Exact inference and learning for cumulative distribution functions on loopy graphs

Jim C. Huang, Nebojsa Jojic and Christopher Meek

NIPS 2010

Presented by Jenny Lam
Previous work

A CDN $\mathcal{G}$ is a bipartite graph $(V, S, E)$ where

- $V$ is the set of variable nodes,
- $S$ is the set of function nodes,
- with $\phi : \mathbb{R}^{|N(\phi)|} \rightarrow [0, 1]$ is a CDF,
- $E$ is the set of edges, connecting functions to their variables.

The joint CDF of this CDN is $F(x) = \prod_{\phi \in S} \phi$. 
CDNs: what are they for?

- PDF models must enforce a normalization constraint.
- PDFs are made more tractable by restricting to, e.g., Gaussians.
- Many non-Gaussian distributions are conveniently parametrized as CDFs.
- CDNs can be used to model heavy-tailed distributions, which are important in climatology and epidemiology.
Inference from joint CDF

Conditional CDF

\[ F(x_B|x_A) = \frac{\partial_{x_A} F(x_A, x_B)}{\partial_{x_A} F(x_A)} \]

Likelihood

\[ P(x|\theta) = \partial_x F(x|\theta) \]

For MLE, need gradient of log likelihood

\[ \nabla_\theta \log P(x|\theta) = \frac{1}{P(x|\theta)} \nabla_\theta P(x|\theta) \]
Mixed derivative of a product

\[ \partial_x [f \cdot g] = \sum_{U \subseteq x} \partial_U f \cdot \partial_{\overline{U}} g \]

which has \(2^{|x|}\) terms. More generally,

\[ \partial_x \prod_{i=1}^{k} f_i = \sum_{U_1, \ldots, U_k} \prod_{i=1}^{k} \partial_{U_i} f_i \]

where we sum over all partitions \(U_1, \ldots, U_k\) of \(x\) into \(k\) subsets. There are \(k^{|x|}\) terms in this sum.
Mixed derivative over a separation

Partition the functions of a CDN into $M_1$ and $M_2$

- with variable sets $C_1$ and $C_2$ and $S_{1,2} = C_1 \cap C_2$
- and $G_1$ and $G_2$ the products of functions in $M_1$ and $M_2$.

Then

$$\partial_x [G_1 G_2] = \sum_{A \subseteq S_{1,2}} \left[ \partial_{x_{C_1 \setminus S_{1,2}}} \partial_{x_A} G_1 \right] \left[ \partial_{x_{C_2 \setminus S_{1,2}}} \partial_{x_{S_{1,2} \setminus A}} G_2 \right]$$
Junction Tree: definition

Let $G = (V, S, E)$ be a CDN.
A tree $T = (C, E)$ is a junction tree for $G$ if

1. $C$ is a cover for $V$:
   each $C_j \in C$ is a subset of $V$ and $\bigcup_j C_j = V$

2. family preservation holds:
   for each $\phi \in S$, there is a $C_j \in C$ such that $\text{scope}(\phi) \subseteq C_j$

3. running intersection property holds:
   if $C_i \in C$ is on the path between $C_j$ and $C_k$, then $C_j \cap C_k \subseteq C_i$
Junction Tree: example
Construction of the junction tree

In implementation

- greedily eliminate the variables with the minimal fill-in algorithm
- construct elimination subsets for nodes in the junction tree using the MATLAB Bayes Net Toolbox (Murphy, 2001)
Decomposition of the joint CDF

Partitioning function of $S$ into $M_j$, the joint CDF is

$$F(x) = \prod_{C_j \in C} \psi_j(x_{C_j}), \quad \text{where } \psi_j \equiv \prod_{\phi \in M_j} \phi$$

Let $r$ be a chosen root of the joint tree. Then

$$F(x) = \psi_r(x_{C_r}) \prod_{k \in E_r} T_r^k(x)$$

where

$$T_r^k(x) = \prod_{j \in \tau_r^k} \psi_j(x_{C_j})$$

and $\tau_r^k$ is the subtree rooted at $k$. 

