Latent Tree Copulas

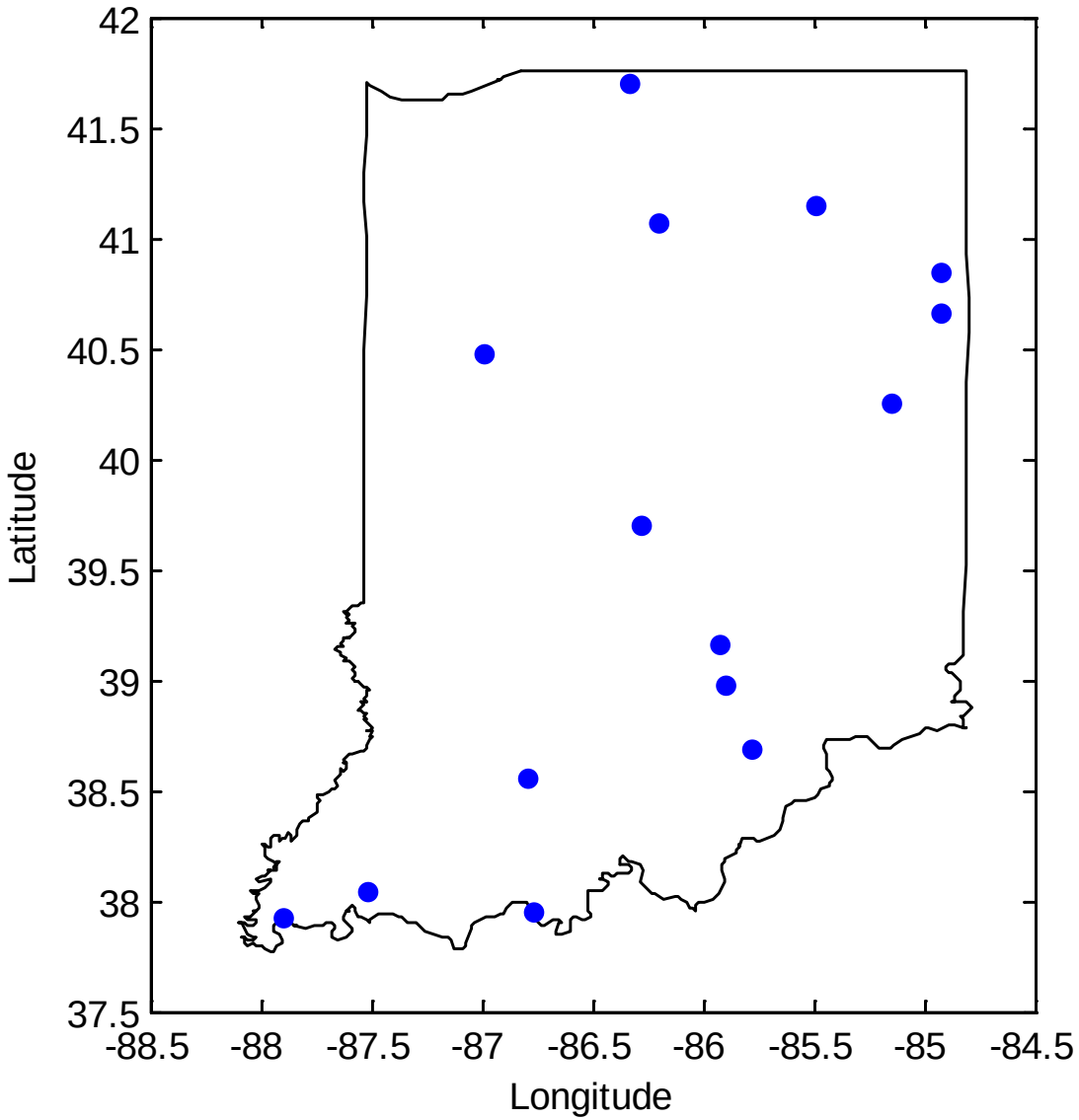
Sergey Kirshner
skirshne@purdue.edu
Purdue University
West Lafayette, IN, USA

Granada, Spain,
September 19, 2012

Coming Attractions

- Want to fit density to model multivariate data?
  - and organize real-valued data into a hierarchy of features?
- New density estimation model based on tree-structured dependence with latent variables
  - Distribution = Univariate Marginals + Copula
  - Hierarchy of variables as a latent tree-copula
  - Parameter estimation and structure learning
    - Efficient inference for Gaussian copulas (100s of variables), several structure learning approaches
    - Variational inference for other copulas (10-30 variables)

Building a Hierarchy of Rainfall Stations

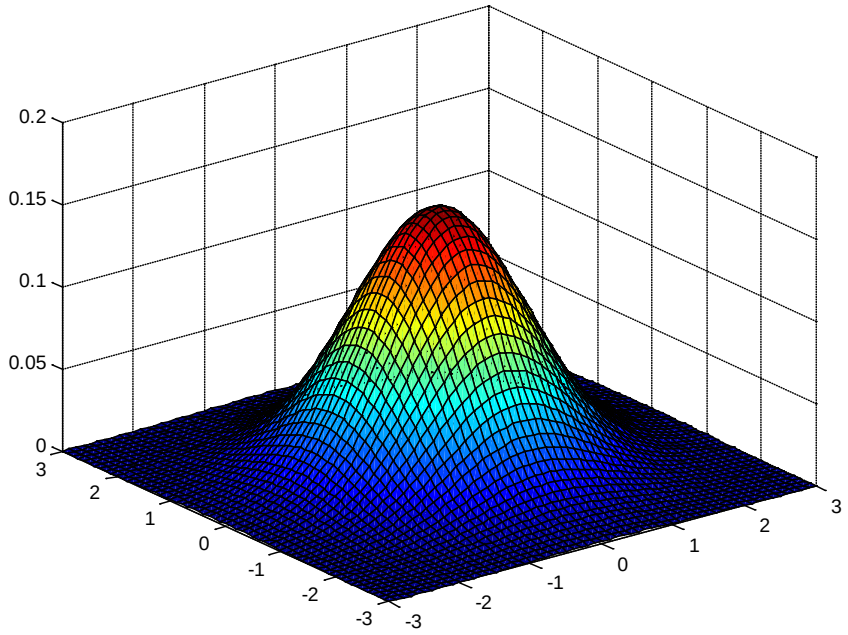


State of Indiana (USA)

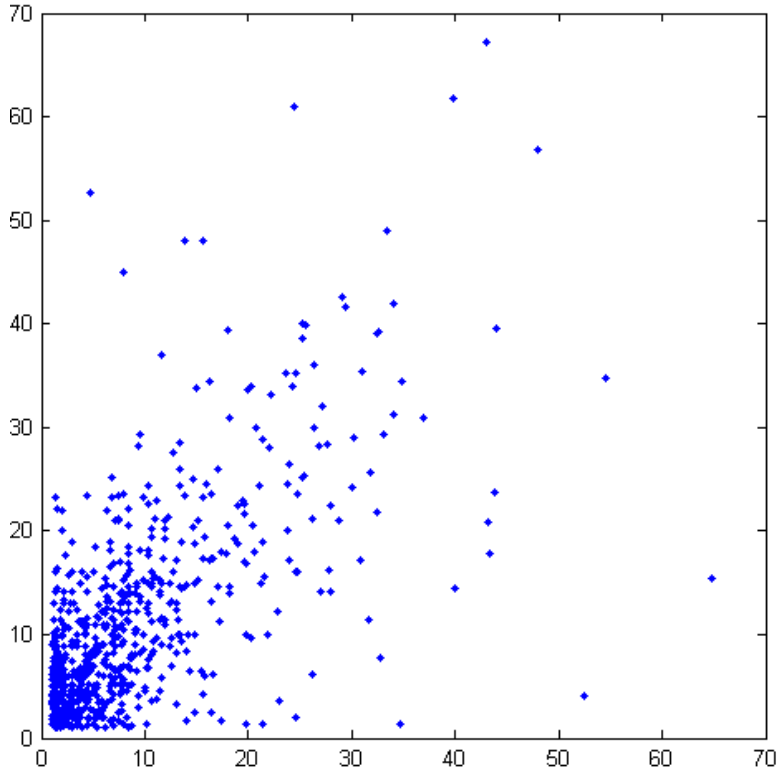
Average monthly observations for 15 rainfall stations 1951-1996 (47 years)

