Exploiting Copula Parameterizations in Graphical Model Construction and Learning

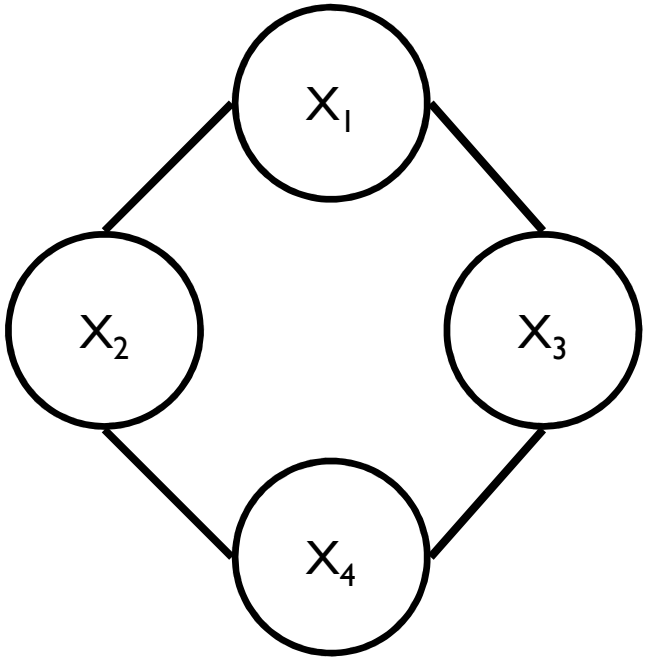
Ricardo Silva,
University College London

NIPS 2011 Workshop on Copulas in Machine Learning – Sierra Nevada, Spain

Joint work with
Robert B. Gramacy,**Charles Blundell and Yee Whye Teh**

Graphical Models

- Languages for encoding conditional independence constraints and factorizations
- Aids parameterization and computation
- From "local" parameterizations to joint distributions



$$X_1 \perp\!\!\!\perp X_4 \mid \{X_2, X_3\}$$

$$P(X_1, \ldots, X_4) \;\alpha\; \Pi\phi_{\theta(A)}(\mathbf{X}_A)$$

This Talk: A Tale of Two Models

- Scope: two main classes of graphical models
  - and how to turn them into copula models
  - and how copula theory helps constructing them
- Learning
  - Bayesian inference, point estimators, a combination of both, MCMC methods, etc.
- Recent work in progress (Buyer Beware)

Focus

I'll take for granted why copula models are good, and focus on why they can make graphical model construction easier

Part I

Trees, Mixtures of Trees and Non-parametric Bayesian Learning

(Silva and Gramacy, 2009)

Contribution

- Bayesian inference in multivariate distributions with Markov random fields

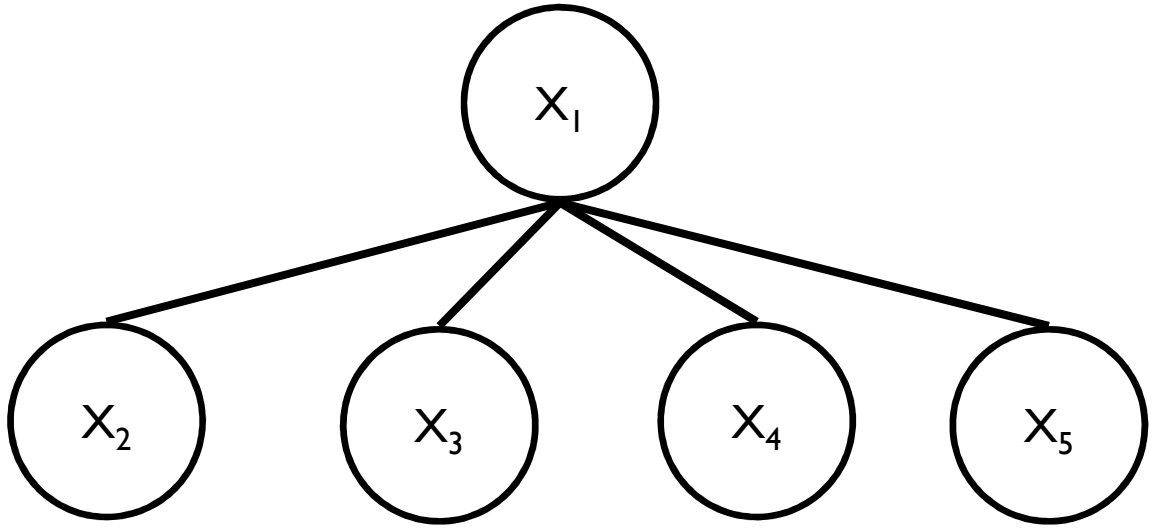
- Wanted: computationally tractable MCMC

$$L(\theta; X_1, \ldots, X_p) = \frac{1}{Z(\theta)} \prod_A \phi_{\theta(A)}(\mathbf{X}_A)$$

For computational tractability, might want to forfeit full generality. Say, use trees.

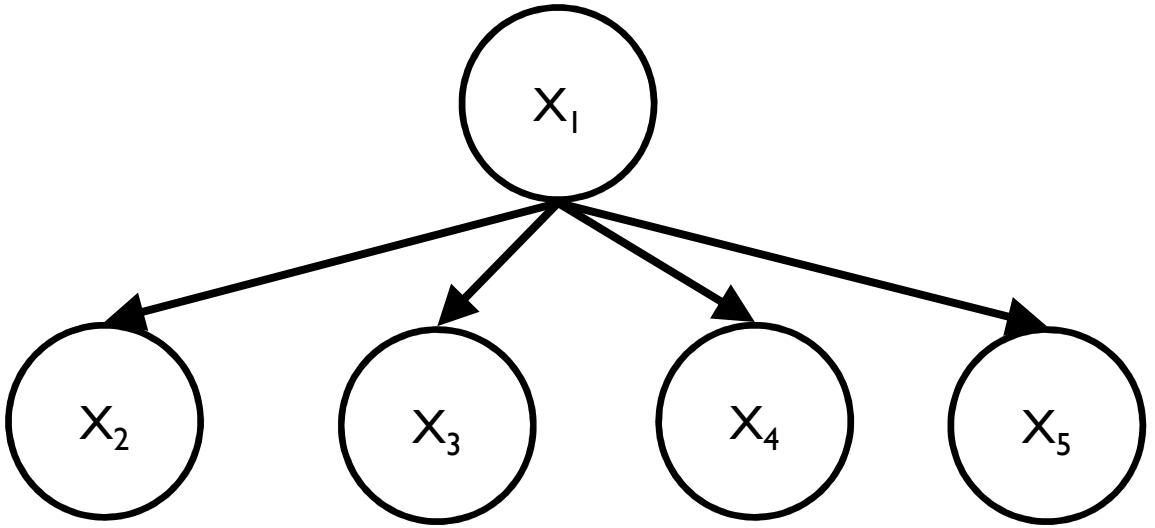
Possibly Nasty Side Effects

- Innocent looking decomposable model:



Possibly Nasty Side Effects

- Alternative encoding (zero mean Gaussian case):



$X_1 \sim N(0, v)$

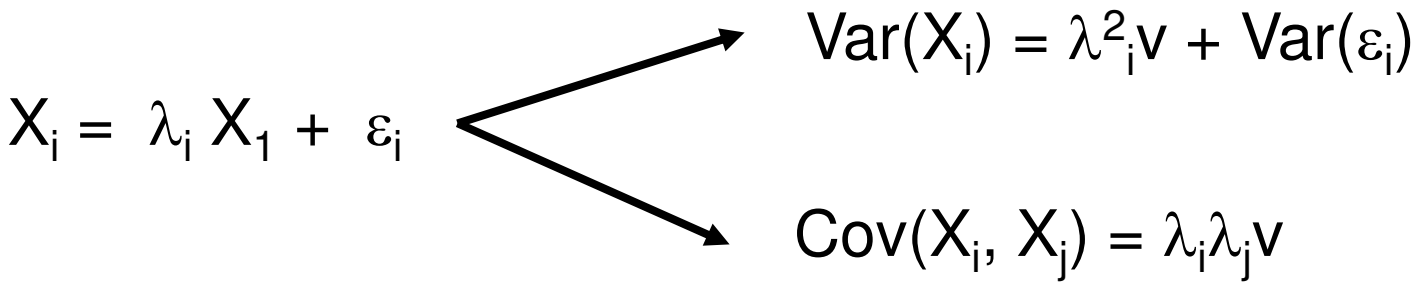
$X_i = \lambda_i X_1 + \varepsilon_i$

$\mathrm{Var}(X_i) = \lambda^2_i v + \mathrm{Var}(\varepsilon_i)$

$\mathrm{Cov}(X_i, X_j) = \lambda_i \lambda_j v$

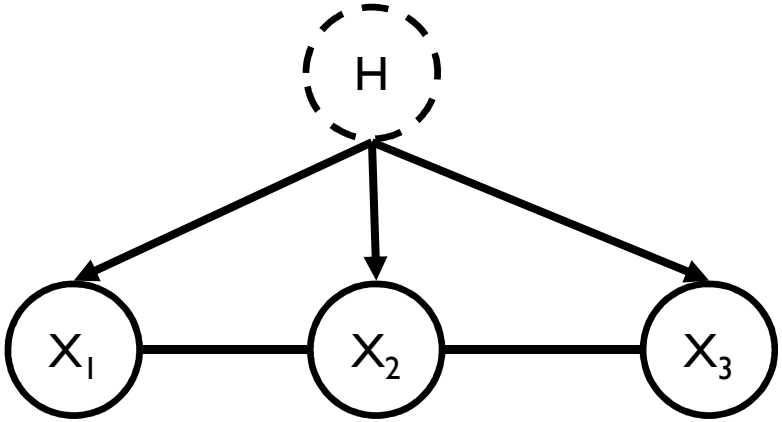
Possibly Nasty Side Effects

- Parameters encode all sorts of features. Independence constraints/factorizations however affect your joint in (possibly) hard-to-understand ways
  - But this distributed encoding given by graphical models is supposed to be a "feature", not a "bug"



Possibly Nasty Side Effects

- Same phenomenon with mixture models



- Marginal distribution over H affects both resulting dependence between variables and their marginal distribution

Now What?

Another type of modularity comes to rescue: parameterize each *univariate marginal*, then parameterize *dependence structure*

Now What?

