Closed-Form Learning of Markov Networks from Dependency Networks

Daniel Lowd, University of Oregon

Dependency networks (DNs):
Easy to learn, ugly semantics.

Markov networks (MNs):
Hard to learn, nice semantics.

Best of both worlds: Learn a DN and convert it into an MN.
How do we convert a DN into an MN?

**KEY IDEA:** We can express probability ratios using only conditional probabilities, which are given by the DN:

\[
\frac{P(A, B)}{P(a^0b^0)} = \frac{P(A|b^0)}{P(a^0|b^0)} \cdot \frac{P(B|A)}{P(b^0|A)}
\]

- \( a^0 \rightarrow A \)
- \( b^0 \rightarrow B \)

Use the conditional probability ratios to construct MN factors:

\[
\phi_1(A) = \frac{P(A|b^0)}{P(a^0|b^0)} \quad \phi_2(A, B) = \frac{P(B|A)}{P(b^0|A)}
\]

**Exact** for consistent DNs!

**Runs in linear time!**
How well does DN2MN work?

**Methods:** Learned DNs on 12 real-world datasets and converted to MNs by both DN2MN and weight learning.

**Results:** DN2MN has similar or better accuracy than weight learning and is orders of magnitude faster.

Complete source code: [http://libra.cs.uoregon.edu](http://libra.cs.uoregon.edu)