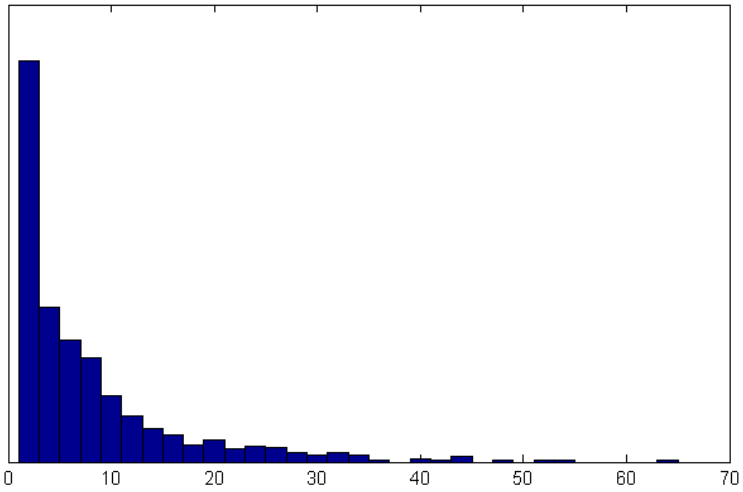
Most Popular Distribution…

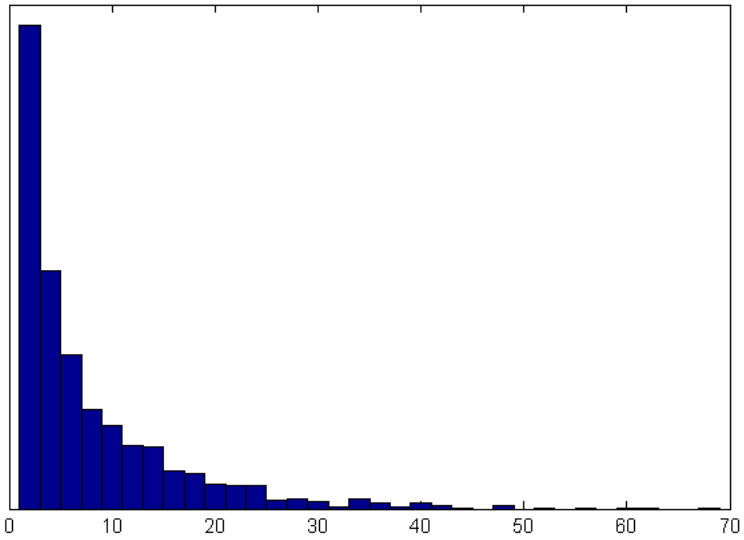
- Interpretable
- Closed under taking marginals
- Generalizes to multiple dimensions
- Models pairwise dependence
- Tractable
- 245 pages out of 691 from *Continuous Multivariate Distributions* by Kotz, Balakrishnan, and Johnson



What If the Data Is NOT Gaussian?







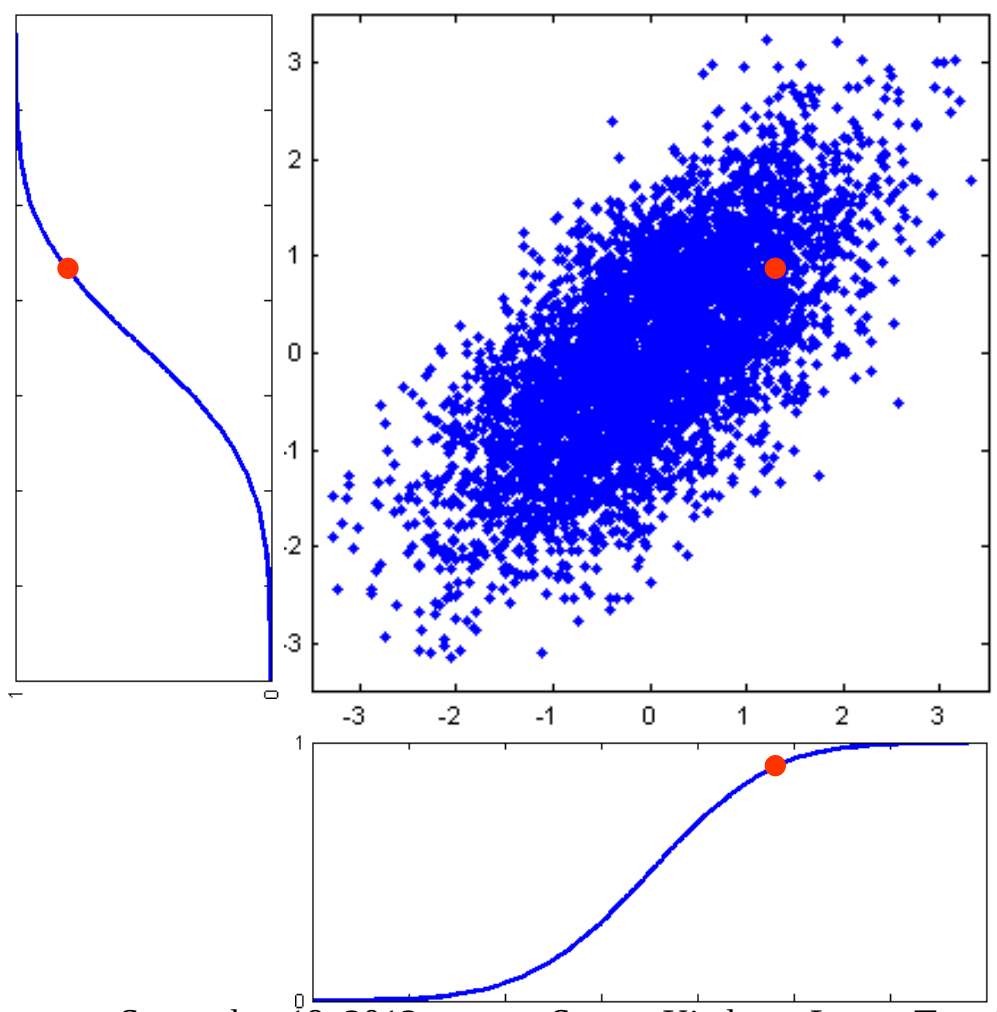
Separating Univariate Marginals

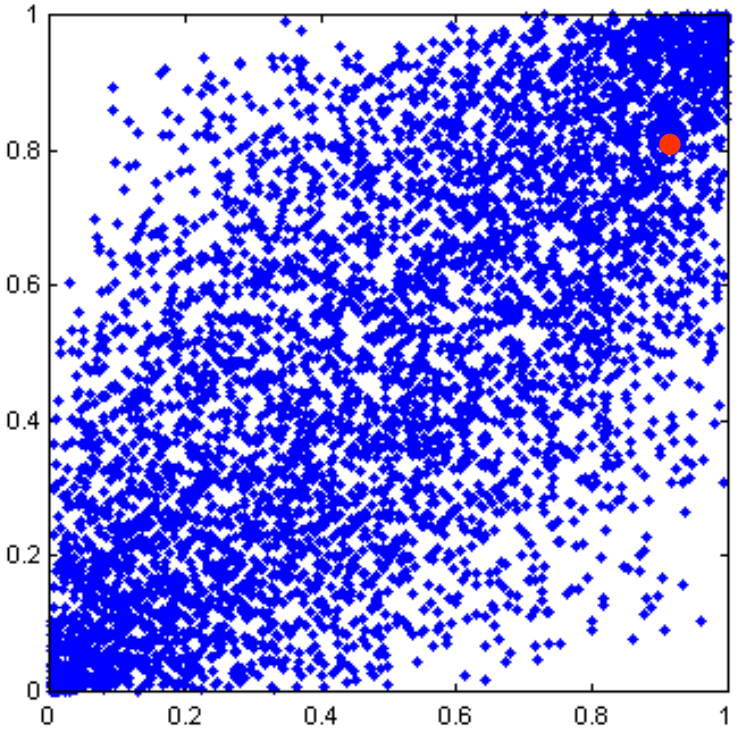
$$\ln p(\boldsymbol{x}) = \sum_{i=1}^{d} \ln p_i(x_i) + \ln \frac{p(\boldsymbol{x})}{p_1(x_1)\dots p_d(x_d)}$$

