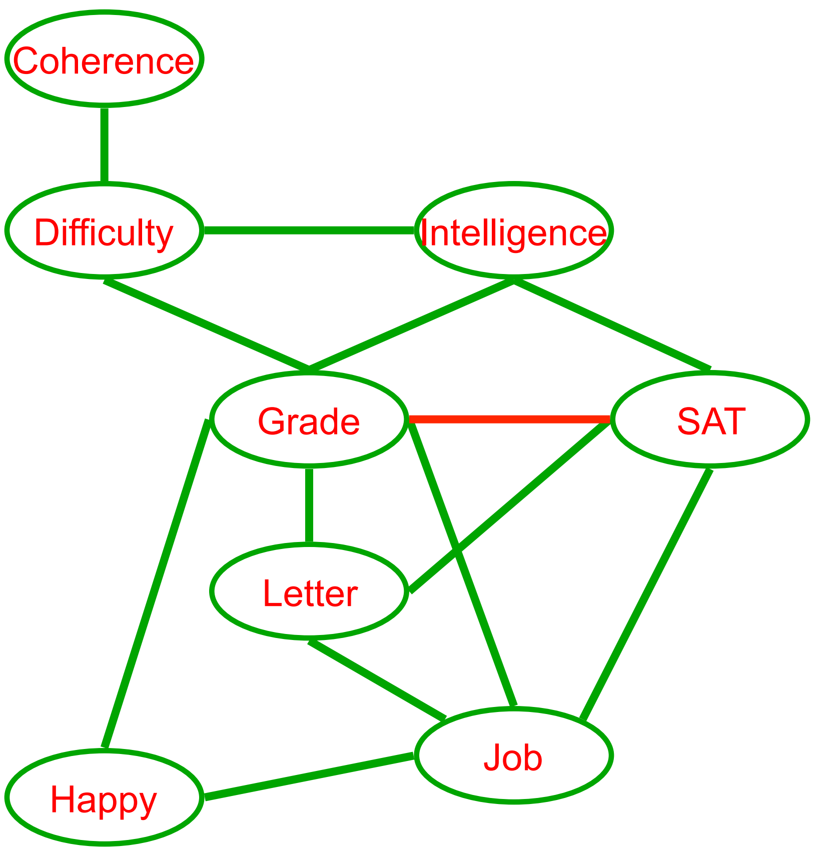


Elimination order:

{C,D,I,S,L,H,J,G}

- **Theorem**: Finding best elimination order is NP-complete:
  - Decision problem: Given a graph, determine if there exists an elimination order that achieves induced width ≤ K
- **Interpretation**:
  - Hardness of finding elimination order in addition to hardness of inference
  - Actually, can find elimination order in time exponential in size of largest clique – same complexity as inference

Minimum (weighted) fill heuristic



Elimination order:

{C,D,I,S,L,H,J,G}

- **Min (weighted) fill heuristic**
  - Often very effective
- Initialize unobserved nodes **X** as unmarked
- For k = 1 to |**X**|
  - O(next) ← unmarked var whose elimination adds fewest edges
  - Mark X
  - Add fill edges introduced by eliminating X
- Weighted version:
  - Consider size of factor rather than number of edges

Demo

- http://www.cs.us.es/~cgdiaz/CIspace/bayes.html

BN: Exact Inference: What you need to know

- Types of queries
  - Conditional probabilities / Marginals
  - maximum a posteriori (MAP)
  - Marginal-MAP
  - Different queries give different answers

- Hardness of inference
  - Exact and approximate inference are NP-hard
  - MAP is NP-complete
  - Conditional Probabilities #P-complete
  - Marginal-MAP is much harder (NP^{PP}-complete)

