Learning Tractable Probabilistic Models

Pedro Domingos
Dept. Computer Science & Eng.
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

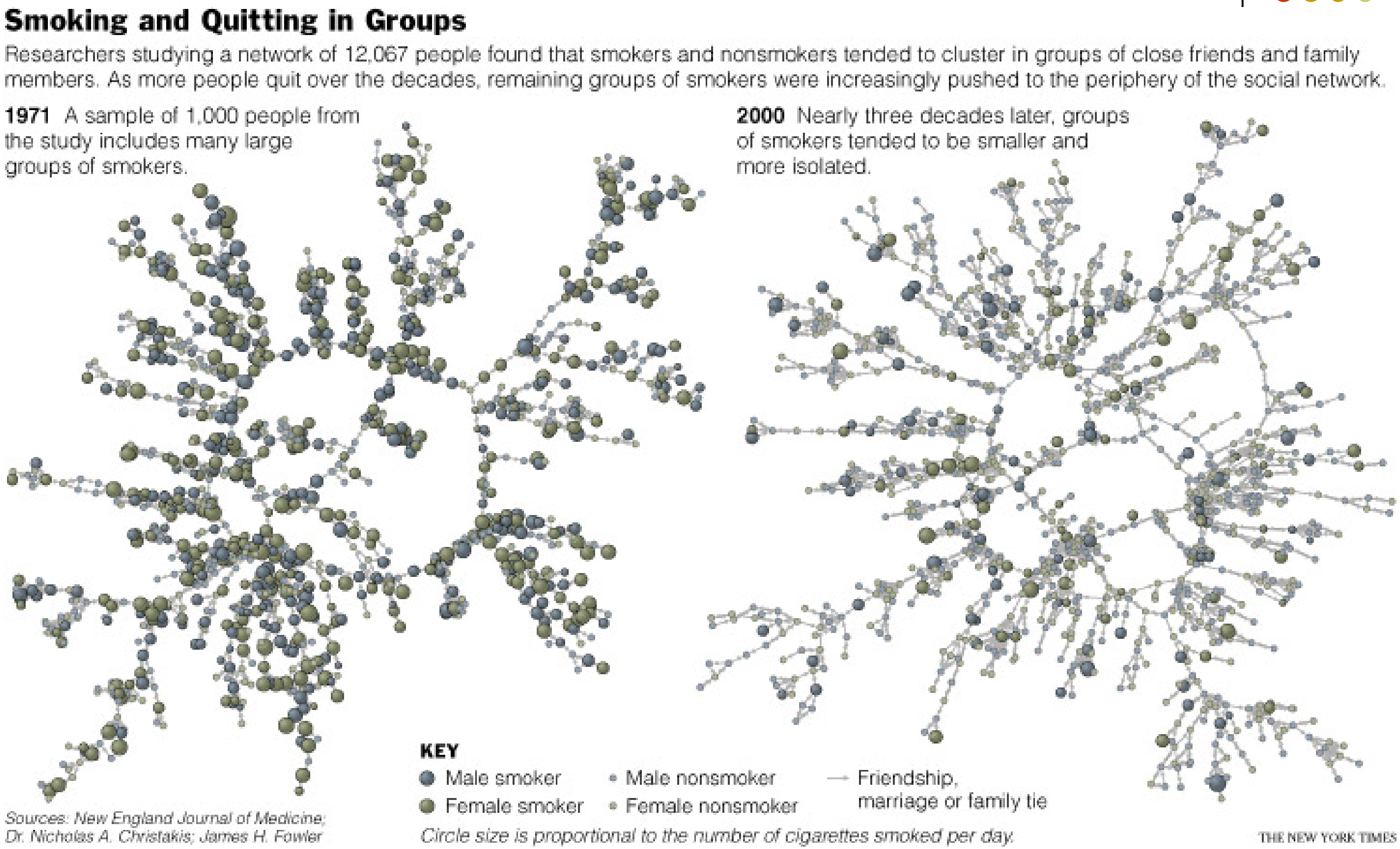
Inference is subroutine of:

- 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



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?

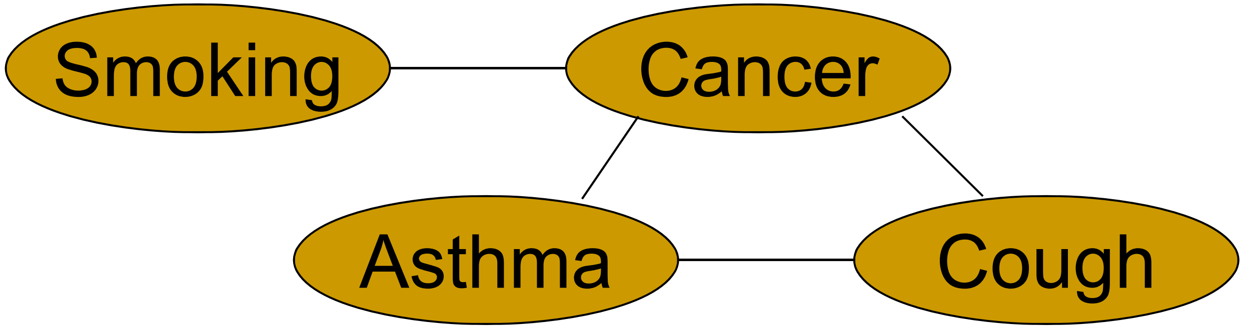
- Correctly handle uncertainty and noise
- Learn with missing data
- Jointly infer multiple variables
- Do inference in any direction
- It's the standard
- Powerful, consistent set of techniques

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



- Potential functions defined over cliques

$$P(x) = \frac{1}{Z} \prod_c \Phi_c(x_c)$$

$$Z = \sum_x \prod_c \Phi_c(x_c)$$

Smoking	Cancer	Φ(S,C)
False	False	4.5
False	True	4.5
True	False	2.7
True	True	4.5

Log-Linear Models

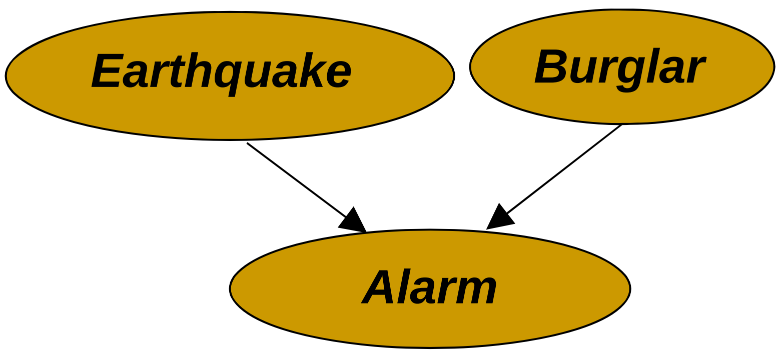
$$P(x) = \frac{1}{Z}\exp\left(\sum_i \underbrace{w_i}_{\text{Weight of Feature } i}\ \underbrace{f_i(x)}_{\text{Feature } i}\right)$$

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

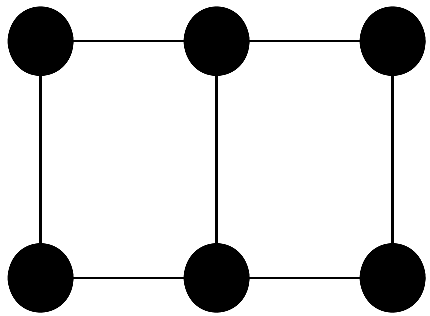
$$w_1 = 0.51$$

Representation and Inference

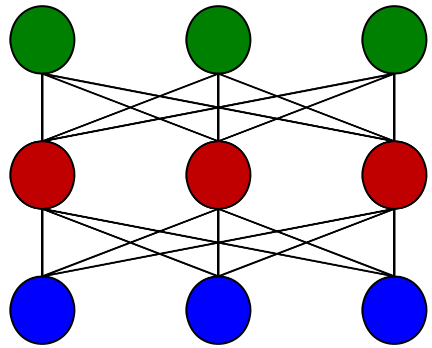
Bayesian Networks



Markov Networks



Deep Architectures



- Advantage: Compact representation
- Inference: ***P(Burglar | 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 → 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:

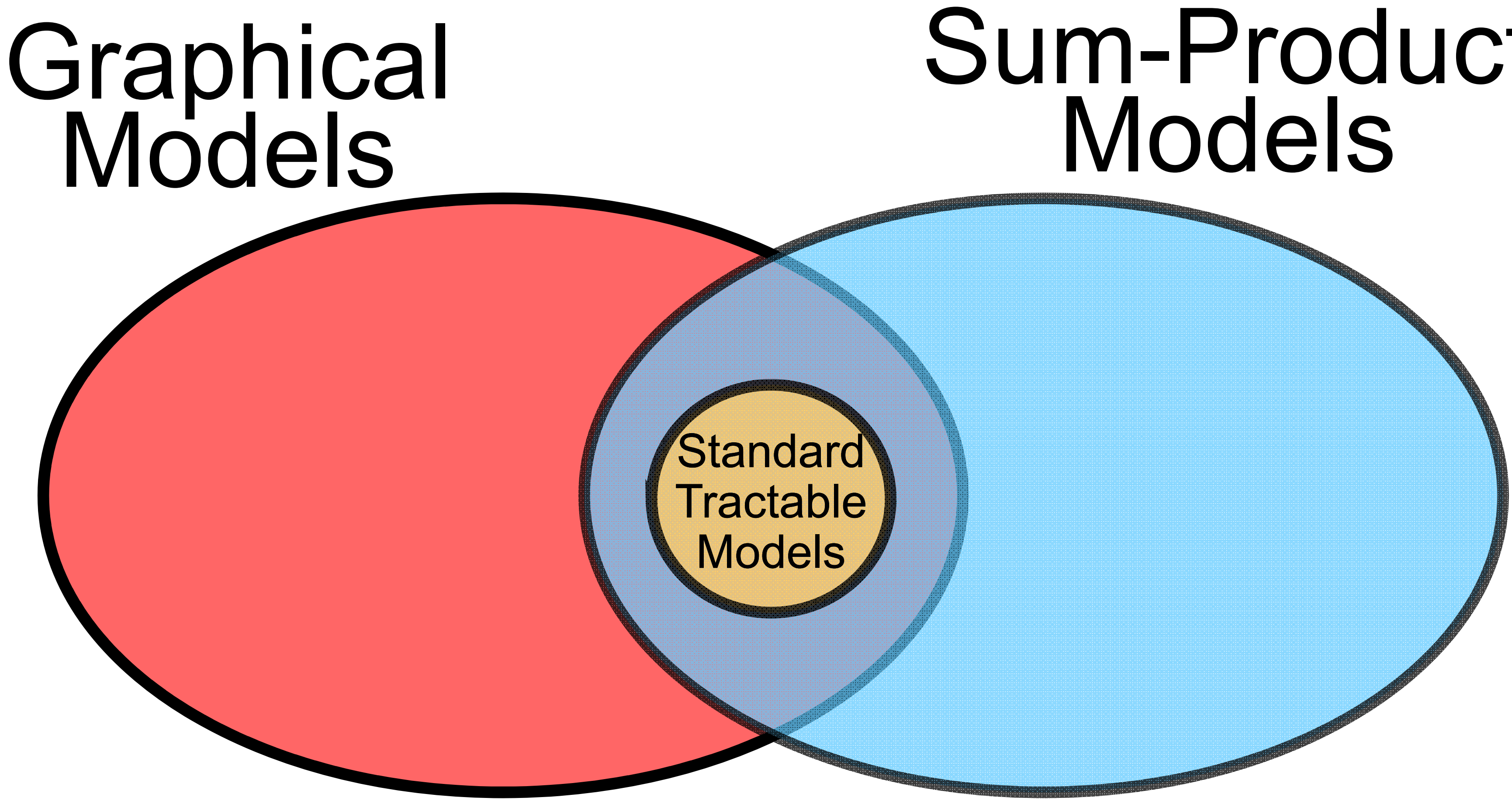
- A sum of efficiently summable functions with the same scope, or
- A product of efficiently summable functions with disjoint scopes,

Then it is also efficiently summable.

