

Markov networks (sequel)

CS B553
Spring 2013

Announcements

- Assignment 2 posted!
 - Implement a Part-of-Speech tagger
 - With Bayes nets and variable elimination
- Quiz 2 next week!



Some facts

- Fact 1: The web contains a tremendous amount of very useful information.
- Fact 2: It is nearly impossible for us to design completely novel assignments.
- Thus there's a good chance that you'll find helpful information about assignments on the web.
- There's an important distinction between *being resourceful* and *plagiarizing or cheating*.

Integrity Expectations

- When you turn in (or present) something, you are implying that you personally created all of it, including:
 - The ideas
 - Source code
 - Report (including all text, figures, tables, etc.)
 - Slides (including all text, figures, etc.)
- For ideas (or text, figures, slides, code, etc.) that came from someone else, you must give credit
 - A citation that identifies the source, and explicitly states which ideas you borrowed from someone else

Situation #1

- You need a description of Markov Nets in your paper. You find a sentence in a paper (or Wikipedia) that explains Markov Nets really well.
 - **Bad:** Copy the sentence word-for-word into your paper, without quotes or a citation.
 - **Bad:** Copy the sentence, rephrasing it a bit to match your style, without quotes or a citation.
 - **Better:** Put the sentence inside quotation marks, and give a citation. Or describe Markov Nets in your own words, based on your own understanding, and give a citation.

Acceptable options

"SIFT is a rotation and scale invariant feature and is robust to some variations of illuminations, viewpoints, and noise." [5]

SIFT [4], the Scale Invariant Feature Transform, is invariant to rotation and scale and is not sensitive to minor changes in illumination, viewpoints, and noise. [5]

[4] Lowe, D. "Distinctive Image Features from Scale-Invariant keypoints", in *International Journal of Computer Vision*, 2004.

[5] Wang, C-C and K-C Wang, "Hand Posture Recognition using Adaboost with SIFT for Human Robot Interaction," In *Proceedings of ICRA*, 2007.

Situation #2

- Your program needs a Variable Elimination algorithm. You find an implementation online.
 - Bad: Copy and paste the VE code into your source code file.
 - Bad: Copy and paste the VE code into your source code file, making changes to it to fit the needs of your application.
 - Better: Copy and paste the VE code into your source code file. Make changes if necessary to fit your application. Add comments to your source code making clear where the code came from, and also mention this in your report.
- (This only prevents you from violating academic integrity expectations. If implementing VE was part of the assignment, you'll still not get a good grade.)

Situation #2

- Your program needs a Variable Elimination algorithm. You find an implementation online.
 - Bad: Copy and paste the VE code into your source code file.
 - Bad: Copy and paste the VE code into your source code file, making changes to it to fit the needs of your application.
 - Better: Copy and paste the VE code into your source code file. Make changes if necessary to fit your application. Add comments to your source code making clear where the code came from, and also mention this in your report.
 - Better yet: Read the code to understand the ideas behind it. Then, without looking at the code, create your own implementation based on your newfound understanding. Cite the code you consulted as a reference.

Situation #3

- You need an algorithm for doing X. You'd like to use the algorithm created or inspired by Y.
 - Bad: Present the idea in your paper (or presentation, etc.) as if it were your own.
 - Bad: Present the idea in your paper (or presentation, etc.), never mention where it came from.
 - Better: Cite the source of the idea, even if you modified it or developed it substantially. Do this even if Y never published the idea.
- Better example:** We present a simple algorithm for computing invariant descriptors. The algorithm was inspired by [4] and [5], and was originally suggested by George W. Bush.

Situation #4

- Your graphical model inference algorithm doesn't work as well as you'd like.
 - Bad: Forge experimental results.
 - Bad: Repeatedly change experimental conditions until you find one that benefits your algorithm. Don't mention this repeated experimentation in your paper.
 - Better: Honestly report the sub-standard results.
 - Better: Try your algorithm in a different setting, or in a different application, or on a different dataset. Find other strengths of your approach (running time, etc).
- It's okay to emphasize the strengths of your work, without misrepresenting or hiding the weaknesses.

Maintaining academic integrity

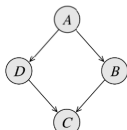
- We look for and prosecute cases of academic integrity
 - Need to ensure fairness to other students
 - Need to maintain the value of an IU degree
- University policy dictates a specific procedure that instructors *must* follow
 - Meeting with student
 - Report to Dean of Students
 - Various opportunities for appeal
- Penalties are case-dependent
 - Typically grade penalty
 - Up to and including expulsion from the university

The bottom line

- As a grad student, your goal is *not* to get good grades
 - Big Secret: Grades are mostly meaningless
 - Your goal is to launch your scientific and professional career (and to get acceptable grades)
- The scientific community is built on trust
 - You need credibility to be able to publish, talk at conferences, get a job, etc.
 - Credibility takes years to develop.
 - Credibility can be destroyed with a single bad decision.
- Please take academic integrity seriously
 - Before cheating, come talk to us to get help!

