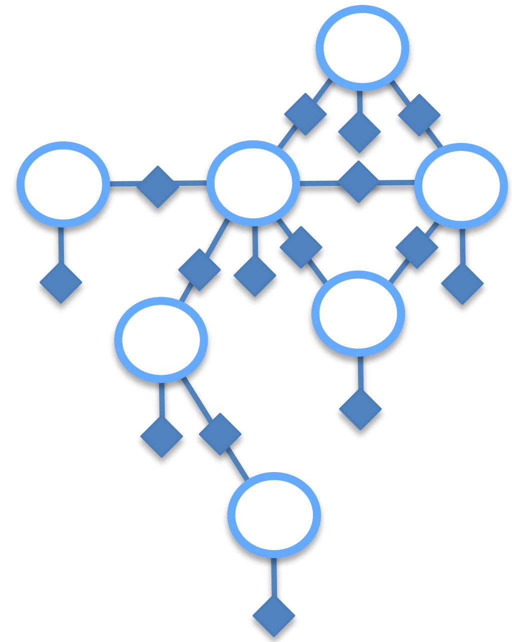
Exact inference and learning for cumulative distribution functions on loopy graphs

Jim C. Huang, Nebojsa Jojic and Christopher Meek

Microsoft Research

- Given a product of functions $F(\mathbf{x}) = \prod_{s \in S} \phi_s(\mathbf{x}_s)$, compute $\partial_{\mathbf{x}} \left[F(\mathbf{x}) \right]$

- e.g.:
  - Combinatorics
  - Statistical physics
  - Computational chemistry



- Graphical models for cumulative distribution functions (CDFs)

Message-passing in trees

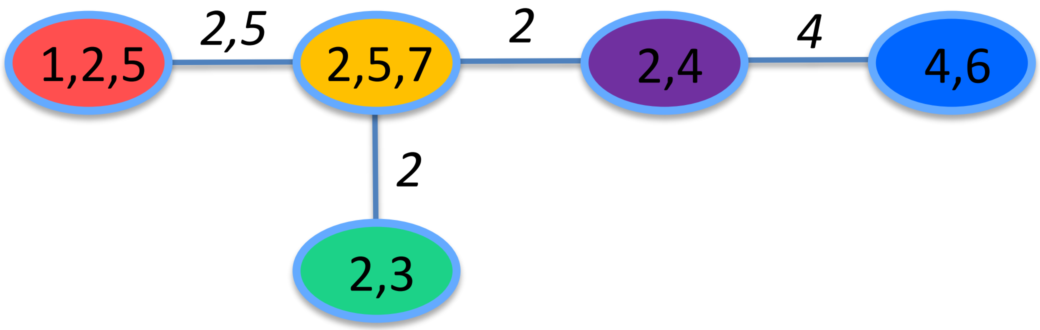
Computing probability density functions (PDFs) $$P(\mathbf{x}) = \partial_{\mathbf{x}} \left[F(\mathbf{x}) \right]$$ (Huang and Frey, 2008)

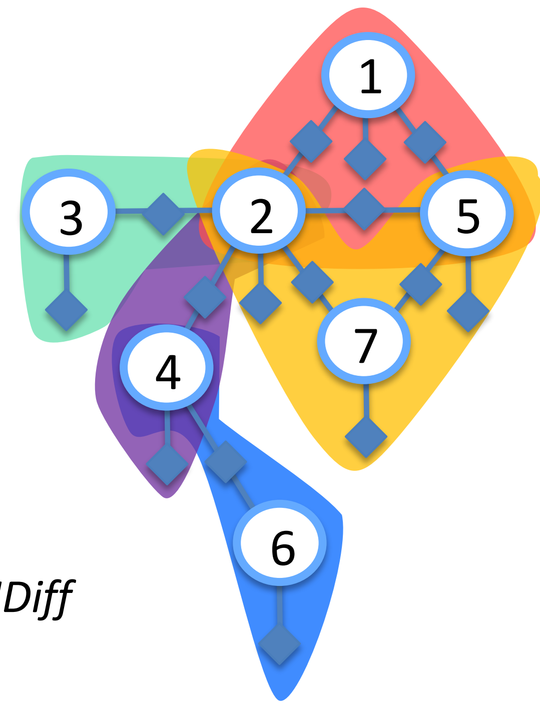
Parameter estimation, or learning $$\nabla_\theta P(\mathbf{x}|\theta) = \nabla_\theta \partial_{\mathbf{x}} \left[F(\mathbf{x}|\theta) \right]$$ (Huang and Jojic, 2010)

JDiff: Message-passing for differentiation on graphs

Divide and conquer

Decompose *global* differentiation into smaller *local* sub-problems





Recursive decomposition is naturally described by a junction tree (Lauritzen and Spiegelhalter, 1988)