Let's shed a tear for budget cuts on the dependence structure, but let's preserve the univariate marginals

Parameterizations

- Parameterization of choice: copula models
  - Kirshner, NIPS 2007
  - Marginal distributions are parameterized independently of dependency structure
  - Deal with intractability by *mixtures of tree-structured distributions*
- Methods: designing good proposals akin to reversible jump MCMC
- Evaluation: different measures on mixing behaviour

Copula Models

- Bivariate case (X_i, X_j)

$a_i = F_i(X_i)$

$$\{F_i(\cdot), F_j(\cdot), C(a_i(\cdot), a_j(\cdot))\}$$

Marginal of X_i

Marginal of X_j

Copula function: cumulative distribution with uniform marginals

- Example: Gaussian copula density

$$\Phi_\rho(u, v) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{u^2 + v^2 - 2\rho uv}{2(1-\rho^2)}\right)$$

Tree-Copula Models

- Multivariate (3+ variables) copulas hard to construct: usually boils down to introducing constraints too
- Alternative: tree-structured copulas (Kirshner, 2007)

Density function of data

Density function of marginal

Tree-copula density

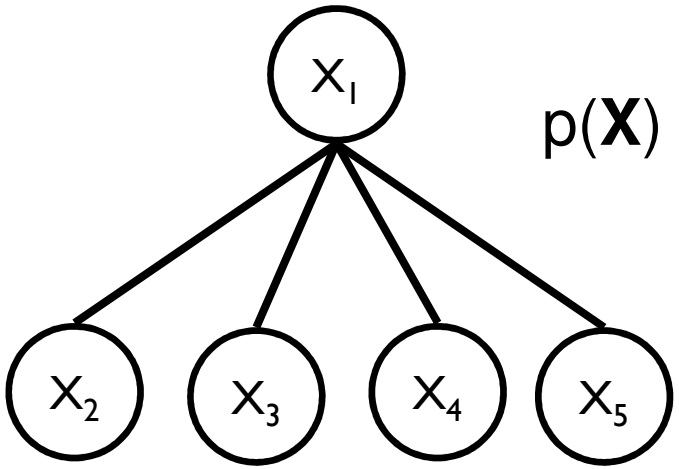
$$p(\mathbf{Y}|\mathcal{T},\Theta) = \left[\prod_{v=1}^{d} f_v(Y_v \mid \Lambda_v)\right] c_{\mathcal{T}}(\mathbf{a})$$

$$c_{\mathcal{T}}(\mathbf{a}) \equiv \prod_{\{u,v\}\in\mathcal{E}(\mathcal{T})} c_{uv}(a_u(Y_u), a_v(Y_v) \mid \Theta_{uv})$$

Uniform random variable

Derivation

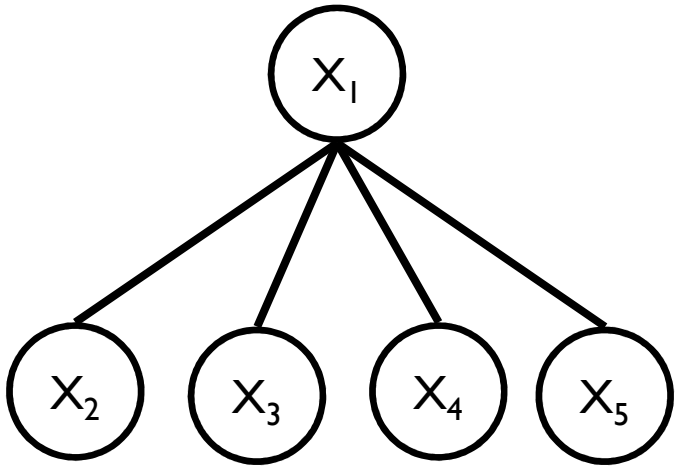
- Follows easily from the alternative marginal parameterization of trees. For instance



$$p(\mathbf{X}) = \frac{p(X_1, X_2)p(X_1, X_3)p(X_1, X_4)p(X_1, X_5)}{p(X_1)^3}$$

Hold On

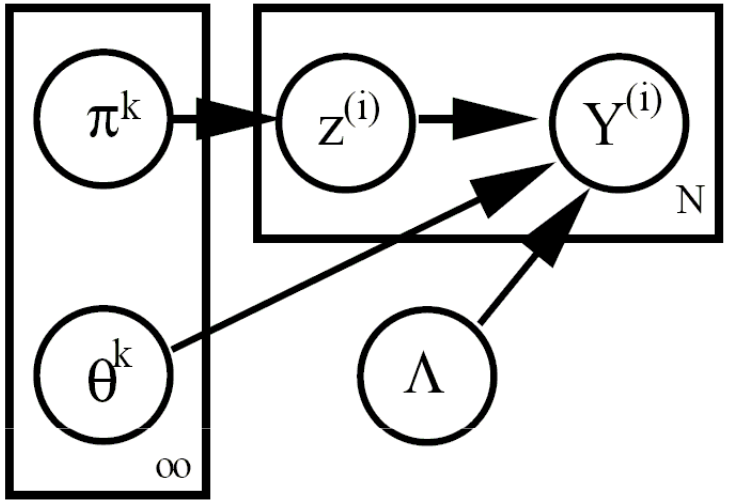
- If we can have a generic marginal parameterization, where do copulas simplify matters in this context?



- $P(X_1, X_2)$ and $P(X_1, X_3)$ have to agree on the same marginal. Copulas give you that for free

- *Variation independence*: no constraints between parameters in different factors

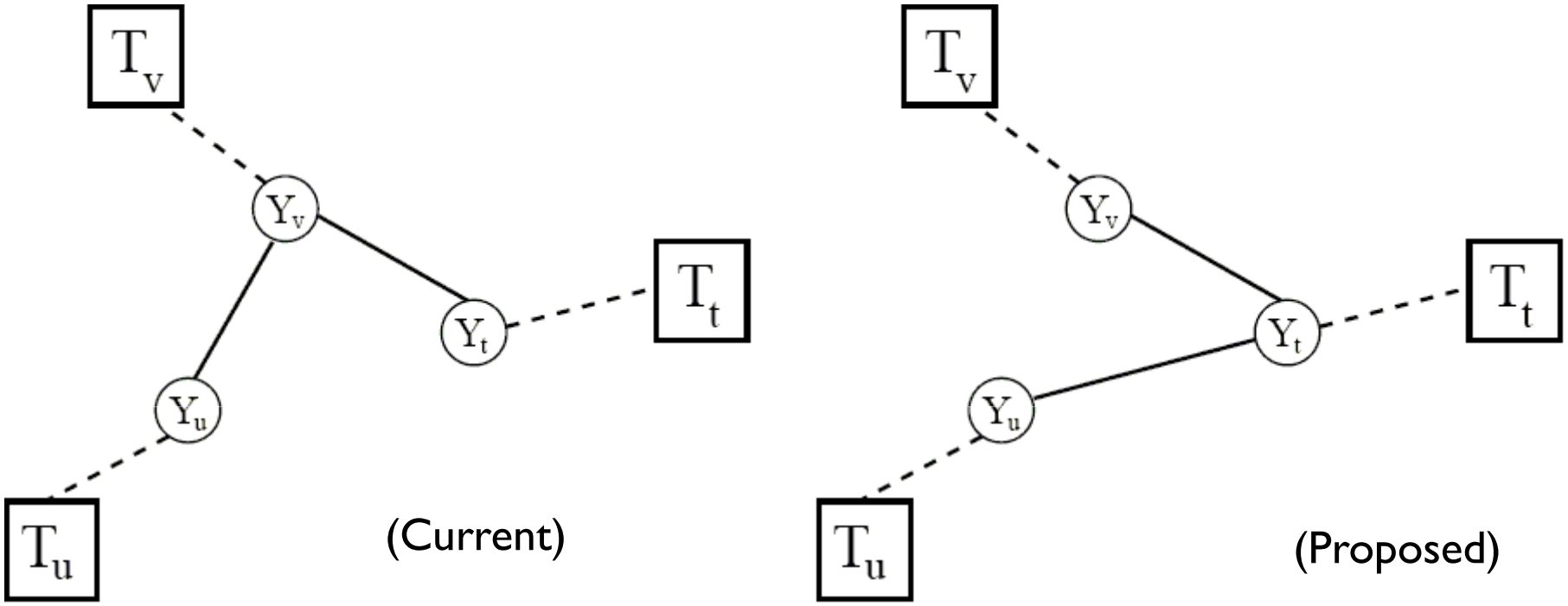
Priors for Mixtures of Tree-Copulas



Dirichlet process mixture when $K \to \infty$

$$
\begin{aligned}
Y^{(i)} \mid z^{(i)}, [\mathcal{T}], \Lambda, [\Theta] &\sim p(\,\cdot \mid \mathcal{T}_{z_i}, \Lambda, \Theta_{z_i}) \\
\Lambda &\sim p_\Lambda(\,\cdot\,) \\
\Theta_z &\sim p_\Theta(\,\cdot\,) \\
z^{(i)} \mid \pi &\sim \text{Discrete}(\,\cdot \mid \pi_1, \ldots, \pi_K) \\
\mathcal{T}_z &\sim T_0(\cdot) \\
\pi &\sim \text{Dirichlet}(\,\cdot \mid \alpha/K, \ldots, \alpha/K)
\end{aligned}
$$

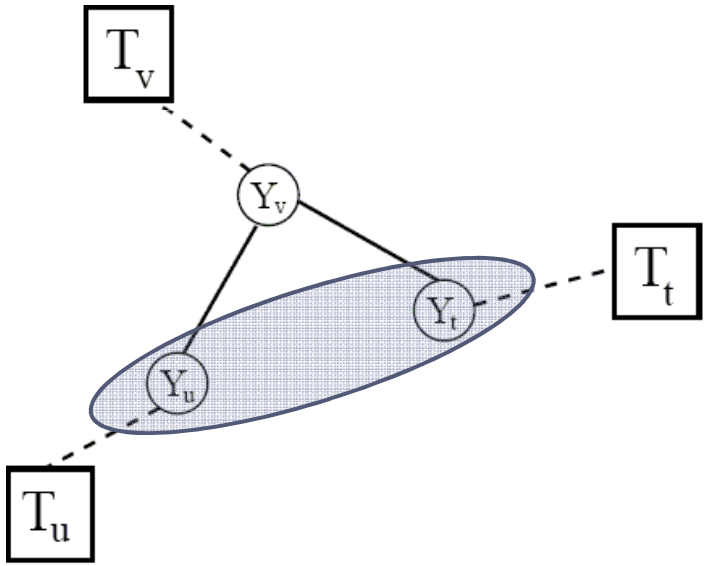
A Family of Proposals: Treeangular Moves



- Local change within a chain $Y_u - Y_v - Y_t$
- Can be made uniform over all possible trees
- Can traverse the whole tree space

Advantages

- "Small" departures from current tree
- Allow the proposal of sensible new parameter values corresponding to the new "active" copula



Idea: there is an *implied* copula function for the joint of Y_u and Y_t. Propose a new "direct" copula based on the implied one.