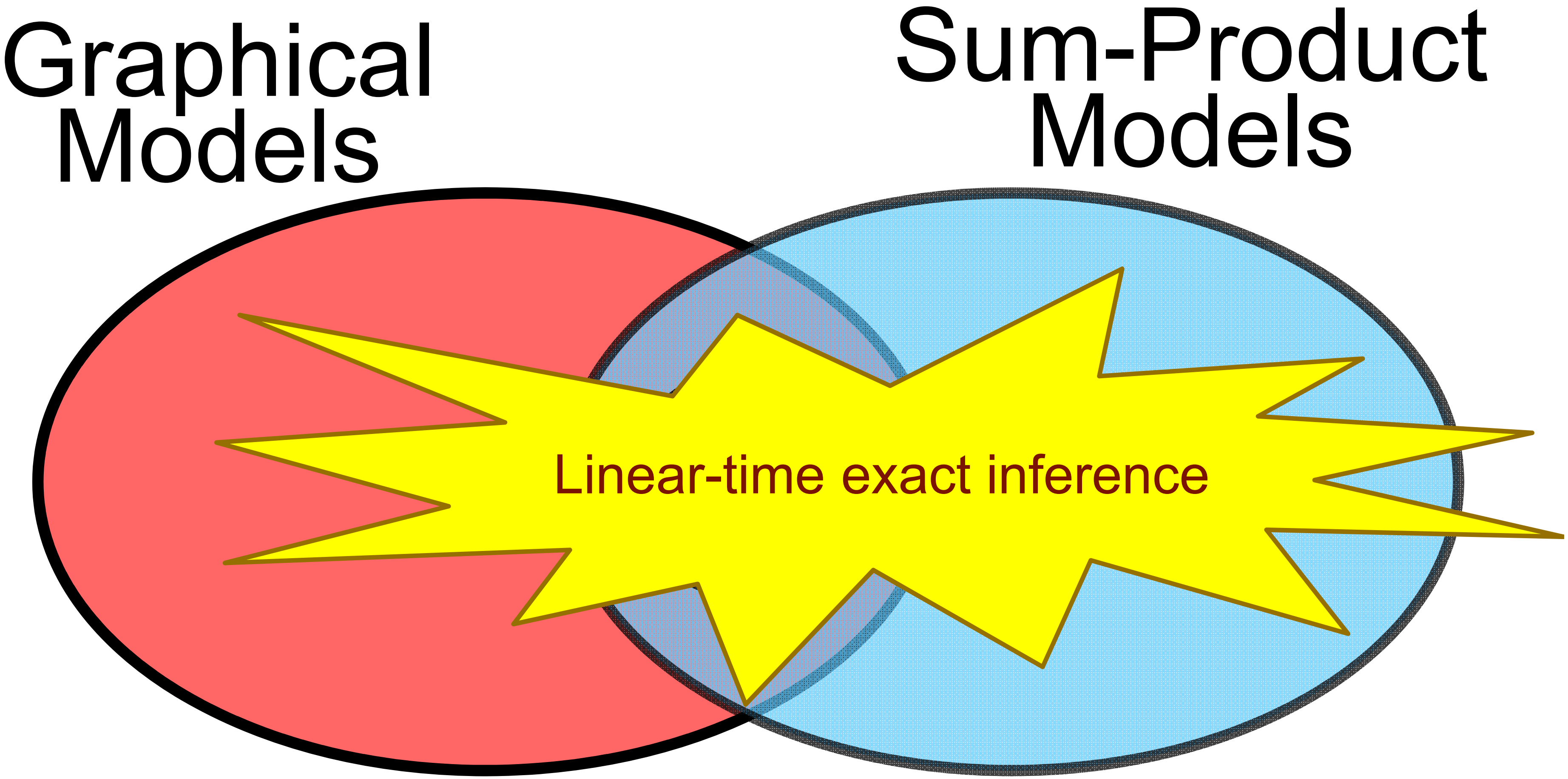
Corollary

Every low-treewidth distribution is efficiently summable, but not every efficiently summable distribution has low treewidth.

Compactly Representable Probability Distributions



Compactly Representable Probability Distributions



Outline

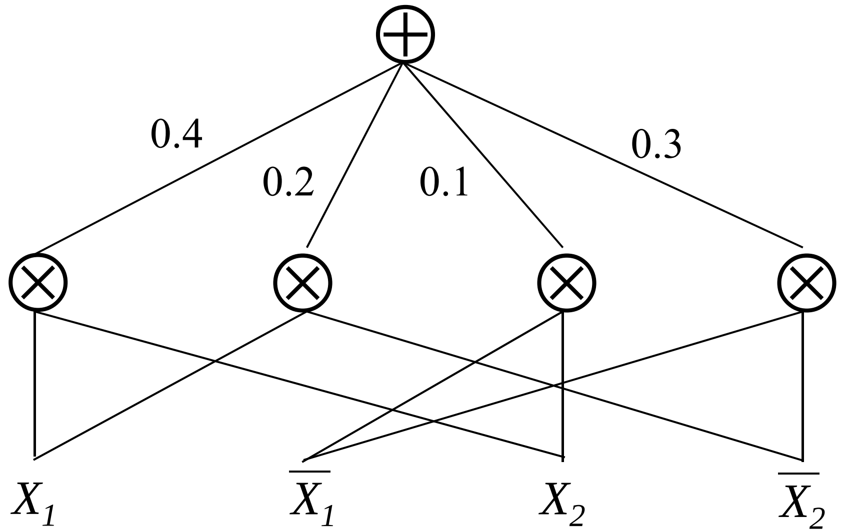
- Motivation
- Probabilistic models
- Standard tractable models
- The sum-product theorem
  - **Bounded-inference graphical models**
  - Feature trees
  - Sum-product networks
  - Tractable Markov logic
- Symmetric models
- Other tractable models

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

X_1	X_2	$P(X)$
1	1	0.4
1	0	0.2
0	1	0.1
0	0	0.3



- 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:

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

Use *inference complexity*:

$$\text{score}(M,T) = \log P(T|M) - k_c\, n_c(M)$$
(log-likelihood – 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)

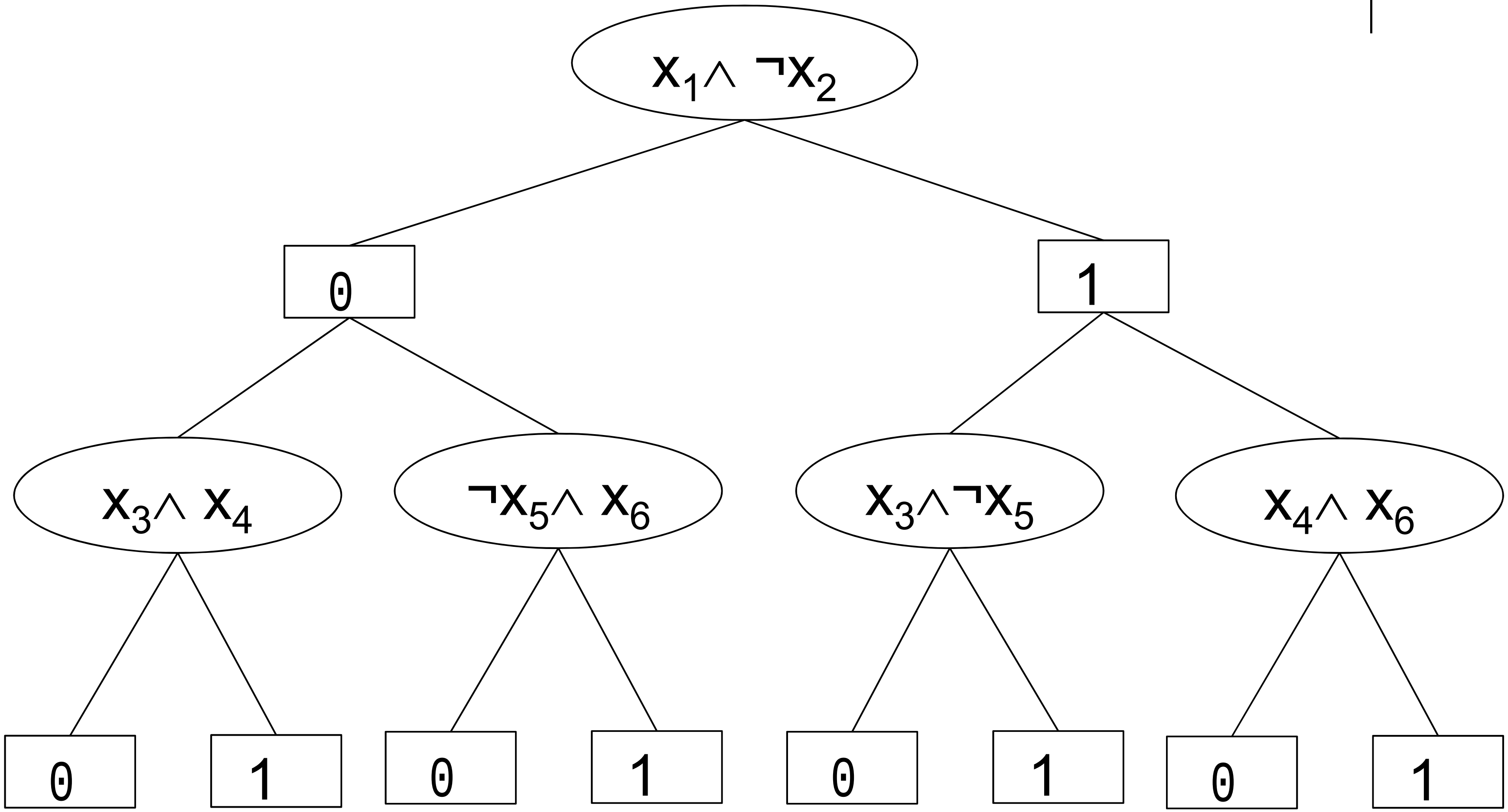
Outline

- 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



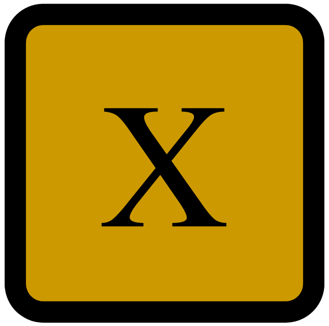
Feature Trees (contd.)