univariate marginals, independent variables,

multivariate dependence term, copula

Monotonic Transformation of the Variables





Copula

Copula C is a multivariate distribution (cdf) defined on a unit hypercube with uniform univariate marginals:

$$C : [0,1]^d \to [0,1]$$
$$C_i(a_i) = a_i, \;\; i = 1, \ldots, d$$

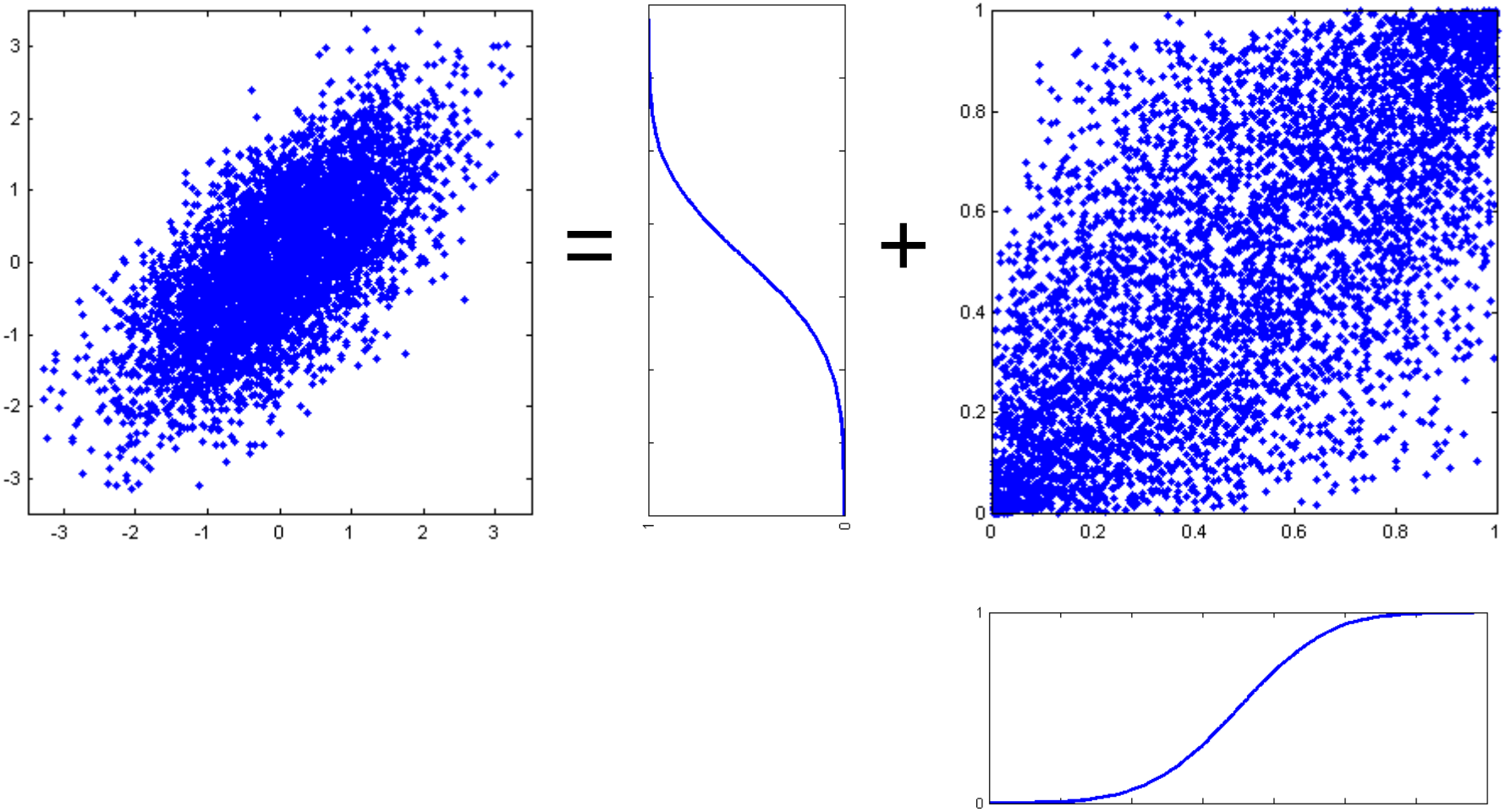
$$F(\boldsymbol{x}) = C(F_1(x_1), \ldots, F_d(x_d)), \;\; \boldsymbol{x} \in \mathbb{R}^d$$
$$C(\boldsymbol{a}) = F\left(F_1^{-1}(a_1), \ldots, F_d^{-1}(a_d)\right)$$

$$a_i = F_i(x_i), \quad i = 1, \ldots, d$$
$$x_i = F_i^{-1}(a_i), \quad i = 1, \ldots, d$$

$$c(\boldsymbol{a}) = \frac{\partial^d C(\boldsymbol{a})}{\partial a_1 \ldots \partial a_d} = \frac{p(\boldsymbol{x})}{\prod_{i=1}^d p_i(x_i)}$$

Sklar's Theorem

[Sklar 59]



Example: Multivariate Gaussian Copula

$$F(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) \qquad \boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_d \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \sigma_{11}^2 & \cdots & \sigma_{11}\sigma_{dd}\rho_{1d} \\ \vdots & \ddots & \vdots \\ \sigma_{11}\sigma_{dd}\rho_{d1} & \cdots & \sigma_{dd}^2 \end{pmatrix}$$

$$F(x_1, \ldots, x_d) = \mathbf{\Phi_R}\left(\frac{x_1 - \mu_1}{\sigma_{11}}, \ldots, \frac{x_d - \mu_d}{\sigma_{dd}}\right)$$

$$a_1 = F_1(x_1) = \mathbf{\Phi}\left(\frac{x_1 - \mu_1}{\sigma_{11}}\right), \ldots, a_d = F_d(x_d) = \mathbf{\Phi}\left(\frac{x_d - \mu_d}{\sigma_{dd}}\right)$$

$$C(\boldsymbol{a}) = F\left(F_1^{-1}(a_1), \ldots, F_d^{-1}(a_d)\right) = \mathbf{\Phi_R}\left(\mathbf{\Phi}^{-1}(a_1), \ldots, \mathbf{\Phi}^{-1}(a_d)\right)$$

