Learning Tractable Probabilistic Models

Pedro Domingos
University of Washington
Outline

- Motivation
- Probabilistic models
- Standard tractable models
- The sum-product theorem
- Bounded-inference graphical models
- Feature trees
- Sum-product networks
- Tractable Markov logic
- Other tractable models
The Hardest Part of Learning Is Inference

Inference is subroutine of:

- Learning undirected graphical models
- Learning discriminative graphical models
- Learning w/ incomplete data, latent variables
- Bayesian learning
- Deep learning
- Statistical relational learning
- Etc.
Goal: Large Joint Models

- Natural language
- Vision
- Social networks
- Activity recognition
- Bioinformatics
- Etc.
Example: Friends & Smokers

Smoking and Quitting in Groups
Researchers studying a network of 12,067 people found that smokers and nonsmokers tended to cluster in groups of close friends and family members. As more people quit over the decades, remaining groups of smokers were increasingly pushed to the periphery of the social network.

1971 A sample of 1,000 people from the study includes many large groups of smokers.

2000 Nearly three decades later, groups of smokers tended to be smaller and more isolated.

Sources: New England Journal of Medicine;
Dr. Nicholas A. Christakis, James H. Fowler

KEY
- Male smoker
- Female smoker
- Male nonsmoker
- Female nonsmoker
- Friendship, marriage or family tie

Circle size is proportional to the number of cigarettes smoked per day.

THE NEW YORK TIMES
Inference Is the Bottleneck

- Inference is \#P-complete
- It’s tough to have \#P as a subroutine
- Approximate inference and parameter optimization interact badly
- An intractable accurate model is in effect an inaccurate model
- What can we do about this?
One Solution: Learn Only Tractable Models

- **Pro:** Inference problem is solved
- **Con:** Insufficiently expressive

Recent development: Expressive tractable models
(theme of this tutorial)
Outline

● Motivation
● **Probabilistic models**
  ● Standard tractable models
  ● The sum-product theorem
  ● Bounded-inference graphical models
  ● Feature trees
  ● Sum-product networks
  ● Tractable Markov logic
  ● Other tractable models
Why Use Probabilistic Models?

- Correctly handle uncertainty and noise
- Learn with missing data
- Jointly infer multiple variables
- Do inference in any direction
- It’s the standard
- Powerful, consistent set of techniques
Probabilistic Models

- Bayesian networks
- Markov networks
- Log-linear models
- Mixture models
- Logistic regression
- Hidden Markov models
- Cond. random fields
- Max. entropy models
- Probabilistic grammars
- Exponential family
- Markov random fields
- Gibbs distributions
- Boltzmann machines
- Deep architectures
- Markov logic
- Etc.
**Markov Networks**

- **Undirected** graphical models

![Graphical model diagram]

- Potential functions defined over cliques

\[ P(x) = \frac{1}{Z} \prod_{c} \Phi_c(x_c) \]

\[ Z = \sum_{x} \prod_{c} \Phi_c(x_c) \]

<table>
<thead>
<tr>
<th>Smoking</th>
<th>Cancer</th>
<th>( \Phi(S,C) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>4.5</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>4.5</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>2.7</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>4.5</td>
</tr>
</tbody>
</table>
Log-Linear Models

\[ P(x) = \frac{1}{Z} \exp \left( \sum_{i} w_i f_i(x) \right) \]

Weight of Feature \( i \)

Feature \( i \)

\[ f_1(\text{Smoking, Cancer}) = \begin{cases} 
1 & \text{if } \neg \text{Smoking } \lor \text{ Cancer} \\
0 & \text{otherwise}
\end{cases} \]

\[ w_1 = 0.51 \]
Representation and Inference

Bayesian Networks

Markov Networks

Deep Architectures

- Advantage: Compact representation
- Inference: $P(\text{Burglar} \mid \text{Alarm}) = ??$
- Need to sum out Earthquake
- Inference cost exponential in treewidth of graph
Learning Graphical Models

- General idea:
  
  **Empirical statistics = Predicted statistics**

- Requires inference!
- Approximate inference is very unreliable
- No closed-form solution (except rare cases)
- Hidden variables $\rightarrow$ No global optimum
- **Result:** Learning is very hard
Outline

● Motivation
● Probabilistic models
● **Standard tractable models**
  ● The sum-product theorem
  ● Bounded-inference graphical models
  ● Feature trees
  ● Sum-product networks
  ● Tractable Markov logic
  ● Other tractable models
Thin Junction Trees