Derivative of the joint CDF

\[
\partial_x F(x) = \partial_x \left[ \psi_r(x_{C_r}) \prod_{k \in \mathcal{E}_r} T^r_k(x) \right]
\]

\[
= \partial_{x_{C_r}} \partial_{x_{C_r}} \left[ \psi_r(x_{C_r}) \prod_{k \in \mathcal{E}_r} T^r_k(x) \right]
\]

\[
= \partial_{x_{C_r}} \left[ \psi_r(x_{C_r}) \partial_{x_{C_r}} \prod_{k \in \mathcal{E}_r} T^r_k(x) \right]
\]

\[
= \partial_{x_{C_r}} \left[ \psi_r(x_{C_r}) \prod_{k \in \mathcal{E}_r} \partial_{x_{\tau^r_k \setminus C_r}} T^r_k(x) \right]
\]

the last equality follows from the running intersection property
Messages to the root of the junction tree

Message from children $k$ to root $r$, where $A \subseteq C_r$

$$m_{k \rightarrow r}(A) \equiv \partial_{x_A} \left[ \partial_{x_{\tau_k \setminus C_r}} T^r_k(x) \right]$$

In particular

$$m_{k \rightarrow r}(\emptyset) = \partial_{x_{\tau_k \setminus C_r}} T^r_k(x)$$

At the root, if $U_r \subseteq E_r$, and $A \subseteq C_r$

$$m_r(A, U_r) \equiv \partial_{x_A} \left[ \psi_r(x_{C_r}) \prod_{k \in E_r} m_{k \rightarrow r}(\emptyset) \right]$$
Messages in the rest of the junction tree

\[ m_i(A, U_i) \equiv \partial_{x_A} \left[ \psi_i(x_{C_i}) \prod_{j \in U_i} m_{j \rightarrow i}(\emptyset) \right] \]

where \( A \subseteq C_i \) and \( U_i \subseteq \mathcal{E}_i \).

\[ m_{j \rightarrow i}(A) \equiv \partial_{x_A} \left[ \partial_{x_{\tau_{j \rightarrow i} \setminus \tau_{i,j}}} T_j^i(x) \right] \]

where \( A \subseteq S_{i,j} \).
Messages in the rest of the junction tree

In terms of messages

\[ m_i(A, U_i) = \partial_{x_A} \left[ \begin{array}{c} \psi_i(x_{C_i}) m_{k\rightarrow i}(\emptyset) \\ \prod_{j \in U_i \setminus \{k\}} m_{j\rightarrow i}(\emptyset) \end{array} \right] = \sum_{B \subseteq A \cap S_{i,k}} m_{k\rightarrow i}(B) m_i(A \setminus B, U_i \setminus \{k\}) \]

\[ m_{j\rightarrow i}(A) = \partial_{x_{A,C_j \setminus S_{i,j}}} \left[ \begin{array}{c} \psi_j(x_{C_j}) \\ \prod_{l \in \mathcal{E}_j \setminus \{i\}} T^j_l(x) \end{array} \right] = m_j(A \cup (C_j \setminus S_{i,j}), \mathcal{E}_j \setminus \{i\}) \]
Gradient of the likelihood

Likelihood

\[ P(x|\theta) = \partial_x [F(x|\theta)] = m_r(C_r, \mathcal{E}_r) \]

Gradient likelihood

\[ \nabla_\theta m_r(C_r, \mathcal{E}_r) \]

decomposed similarly to \( m_r(C_r, \mathcal{E}_r) \) in the junction tree:

- \( g_i \equiv \nabla_\theta m_i \)
- \( g_{j \rightarrow i} \equiv \nabla_\theta m_{j \rightarrow i} \)
JDiff algorithm: outline

for each cluster (from leaf to root):

1. compute derivative within cluster
2. compute messages from children
3. send messages to parent
foreach Node \( j \in C \) do
\[
U_j \leftarrow \emptyset; \quad \psi_j \leftarrow \prod_{s \in M_j} \phi_s;
\]
end