Separating Univariate Marginals

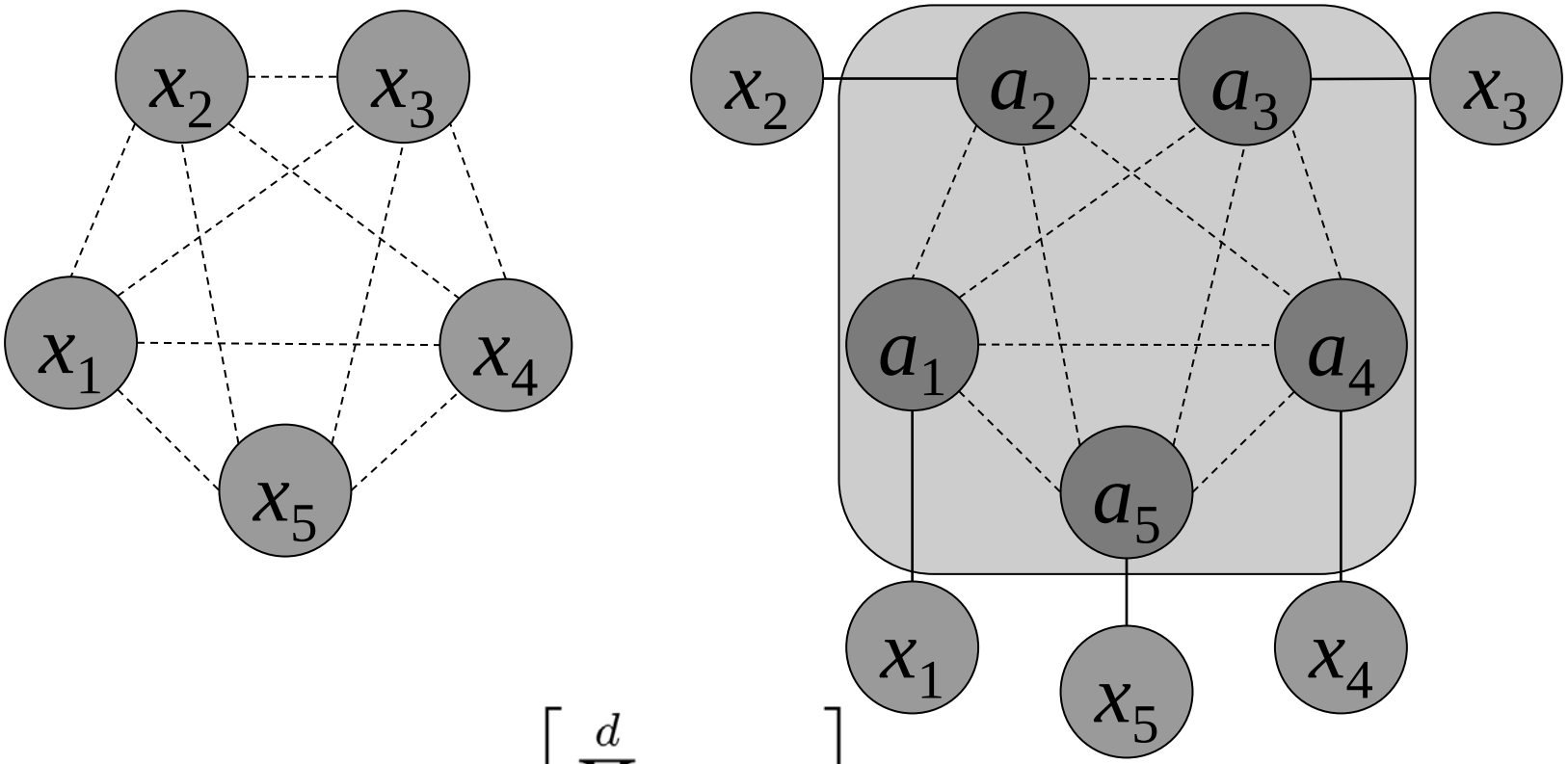
$$\ln p(\boldsymbol{x}) = \sum_{i=1}^{d} \ln p_i(x_i) + \ln \frac{p(\boldsymbol{x})}{p_1(x_1)\dots p_d(x_d)}$$

$$\ln p(\mathcal{D}) = \sum_{n=1}^{N}\sum_{i=1}^{d} \ln p_i(x_i^n) + \sum_{n=1}^{N} \ln c\left(F_1(x_1^n), \dots, F_d(x_d^n)\right)$$

1. Fit univariate marginals (parametric or non-parametric)
2. Replace data points with cdf's of the marginals
3. Estimate copula density

Inference for the margins [Joe and Xu 96]; canonical maximum likelihood [Genest et al 95]

Graphical Model Using a Copula



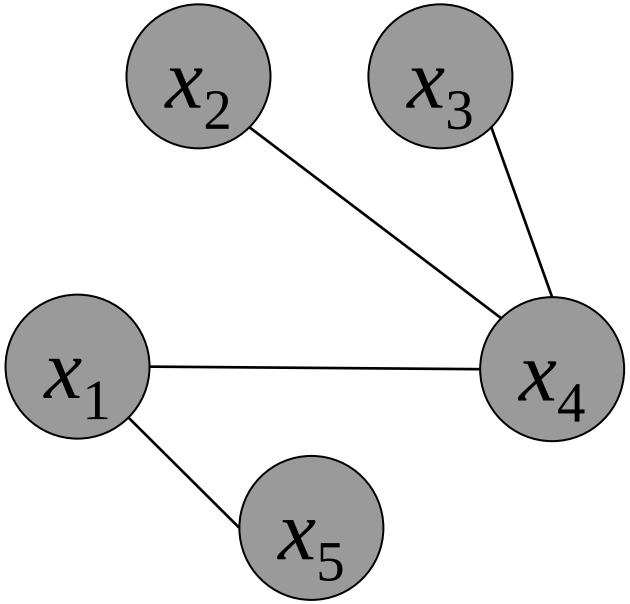
$$f(x_1, \ldots, x_d) = \left[\prod_{i=1}^{d} f_i(x_i) \right] c(F_1(x_1), \ldots, F_d(x_d))$$

$$= \left[\prod_{i=1}^{d} \phi_i(x_i, a_i) \right] c(a_1, \ldots, a_d)$$

Graphical Model Approaches to Estimating Copulas

- Vines [Bedford and Cooke 02]
- Trees [Kirshner 08]
- Nonparanormal model [Liu et al 09]
- Copula Bayesian networks [Elidan 10]

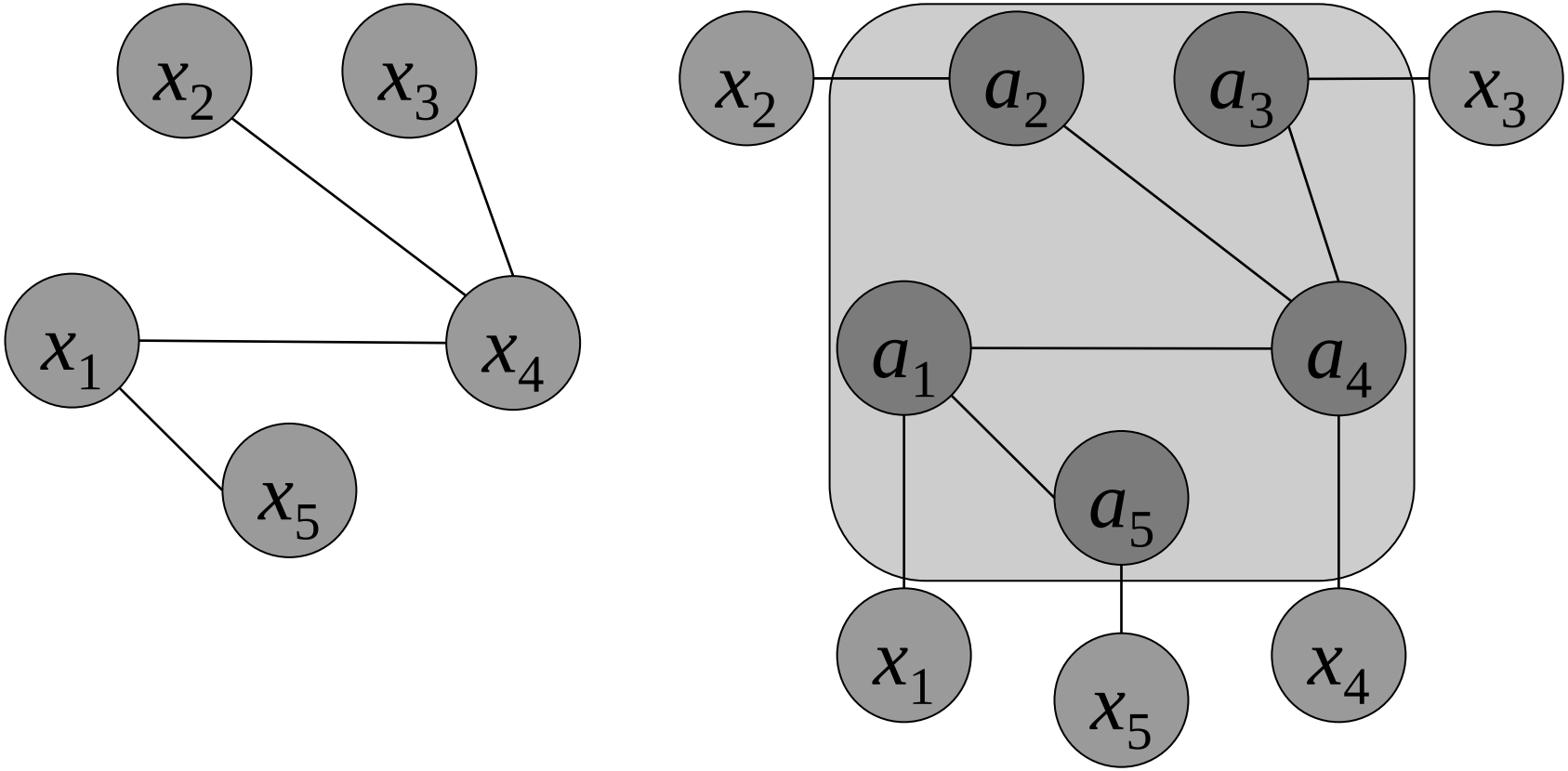
Tree-Structured Densities



$$p_{\mathcal{E}}(\boldsymbol{x}) = \left[\prod_{v \in \mathcal{V}} p_v(x_v)\right] \left[\prod_{\{u,v\} \in \mathcal{E}} \frac{p_{uv}(x_u, x_v)}{p_u(x_u)\, p_v(x_v)}\right]$$