[Karger & Srebro, SODA-01; Bach & Jordan, NIPS-02; Narasimhan & Bilmes, UAI-04; Chechetka & Guestrin, NIPS-07]

- **Junction tree**: obtained by triangulating the Markov network
- Inference is exponential in **treewidth** (size of largest clique in junction tree)
- **Solution**: Learn only low-treewidth models
- **Problem**: Too restricted (treewidth ≤ 3)
Very Large Mixture Models
[Lowd & Domingos, ICML-05]

- Just learn a naive Bayes mixture model with lots of components (hundreds or more)
- Inference is linear in model size (no worse than scanning training set)
- Compared to Bayes net structure learning:
  - Comparable data likelihood
  - Better query likelihood
  - Much faster & more reliable inference
- Problem: Curse of dimensionality
Outline

- Motivation
- Probabilistic models
- Standard tractable models
- **The sum-product theorem**
- Bounded-inference graphical models
- Feature trees
- Sum-product networks
- Tractable Markov logic
- Other tractable models
Efficiently Summable Functions

A function is efficiently summable iff its sum over any subset of its scope can be computed in time polynomial in the cardinality of the subset.
The Sum-Product Theorem

If a function is:
- A sum of efficiently summable functions with the same scope, or
- A product of efficiently summable functions with disjoint scopes,
Then it is also efficiently summable.
Corollary

Every low-treewidth distribution is efficiently summable, but not every efficiently summable distribution has low treewidth.
Compactly Representable Probability Distributions

Graphical Models

Sum-Product Models

Standard Tractable Models
Compactly Representable Probability Distributions

Graphical Models

Sum-Product Models

Linear-time exact inference
Outline

- Motivation
- Probabilistic models
- Standard tractable models
- The sum-product theorem
  - Bounded-inference graphical models
  - Feature trees
  - Sum-product networks
  - Tractable Markov logic
- Symmetric models
- Other tractable models
Arithmetic Circuits
[Darwiche, JACM, 2003]

- Inference consists of sums and products
- Can be represented as an arithmetic circuit
- Complexity of inference = Size of circuit
Arithmetic Circuit

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$P(X)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.4</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.3</td>
</tr>
</tbody>
</table>

- Rooted DAG of sums and products
- Leaves are indicator variables
- Computes marginals in linear time
- Graphical models can be compiled into ACs
Learning Bounded-Inference Graphical Models [Lowd & D., UAI-08]

- Use standard Bayes net structure learner (with context-specific independence)
- **Key idea:** Instead of using *representation complexity* as regularizer:

\[
\text{score}(M,T) = \log P(T|M) - k_p \ n_p(M) \\
(\text{log-likelihood} - \#\text{parameters})
\]

Use *inference complexity*:

\[
\text{score}(M,T) = \log P(T|M) - k_c \ n_c(M) \\
(\text{log-likelihood} - \text{circuit size})
\]
Learning Bounded-Inference Graphical Models (contd.)

- Incrementally compile circuit as structure added (splits in decision trees)
- Compared to Bayes nets w/ Gibbs sampling:
  - Comparable data likelihood
  - Better query likelihood
  - Much faster & more reliable inference
- Large treewidth (10’s – 100’s)
Outline

- Motivation
- Probabilistic models
- Standard tractable models
- The sum-product theorem
  - Bounded-inference graphical models
  - Feature trees
  - Sum-product networks
  - Tractable Markov logic
- Symmetric models
- Other tractable models
Feature Trees
[Gogate, Webb & D., NIPS-10]

- Thin junction tree learners work by repeatedly finding a subset of variables $A$ such that
  \[ P(B, C | A) \approx P(B | A) \, P(C | A) \]
  where $A, B, C$ is a partition of the variables
- LEM algorithm: Instead find a feature $F$ s.t.
  \[ P(B, C | F) \approx P(B | F) \, P(C | F) \]
  and recurse on variables and instances
- Result is a tree of features
A Feature Tree

$X_1 \land \neg X_2$

$X_3 \land X_4$

$\neg X_5 \land X_6$

$X_3 \land \neg X_5$

$X_4 \land X_6$
Feature Trees (contd.)

- High treewidth because of context-specific independence
- More flexible than decision tree CPDs
- PAC-learning guarantees
- Outperforms thin junction trees and other algorithms for learning Markov networks
- More generally: Feature graphs
Outline

- Motivation
- Probabilistic models
- Standard tractable models
- The sum-product theorem
  - Bounded-inference graphical models
  - Feature trees
  - **Sum-product networks**
  - Tractable Markov logic
- Symmetric models
- Other tractable models
A Univariate Distribution Is an SPN
A Product of SPNs over Disjoint Variables Is an SPN
A Weighted Sum of SPNs over the Same Variables Is an SPN

Sums out a mixture variable
Recurse Freely . . .
All Marginals Are Computable in Linear Time

\[ P(X=0) = 0.26 \]
All MAP States Are Computable in Linear Time

$$\max_y P(X=0, Y=y) = 0.12$$
All MAP States Are Computable in Linear Time

\[
\max_y P(X=0, Y=y) = 0.12
\]
What Does an SPN Mean?