(A link to Reversible Jump MCMC)

- Full proposal: treeangular move + parameter proposal

Implied Copulas: Proposals

- Usually it is not possible to analytically compute the implied copula
- Take advantage of the treeangular move: tabulate numerical computations!
- Take advantage of the copula parameterization: map parameters into a dependence measure space

Proposal Template

treeangle move $Y_u - Y_v - Y_t \rightarrow Y_u - Y_t - Y_v$

1. calculate the rank correlation ρ_{uv} corresponding to θ_{uv}
2. calculate the rank correlation ρ_{ut} corresponding to the copula function implied by $Y_u - Y_v - Y_t$
3. find a ρ^0_{ut} that trades-off the following:
   (a) ρ^0_{ut} is "close" to ρ_{ut}
   (b) ρ_{uv} is "close" to the implied rank correlation of Y_u and Y_v as given by the path $Y_u - Y_t - Y_v$ and parameters θ_{tv} and $\theta(\rho^0_{ut})$
4. calculate θ^0_{ut} from ρ^0_{ut}
5. propose $\theta^\star_{ut}$ from a distribution parameterized by θ^0_{ut}

One Step Further: MCMC + Deterministic

- Propose treeangular moves by numerically *integrating out* copula parameters in the posterior joint of (θ_{uv}, θ_{vt}, θ_{ut})
- For one-parameter copulas, this boils down to three one-dimensional integrals (solved by a deterministic method)
- Account for numerical mistakes by Metropolis-Hastings

Experiments

Table 1: Comparison of the ratio of the average effective sample sizes for the different algorithms (H, S, T, E) in 9 different datasets, as explained in the text. In the table, d refers to the number of variables and N to the sample size. The respective proportion of accepted trees is given as the last three columns.

Dataset	d	N	H/S	H/T	H/E	T/S	T/E	H/Sn	H/Tn	H/En	AccS	AccT	AccH
CLOUD	10	1024	2.35	1.06	4.27	4.30	8.01	3.17	1.36	5.50	0.010	0.036	0.031
CONCRETE	9	1030	3.23	1.98	1.20	1.83	0.74	4.25	2.58	1.78	0.066	0.102	0.137
ECOLI	5	336	1.62	1.43	1.03	1.20	0.82	3.03	2.70	1.84	0.168	0.204	0.245
FIRE	7	517	1.34	1.15	2.17	1.14	2.15	1.79	1.58	3.07	0.147	0.178	0.218
GLASS	8	214	0.81	0.84	0.89	1.03	1.19	1.09	1.14	1.14	0.112	0.172	0.214
SEG	16	205	9.16	1.13	8.02	9.17	8.55	10.5	1.36	8.23	0.047	0.088	0.141
VOWEL	10	990	1.17	1.84	1.23	1.03	0.95	1.71	2.42	1.98	0.070	0.091	0.141
WDBC	30	569	6.38	3.04	4.86	5.52	6.96	7.00	3.22	4.02	0.028	0.088	0.110
YEAST	6	1484	0.94	0.67	1.18	1.40	1.77	1.48	1.08	1.80	0.208	0.262	0.315

Summary: Part I

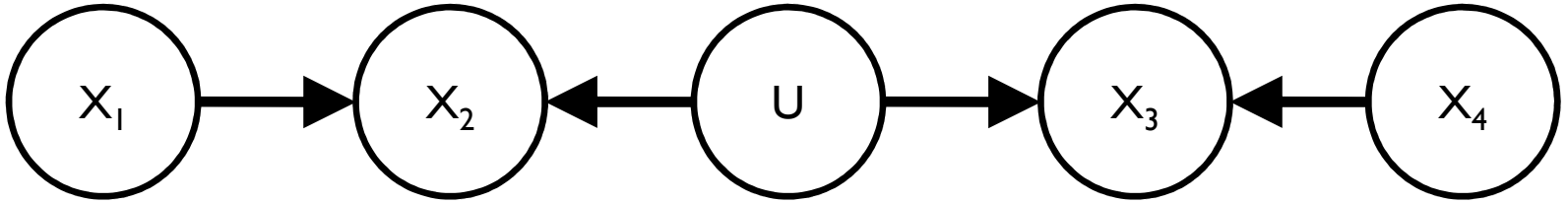
- Towards expanding Bayesian multivariate analysis
- Smart proposals make a difference
- Simple extension: constraining tree mixtures by forbidding some edges
- Hierarchical models (for marginals and copulas) are particularly of interest

Part II

Mixed Graph Modeling: Non-monotonic Independence Families, and Products of Copulas

(Silva, Blundell and Teh, 2011)

Directed Graphical Models



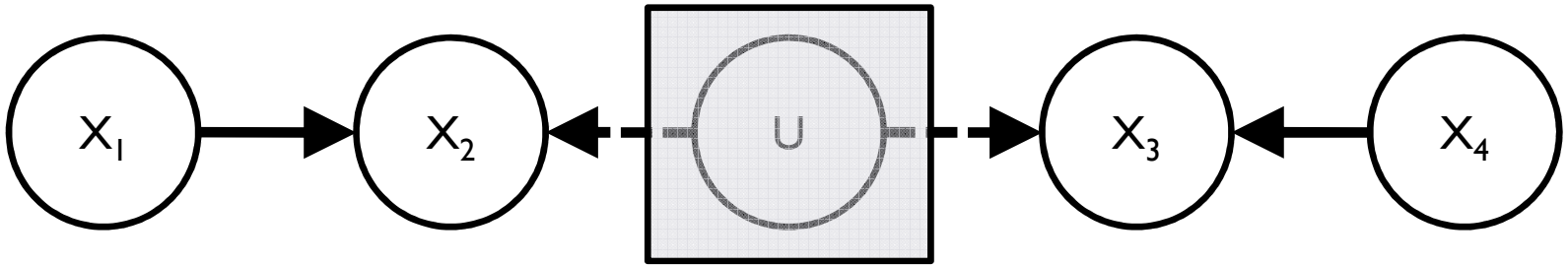
$$X_2 \perp\!\!\!\perp X_4$$

$$X_2 \not\perp\!\!\!\perp X_4 \mid X_3$$

$$X_2 \perp\!\!\!\perp X_4 \mid \{X_3, U\}$$

...

Marginalization



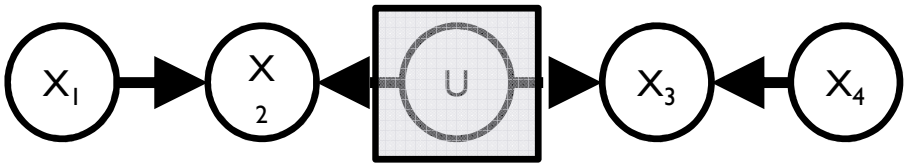
$$X_2 \perp\!\!\!\perp X_4$$

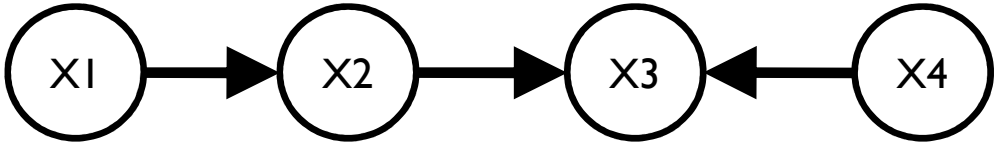
$$X_2 \not\perp\!\!\!\perp X_4 \mid X_3$$

~~$X_2 \perp\!\!\!\perp X_4 \mid \{X_3, U\}$~~

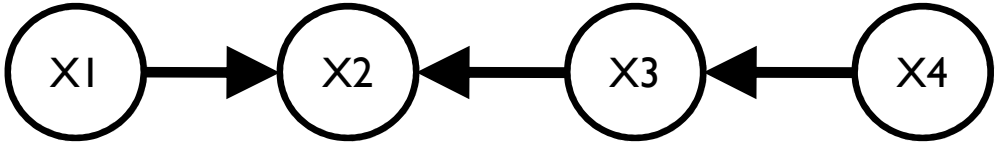
...

Marginalization

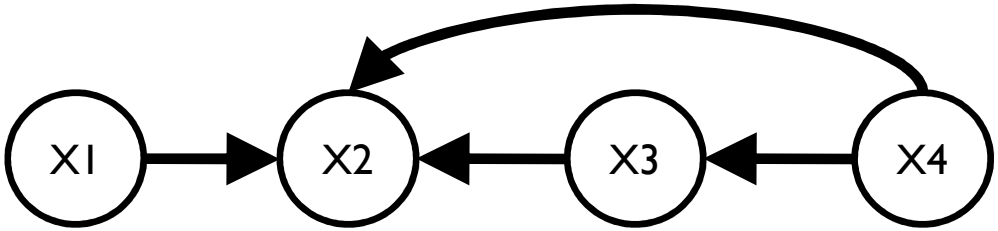




? No: $X_1 \perp\!\!\!\perp X_3 \mid X_2$



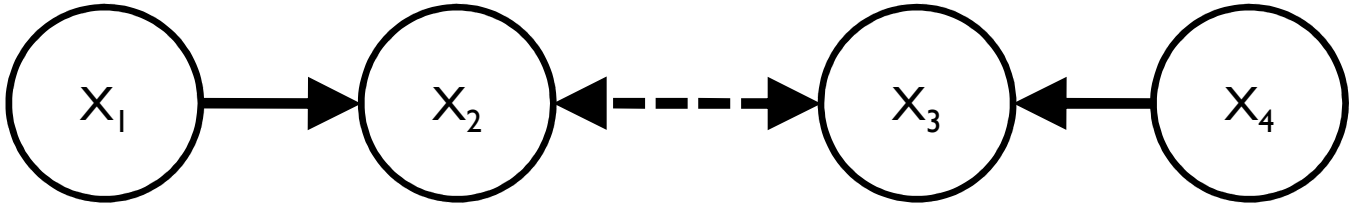
? No: $X_2 \perp\!\!\!\perp X_4 \mid X_3$