Bayes networks

- Bayes networks model dependencies between variables as directed graphs
 - Nodes represent random variables
 - Edges represent direct correlation
- We can write a joint probability distribution as,

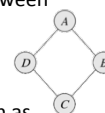


$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | \text{Pa}(X_i))$$

Markov networks

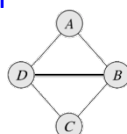
- Markov networks model dependencies between variables as undirected graphs
 - Nodes represent random variables
 - Edges represent direct correlation
- We can write a joint probability distribution as,

$$P(\mathbf{X}) = P(X_1, \dots, X_N) = \frac{1}{Z} \phi_1(\mathbf{A}_1) \cdot \phi_2(\mathbf{A}_2) \cdot \dots \cdot \phi_N(\mathbf{A}_N)$$
 - Where Z is a normalizing constant, and A_1, \dots, A_N are cliques (complete subgraphs) of the graph, and ϕ_i are factors



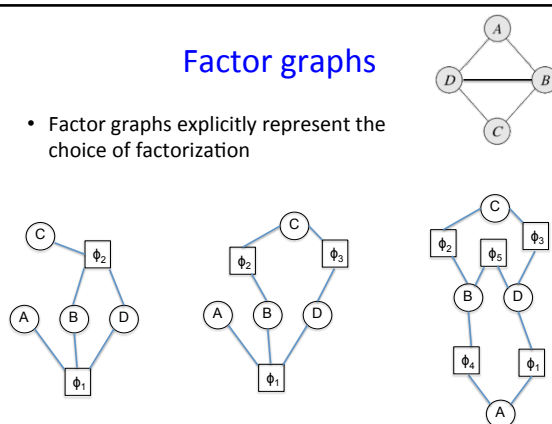
Cliques in a graph

- How many cliques are in this graph?
- Technically there are 12:
 - Two 3-cliques, five 2-cliques, four 1-cliques, one 0-clique
 - So the joint distribution could include these 12 factors
- It's always possible to factor over just the *maximal cliques* of the graph
 - Without loss of representational power
- For some distributions, it may be possible to factor over smaller cliques of the graph
 - There's no way of telling from the Markov network



Factor graphs

- Factor graphs explicitly represent the choice of factorization



Log-linear models

- From the Gibbs distribution,

$$P(\mathbf{X}) = P(X_1, \dots, X_N) = \frac{1}{Z} \phi_1(\mathbf{A}_1) \cdot \phi_2(\mathbf{A}_2) \cdot \dots \cdot \phi_N(\mathbf{A}_N)$$

- We can take logarithms,

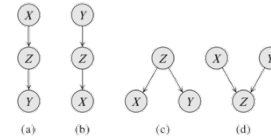
$$P(X_1, \dots, X_N) = \frac{1}{Z} \exp(\log \phi_1(\mathbf{A}_1) + \log \phi_2(\mathbf{A}_2) + \dots + \log \phi_N(\mathbf{A}_N))$$

$$= \frac{1}{Z} \exp\left(-\sum_i f_i(\mathbf{A}_i)\right)$$

where $f_i(\mathbf{A}_i) = -\log \phi_i(\mathbf{A}_i)$ is called an *energy function*

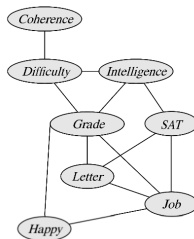
Bayes net independencies

- Variables X and Y are independent conditioned on a set of observed variables Z if there is no *active trail* between X and Y
- A trail is active given a set of nodes Z if
 - In any “v-structure” (type (d) below) along the trail, the middle node or one of its descendants is in Z
 - No other node along the trail is in Z

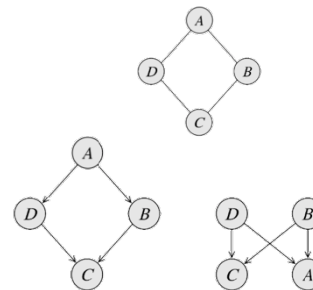


Markov net independencies

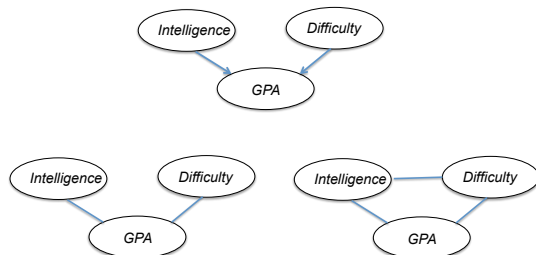
- Variables X and Y are independent conditioned on a set of variables Z if there is no *active path* between X and Y
- A path is active given a set of nodes Z if the path does not traverse any node in Z



Some Markov nets cannot be represented as Bayes nets...