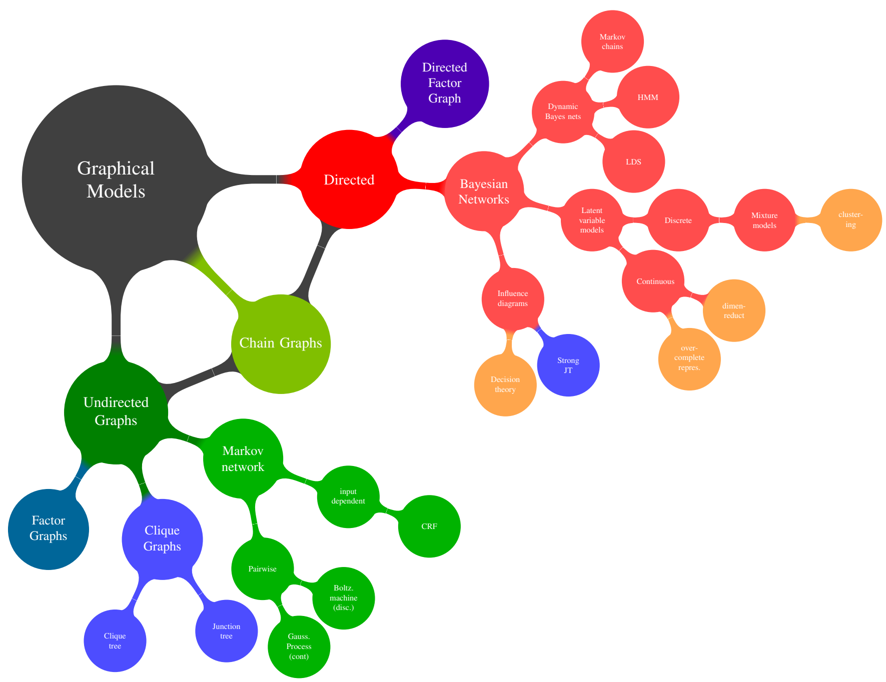
- Variable elimination algorithm
  - Eliminate a variable:
    - Combine factors that include this var into single factor
    - Marginalize/Maximize var from new factor
  - Efficient algorithm ("only" exponential in induced-width, not number of variables)
    - If you hear: "Exact inference only efficient in tree graphical models"
    - You say: "No! Any graph with low induced width"

- Elimination order is important!
  - NP-complete problem
  - Many good heuristics

Main Issues in PGMs

- Representation
  - How do we store $P(X_1, X_2, \ldots, X_n)$
  - What does my model mean/imply/assume? (Semantics)
- Learning
  - How do we learn parameters and structure of $P(X_1, X_2, \ldots, X_n)$ from data?
  - What model is the right for my data?
- Inference
  - How do I answer questions/queries with my model? such as
  - Marginal Estimation: $P(X_5 \mid X_1, X_4)$
  - Most Probable Explanation: argmax $P(X_1, X_2, \ldots, X_n)$

New Topic: Markov Nets / MRFs



Synonyms

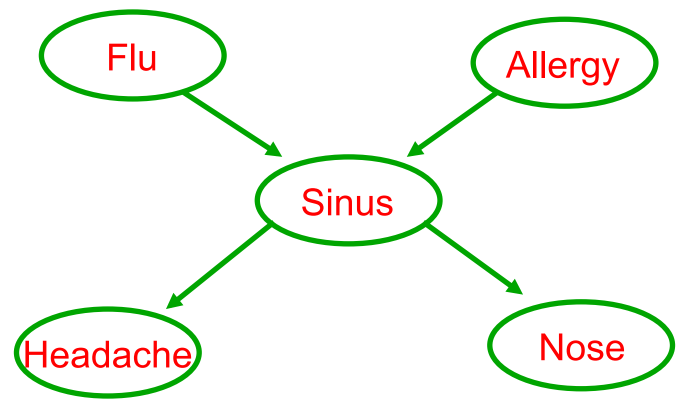
- Markov Networks

- Markov Random Fields
- Gibbs Distribution

- In vision literature
  - MAP inference in MRFs = Energy Minimization

A general Bayes net

- Set of random variables
- Directed acyclic graph
  - Encodes independence assumptions
- CPTs
  - Conditional Probability Tables
- Joint distribution:



$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P\left(X_i \mid \mathbf{Pa}_{X_i}\right)$$

Markov Nets

- Set of random variables
- Undirected graph
  - Encodes independence assumptions
- Unnormalized Factor Tables
- Joint distribution:
  - Product of Factors

Local Structures in BNs

- Causal Trail
  - $X \rightarrow Y \rightarrow Z$
- Evidential Trail
  - $X \leftarrow Y \leftarrow Z$
- Common Cause
  - $X \leftarrow Y \rightarrow Z$
- Common Effect (v-structure)
  - $X \rightarrow Y \leftarrow Z$

Local Structures in MNs

- On board

Active Trails and Separation

- A path X_1 – … – X_k is **active** when set of variables **Z** are observed
  - if none of $X_i \in \{X_1,\ldots,X_k\}$ are observed (are part of **Z**)

- Variables **X** are **separated** from **Y** given **Z** in graph
  - If no active path between any $X \in \mathbf{X}$ and any $Y \in \mathbf{Y}$ given **Z**

Independence Assumptions in MNs

- **Separation** defines global independencies
- **Pairwise Markov Independence**:
  - Pairs of non-adjacent variables A,B are independent given all others
- **Markov Blanket**:
  - Variable A independent of rest given its neighbors