Tree-Structured Copulas

[Kirshner 08]

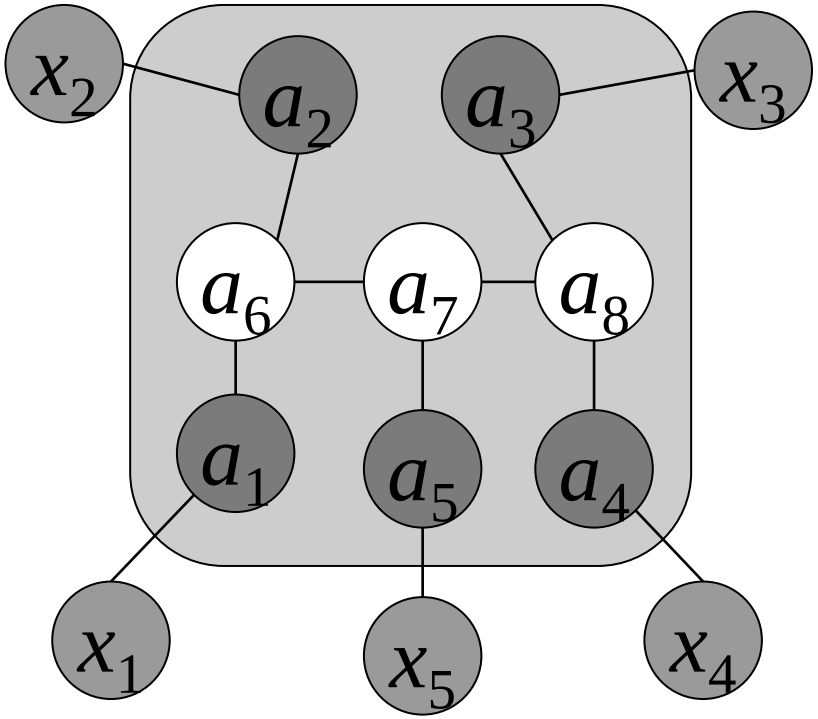


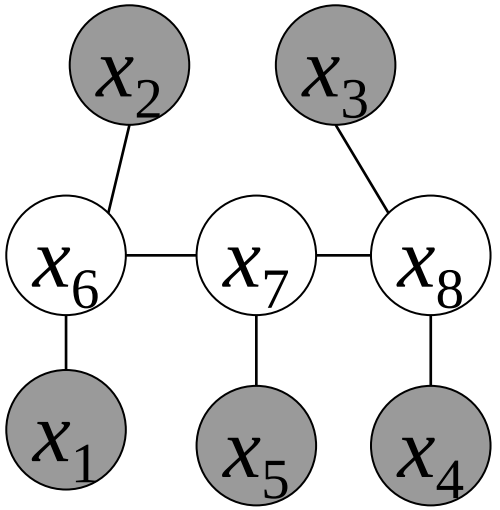
$$c_{\mathcal{E}}(\boldsymbol{a}) = \prod_{\{u,v\}\in\mathcal{E}} c_{uv}(a_u, a_v)$$

Using Tree-Structured Copulas

- Tree-structured copulas are convenient, but are restrictive
  - True distribution may require much larger cliques to decompose
- Can approximate other dependencies using latent variables
  - Mixtures [Kirshner 08]: discrete latent variables
  - Latent tree copulas: continuous random variables embedded in copula trees

Latent Tree Copulas

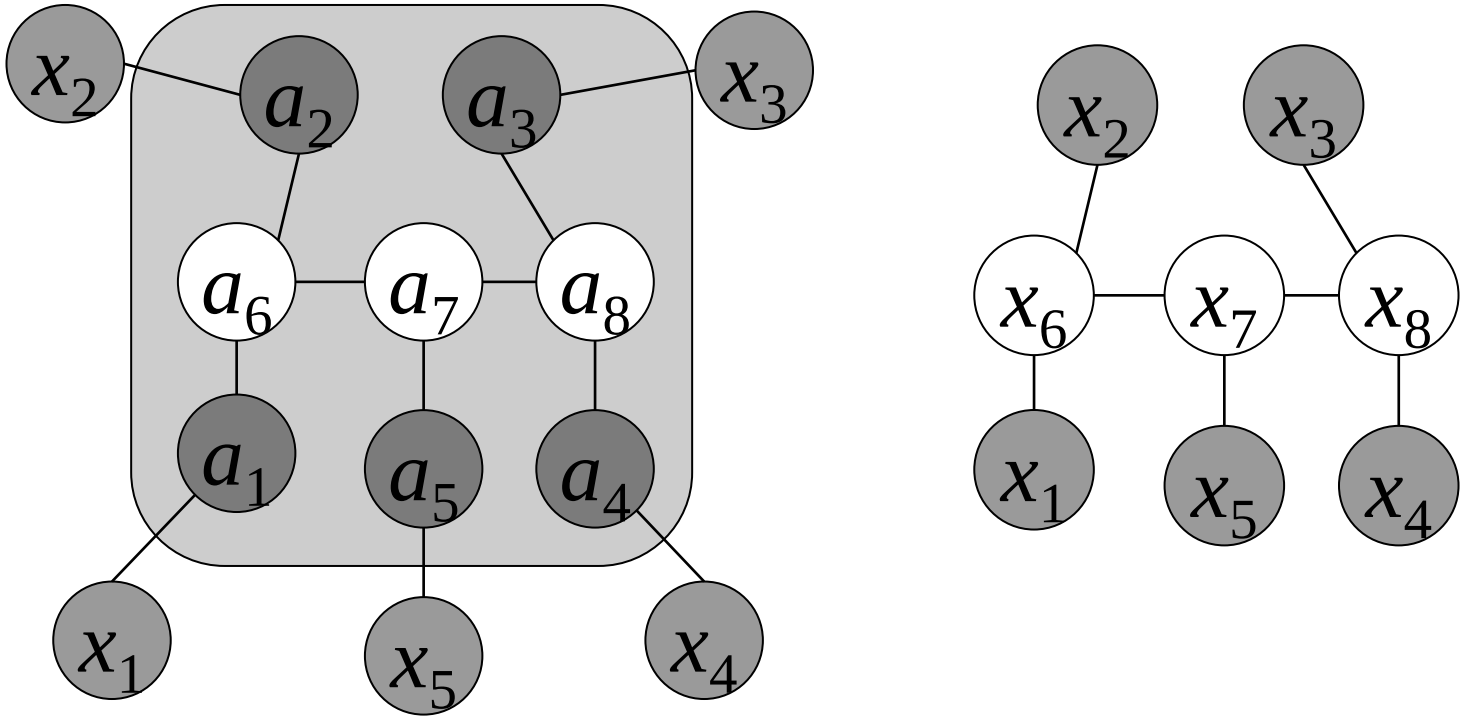




- Defined as a tuple of variables, tree structure, and bivariate copulas

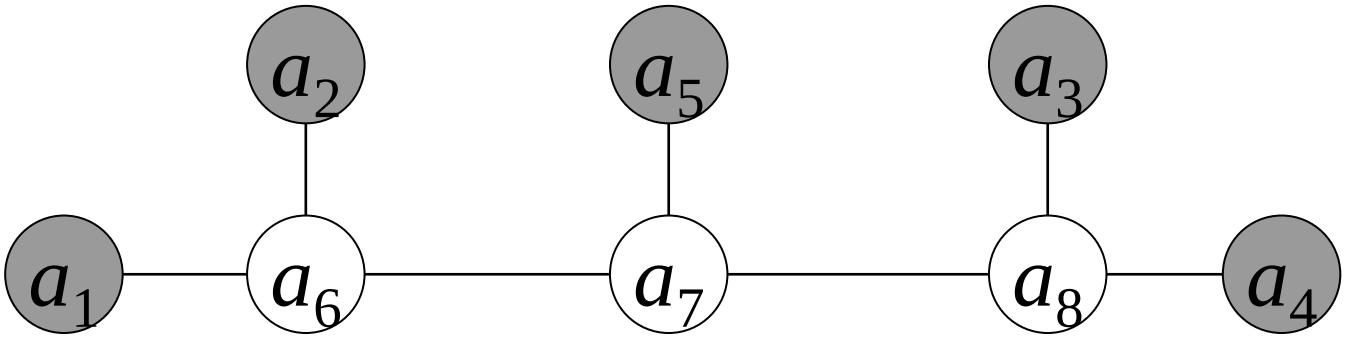
$$c_{LT}(\boldsymbol{a}) = \int\!\!\cdots\!\!\int_{\mathbb{I}^{t-d}} \prod_{\{u,v\}\in\mathcal{E}} c_{uv}(a_u, a_v)\, \mathrm{d}a_{d+1} \ldots \mathrm{d}a_t$$