1 foreach Subset \( A \subseteq C_j \) do
\[
m_j(A, \emptyset) \leftarrow \partial_{x_A}[\psi_j];
\]
\[
g_j(A, \emptyset) \leftarrow \nabla_\theta \partial_{x_A}[\psi_j];
\]
end

2 foreach Neighbor \( k \in \mathcal{E}_j \cap \tau^j_k \) do
\[
S_{j,k} \leftarrow C_j \cap C_k;
\]
foreach Subset \( A \subseteq C_j \) do
\[
m_j(A, U_j \cup k) \leftarrow \sum_{B \subseteq A \cap S_{j,k}} m_{k \rightarrow j}(B)m_j(A \setminus B, U_j);
\]
\[
g_j(A, U_j \cup k) \leftarrow \sum_{B \subseteq A \cap S_{j,k}} m_{k \rightarrow j}(B)g_j(A \setminus B, U_j) + g_{k \rightarrow j}(B)m_j(A \setminus B, U_j);
\]
end
\[
U_j \leftarrow U_j \cup k;
\]
end
if \( j \neq r \) then
\[
k \leftarrow \{l \mid \mathcal{E}_j \cap \tau^l_j \neq \emptyset\}; \quad S_{j,k} \leftarrow C_j \cap C_k;
\]
foreach Subset \( A \subseteq S_{j,k} \) do
\[
m_{j \rightarrow k}(A) \leftarrow m_j(A \cup C_j \setminus S_{j,k}, \mathcal{E}_j \setminus k);
\]
\[
g_{j \rightarrow k}(A) \leftarrow g_j(A \cup C_j \setminus S_{j,k}, \mathcal{E}_j \setminus k);
\]
end
else
\[
return (m_r(C_r, \mathcal{E}_r), g_r(C_r, \mathcal{E}_r))
\]
end
end
Complexity of JDiff

O-notation of number of steps/terms in each inner loop for fixed $j$:

1. \[
\sum_{k=1}^{C_j} \binom{|C_j|}{k} |M_j|^k = (|M_j| + 1)|C_j|
\]

2. \[
(|E_j| - 1) \max_{k \in E_j} \sum_{l=0}^{S_{j,k}} \binom{|S_{j,k}|}{l} 2|C_j \setminus S_{j,k}| 2^l
\]

3. $2|S_{j,k}|$

Total. Exponential in tree-width of graph

\[
O \left( \max_j (|M_j| + 1)|C_j| + \max_{(j,k) \in E} (|E_j| - 1)2|C_j \setminus S_{j,k}| 3|S_{j,k}| \right)
\]
Application: symbolic differentiation on graphs

Computation of $\partial_x F(x)$ on CDNs

- Grids: $3 \times 3$ to $9 \times 9$
- Cycles: 10 to 20 nodes

<table>
<thead>
<tr>
<th></th>
<th>JDiff</th>
<th>Mathematica</th>
<th>D*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grids</td>
<td>1 s. – 20 min.</td>
<td>6.2 s. - $\infty$</td>
<td>9.2 s. - $\infty$</td>
</tr>
<tr>
<td>Cycles</td>
<td>0.81 s. – 2.83 s.</td>
<td>1.2 s. – 580 s.</td>
<td>6.7 s. – 12.7 s.</td>
</tr>
</tbody>
</table>
Application: modeling heavy-tailed data

- Rainfall: 61 daily measurements of rainfall at 22 sites in China
- H1N1: 29 weekly mortality rates in 11 cities in the Northeastern US during the 2008-2009 epidemic
Application: modeling heavy-tailed data

Average test log-likelihoods on leave-one-out cross-validation

Rainfall data

H1N1 mortality
Future work

- Develop compact models (bounded treewidth) for applications in other areas (seismology)
- Study connection between CDNs and other copula-based algorithms
- Develop faster approximate algorithms