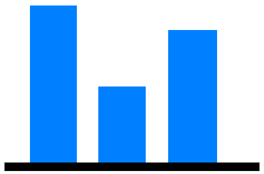
- High treewidth because of context-specific independence
- More flexible than decision tree CPDs
- PAC-learning guarantees
- Outperforms thin junction trees and other algorithms for learning Markov networks
- More generally: Feature graphs

Outline

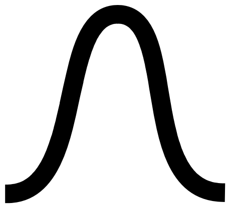
- Motivation
- Probabilistic models
- Standard tractable models
- The sum-product theorem
  - Bounded-inference graphical models
  - Feature trees
  - **Sum-product networks**
  - Tractable Markov logic
- Symmetric models
- Other tractable models

A Univariate Distribution Is an SPN

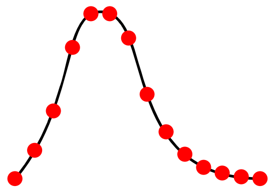




Multinomial



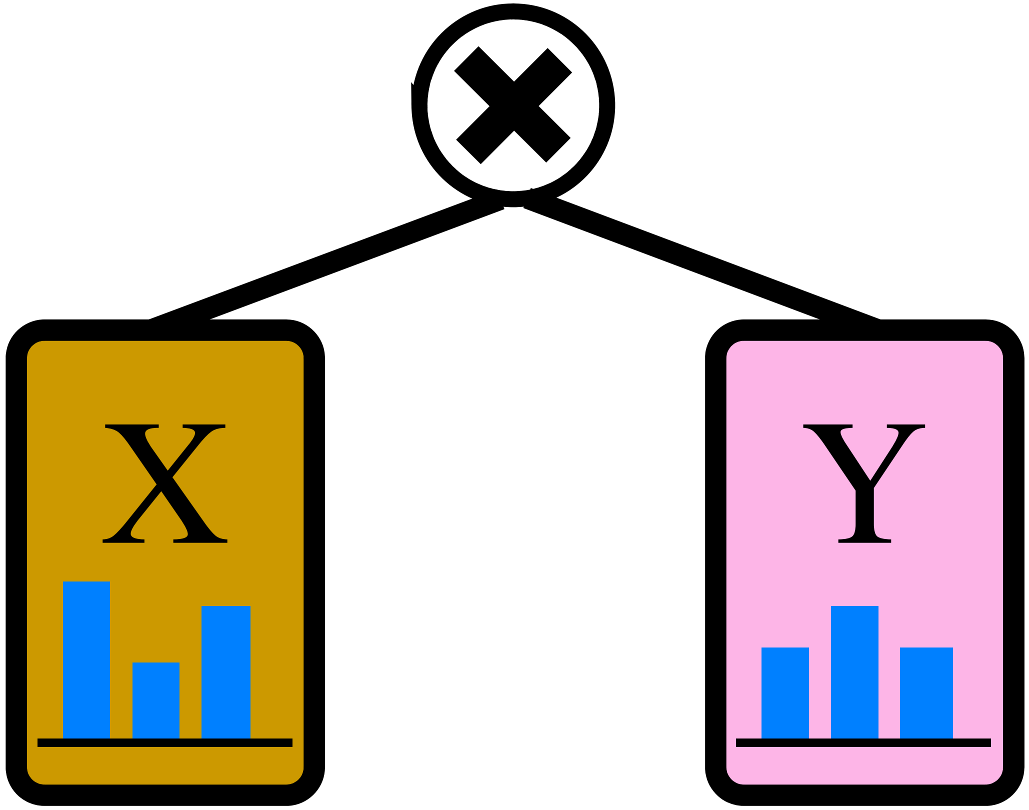
Gaussian



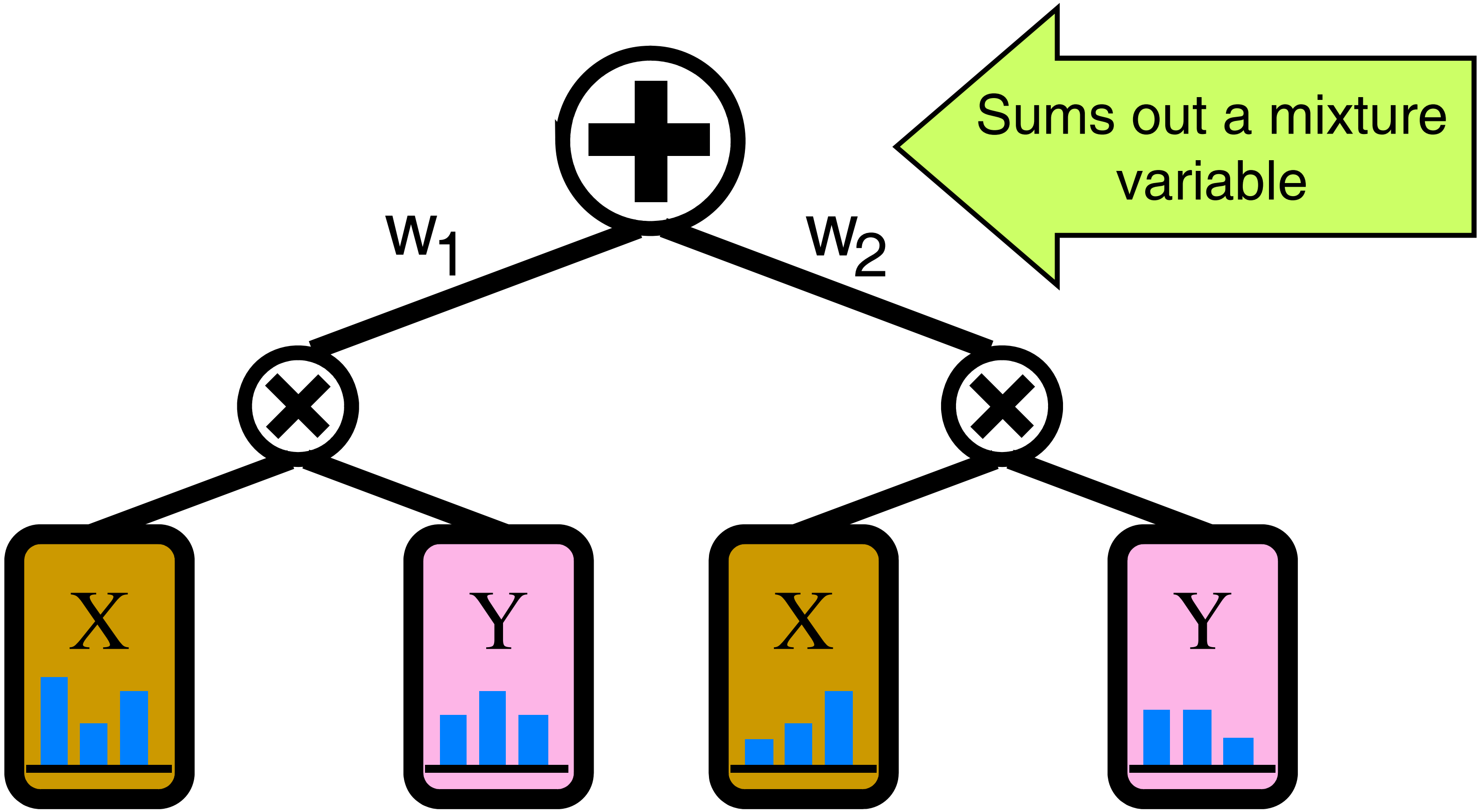
Poisson

. . .

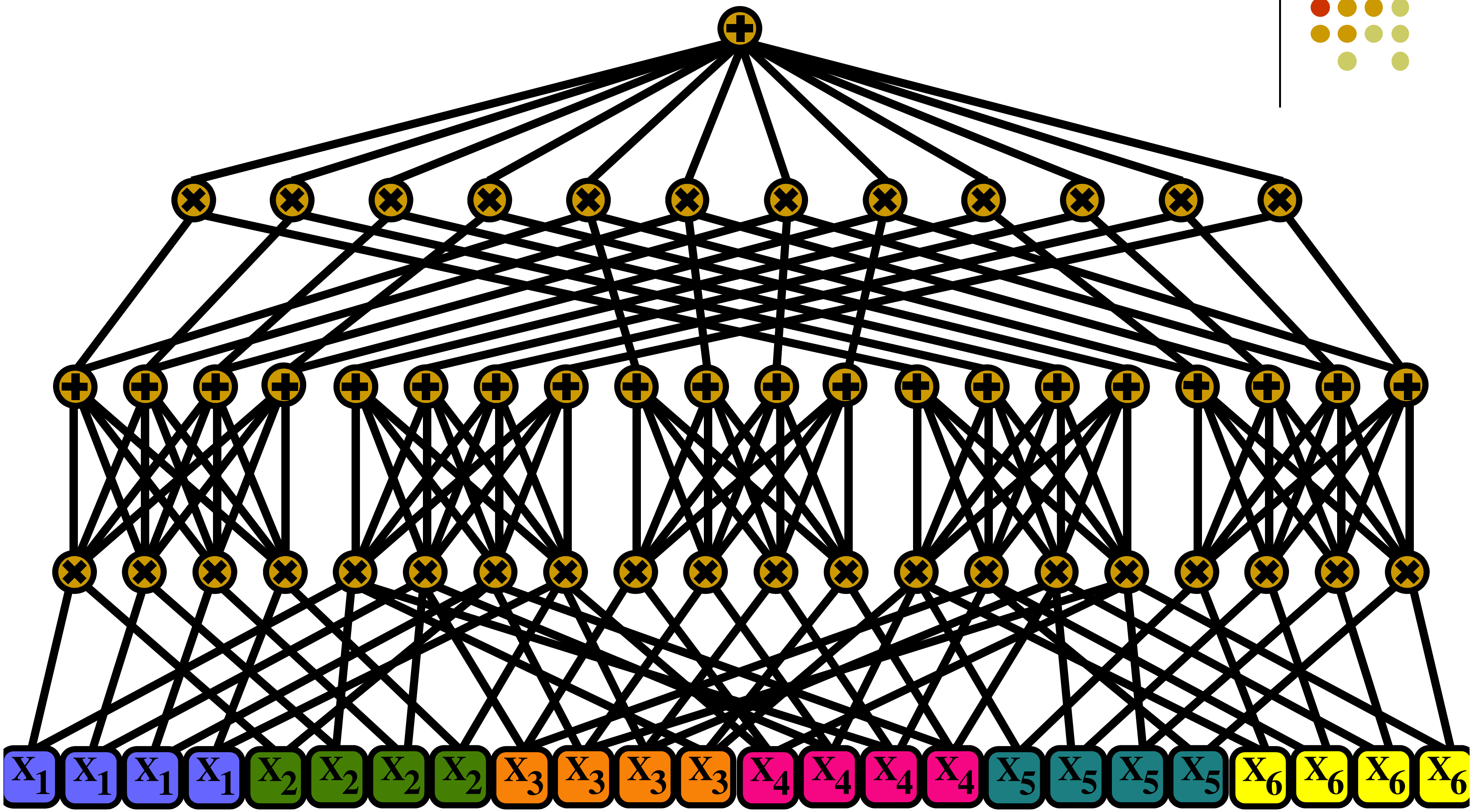
A Product of SPNs over Disjoint Variables Is an SPN



