Sum-Product Copulas

UNIVERSITY OF CAMBRIDGE
Department of Engineering

Ramona Comanescu, Robert Peharz, José Miguel Hernández-Lobato

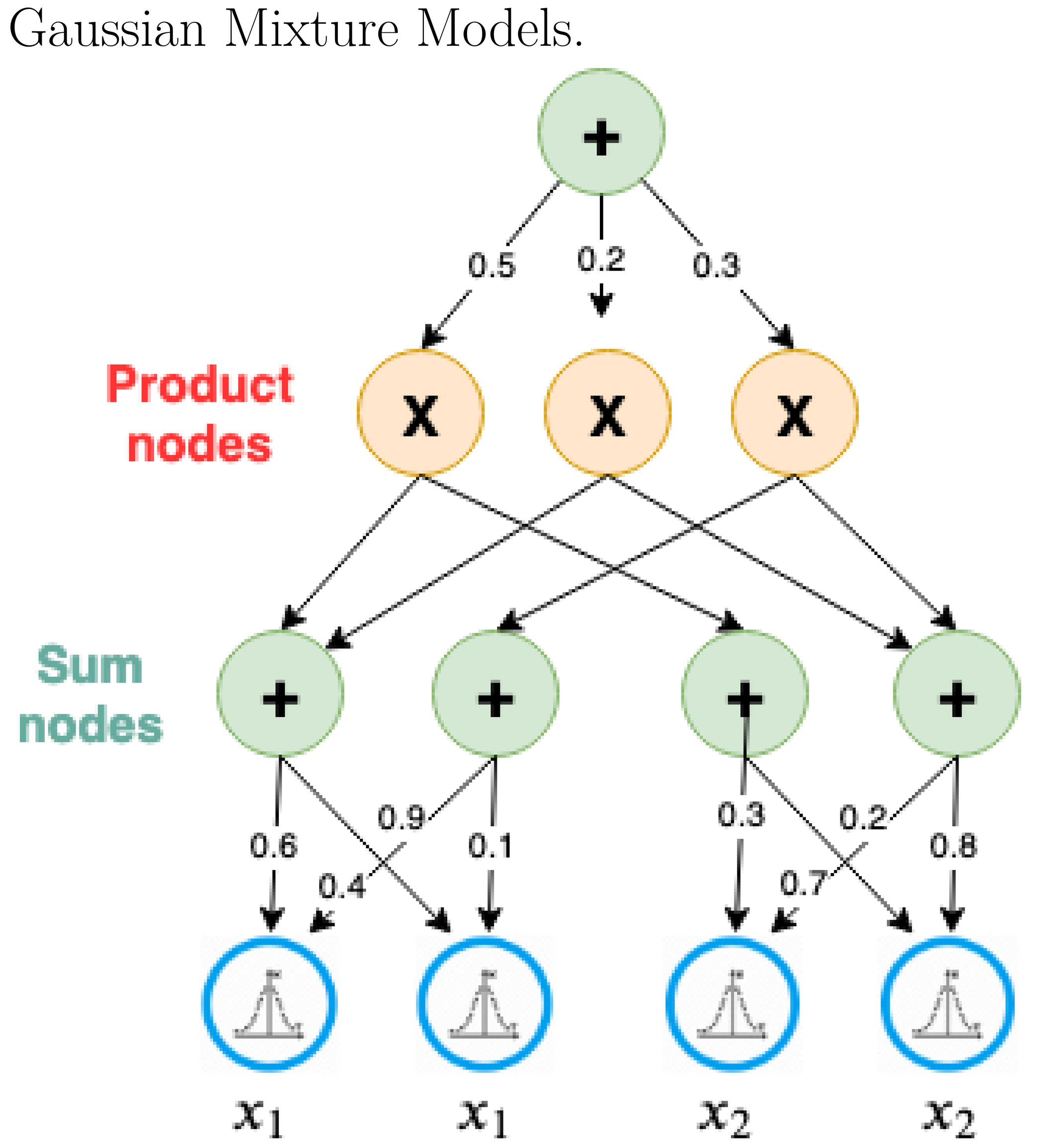
rgc35@cam.ac.uk

Motivation

We aim to combine the advantages of sum product networks and copulas, both effective **frameworks for modeling multivariate distributions**. We will distort sum-product networks so that they satisfy the copula constraints.