Latent Tree Copulas



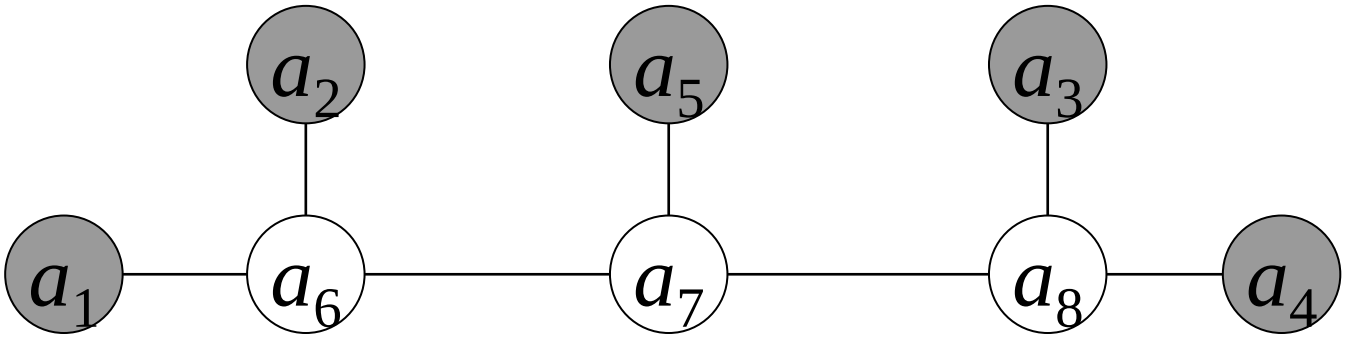
- Defined as a tuple of variables, tree structure, and bivariate copulas
- "Siblings" of latent tree models (LTMs) for categorical variables [e.g., Zhang 02, 04]

Inference



- Good news: posterior distribution is also tree-structured
  - Fairly easy to carry out inference for LTMs
- Bad news: Latent variables are continuous: infinite number of possible values
  - Need to estimate the joint posterior densities

Inference



- Easy for Gaussian copulas
  - Apply inverse standard normal CDF; use belief propagation on jointly Gaussian distribution
- Difficult for non-Gaussian copulas
  - May have no exact form for the posterior!

Inference for non-Gaussian Case

- Variational approach:
  - Approximate the posterior distribution using a tree-structured distribution over piece-wise uniform variables
  - Essentially, approximate using the tree over categorical variables

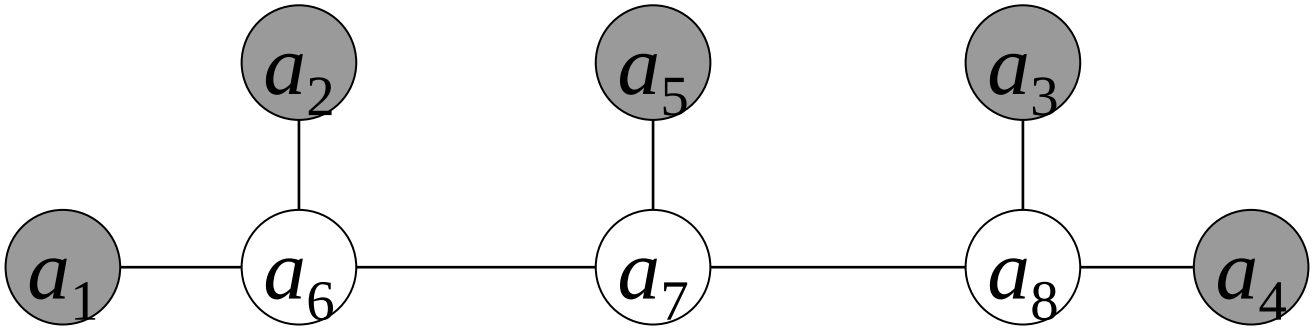
$$q(\boldsymbol{a}_H) = \prod_{u \in H} q_u(a_u) \left[\prod_{\{u,v\} \in \mathcal{E}_H} \frac{q_{uv}(a_u, a_v)}{q_u(a_u)\, q_v(a_v)} \right]$$

$$q_{uv}(a_u, a_v) = p_{uv}(i,j) \geq 0 \text{ for } a_u \in \mathbb{I}_i,\ a_v \in \mathbb{I}_j,$$

$$q_u(a_u) = p_u(i) \text{ for } a_u \in \mathbb{I}_i, \quad \text{where } \mathbb{I}_i = \left(\frac{i-1}{K}, \frac{i}{K} \right]$$

$$\operatorname*{argmin}_{q \in \mathcal{Q}} D\left(q^n(\boldsymbol{a}_H^n) \parallel c_{LT}\left(\boldsymbol{a}_H^n | \boldsymbol{a}_O^n, \boldsymbol{\theta}' \right) \right)$$

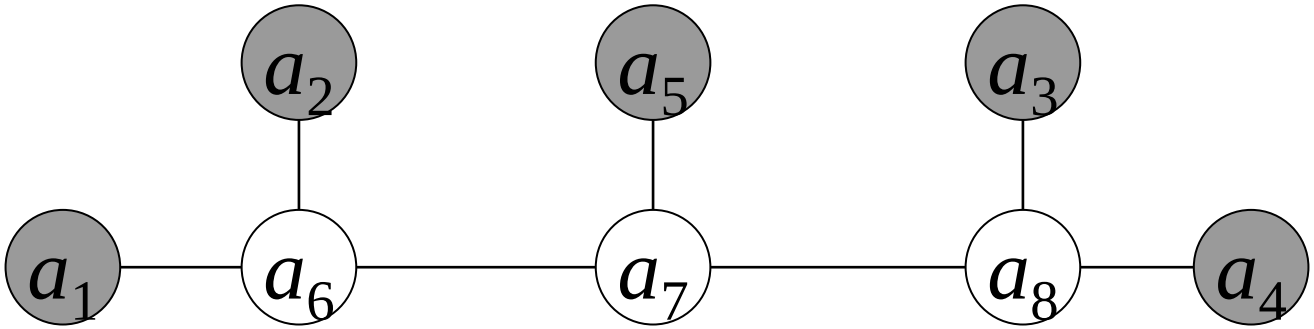
Parameter Estimation with Known Structure



- (Variational) EM
  - E-step: minimize KL divergence
  - M-step: maximize the expected compete-data log-likelihood

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{n} \int_{\mathbb{I}^{t-d}} q^n(\boldsymbol{a}_H^n) \ln \frac{c_{LT}\left(\boldsymbol{a}_O^n, \boldsymbol{a}_H^n | \boldsymbol{\theta}'\right)}{q^n\left(\boldsymbol{a}_H^n\right)} \mathrm{d}\boldsymbol{a}_H^n$$
$$+ \sum_{i=1}^{n} D\left(q^n(\boldsymbol{a}_H^n) \parallel c_{LT}\left(\boldsymbol{a}_H^n | \boldsymbol{a}_O^n, \boldsymbol{\theta}'\right)\right)$$