? OK, but not ideal

$X_2 \not\perp\!\!\!\perp X_4$

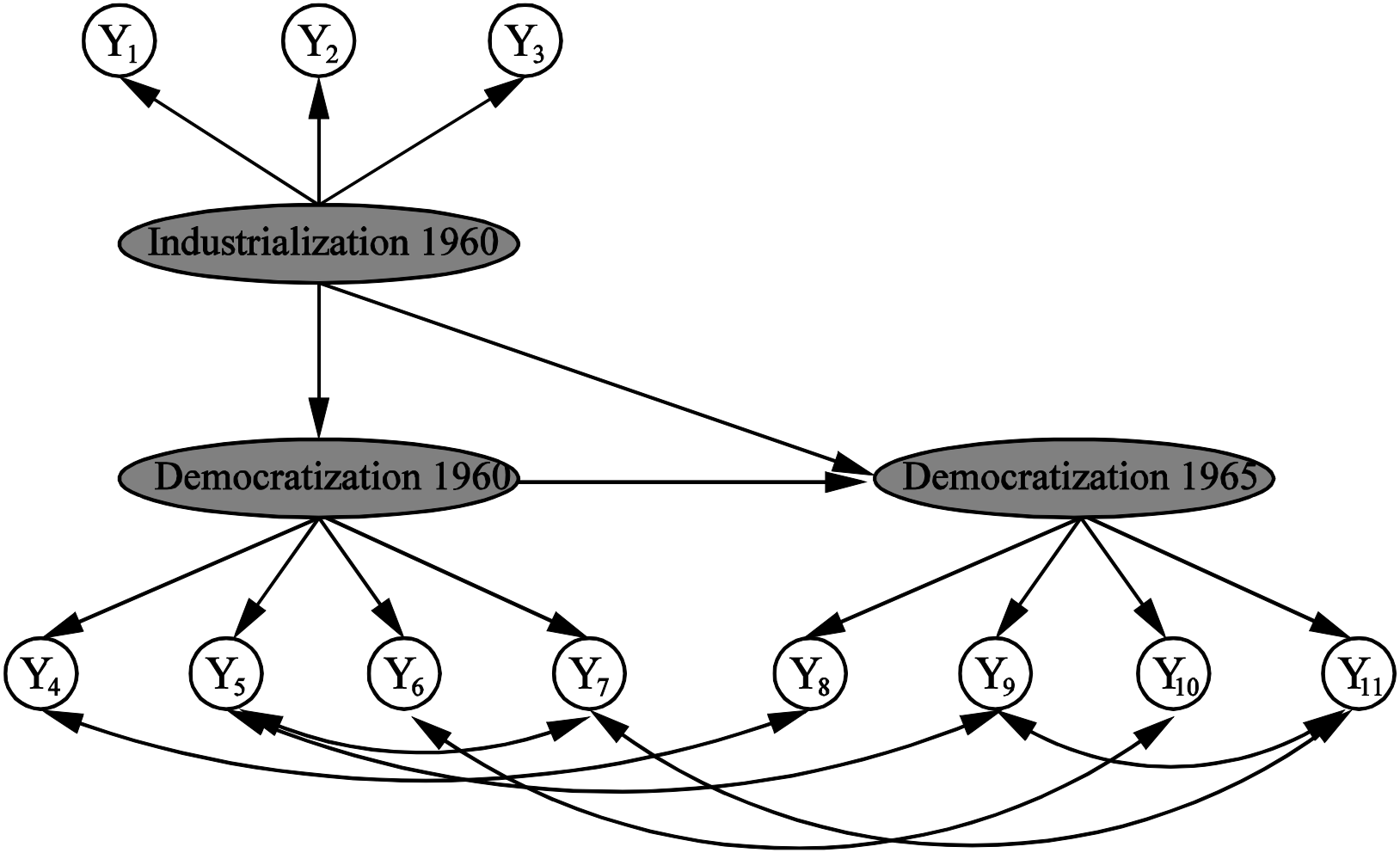
The Acyclic Directed Mixed Graph (ADMG)



- "Mixed" as in directed + bi-directed
  - See also: chain graphs
- "Directed" for obvious reasons
- "Acyclic" for the usual reasons
- Independence model is
  - Closed under marginalization (generalize DAGs)
  - Different from chain graphs/undirected graphs
  - Analogous inference calculus as DAGs: m-separation

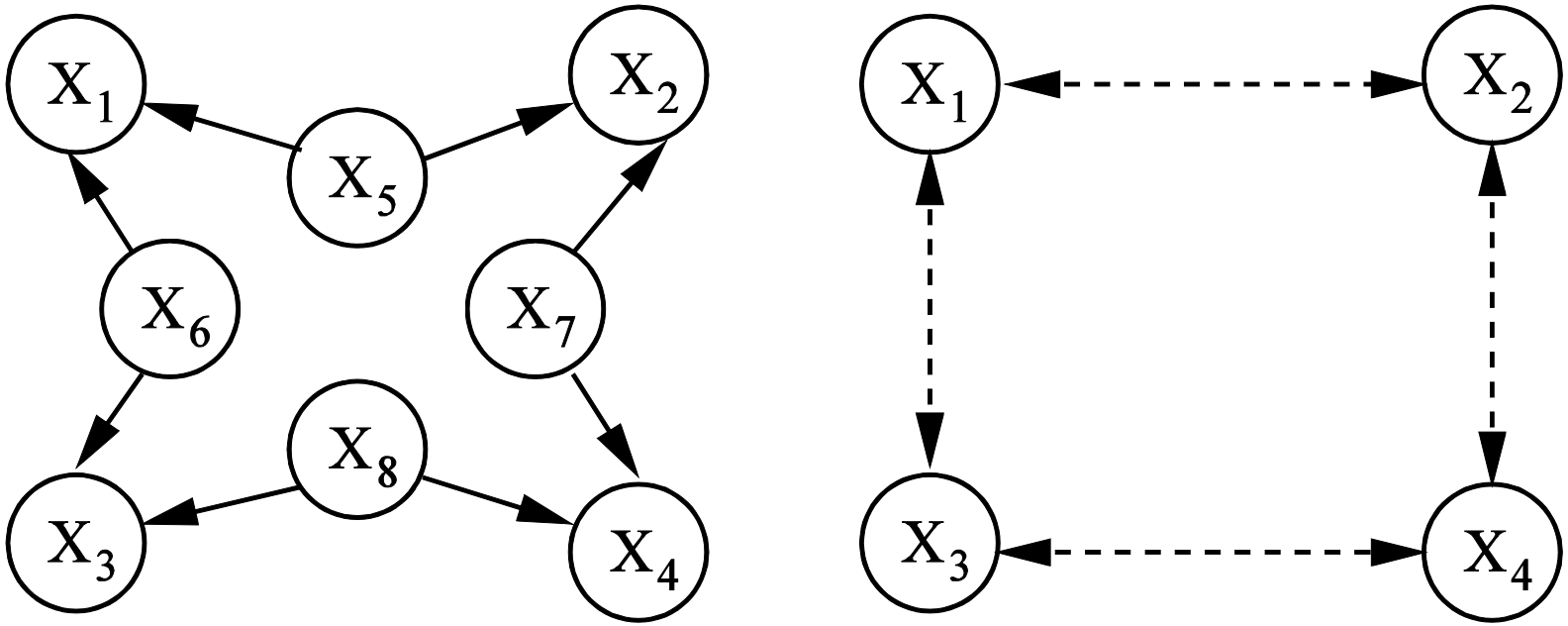
(Richardson and Spirtes, 2002)

Why Do We Care?

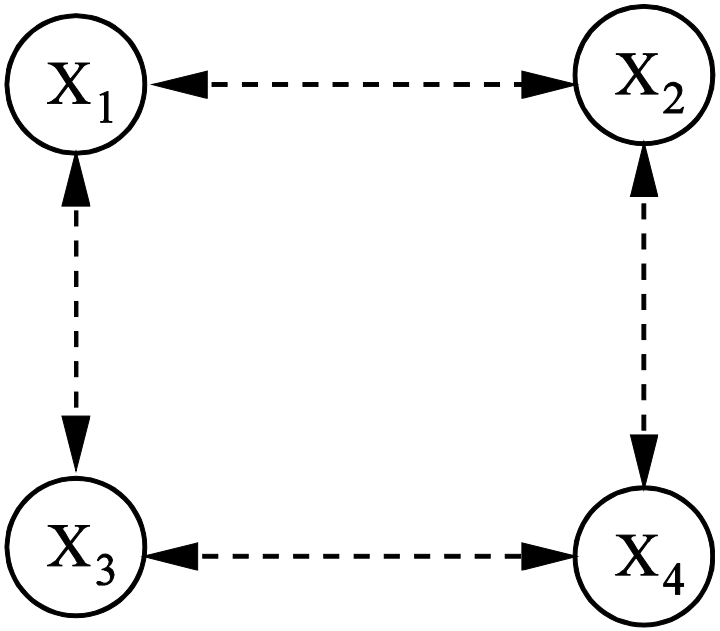


(Bollen, 1989)

The Gaussian Bi-directed Model



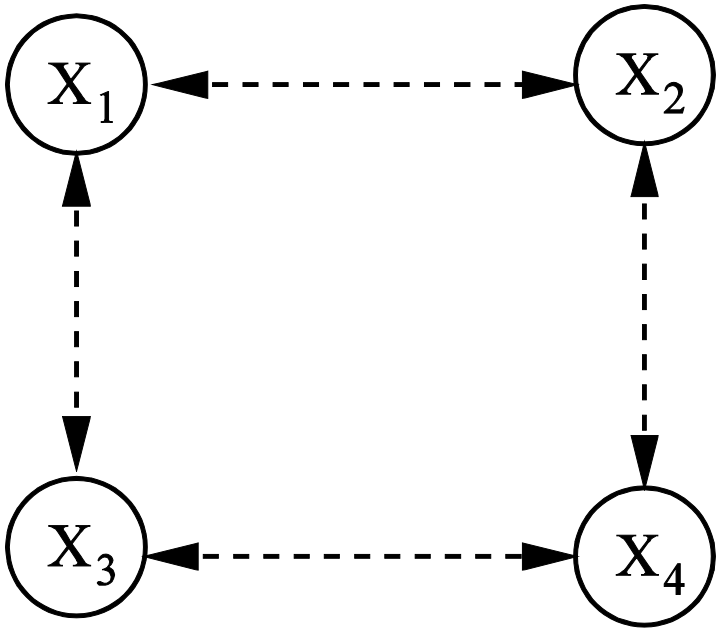
The Gaussian Bi-directed Case



$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & 0 \\ \sigma_{12} & \sigma_{22} & 0 & \sigma_{24} \\ \sigma_{13} & 0 & \sigma_{33} & \sigma_{34} \\ 0 & \sigma_{24} & \sigma_{34} & \sigma_{44} \end{bmatrix}$$

(Drton and Richardson, 2003)

Binary Bi-directed Case: the Constrained Moebius Parameterization



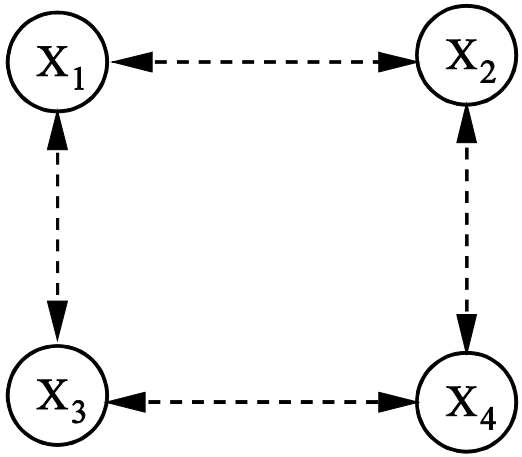
$$q_A \equiv P(X_A = 0)$$

$$P(X_A = 0, X_{V \setminus A} = 1) = \sum_{B : A \subseteq B} (-1)^{|B \setminus A|} q_B$$

(Drton and Richardson, 2008)

Binary Bi-directed Case: the Constrained Moebius Parameterization

- *Disconnected sets* are marginally independent. Hence, define q_A for connected sets only



