Scalable Learning of Markov Networks

Daniel Lowd
University of Oregon

Joint work with:
Jesse Davis, Katholieke Universiteit Leuven
Problem Setting

**Goal:** Learn the uncertain relationships among many random variables

**Applications:** medical diagnosis, fault diagnosis, web usage analysis, bioinformatics, recommender systems, etc.

**State-of-the-art:** Markov networks are a very flexible way to represent these relationships, but are hard to learn.
Markov Networks: Representation

Flu → Wheeze → Asthma

Variables

Smoke ∧ Cancer

Weight of Feature $i$

Feature $i$

$$P(x) = \frac{1}{Z} \exp \left( \sum w_i f_i(x) \right)$$

(aka Markov random fields, Gibbs distributions, log-linear models, exponential models, maximum entropy models)
Problem Definition

Training Data

<table>
<thead>
<tr>
<th>F</th>
<th>W</th>
<th>A</th>
<th>S</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Markov Network Structure

\[ P(F, W, A, S, C) \]
The Challenge

Key Idea: Local models $\rightarrow$ Global model

Learn local distributions to predict each variable separately, then combine them into a global, joint distribution.
General Strategy: Local Models

Learn a “local model” to predict each variable given the others:

- \( P(A|B): \)
- \( P(B|A): \)
- \( P(C|B,D): \)
- \( P(D|C): \)

Combine the models into a Markov network:
L1 Structure Learning
[Ravikumar et al., 2009]

**Given:** Set of variables = \{F, W, A, S, C\}

**Do:** L1 logistic regression to predict each variable

Construct pairwise features between target and each variable with non-zero weight

**Model** = \{S ∧ C, ... , W ∧ F\}

**Downside:** Algorithm restricted to pairwise features
**DTSL: Decision Tree Structure Learning**  
[Lowd and Davis, ICDM 2010]

**Given:** Set of variables = \{F, W, A, S, C\}

**Do:** Learn a decision tree to predict each variable

P(C | F, S) = ….  

\[
P(F | C, S) = \ldots
\]

Construct a feature for each leaf in each tree:

- F ∧ C
- F ∧ ¬C
- ¬F ∧ S ∧ C
- ¬F ∧ S ∧ ¬C
- ¬F ∧ ¬S ∧ C
- ¬F ∧ ¬S ∧ ¬C

Can also consider sub-features, for finer-grained granularity.
Accuracy

Local methods beat global search on 11 out of 13 datasets: DTSL is best on 5; L1 on 6
Do long features matter?

DTSL Feature Length

- MSNBC
- Plants
- Book
- EachMovie
- NLTS
- KDD Cup
- 20 Newsgroups
- WebKB
- MSWeb
- Reuters
- Audio
- Jester
- Netflix

DTSL Better

L1 Better
Overview

• Using local models to construct a global model is an effective strategy.
• L1 and DTSL work best on different datasets:
  – L1 can handle independent influence
  – DTSL can handle complex interactions
• DTSL is really fast
  – Bottleneck is now weight learning!
    (Even though we’re optimizing pseudo-likelihood.)
Can we do better?

- DTSL and L1 only use the structure of the local models, completely ignoring the parameters.

- Can we directly convert a set of conditional distributions into a single joint distribution without relearning weights?

Yes. In closed form. In linear time.
From Conditional Distributions to a Joint Distribution  

[Lowd, UAI 2012]

KEY IDEA: Compute relative joint probability by multiplying and dividing conditional probabilities.

• The conditional distribution $P(A|B,C,D)$ tells us the relative probabilities of instances that only differ in A:

$$
\frac{P(a^1, b, c, d)}{P(a^0, b, c, d)} = \frac{P(a^1|b, c, d)P(b, c, d)}{P(a^0|b, c, d)P(b, c, d)} = \frac{P(a^1|b, c, d)}{P(a^0|b, c, d)}
$$

• If two instances differ in multiple variables, we repeat the process multiple times:

$$
\frac{P(a^1, b^1, c, d)}{P(a^0, b^0, c, d)} = \frac{P(a^1|b^0, c, d)P(b^1, a^1, b, c)}{P(a^0|b^0, c, d)P(b^0, a^1, b, c)}
$$

• We can use this idea to construct a set of factors for a Markov network. Works for any conditional distributions.
Empirical Comparison

Question: Which method is faster and more accurate – directly converting distributions or relearning all weights from data?

Answer:
1. With decision trees, both methods have similar accuracy but conversion is 60 times faster.
2. With L1 logistic regression, conversion is more accurate and 360 times faster.
Conclusion

• Markov networks are powerful, but hard to learn.
• Probabilistic classifiers are limited, but easy to learn.
• Best of both worlds: Local $\rightarrow$ Global
  – Learn a local distribution for each variable (very fast!)
  – Combine them into a global distribution (even faster!)
  – Faster and more accurate than global search

Code available: libra.cs.uoregon.edu