Parameter Estimation with Known Structure

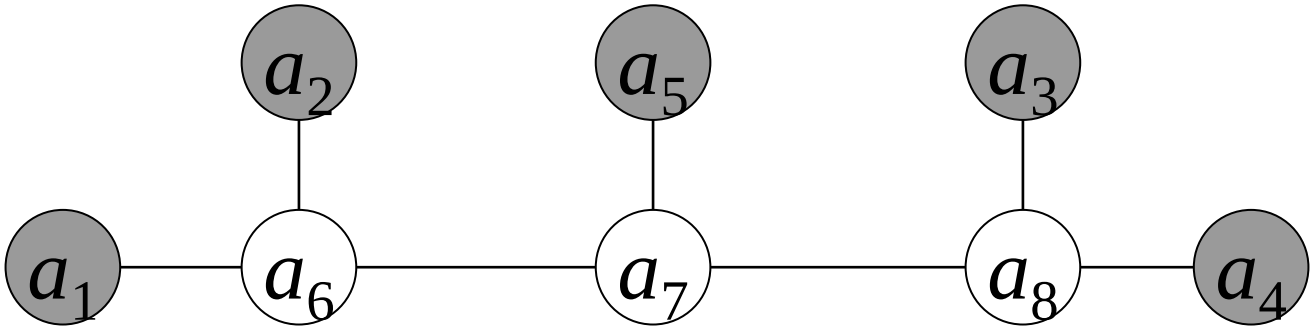


- Gaussian copula case: EM $q^n\left(\boldsymbol{a}_H^n\right) = c\left(\boldsymbol{a}_H^n | \boldsymbol{a}_O^n | \boldsymbol{\theta}\right)$
  - E-step: closed form inference, *O(Nt)* per iteration
  - M-step: maximize the expected compete-data log-likelihood

$$l\left(\boldsymbol{\theta}\right) = \sum_{i=1}^{n} \int_{\mathbb{I}^{t-d}} q^n\left(\boldsymbol{a}_H^n\right) \ln \frac{c_{LT}\left(\boldsymbol{a}_O^n, \boldsymbol{a}_H^n | \boldsymbol{\theta}'\right)}{q^n\left(\boldsymbol{a}_H^n\right)} \mathrm{d}\boldsymbol{a}_H^n$$

$$+ \sum_{i=1}^{n} D\left(q^n\left(\boldsymbol{a}_H^n\right) \| \, c_{LT}\left(\boldsymbol{a}_H^n | \boldsymbol{a}_O^n, \boldsymbol{\theta}'\right)\right)$$

Parameter Estimation with Known Structure



- Non-gaussian copula case: variational EM
  - E-step: approximate inference, $O(sN|E|k^2) + |E|$ bivariate integrals per iteration
  - M-step: approximate maximization, need to update $|E|$ bivariate copula parameters

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{n} \int_{\mathbb{I}^{t-d}} q^n(\boldsymbol{a}_H^n) \ln \frac{c_{LT}\left(\boldsymbol{a}_O^n, \boldsymbol{a}_H^n | \boldsymbol{\theta}'\right)}{q^n\left(\boldsymbol{a}_H^n\right)} \mathrm{d}\boldsymbol{a}_H^n$$

$$+ \sum_{i=1}^{n} D\left(q^n(\boldsymbol{a}_H^n) \parallel c_{LT}\left(\boldsymbol{a}_H^n | \boldsymbol{a}_O^n, \boldsymbol{\theta}'\right)\right)$$

Unknown Structure

- Gaussian LTCs: same as for tree-structured Gaussians
  - Size of possible trees can be limited
  - e.g., can use information distances [Choi et al 2011]
- Non-Gaussian LTCs: need to restrict the space of possible models
  - Very large space of structures/copula families
  - Fix the bivariate copula family
  - Consider only binary latent tree copulas
    - Observed nodes = leafs
    - Motivation: Any Gaussian LTC is equivalent to some binary LTC

Bottom-up Binary LTC Learning

[Similar to Harmeling and Williams 11]

- Initialize the subtrees to consist of individual variables (variable = root of a subtree)
- Iterate until all variables are in one tree
  - Estimate mutual informations (Mis) between the root nodes
  - Pick the pair of roots with the largest MI
  - Merge the subtrees by creating a new latent root node
  - Re-estimate parameters (EM)

Bottom-up Binary LTC Learning

[Similar to Harmeling and Williams 11]

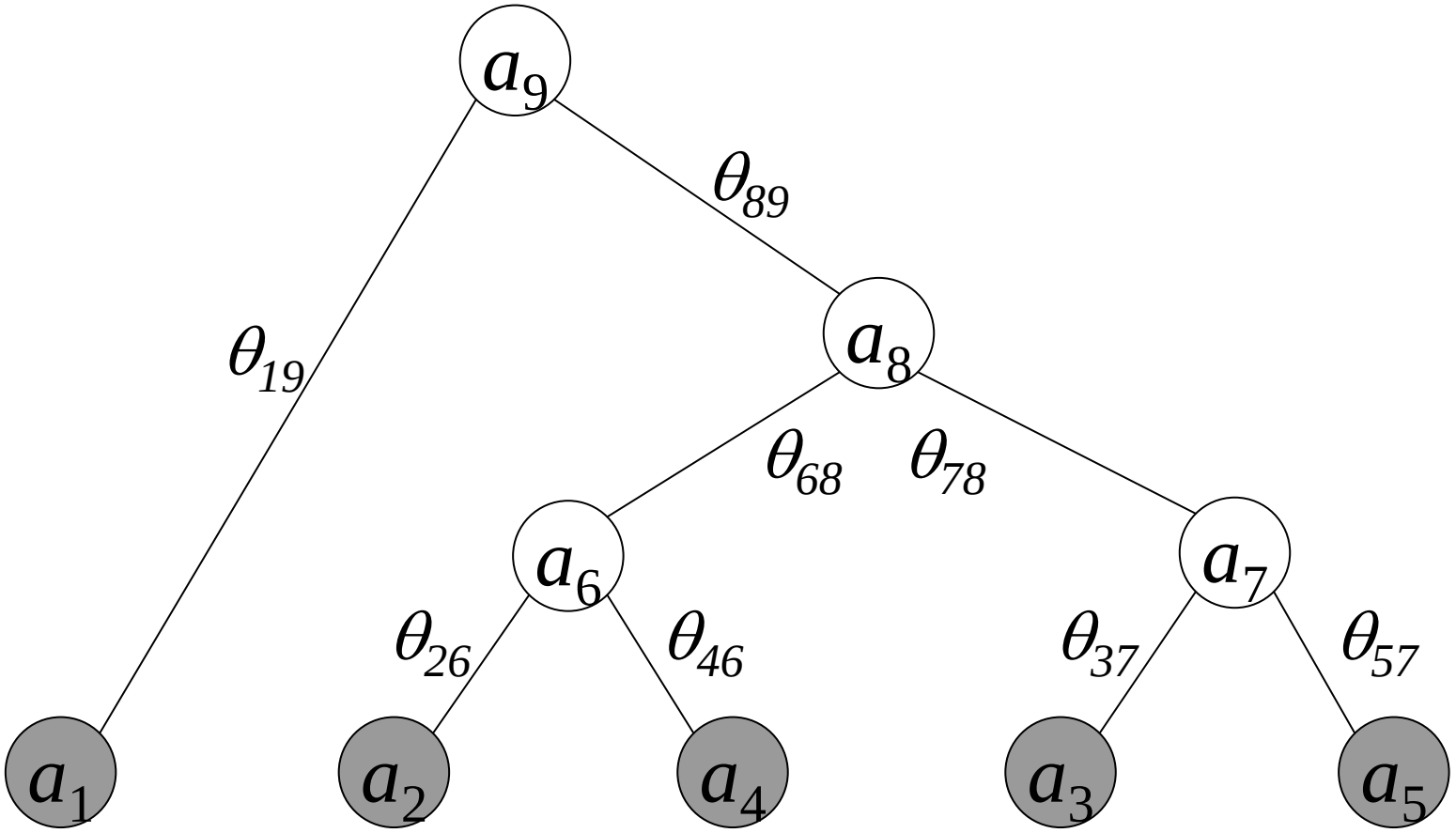
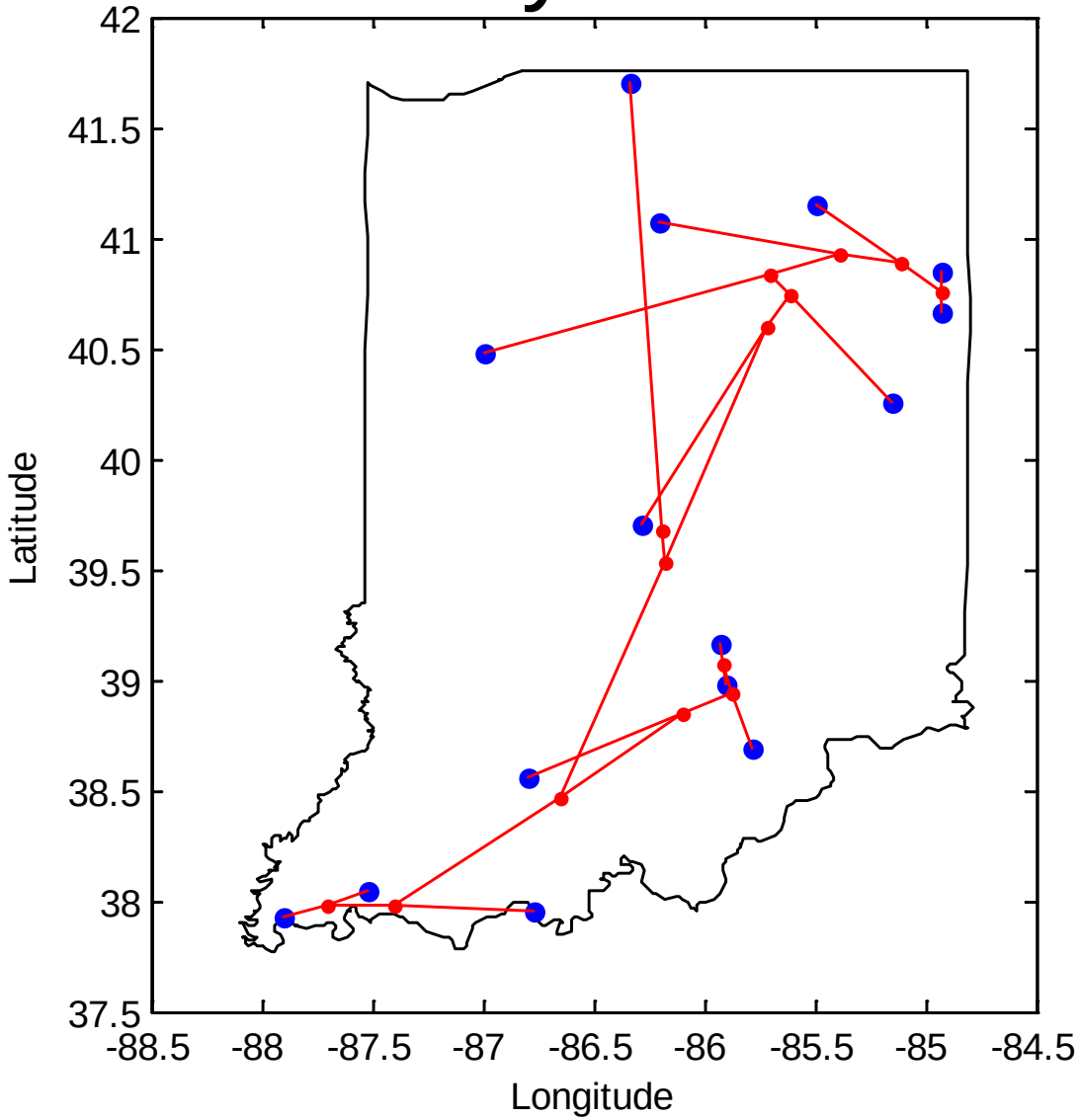