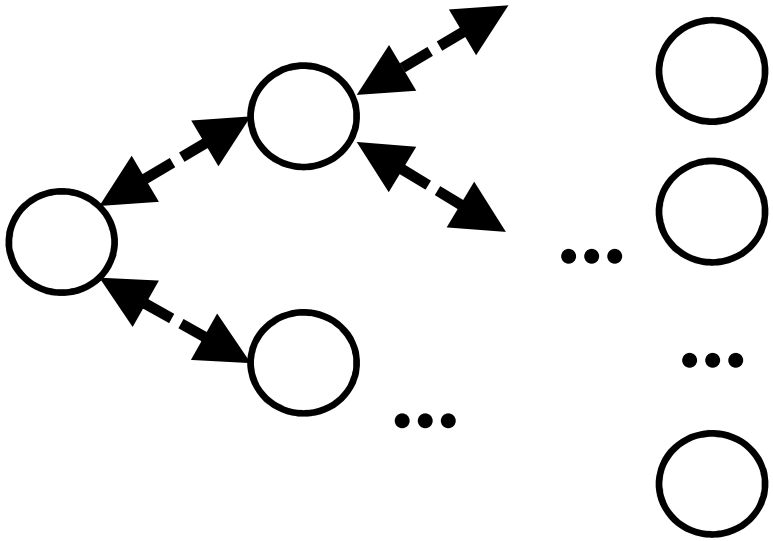
$$P(X_1 = 0, X_4 = 0) = P(X_1 = 0)P(X_4 = 0)$$

$$q_{14} = q_1 q_4$$

(However, notice there *is* a parameter q_{1234})

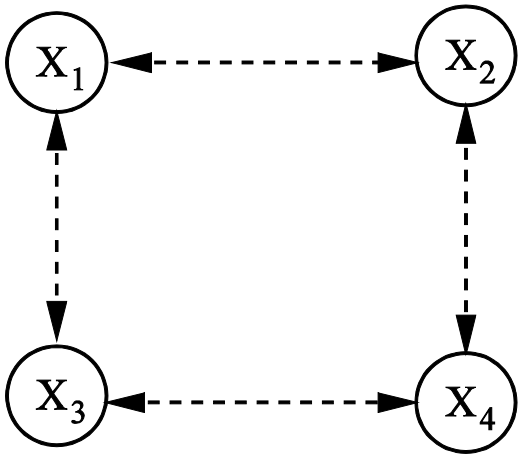
Binary Bi-directed Case: the Constrained Moebius Parameterization

- The good:
  - this parameterization is *complete*. *Every single binary bi-directed model* can be represented with it
- The bad:
  - Moebius inverse is intractable, and number of connected sets can grow exponentially even for trees



The Cumulative Distribution Network (CDN) Approach

- Parameterizing cumulative distribution functions (CDFs) by a product of functions defined over subsets
  - Sufficient condition: each factor is a CDF itself
  - Independence model: the "same" as the bi-directed graph... but with extra constraints



$$F(X_{1234}) = F_1(X_{12})F_2(X_{24})F_3(X_{34})F_4(X_{13})$$

$$X_1 \perp\!\!\!\perp X_4$$

$$X_1 \not\perp\!\!\!\perp X_4 \mid X_2 \quad \textbf{etc}$$

(Huang and Frey, 2011)

CDF $\Leftrightarrow$ PMF Relations

- $P(X = x_i) = P(X \leq x_i) - P(X \leq x_{i-1})$

- $P(X = x_i, Y = y_j) =$

  $P(X \leq x_i, Y \leq y_j) - P(X \leq x_{i-1}, Y \leq y_j)$
  $- P(X \leq x_i, Y \leq y_{j-1}) + P(X \leq x_{i-1}, Y \leq y_{j-1})$

- $P(X \leq x_i) = P(X \leq x_i, Y \leq \infty)$

Relationship

- CDN: the resulting PMF (usual CDF2PMF transform)

$$\sum_{z_1=0}^{1} \cdots \sum_{z_d=0}^{1} (-1)^{z_1+z_2+\ldots z_d} F(x_1 - z_1, \ldots, x_d - z_d)$$

- Moebius: the resulting PMF is equivalent

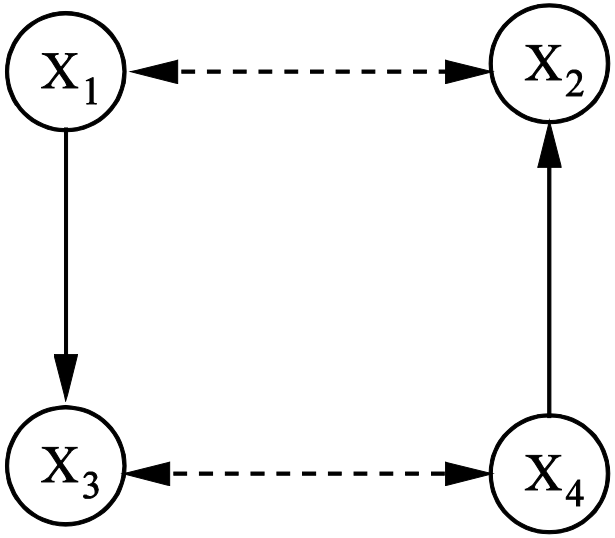
$$P(X_A = 0, X_{V \setminus A} = 1) = \sum_{B: A \subseteq B} (-1)^{|B \setminus A|} q_B$$

- Notice: $q_B = P(X_B = 0) = P(X_{\setminus B} \leq 1, X_B \leq 0)$
- However, in a CDN, parameters further factorize over cliques

$$q_{1234} = q_{12} q_{13} q_{24} q_{34}$$

The Mixed CDN Model (MCDN)

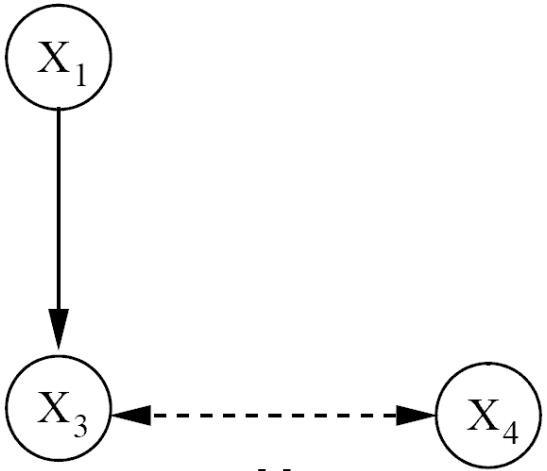
- How to construct a distribution Markov to this?



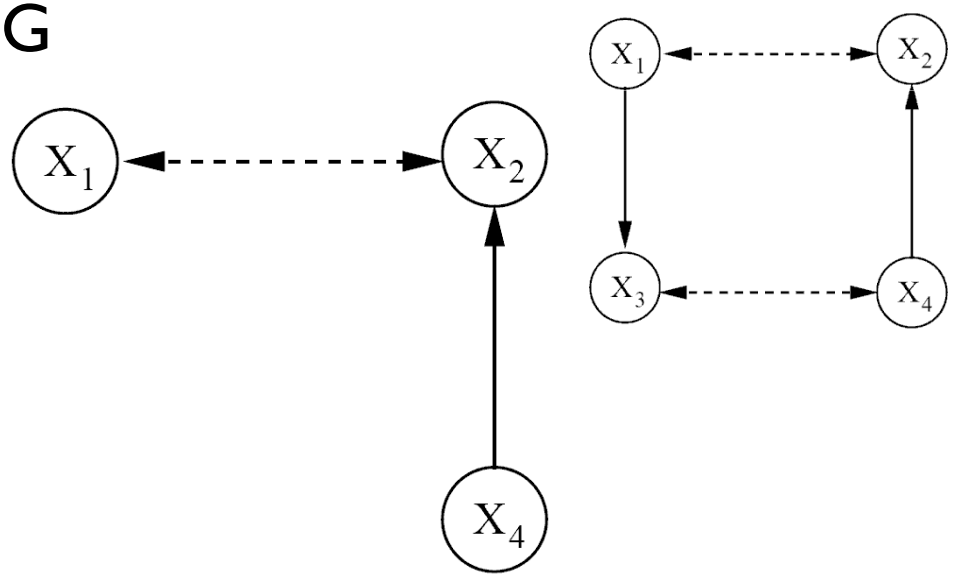
- The binary ADMG parameterization by Richardson (2009) is complete. Many nice properties, but with the same computational difficulties
  - And how to easily extend it to non-Gaussian, infinite discrete cases, etc.?

Step 1: The High-level Factorization

- Define a set of $P_i(\cdot \mid \cdot)$ Markov with respect to subgraph G_i – the graphs we obtain from looking at bi-directed components
- We can show the resulting distribution is Markov with respect to the ADMG