Products = Features
Sums = Clusters
Special Cases of SPNs

- Hierarchical mixture models
- Thin junction trees (e.g.: hidden Markov models)
- Non-recursive probabilistic context-free grammars
- Etc.
Discriminative SPNs

[Gens & D., NIPS-12; Best Student Paper Award]
Discriminative Training

\[ \nabla \log P(y|x) = \nabla \log \frac{P(y, x)}{P(x)} = \]

\[ \nabla \log \sum_h P(y, h, x) - \nabla \log \sum_{y', h} P(y', h, x) \]

Tractable!
Backpropagation

For each sum child $j$:

$$\frac{\partial S}{\partial S_j} \leftarrow \frac{\partial S}{\partial S_j} + w_{n,j} \frac{\partial S}{\partial S_n}$$

$$\frac{\partial S}{\partial w_{n,j}} \leftarrow S_j \frac{\partial S}{\partial S_n}$$
Backpropagation

For each product child $j$:

$$\frac{\partial S}{\partial S_j} \leftarrow \frac{\partial S}{\partial S_j} + \frac{\partial S}{\partial S_k} \prod_{k \in Ch(n) \setminus \{j\}} S_k$$
Problem: Gradient Diffusion
Solution: Hard Inference

Soft Inference
(Marginals)

Hard Inference
(MAP States)
Hard Gradient

\[ \nabla \log \tilde{P}(y|\mathbf{x}) = \nabla \log \frac{\tilde{P}(y, \mathbf{x})}{\tilde{P}(\mathbf{x})} = \]

\[ \nabla \log \left( \max_{\mathbf{h}} P(y, \mathbf{h}, \mathbf{x}) \right) - \nabla \log \left( \max_{\mathbf{y}', \mathbf{h}} P(y', \mathbf{h}, \mathbf{x}) \right) \]
Hard Gradient

\[ \nabla \log \tilde{P}(y|x) = \nabla \log \frac{\tilde{P}(y, x)}{\tilde{P}(x)} = \]

\[ \nabla \log \begin{pmatrix} f_1(X) \\ f_2(X) \end{pmatrix} \]

Number with correct label - Number with model guess

\[ \frac{\partial}{\partial w_i} \log \tilde{P}(y|x) = \frac{\Delta c_i}{w_i} \]
Empirical Evaluation: Object Recognition

CIFAR-10
32x32 pixels
50k training exs.
10k test exs.

STL-10
96x96 pixels
5k training exs.
8k test exs.
100k unlabeled exs.
Feature Extraction

32x32 → 27x27xK → GxGxK

Triangle encoding

K-means

Max-pooling

[Coates et al., AISTATS 2011]
Architecture

Classes

Parts

Mixture

Location

$e^{x_{ij} \cdot f_{cpt}}$

WxWxK

GxGxK
CIFAR-10 Results

Accuracy vs. Dictionary Size (K)

- SPN
- Pooling
- SVM
- Autoencoder
- RBM

4x4xK
STL-10 Results

- 1-layer Vector Quantization: 54.9%
- 1-layer Sparse Coding: 59.0%
- 3-layer Learned Receptive Field: 60.1%
- Discriminative SPN: 62.3%

without unlabeled data
Generative Weight Learning

[Poon & D., UAI-11; Best Paper Award]

- Model joint distribution of all variables
- Algorithm: **Online hard EM**
- Sum node maintains counts for each child
- For each example
  - Find MAP instantiation with current weights
  - Increment count for each chosen child
  - Renormalize to set new weights
- Repeat until convergence
Empirical Evaluation: Image Completion

- Datasets: Caltech-101 and Olivetti
- Compared with DBNs, DBMs, PCA and NN
- SPNs reduce MSE by ~1/3
- Orders of magnitude faster than DBNs, DBMs
Architecture
Example Completions