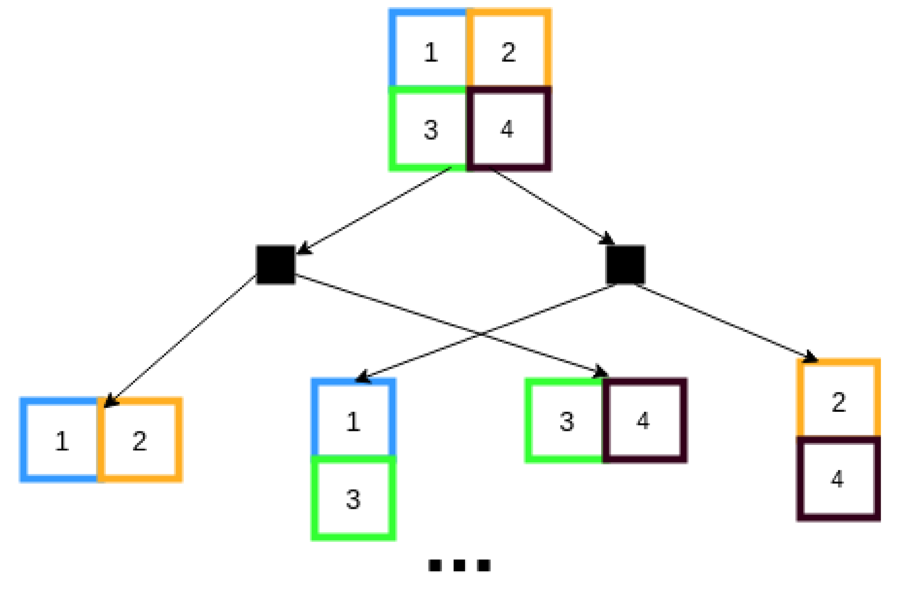
Sum-Product Networks

Sum-product networks (SPNs) [Poon and Domingos, 2011] are a prominent class of **tractable probabilistic model**, facilitating **fast, exact inference**. They can be seen as a generalization of Gaussian Mixture Models.



Learning structure is hard

Recursively divide root into random **region graphs**.



Copulas

Copula Function:

$$C(u_1, \dots, u_N) = P(U_1 \leq u_1, \dots, U_N \leq u_N)$$

Sklar's Theroem [Sklar, 1959]: For any multivariate distribution, there exists C :

$$F(x_1, \dots, x_D) = C(F_1(x_1), \dots, F_D(x_D))$$

Gaussian Copula and the Financial Crisis:
Asymptotic tail independence, unable to give sufficient weight to scenarios where many joint defaults occur.
Known Copula families are typically limited to a small number of dimensions and do not capture dependencies in multimodal distributions.

Copula-based Factorization of Joint Densities:

$$f(z_1, z_2, \dots z_d) = \prod_d f_d(z_d) \times \underbrace{c(u_1, u_2, \dots \dots u_d)}_{\text{Copula pdf}},$$

where $f_d(x_d)$ are marginal PDFs, $u_d = F_d(z_d)$.

Copula function from known joint densities

$$c(u_1, u_2 \cdots u_d)$$
$$= \frac{f\left(F_1^{-1}(u_1), \dots \dots F_d^{-1}(u_d)\right)}{f_1\left(F_1^{-1}(u_1)\right) \times f_2\left(F_2^{-1}(u_1)\right) \cdots f_d\left(F_d^{-1}(u_d)\right)}$$

Satisfying the copula constraints with SPNs

$$c_{SPN}(u_1, u_2, \dots u_d; \theta) = \frac{\psi(y_1, y_2, \dots y_d; \theta)}{\prod_{d=1}^{D} \psi_j(y_d)}$$

where ψ is the SPN joint, $y_d = \Psi_d^{-1}(u_d)$, ψ_d and Ψ_d^{-1} denote the marginal and inverse CDF of the SPN along the d^{th} dimension.

Optimize:

$$\max_{\theta, \mathbf{y}} \sum_{n=1}^{N} \left(\log \psi(y_{n,1}, \dots, y_{n,d}; \theta) - \sum_{d=1}^{D} \log(\psi_d(y_{n,d})) \right),$$

$\forall_{n,d}$ s.t. $\Psi_d(y_{n,d}) = u_{n,d}$,

where $\Psi_d(y_{n,d})$ are obtained from the SPN marginals.