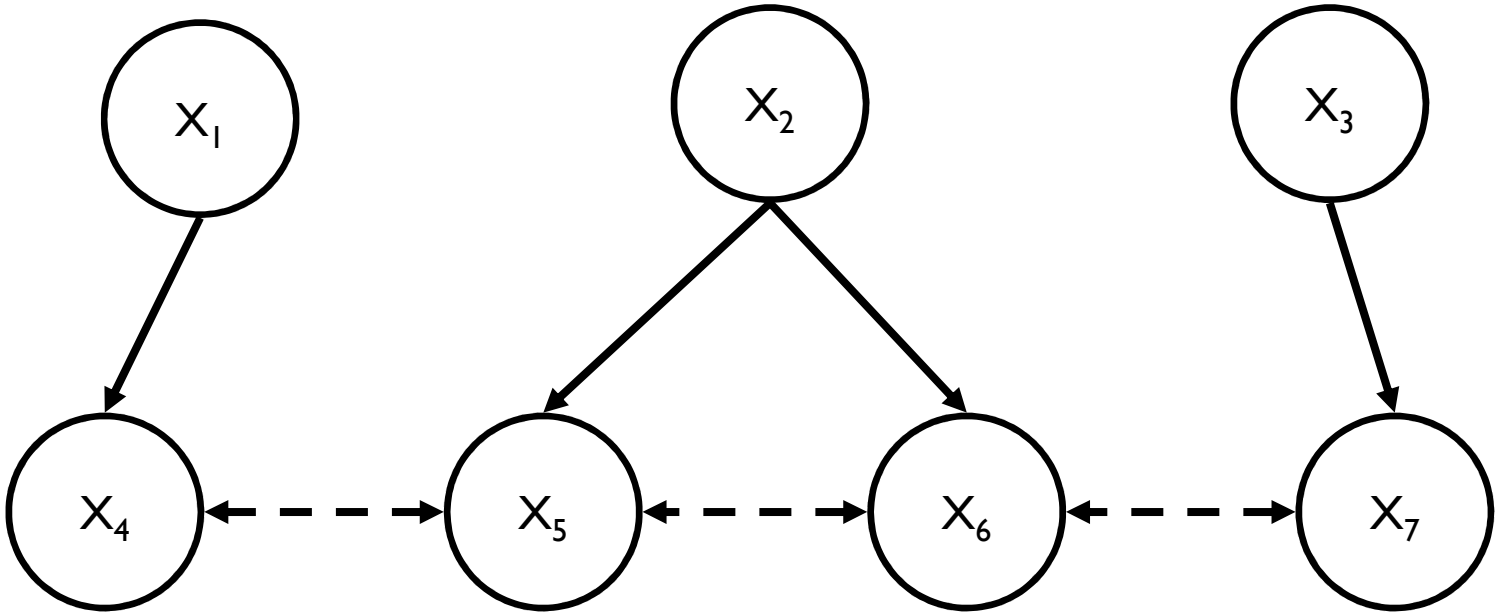
$X_4 \perp\!\!\!\perp X_1$



$X_4 \perp\!\!\!\perp X_1$

Step 2: Parameterizing Components

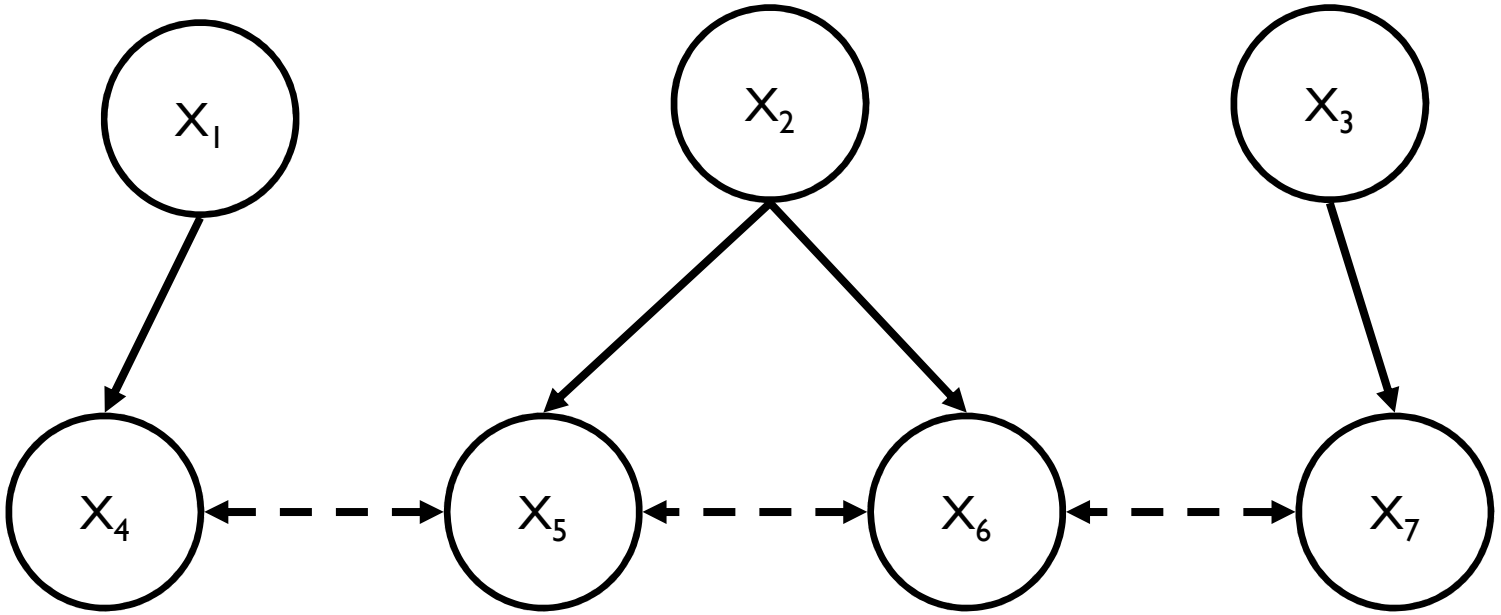
- For this talk, I'll consider only the case where there are no directed edges within a bi-directed component



Step 2: Parameterizing Components

- Multiply conditional CDFs using bi-directed cliques

$$F_i(x_{D_i} \mid pa_{\mathcal{G}}(X_{D_i})) \equiv \prod_{X_S \in \mathcal{C}_i} F_S(x_S \mid pa_{\mathcal{G}}(X_{D_i}))$$

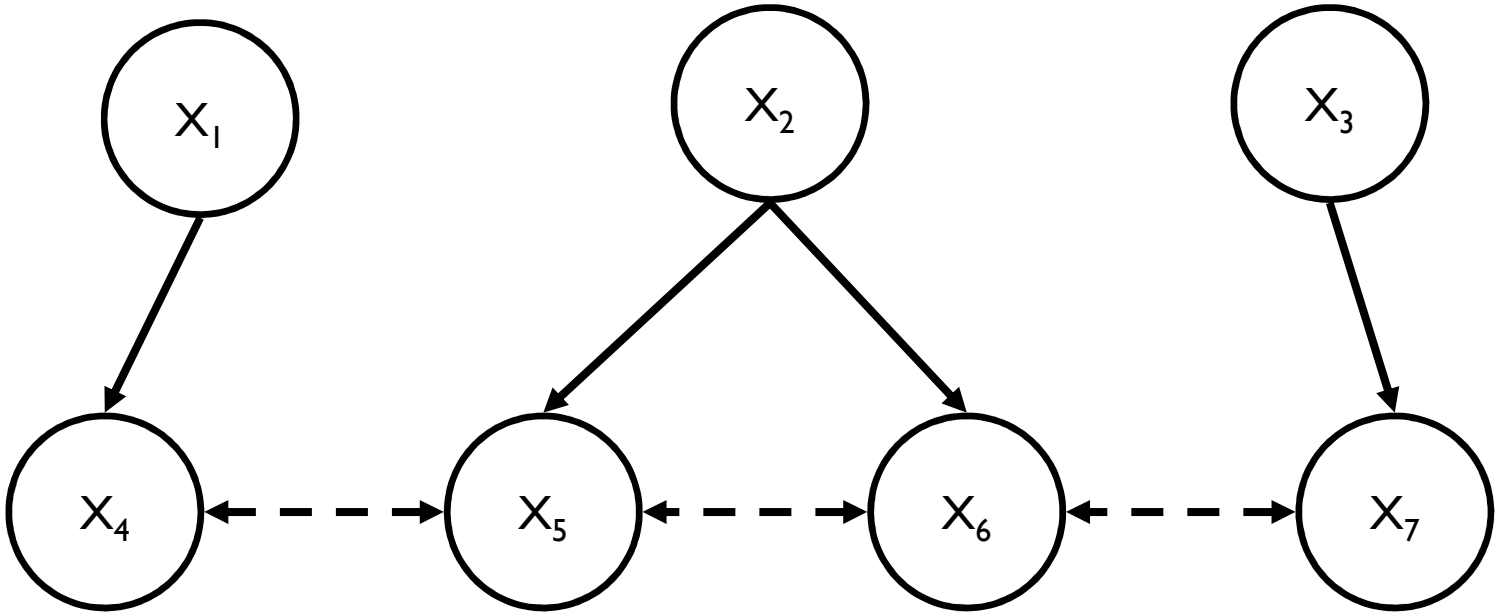


$$F(X_4, X_5, X_6, X_7 \mid X_1, X_2, X_3) =$$

$$F(X_4, X_5 \mid X_1, X_2)F(X_5, X_6 \mid X_2)F(X_6, X_7 \mid X_2, X_3)$$

Step 2: Parameterizing Components

- Needs "local factor restrictions" though



Follows from factorization: $X_4 \perp\!\!\!\perp X_7 | X_1$

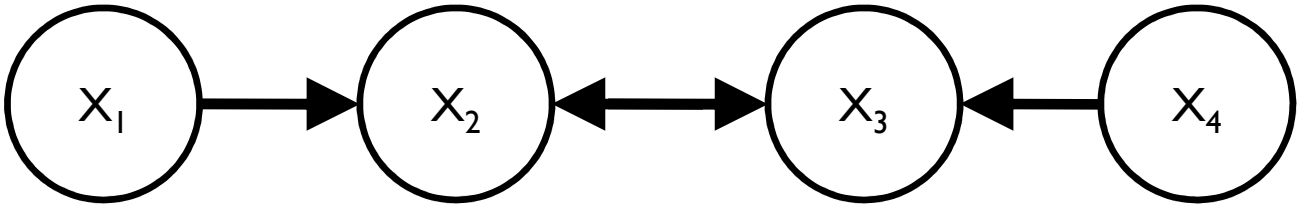
Needs further care: $X_5 \perp\!\!\!\perp X_1 | X_2$

Step 2a: A Copula Formulation

Implementing the local factor restriction could be potentially complicated, but the problem can again be easily tackled by adopting a copula formulation.

Step 2a: A Copula Formulation

- The idea is to use a conditional marginal $F_i(X_i \mid pa(X_i))$ within a copula
- Example



$$U_2(x_1) \equiv P_2(X_2 \le x_2 \mid x_1) \qquad U_3(x_4) \equiv P_2(X_3 \le x_3 \mid x_4)$$

$$P(X_2 \le x_2, X_3 \le x_3 \mid x_1, x_4) = C(U_2(x_1), U_3(x_4))$$

- Check:

$$P(X_2 \le x_2 \mid x_1, x_4) = C(U_2(x_1), 1) = C(U_2(x_1))$$
$$= U_2(x_1) = P_2(X_2 \le x_2 \mid x_1)$$

Step 2a: A Copula Formulation

- Not done yet! We need this

$$F_i(x_{D_i} \mid pa_{\mathcal{G}}(X_{D_i})) \equiv \prod_{X_S \in \mathcal{C}_i} F_S(x_S \mid pa_{\mathcal{G}}(X_{D_i}))$$

- Product of copulas is not a copula

- However, results in the literature are helpful here. It can be shown that plugging in $U_i^{1/d(i)}$, instead of U_i will turn the product into a copula
  - where *d(i)* is the number of bi-directed cliques containing X_i

Liebscher (2008)

Liebscher's Construction for Products

- Special case is easy to understand:

$$C(u_1, \ldots, u_p) = \prod_f C_f(u_1^{w(1f)}, u_2^{w(2f)}, \ldots u_p^{w(pf)})$$

$$\sum_f w(if) = 1,\ 0 \leq w(if) \leq 1$$

where each $C_f(\ .)$ is a copula function

- It is not hard to verify that $C(u_1, \ldots, u_p)$ is a CDF, and that $C(u_i) = u_i$

Parameter Learning

- For the purposes of illustration, assume a finite mixture of experts for the conditional marginals for continuous data

$$f_v(x_v \mid pa_{\mathcal{G}}(X_v)) = \sum_{z=1}^{K} \pi_{z;v} \mathcal{N}(x_v;\ \mu_{z;v}, \sigma^2_{z;v})$$

$$\mu_{z;v}(pa_{\mathcal{G}}(X_v)) = \theta_{v0} + \theta_v^{\mathsf{T}} pa_{\mathcal{G}}(X_v)$$

$$\pi_{z;v}(pa_{\mathcal{G}}(X_v)) \propto \exp(w_{v0} + w_v^{\mathsf{T}} pa_{\mathcal{G}}(X_v))$$

- For discrete data, just use the standard CPT formulation found in Bayesian networks

Parameter Learning

- Copulas: we use a bi-variate formulation only (so we take products "over edges" instead of "over cliques").
- In the experiments: Frank copula

$$C_F(u_i, u_j; \alpha) = -\frac{1}{\alpha} \ln \left(1 + \frac{(e^{-\alpha u_i} - 1)(e^{-\alpha u_j} - 1)}{e^{-\alpha} - 1} \right)$$

Parameter Learning

- Suggestion: two-stage quasiBayesian learning
  - Analogous to other approaches in the copula literature (inference function for margins, Joe 1997)
  - Fit marginal parameters using the posterior expected value of the parameter for each individual mixture of experts
  - Plug those in the model, then do MCMC on the copula parameters
- Relatively efficient, decent mixing even with random walk proposals
  - Nothing stopping you from using a fully Bayesian approach, but mixing might be bad without some smarter proposals
- Notice: needs constant CDF-to-PDF/PMF transformations!