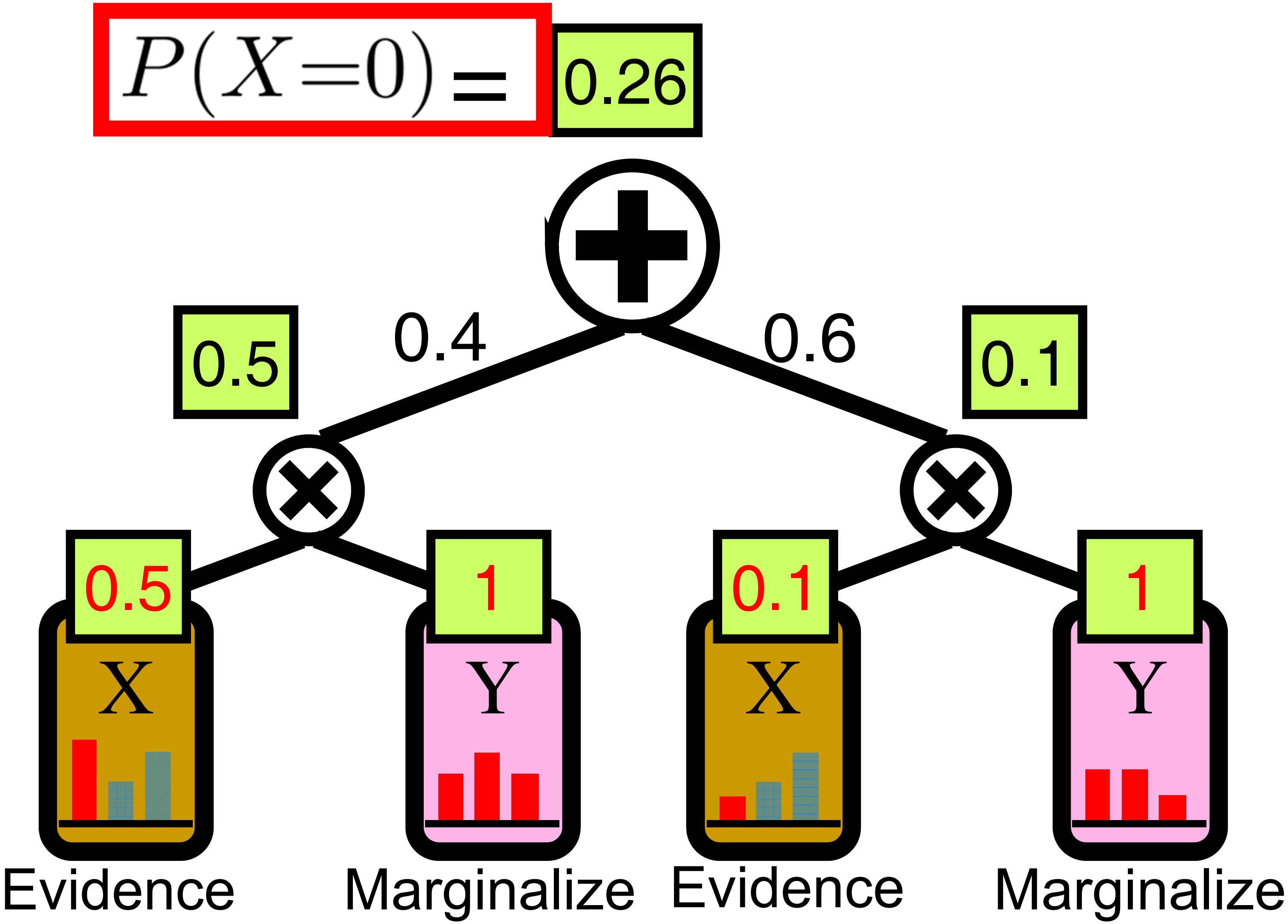
A Weighted Sum of SPNs over the Same Variables Is an SPN



Recurse Freely . . .

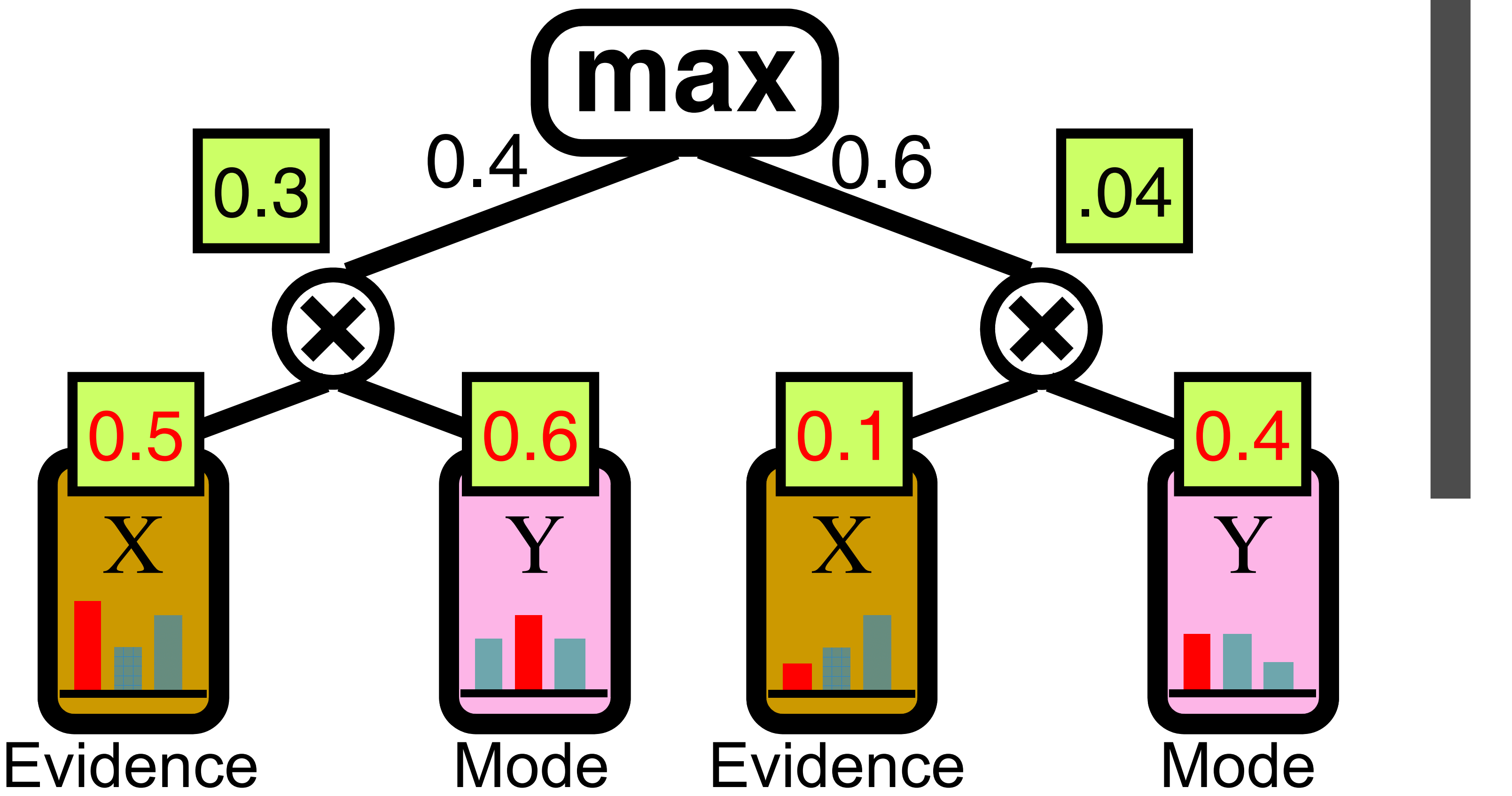


All Marginals Are Computable in Linear Time



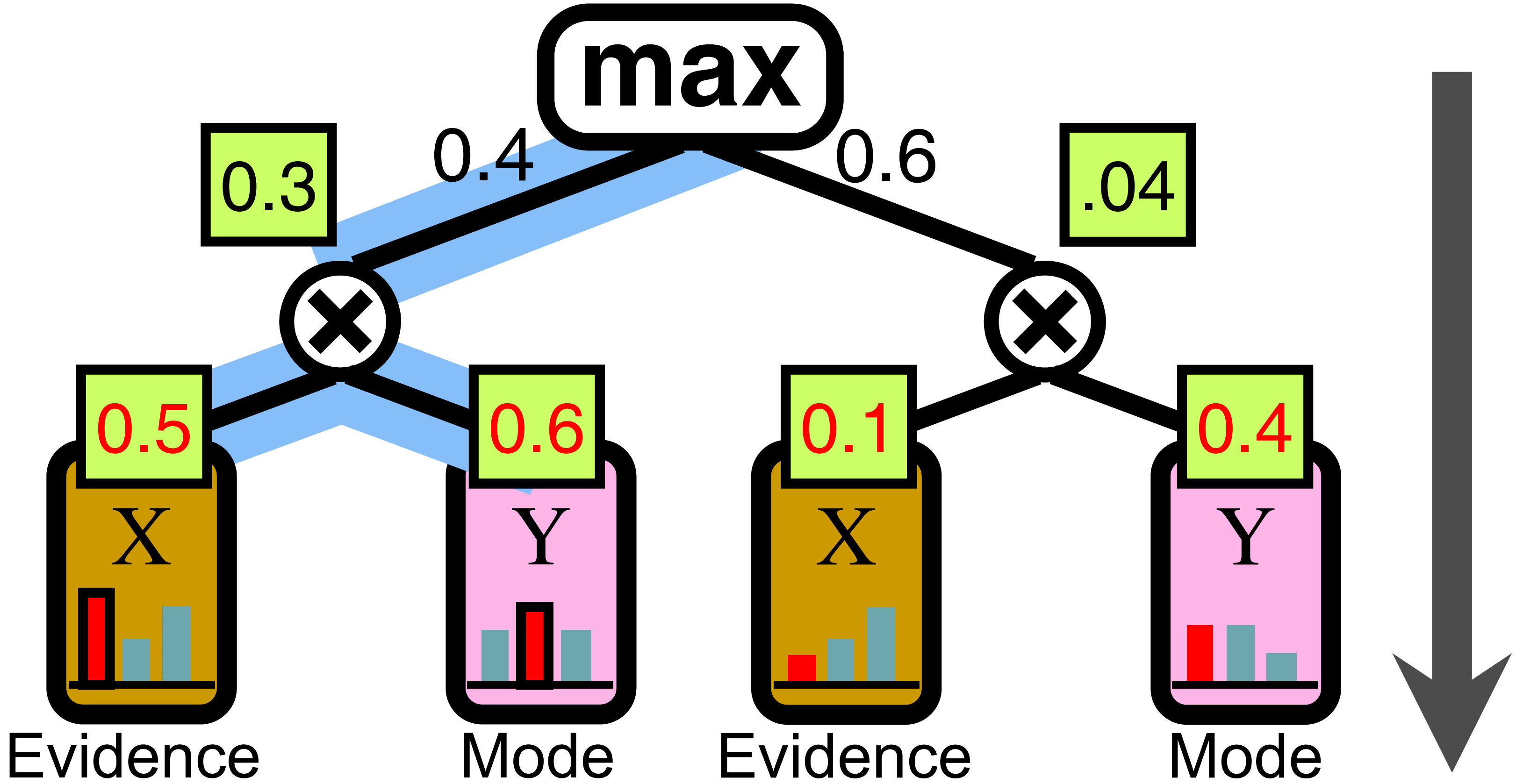
All MAP States Are Computable in Linear Time