SPN Superpowers

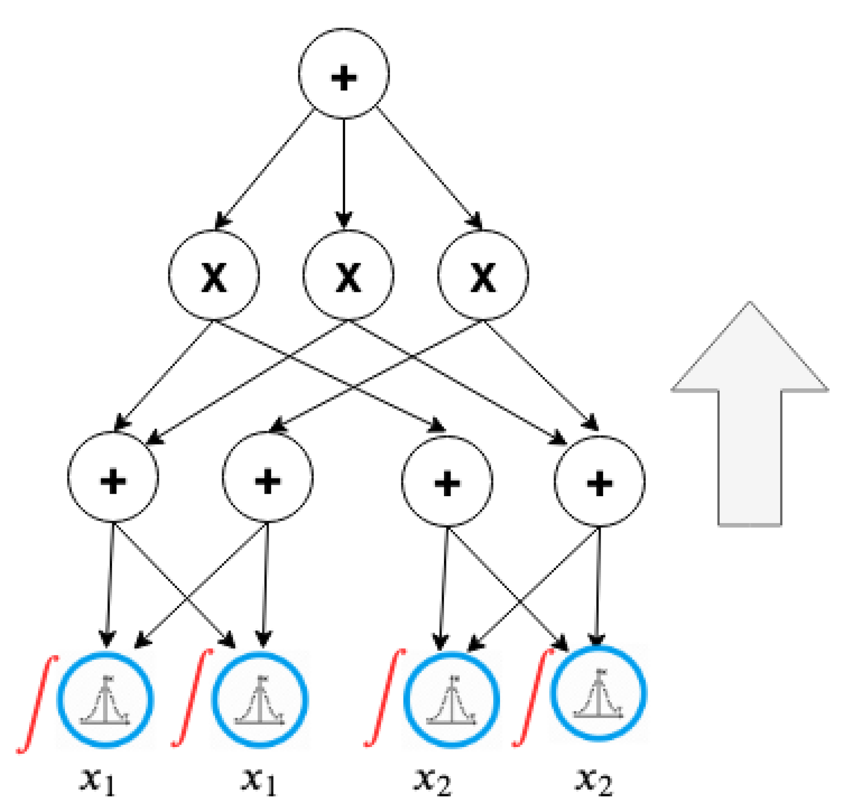
- Easy to: sample, learn parameters, marginalise missing R.V, infer missing R.V., compute likelihood.

Differentiation:
Evaluate root in one **upward pass**, followed by one **downward pass**:

$$\frac{\partial S(\mathbf{x})}{\partial w_{ij}} = \frac{\partial S(\mathbf{x})}{\partial S_i(\mathbf{x})} S_j(\mathbf{x})$$

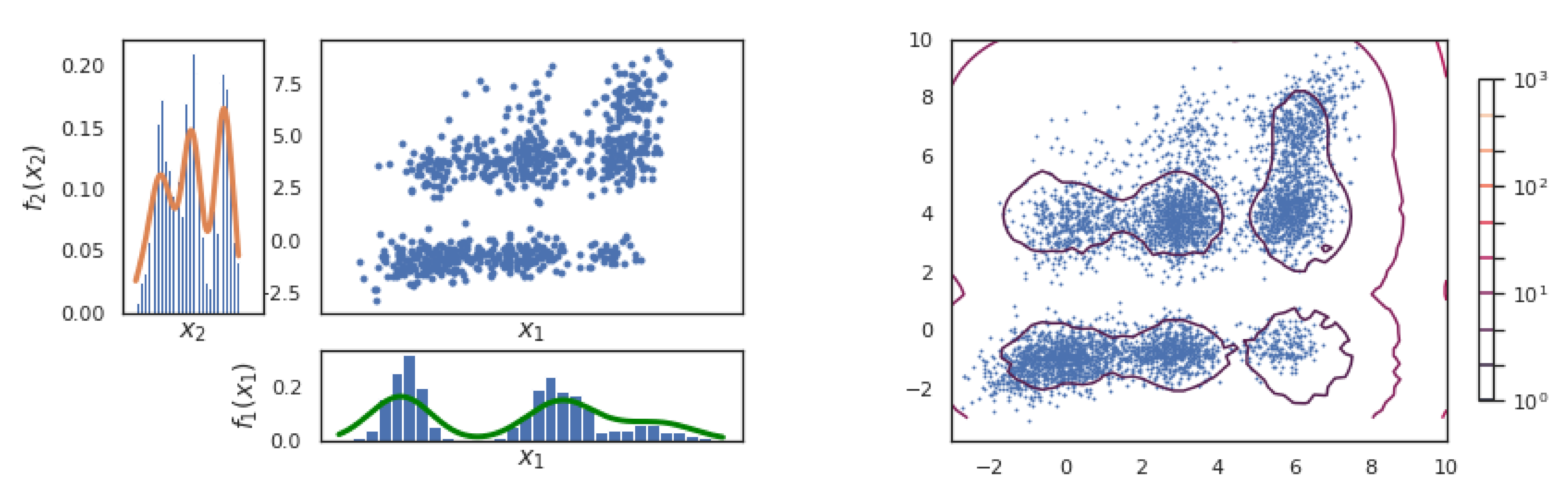
Marginalization:

$$\int \int p(x1, x2)$$



Density estimation with copulas

- Estimate $u_d = F_d(z_d)$ and $f_d(z_d)$
- Create random SPN structure over random variables (R.V.)
- Learn SPN over y_d, constraining $\Psi_d(y_d) = u_d$. RMSE: 0.07.
- Experiment over synthetic data with dependent R.V. and multimodal marginals



(a) Synthetic two dimensional data and their marginals.

(b) Sum-Product Copula fit

Further work

- Evaluate on image tasks
- Improve SPN implementation: probabilistic dropout, full Gaussian distributions
- Different constraint optimization methods
- Compare with copula bayesian neurol networks and tree-structured copulas

References

Poon, H. and Domingos, P. (2011). Sum-product networks: A new deep architecture. In *ICCV*.

Sklar, M. (1959). Fonctions de repartition an dimensions et leurs marges.