Experiments

Data set	**Data type**	**#V**	**#D**	$\mathbb{E}[\#\leftrightarrow]$	$\mathbb{E}[\#\rightarrow]$
SPECT	Binary	23	267	4.1	25.6
Breast cancer wisconsin	Ordinal	10	683	5.1	16.3
Soybean (large)	Ordinal	33	266	9.3	39.8
Parkinsons	Continuous	15	5875	8.9	18.2
Ionosphere	Continuous	32	351	12.4	32.8
Wine quality (red)	Continuous	11	1599	5.7	7.5
Wine quality (white)	Continuous	11	4898	7.3	14.5

Experiments

Data set	Gaussian/probit	Copula MCDN	Difference
SPECT	-11.32	-11.11	0.21 ± 0.06 $\star$
Breast cancer wisconsin	-12.60	-12.77	-0.17 ± 0.11
Soybean (large)	-20.17	-17.71	2.46 ± 0.20 $\star$
Parkinsons	-11.65	-3.48	8.17 ± 0.28 $\star$
Ionosphere	-41.10	-27.45	13.64 ± 0.67 $\star$
Wine quality (red)	-13.72	-11.25	2.47 ± 0.10 $\star$
Wine quality (white)	-13.76	-12.11	1.65 ± 0.09 $\star$

Summary: Part II

- General toolbox for construction for ADMG models
- Bayesian learning requires some extra work in case where CDF to PDF transformations are intractable
- Structure learning: how would this parameterization help?
- Empirical applications in problems with extreme value issues, exploring non-independence constraints, relations to effect models in the potential outcome framework etc.

Part III

Work in Progress

(You have been warned)

Parameter Learning for CDNs

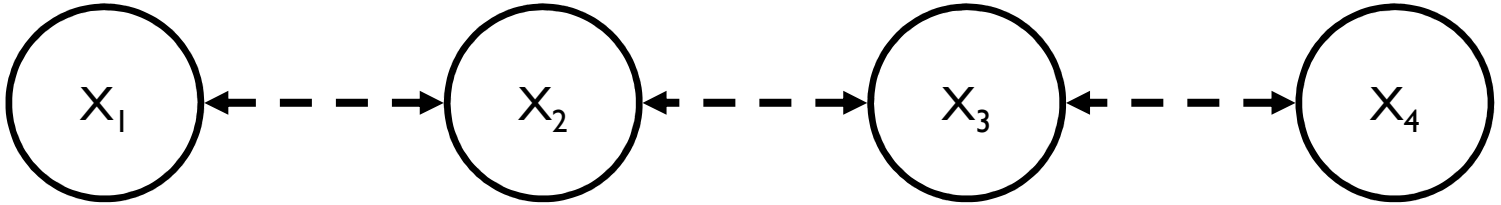
- Unsurprisingly, finding the maximum likelihood estimator requires computing the likelihood function

- For variables in {0, 1, 2, …}, mass function is given by:

$$\sum_{z_1=0}^{1} \cdots \sum_{z_d=0}^{1} (-1)^{z_1+z_2+\ldots z_d} F(x_1 - z_1, \ldots, x_d - z_d)$$

- Intractable in general. Without special knowledge of the function, sampling from it is intractable too!

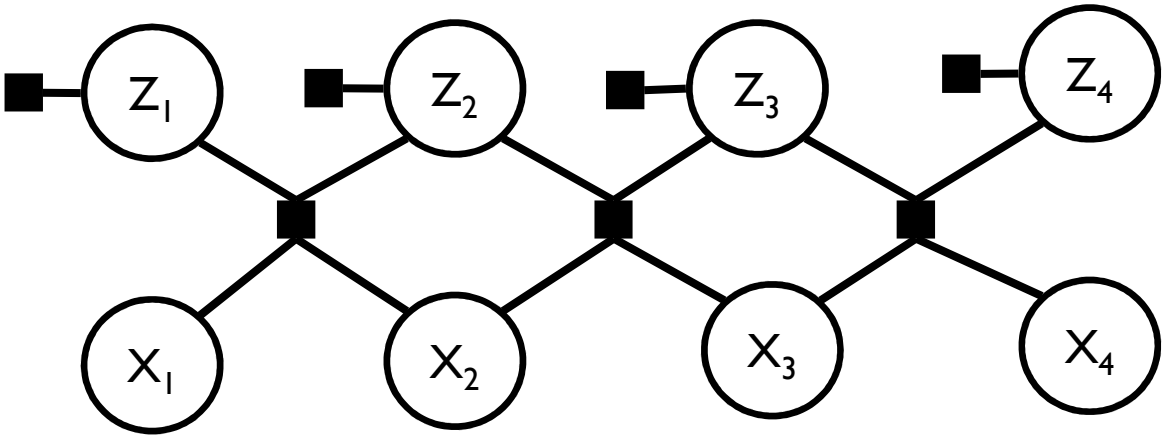
Exploit Factorizations: Dynamic Programming (Huang and Frey, 2011)



$$C_{12}(x_1, x_2)C_{23}(x_2, x_3)C_{34}(x_3,x_4)$$

- Pretend **z** is a hidden vector

$$\sum_{\mathbf{z}} \{\Pi_i(-1)^{z_i}\}\, C_{12}(x_1 - z_1, x_2 - z_2)C_{23}(x_2 - z_2, x_3 - z_3)C_{34}(x_3 - z_3,x_4 - z_4)$$



(Notice: this is NOT a joint distribution)

Alternative Fitting Methods

- Abandon maximum likelihood. Marginal composite likelihood is straightforward

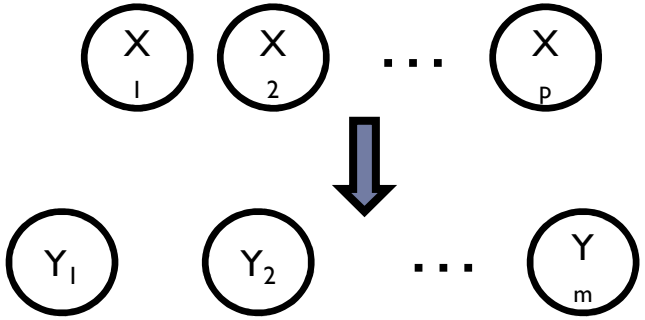
$$S(\theta) = \sum_A w_A \log P(\mathbf{X}_A; \theta)$$

- Small marginals can be calculate trivially by brute force. Larger marginals might be tractable to calculate using Huang and Frey's DP method

- Tap into the copula literature. Theoretical results for general copula fitting by IFM + CL, including covariates: Zhao and Joe (2005)

Some Open Questions

- Multi-output prediction problems

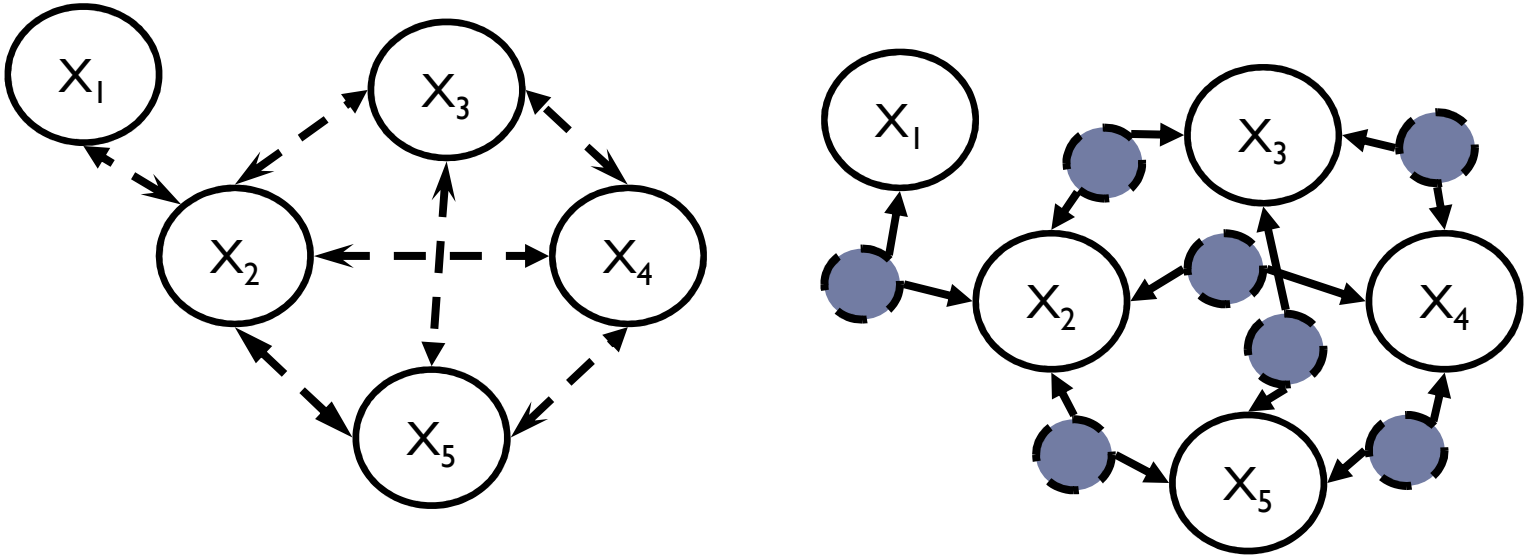


- Which sparse structures could be used as alternatives to conditional random fields?

- Why/when sparsity?

Care Needs to be Taken

- Watch out for those pairwise models with fixed exponentiation

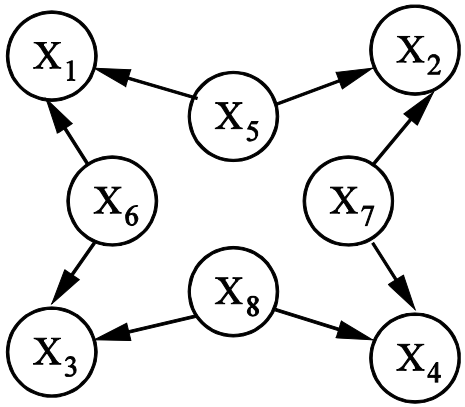


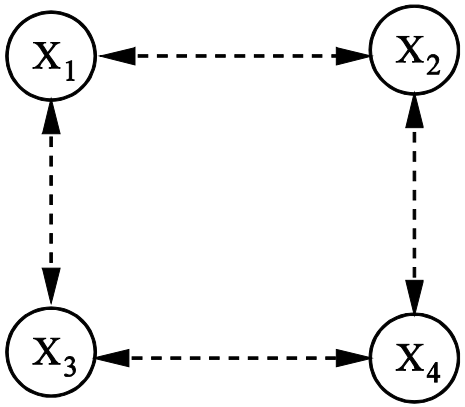
Recall: single factors $C_{ij}(u_i^{1/\#n(i)}, u_j^{1/\#n(j)})$

Pairwise marginal: $u_i^{1 - 1/\#n(i)} u_j^{1 - 1/\#n(j)} C_{ij}(u_i^{1/\#n(i)}, u_j^{1/\#n(j)})$

Another Problem: Sampling

- What if I have a bi-directed graph with a large treewidth?
  - Calculating the likelihood function is intractable. Is Bayesian inference as problematic as in Markov random fields?
- Auxiliary variable schemes: there is an implicit latent variable construction. Can we Gibbs sample our way out of it?