And some Bayes nets cannot be represented as Markov nets...



Converting between Markov and Bayes net representations

- Some graphical models cannot be converted from one representation to the other
- What properties of a graphical model determine this “compatibility?”

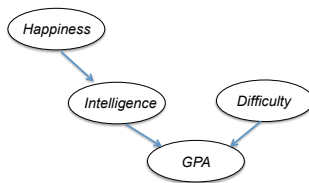
Moral graph

- In a *moral graph*, all parents are connected together
 - That is, if some node X has a set of parents Y , then there are edges directly connecting the nodes of Y together
- The *moralized graph* of a dag is an undirected graph that includes an edge between X and Y if either:
 - there's an edge between X and Y (or Y and X) in the dag, or
 - X and Y are both parents of the same node

Bayes nets to Markov nets

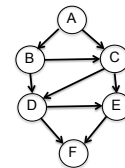
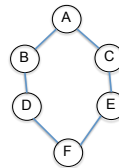
- For any Bayes net G , we can find a moralized graph $M[G]$. The independence assumptions of the Markov net $M[G]$ are a subset of those of G .
 - But G may have additional independencies not in $M[G]$
- If a Bayes net G is moral, then $M[G]$ is a perfect map.
 - I.e. $M[G]$ has exactly the same independencies as G

Is this graph moral?



Markov nets to Bayes nets

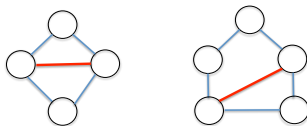
- What edges would we need in a Bayes net "translation" of this Markov net?



- The right graph expresses a subset of the independencies of the left graph

Chordal graphs

- A *chord* in a graph is an edge connecting two non-consecutive nodes of a loop



- In a *chordal graph*, every loop of 4 or more nodes has a chord
 - I.e., the longest minimal loop is a triangle

Markov nets to Bayes nets

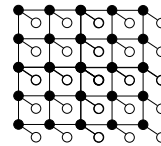
- For any Markov net H , we can construct a Bayes net that encodes a subset of H 's independencies
 - This Bayes net will be chordal
- A Bayes net that is a *perfect map* for a given Markov net H exists if and only if H is a chordal graph.

Summary

- We can “perfectly” convert...
 - a Bayes net to a Markov net iff the Bayes net is moral
 - a Markov net to a Bayes net iff the Markov net is chordal
 - where “perfectly” means that the two graphs express exactly the same conditional independencies

Pairwise Markov networks

- In a pairwise Markov network (aka pairwise Markov Random Field or MRF), the max clique size is 2
 - Grid graphs are an especially popular special case



MRFs for images

- MRFs have revolutionized computer vision and image processing over the last decade
- A sample application: Image reconstruction
 - Given a noisy image, estimate the original noise-free image

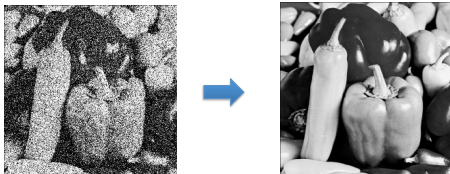


Image reconstruction

- In probabilistic terms, we're given a noisy image N and we want to estimate the original image O ,

$$P(O|N) \propto P(N|O)P(O)$$

- E.g. estimate the most likely original image,

$$O^* = \arg \max_O P(N|O)P(O)$$

- We might assume that noise at each pixel is independent given original pixel data,

$$P(N|O) = \prod_x \prod_y P(N_{x,y} | O_{x,y})$$

What about the prior, $P(O)$?

$$O^* = \arg \max_O P(N|O)P(O)$$

- Prior $P(O)$ very difficult to define
 - For a 1 megapixel camera, there are $2^{24000000}$ possible images, so the joint distribution has this many entries!
- Intuitively, captures the fact that some images are much more likely than others

$$P(\text{pepper image}) > P(\text{noise image})$$

- Simple prior: Encourage “smoothness”
 - Neighboring pixels should generally have similar intensities

Application: Image reconstruction

- Given a noisy image, infer original image
- Express problem naturally in terms of an MRF
 - Image is stored as a sampled function on a grid
 - We can observe noisy pixel values, and we'd like to estimate the original, clean pixel values
 - Important constraint: In images of real-world scenes, one pixel's color is correlated with that of its neighbors
 - The pairwise factors model this constraint
- Problem can be solved by doing inference on the Markov network