$$\boxed{\max_{y} P(X{=}0, Y{=}y) =} \; 0.12$$

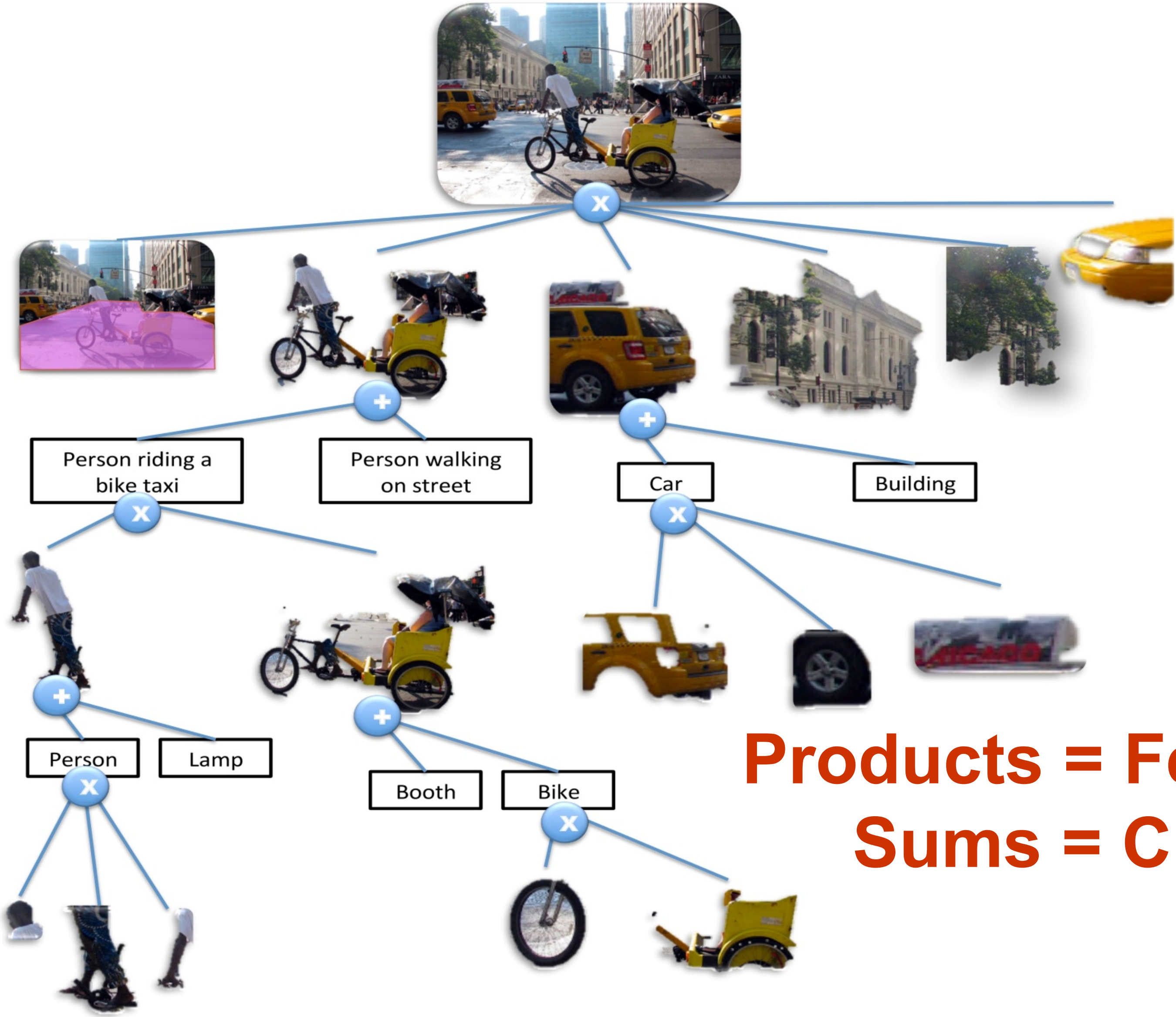


All MAP States Are Computable in Linear Time

$$\max_{y} P(X{=}0, Y{=}y) = \boxed{0.12}$$



What Does an SPN Mean?



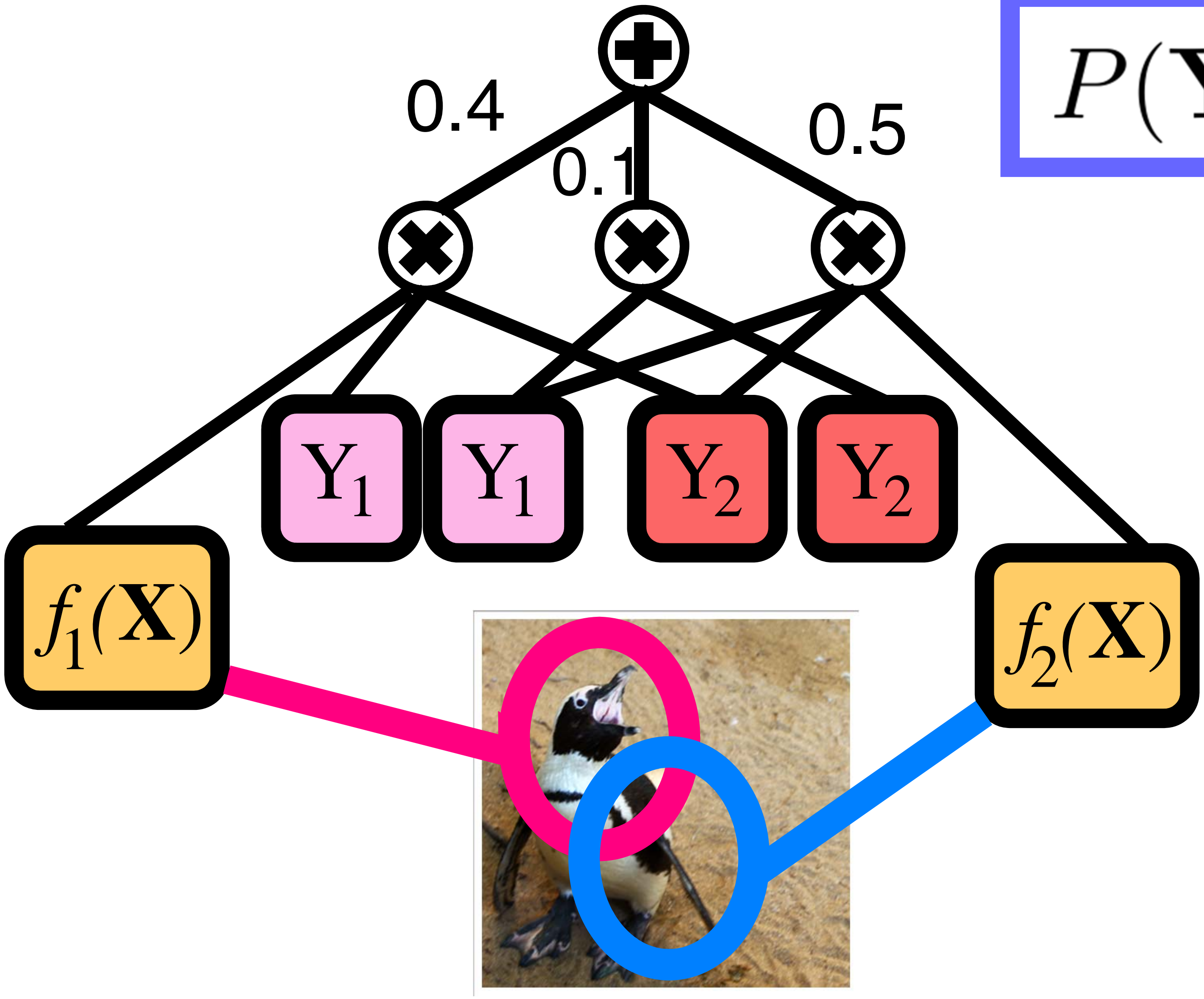
Products = Features
Sums = Clusters

Special Cases of SPNs

- Hierachical 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]



$$P(\mathbf{Y}|\mathbf{X})$$

$\mathbf{H}$ Hidden

$\mathbf{Y}$ Query

$f(\mathbf{X})$ Features (non-negative)

$\mathbf{X}$ Evidence

Discriminative Training

$$\nabla \log P(\mathbf{y}|\mathbf{x}) = \nabla \log \frac{P(\mathbf{y},\mathbf{x})}{P(\mathbf{x})} =$$

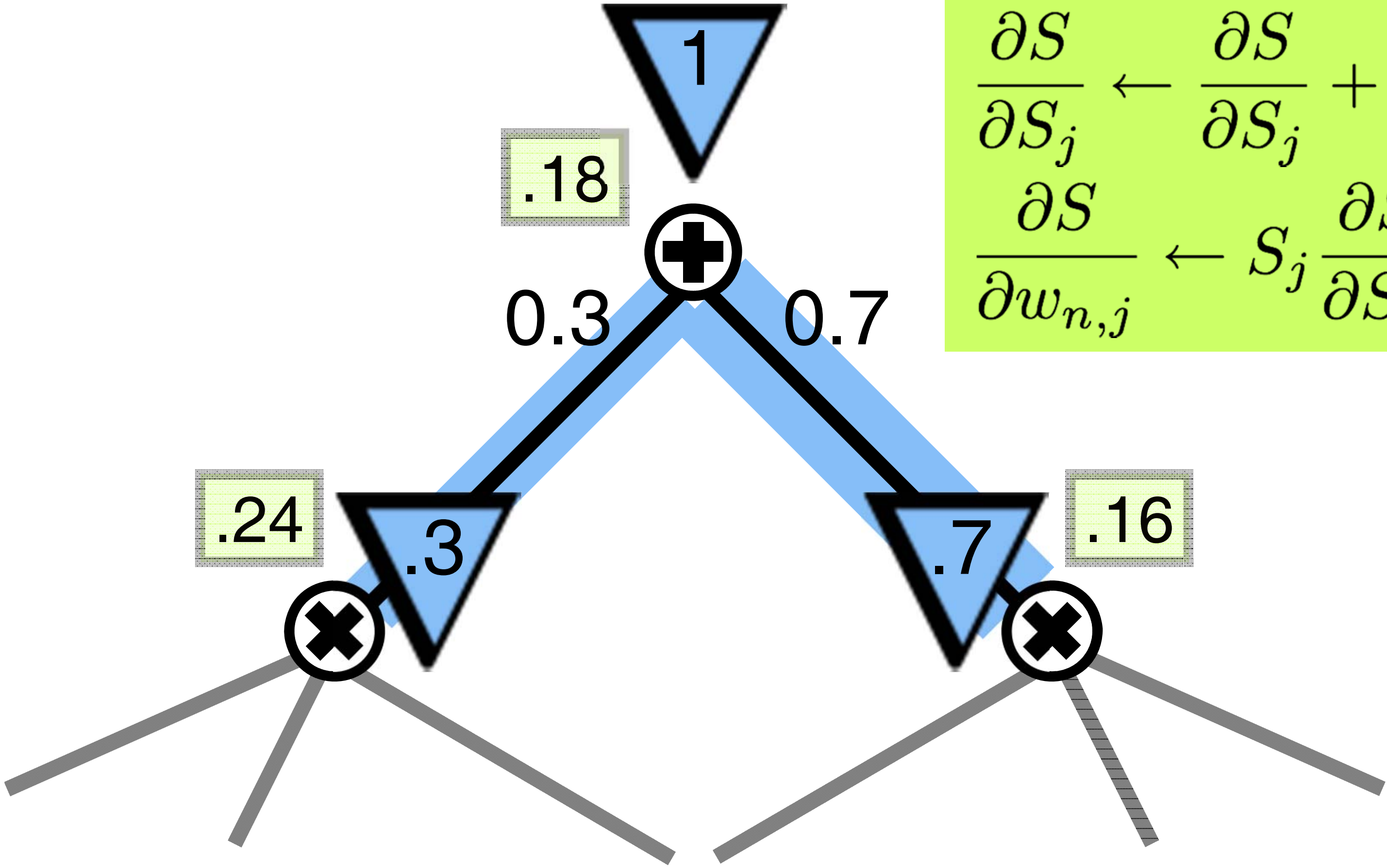
$$\nabla \log \sum_{\mathbf{h}} P(\mathbf{y},\mathbf{h},\mathbf{x}) - \nabla \log \sum_{\mathbf{y}',\mathbf{h}} P(\mathbf{y}',\mathbf{h},\mathbf{x})$$