More Exploitation of Copula Theory

- The Laplace transform of a CDF of a positive random variable with CDF G(h):

$$\phi(x) = \int \exp(-sh) dG(h), s \geq 0$$

- Relation to the marginal CDF of a random variable

$$F(x) = \int F(x \mid h) dG(h) = \int \exp(\log F(x \mid h)) dG(h) = \phi(-\log F(x \mid h))$$

- So

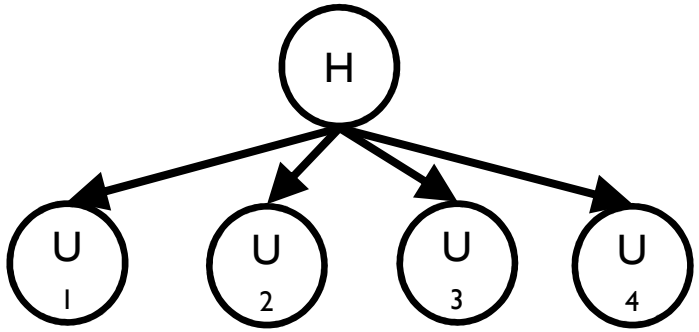
$$G = \exp\{\phi^{-1}(F)\}$$

Archimedean Copulas

- Restricted to "constant dependency" over multiple variables (nested versions exist)

$$C(\mathbf{u}) = \psi^{-1}(\psi(u_1) + \psi(u_2) + \ldots + \psi(u_p))$$

- Analogy: exchangeable "rank-one" latent variable model with restriction on parameters (Gaussian instead of uniform in this example)



$$U_i = \sqrt{\rho} H + \varepsilon_i$$

$$H \sim N(0,1)$$

$$\varepsilon_i \sim N(0, 1-\rho^2)$$

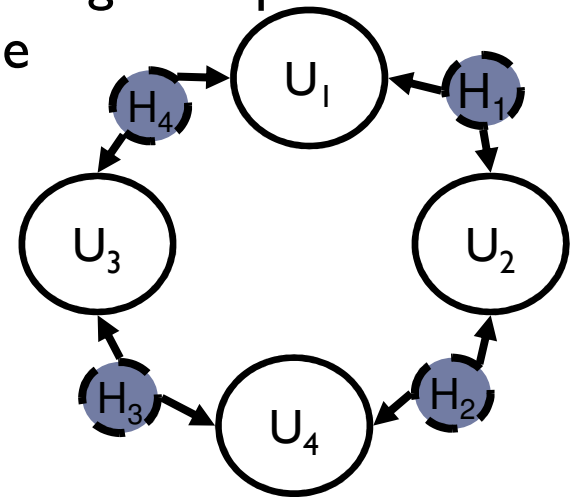
A Sampling Procedure for a Single Archimedean Copula

- In the Archimedean case, we can sample *H* from the inverse Laplace transform of ψ^{-1}
- We can then sample each U_i independently conditioned on *H* using a simple function of ψ
- Laplace transform might not be easy to find. Simple in some cases
  - In the Frank copula, it boils down to a discrete distribution
- Other numerical methods available (Hofert, 2008)

Product Case

$$C(u) = \prod_f C_f(u_1^{w(1,f)}, \ldots, u_p^{w(p,f)}) = \prod_f \int_0^\infty C_f(u_1^{w(1,f)}, \ldots, u_p^{w(p,f)} \mid H_f) dG(H_f)$$

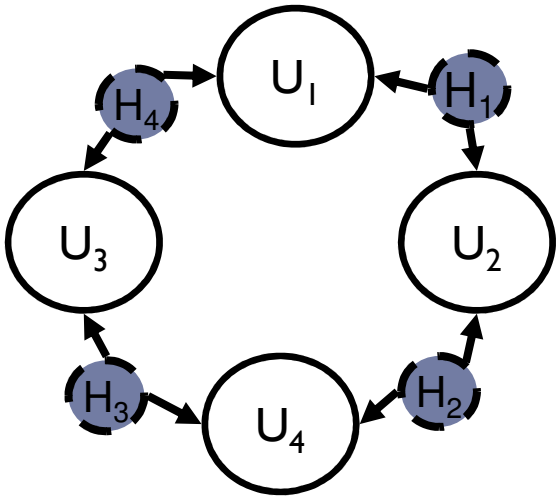
- Sampling scheme: sample (**H**, θ) given data, keep θ samples
- $C_f(u_1^{w(1,f)}, \ldots, u_p^{w(p,f)} \mid H_f)$ factorizes into univariate CDFs: easy to convert to product of PMFs/PDFs by differentiating each product of factors that depends only in a single u_i at a time

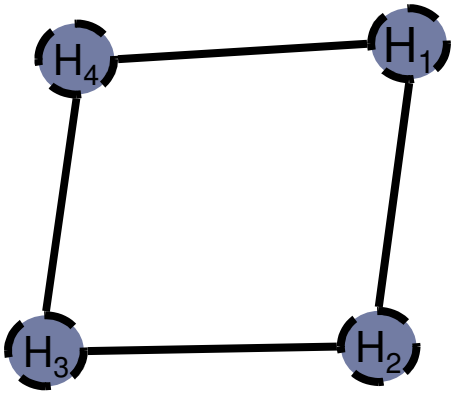


$$g_1(h_1) g_2(h_2) \times \ldots \left\{ \frac{\partial}{\partial u_1} t_{11}(u_1, h_1) t_{14}(u_1, h_4) \right\} \times \left\{ \frac{\partial}{\partial u_2} t_{21}(u_2, h_1) t_{22}(u_2, h_2) \right\} \times \ldots$$

Product Case

- Gibbs-like scheme for H_f given the others





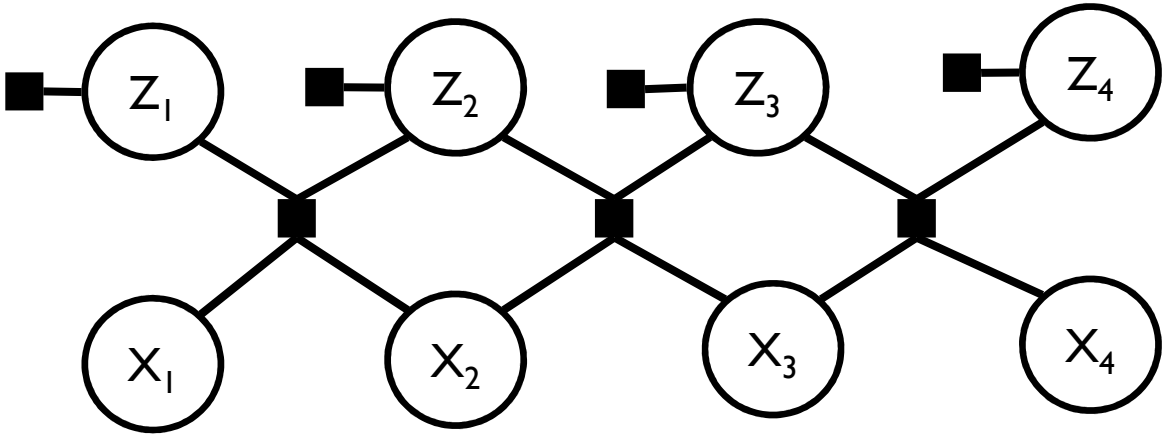
- Conditioned on X, there is a "dual" Markov network for **H**. Small neighbourhoods might speed up computation.

Remind me Again: Why not Latent Variables from the Very Start?

- Copula motivation: CDN allows easy copula construction. The Laplace transform might be hard to find, and it is not necessary if sampling is not necessary
- Silva and Ghahramani (2009): the collapsed Gibbs sampler interpretation for a mixed graph model
  - Gains of an order of magnitude in effective sample size, even for small networks
- Partial collapse: keep a tractable subset of factors where DP can be used, make the rest of the latents explicit
  - Question: good ways of doing this?

What About Prediction?

- Finding the MAP assignment of a CDN model: finding the maximum of a "marginal"



- Alternative: a stochastic EM approach
  - Use sampler to generate p(H | X) (hard)
  - Maximize approximation to $E_{p(H \mid X)}[\log(P(X \mid H)]$ ("easy")
  - Iterate

Conclusion

- Using graphical model decompositions is great as a way of constructing copulas
  - Copula Bayesian networks, vines and many other interesting contributions today
- Likewise, using results from copula theory helps us to better understand graphical models and to avoid reinventing some wheels
  - Both in model construction and estimation
- Thanks to the organizers and the audience. Also thanks to Sergey Kirshner and Thomas Richardson for several useful discussions and code

References

- K. Bollen (1989). *Structural Equations with Latent Variables*. Wiley & Sons
- M. Drton and T. Richardson (2003). "A new algorithm for maximum likelihood estimation in Gaussian graphical models for marginal independence". UAI.
- M. Drton and T. Richardson (2008). "Binary models for marginal independence". JRSS B 70, 287-309
- H. Joe (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall.
- S. Kirshner (2007). "Learning with tree-averaged densities and distributions". NIPS.
- E. Liebscher (2008). "Construction of asymmetric multivariate copulas". Journal of Multivariate Analysis 99, 2234-2250.
- M. Hofert (2008). "Sampling Archimedean copulas". CSDA 52, 5163-5174.
- J. Huang and B. Frey (2011). "Cumulative distribution networks and the derivative-sum-product: models and inference for cumulative distribution functions on graphs". JMLR 12, 301-348.
- T. Richardson (2009). "A factorization criterion for acyclic directed mixed graphs". UAI.
- T. Richardson and P. Spirtes (2002). "Ancestral graph Markov models". Annals of Statistics 30, 962-1030.
- R. Silva and R. B. Gramacy (2009). "MCMC methods for Bayesian mixtures of copulas". AISTATS
- R. Silva and Z. Ghahramani (2009). "The hidden life of latent variables: Bayesian learning for Gaussian mixed graph models". JMLR 10, 1187-1238
- R. Silva, C. Blundell and Y.-W. Teh (2011). "Mixed cumulative distribution networks". AISTATS
- Y. Zhao and H. Joe (2005). "Composite likelihood estimation in multivariate data analysis". The Canadian Journal of Statistics 33, 335-356.