Message-passing algorithm for differentiation on graphs: *JDiff*

Complexity: Exponential in tree-width of graph

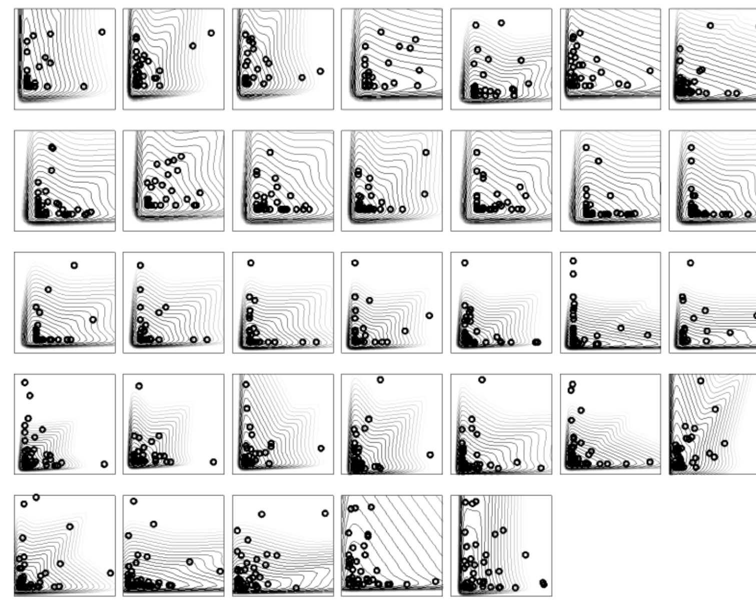
Cumulative distribution networks

Well-suited for statistical modelling of heavy-tailed data

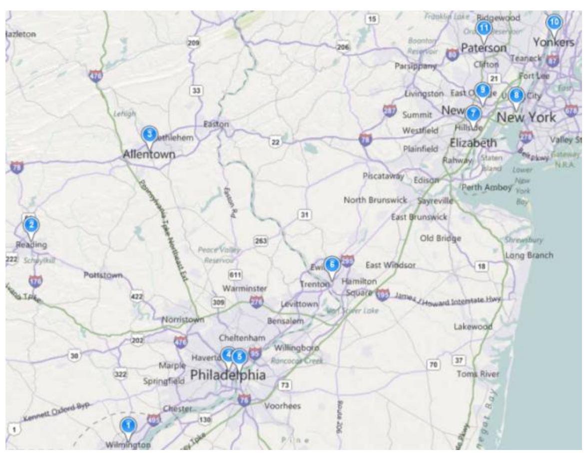
Given measurements at different spatial locations…

e.g.: Modelling rainfall data

Learn a joint probability density model

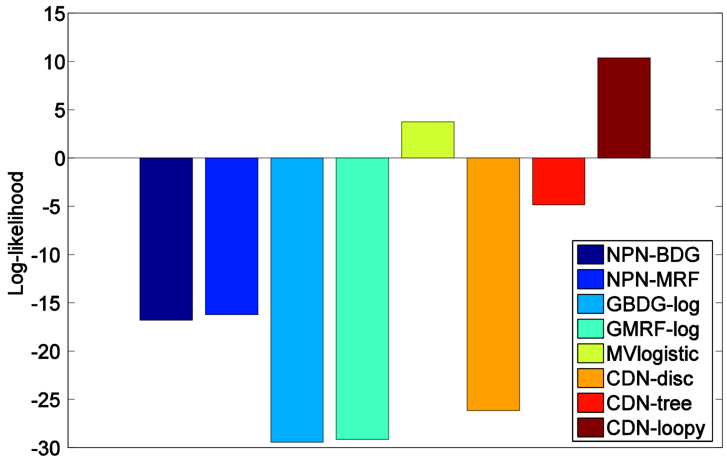


e.g.: Modelling H1N1 mortality

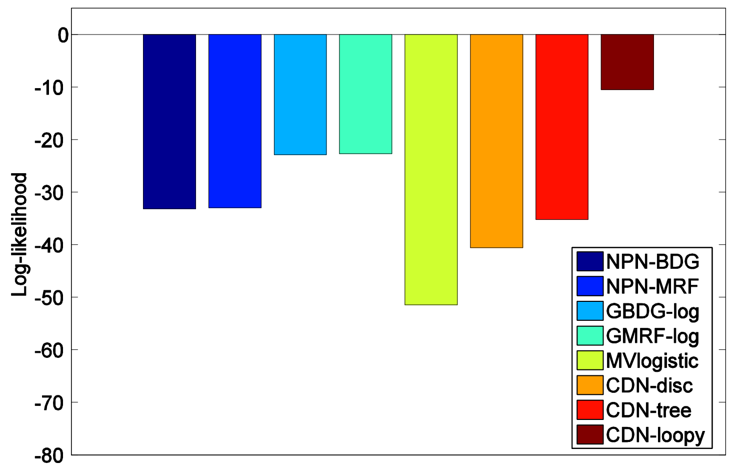


Learning loopy cumulative distribution networks

Test likelihoods on real, heavy-tailed data:



Rainfall data



H1N1 mortality

Symbolic differentiation on graphs:

	JDiff	Mathematica	D*
Grids	1 s. – 20 min.	6.2 s. - ∞	9.2 s. - ∞
Cycles	0.81 s. – 2.83 s.	1.2 s. – 580 s.	6.7 s. – 12.7 s.

Come see our poster #87 tonight!