Original

SPN

DBM

DBN

PCA

Nearest Neighbor
## Weight Learning: Summary

<table>
<thead>
<tr>
<th>Update</th>
<th>Soft Inference (Marginals)</th>
<th>Hard Inference (MAP States)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generative EM</td>
<td>$\Delta w_i \propto w_i \frac{\partial S}{\partial S_k}$</td>
<td>$\Delta w_i = c_i$</td>
</tr>
<tr>
<td>Generative Gradient</td>
<td>$\Delta w_i = \eta \frac{\partial S}{\partial S_k} S_i$</td>
<td>$\Delta w_i = \eta \frac{c_i}{w_i}$</td>
</tr>
<tr>
<td>Discriminative Gradient</td>
<td>$\Delta w_i = \eta \left( \frac{S_i}{S} \frac{\partial S}{\partial S_k} \bigg</td>
<td>_{\text{true label}} - \frac{S_i}{S} \frac{\partial S}{\partial S_k} \bigg</td>
</tr>
</tbody>
</table>
Structure Learning
[Gens & D., ICML-13; no best paper award]
Empirical Evaluation

- 20 varied real-world datasets
  - 10s-1000s of variables
  - 1000s-100,000s of samples
- Compared with state-of-the-art Bayesian network and Markov random field learners
  - Likelihood: typically comparable
  - Query accuracy: much higher
  - Inference: orders of magnitude faster
Outline

- Motivation
- Probabilistic models
- Standard tractable models
- The sum-product theorem
  - Bounded-inference graphical models
  - Feature trees
  - Sum-product networks
  - Tractable Markov logic
- Symmetric models
- Other tractable models
Tractable Markov Logic
[D. & Webb, AAAI-12]

- Tractable representation for statistical relational learning
- Three types of weighted rules and facts
  - **Subclass:** \(\text{Is} (\text{Family}, \text{SocialUnit})\)
    \(\text{Is} (\text{Smiths}, \text{Family})\)
  - **Subpart:** \(\text{Has} (\text{Family}, \text{Adult}, 2)\)
    \(\text{Has} (\text{Smiths}, \text{Anna}, \text{Adult1})\)
  - **Relation:** \(\text{Parent} (\text{Family}, \text{Adult}, \text{Child})\)
    \(\text{Married} (\text{Anna}, \text{Bob})\)
Restrictions

- One top class
- One top object (all others are subparts)
- Relations must be among subparts of some object
- Subclasses are mutually exclusive
- Objects do not share subparts
TML Semantics

\[ Z(X, C) = \left( \sum_{S} e^{w_s} Z(X, S) \right) \times \left( \prod_{P} Z(P(X), C_{P})^{n_{P}} \right) \times \left( \prod_{R} (1 + e^{w_{R}}) \right) \]

\[ Z(KB) = Z(\text{TopObject}, \text{TopClass}) \]
**Tractability**

**Theorem:** The partition function of every TML knowledge base can be computed in time and space polynomial in the size of the knowledge base.

\[
\text{Time} = \text{Space} = O(\#\text{Rules} \times \#\text{Objects})
\]
Why TML Is Tractable

KB structure is isomorphic to Z computation:
• Parts = Products
• Classes = Sums
Why TML Is Tractable

KB structure is isomorphic to \( Z \) computation:

- **Parts = Products**
- **Classes = Sums**
Why TML Is Tractable

KB structure is isomorphic to $\mathbb{Z}$ computation:
- Parts = Products
- Classes = Sums
Expressiveness

The following can be compactly represented in TML:

- Junction trees
- Sum-product networks
- Probabilistic context-free grammars
- Probabilistic inheritance hierarchies
- Etc.
Learning Tractable MLNs

Alternate between:

- Dividing / aggregating the domain into subparts
- Inducing class hierarchies over similar subparts
Other Sum-Product Models

- Relational sum-product networks
- Tractable probabilistic knowledge bases
- Tractable probabilistic programs
- Etc.
What If This Is Not Enough?

Use variational inference, with the most expressive tractable representation available as the approximating family

[Lowd & D., NIPS-10]
Outline

- Motivation
- Probabilistic models
- Standard tractable models
- The sum-product theorem
- Bounded-inference graphical models
- Feature trees
- Sum-product networks
- Tractable Markov logic
- Other tractable models
Other Tractable Models

- Symmetry
  - Liftable models
  - Exchangeable models
- Submodularity
- Determinantal point processes
- Etc.
- Several papers at ICML-14
- Workshop on Thursday
Summary

- Intractable inference is the bane of learning
- Tractable models avoid it
- Standard ones are too limited
- We have powerful new tractable classes
  - Sum-product theorem
  - Symmetry
  - Etc.
Summary

Graphical Models

Tractability

Expressiveness
Summary

New Tractable Classes

Graphical Models

Tractability

Expressiveness