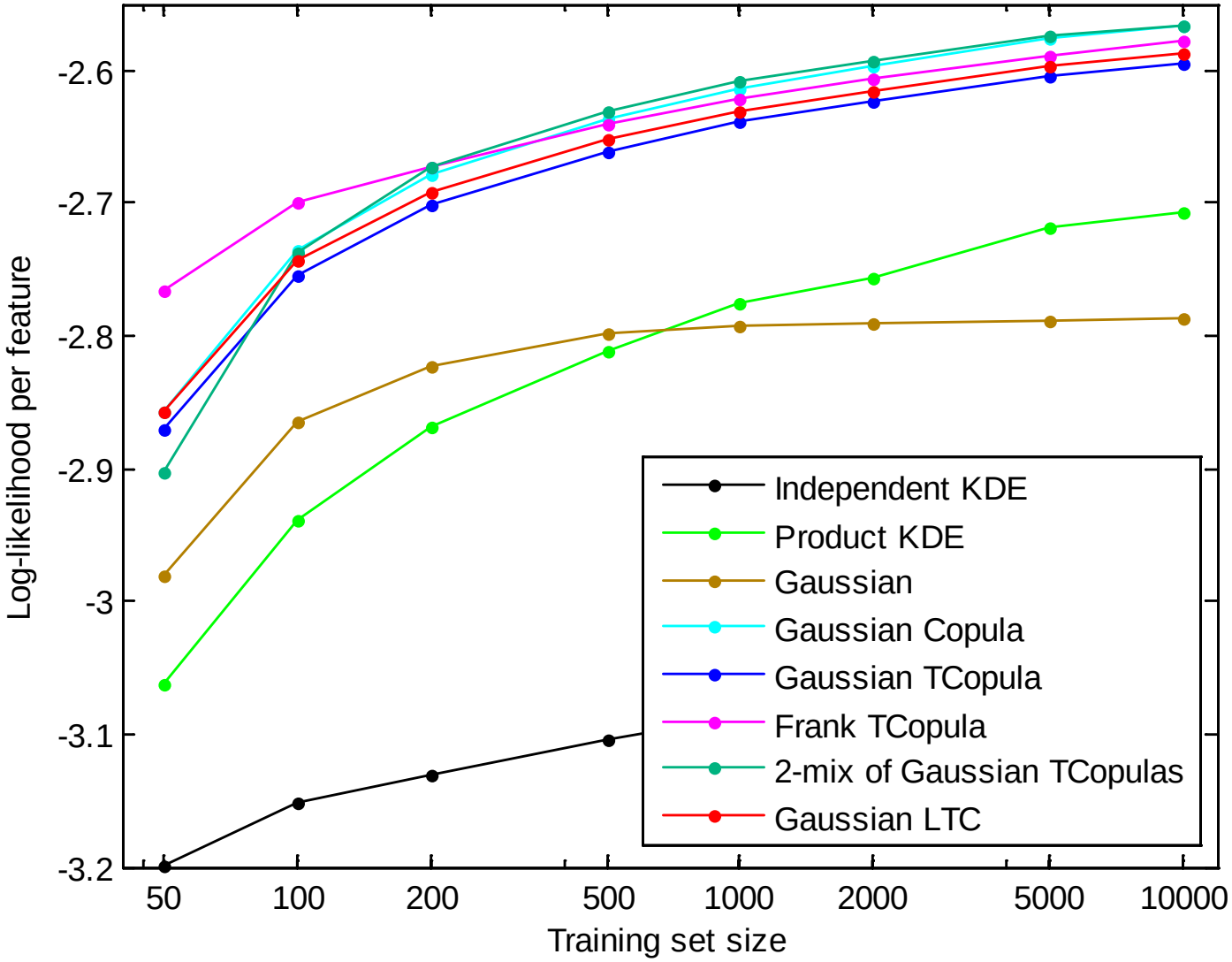
Illustration for Building of Hierarchy of Rainfall Stations



State of Indiana (USA)

Average monthly observations for 15 rainfall stations 1951-1996 (47 years)

Experiments: Log-Likelihood on Test Data



UCI ML Repository MAGIC data set

12000 10-dimensional vectors

2000 examples in test sets

Average over 10 partitions

Summary

- Multivariate distribution = univariate marginals + copula
- New model: tree-structured multivariate distribution with marginally uniform latent variables (latent tree copula, LTC)
  - Sufficient to employ only bivariate copula families!
- Closed form inference for Gaussian copulas (efficient)
- Variational inference for non-Gaussian copulas (slow)
- Parameter estimation using the EM algorithm
- Bottom-up structure learning for bivariate LTCs
- Can be used for parsimonious multivariate density estimation or to structure variables into hierarchies

Thank you!

Software:

http://www.stat.purdue.edu/~skirshne/LTC/index.html

Support:

US National Science Foundation Award AGS-1025430

Questions?

See me at the poster tonight for more details