Correct label

Tractable!

Best guess

Backpropagation

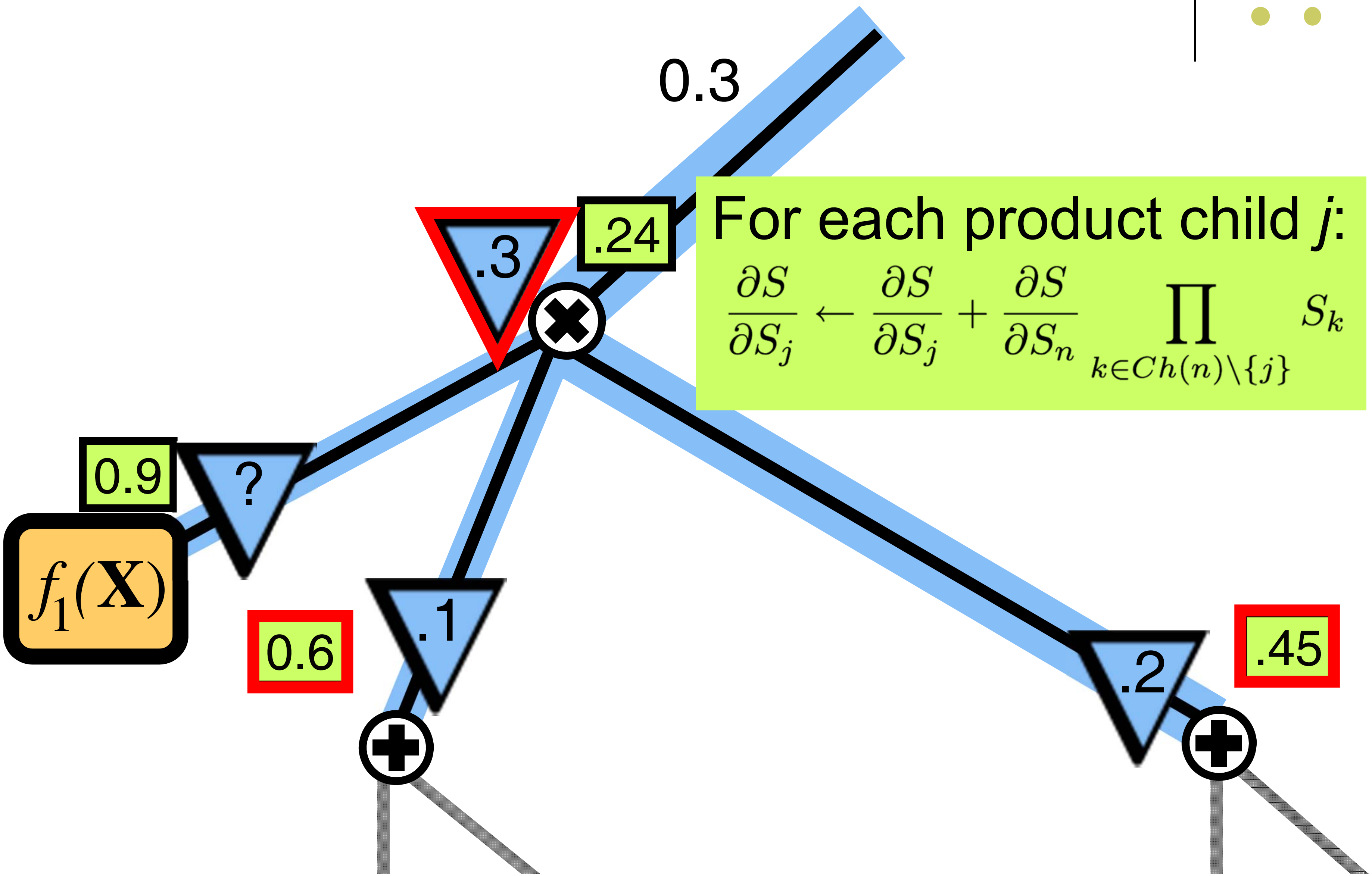


For each sum child j:

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

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

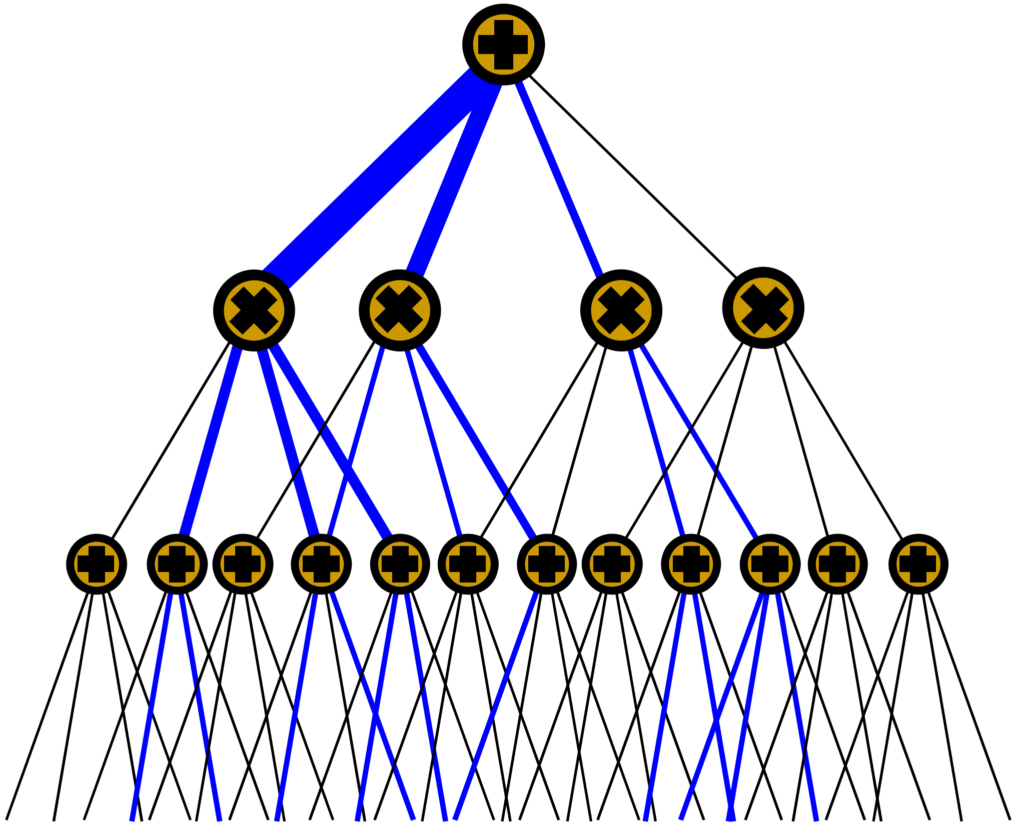
Backpropagation



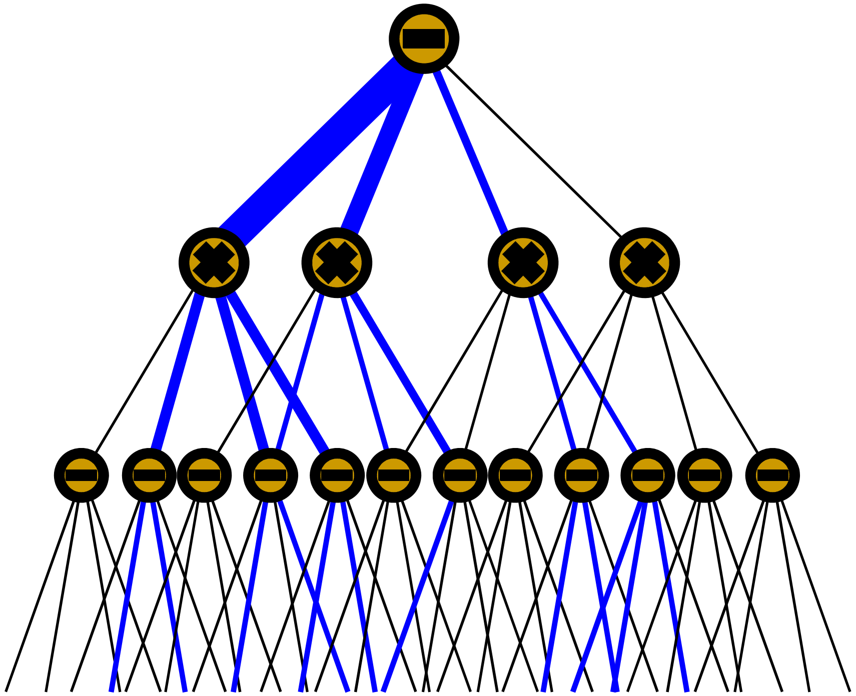
For each product child *j*:

$$\frac{\partial S}{\partial S_j} \leftarrow \frac{\partial S}{\partial S_j} + \frac{\partial S}{\partial S_n} \prod_{k \in Ch(n)\setminus\{j\}} S_k$$

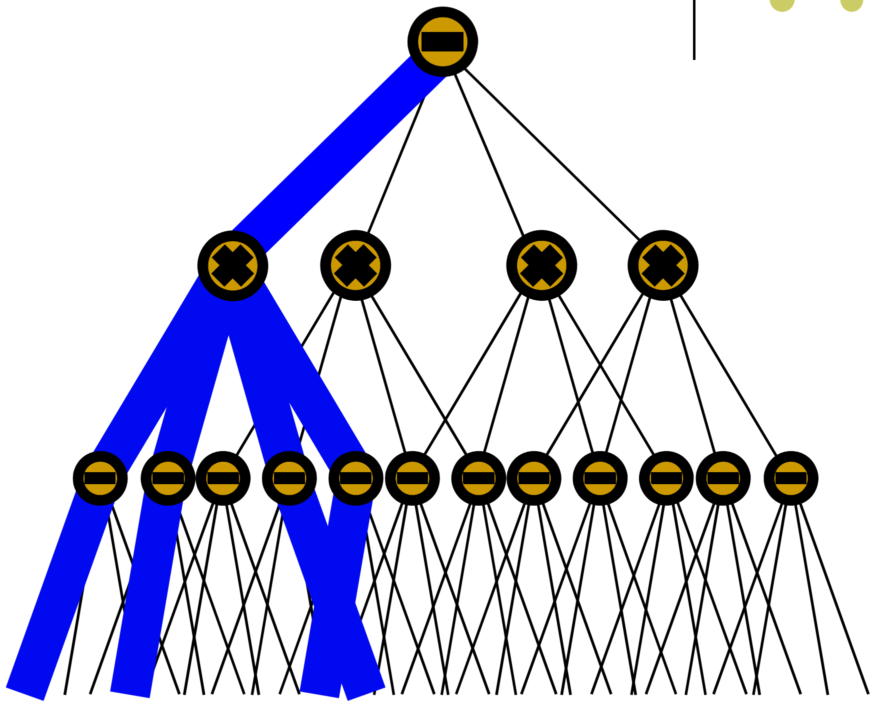
Problem: Gradient Diffusion



Solution: Hard Inference



Soft Inference
(Marginals)

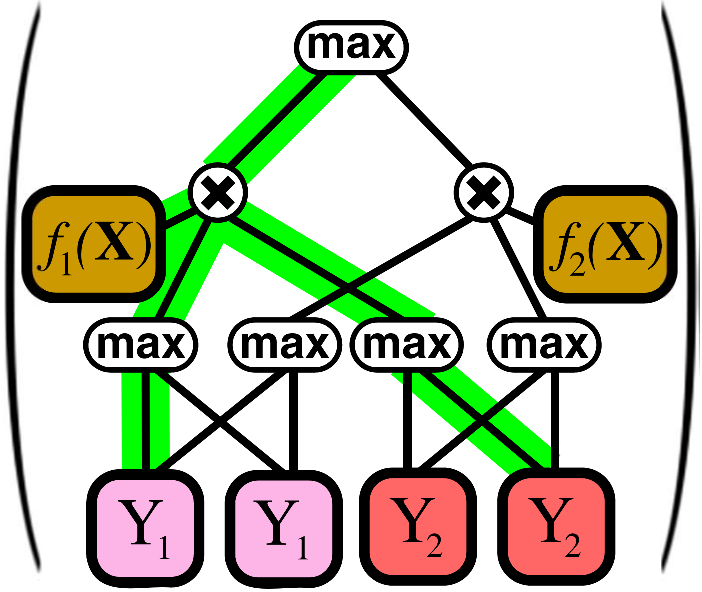


Hard Inference
(MAP States)

Hard Gradient

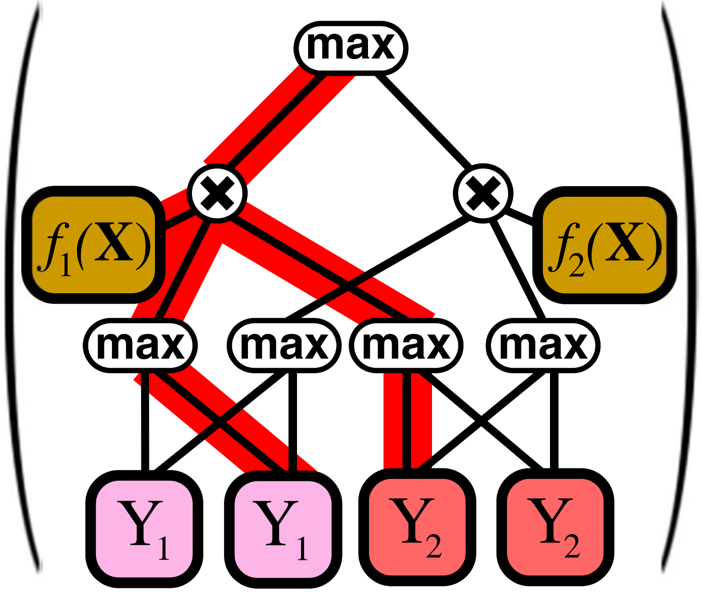
$$\nabla \log \tilde{P}(\mathbf{y}|\mathbf{x}) = \nabla \log \frac{\tilde{P}(\mathbf{y}, \mathbf{x})}{\tilde{P}(\mathbf{x})} =$$

$\nabla \log$



-

$\nabla \log$

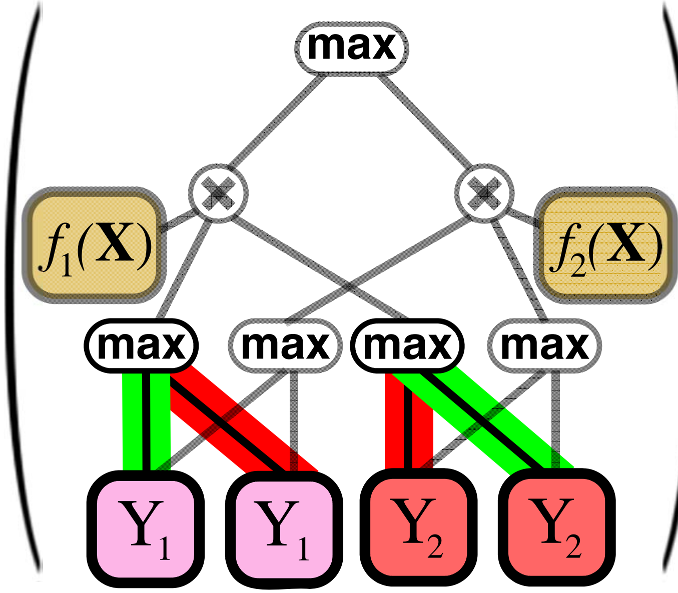


$$\max_{\mathbf{h}} P(\mathbf{y}, \mathbf{h}, \mathbf{x}) \qquad \max_{\mathbf{y}', \mathbf{h}} P(\mathbf{y}', \mathbf{h}, \mathbf{x})$$

Hard Gradient

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

$\nabla \log \Big($

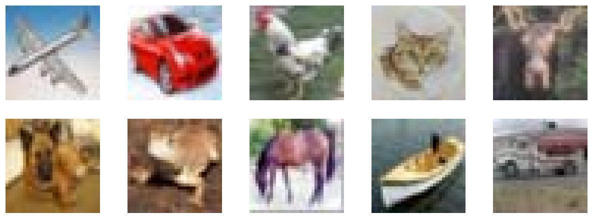


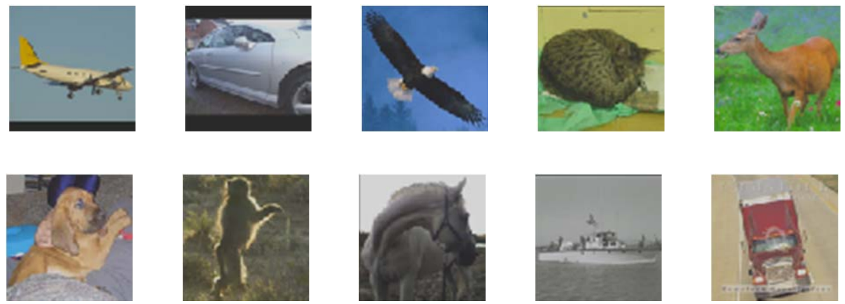
$\Big)$

Number with correct label – Number with model guess

$$\frac{\partial}{\partial w_i} \log \tilde{P}(\mathbf{y}|\mathbf{x}) = \frac{\Delta c_i}{w_i}$$

Empirical Evaluation: Object Recognition





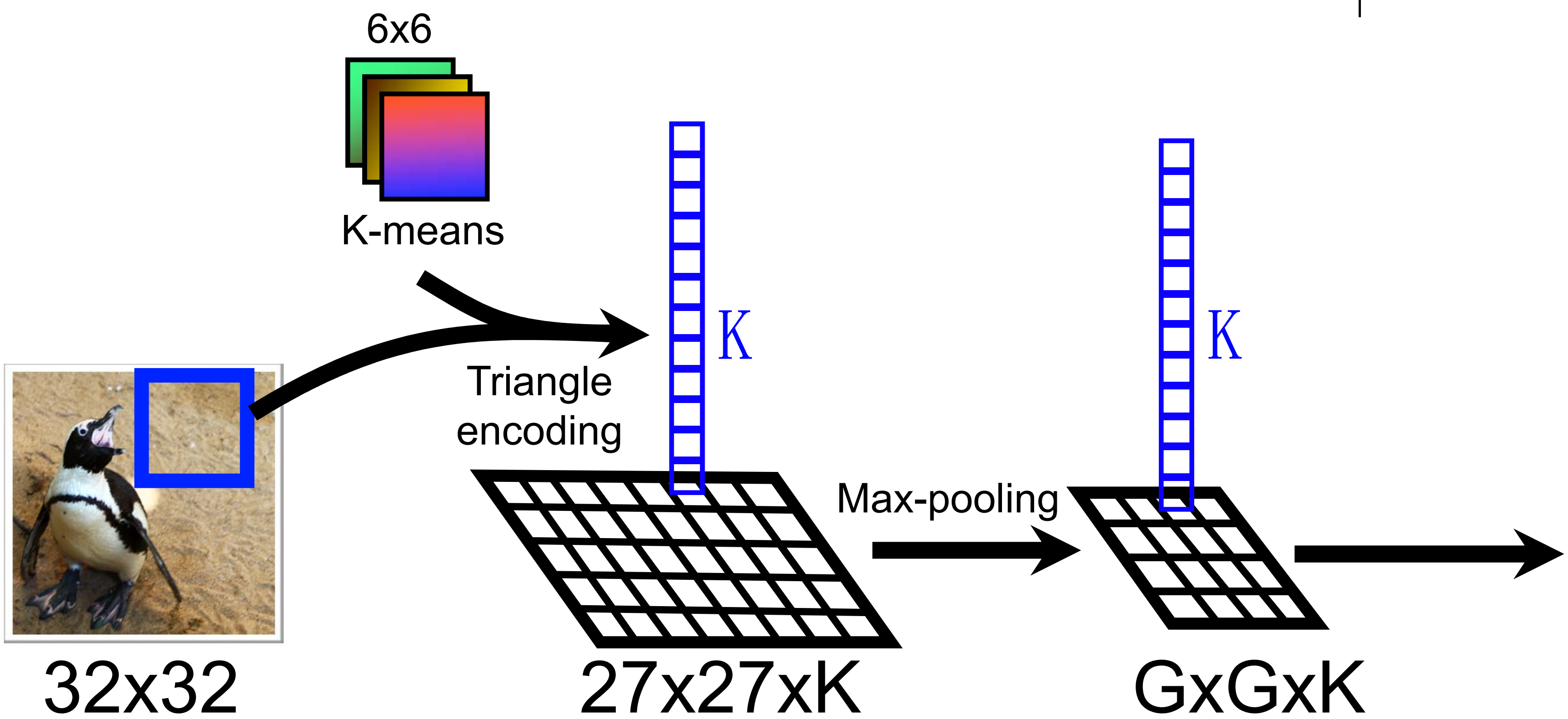
CIFAR-10

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

STL-10

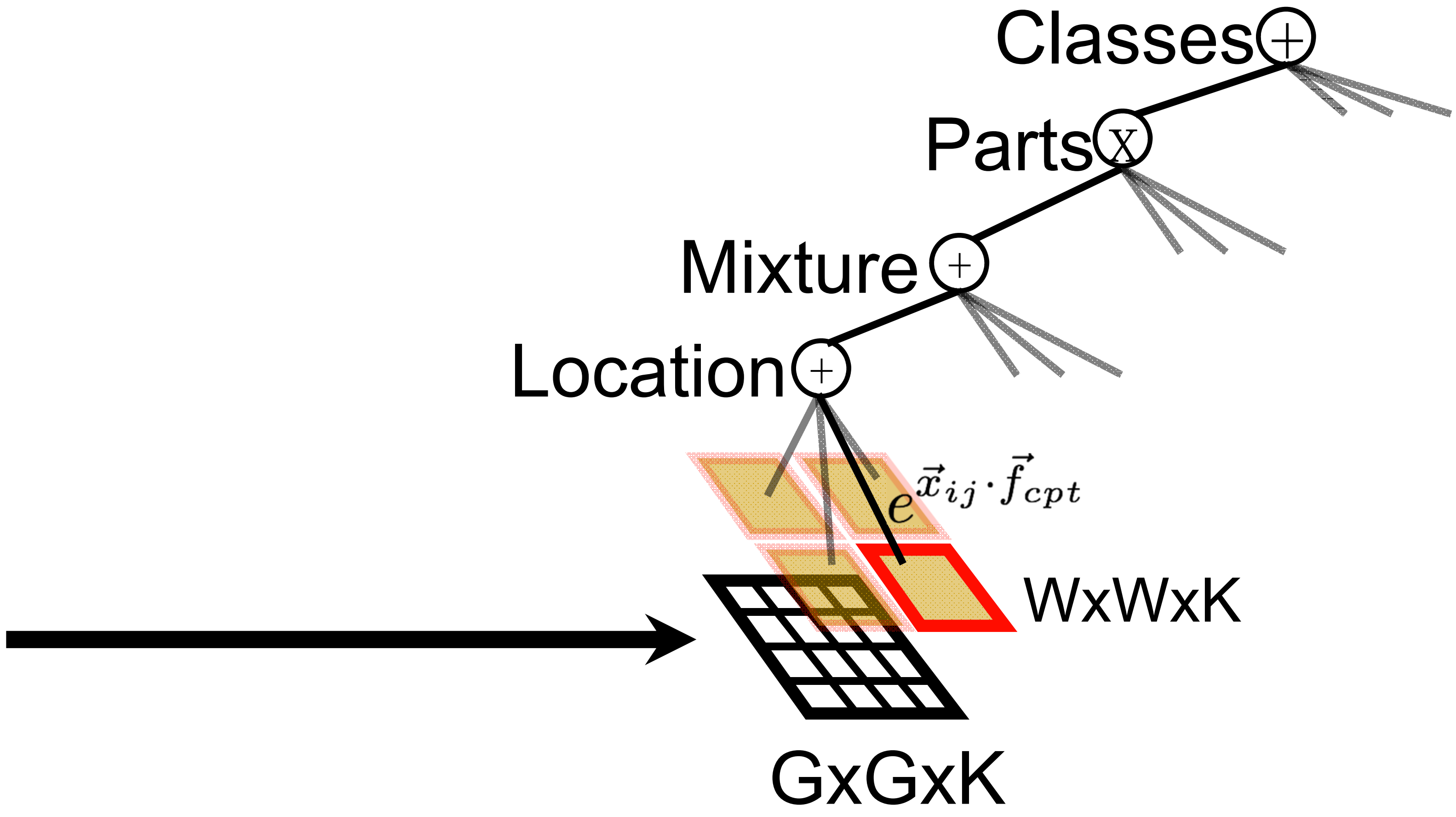
96x96 pixels
5k training exs.
8k test exs.
100k unlabeled exs.

Feature Extraction

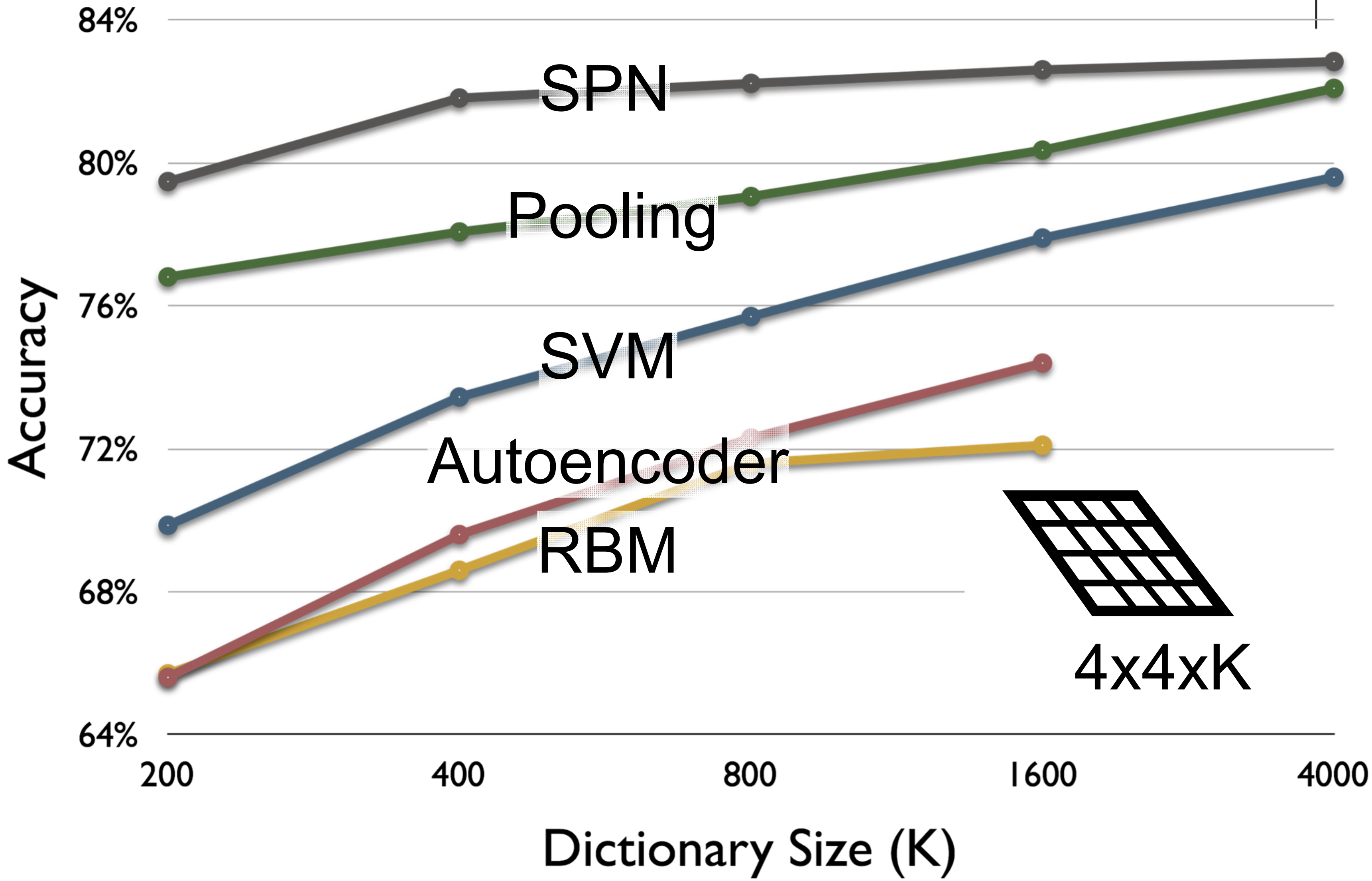


[Coates et al., AISTATS 2011]

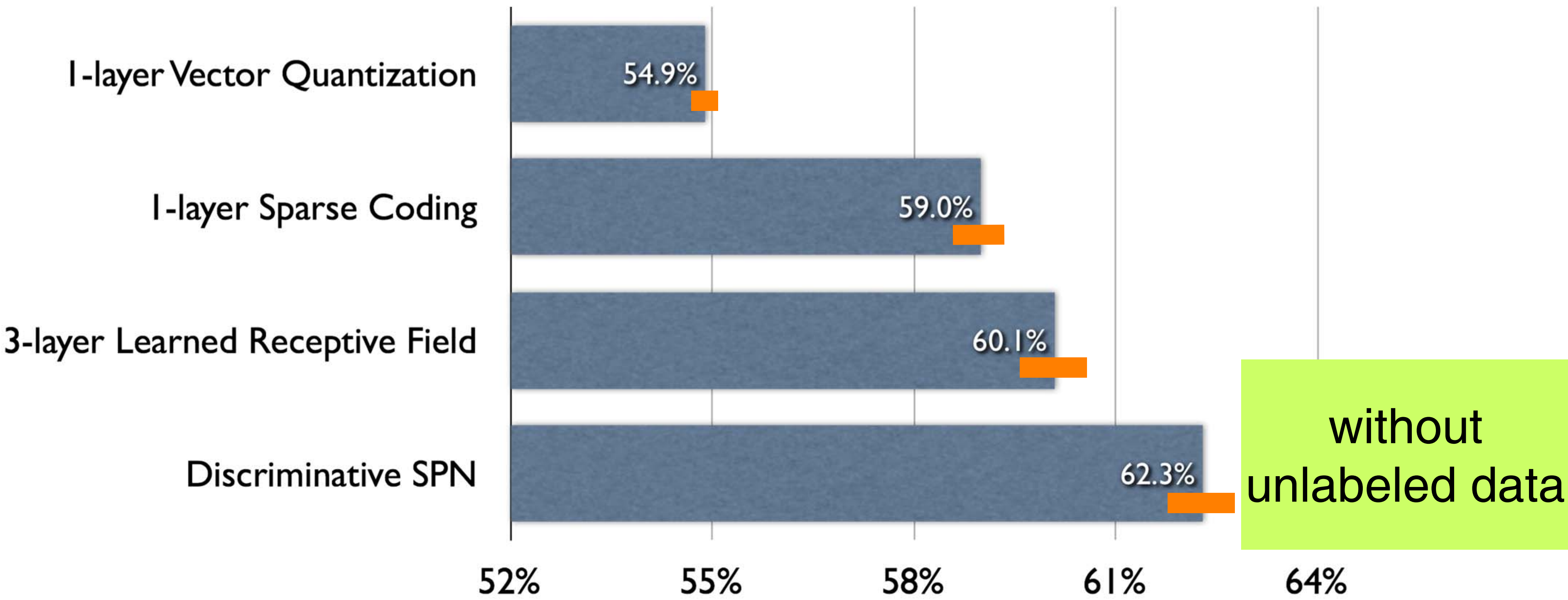
Architecture



CIFAR-10 Results



STL-10 Results



Generative Weight Learning

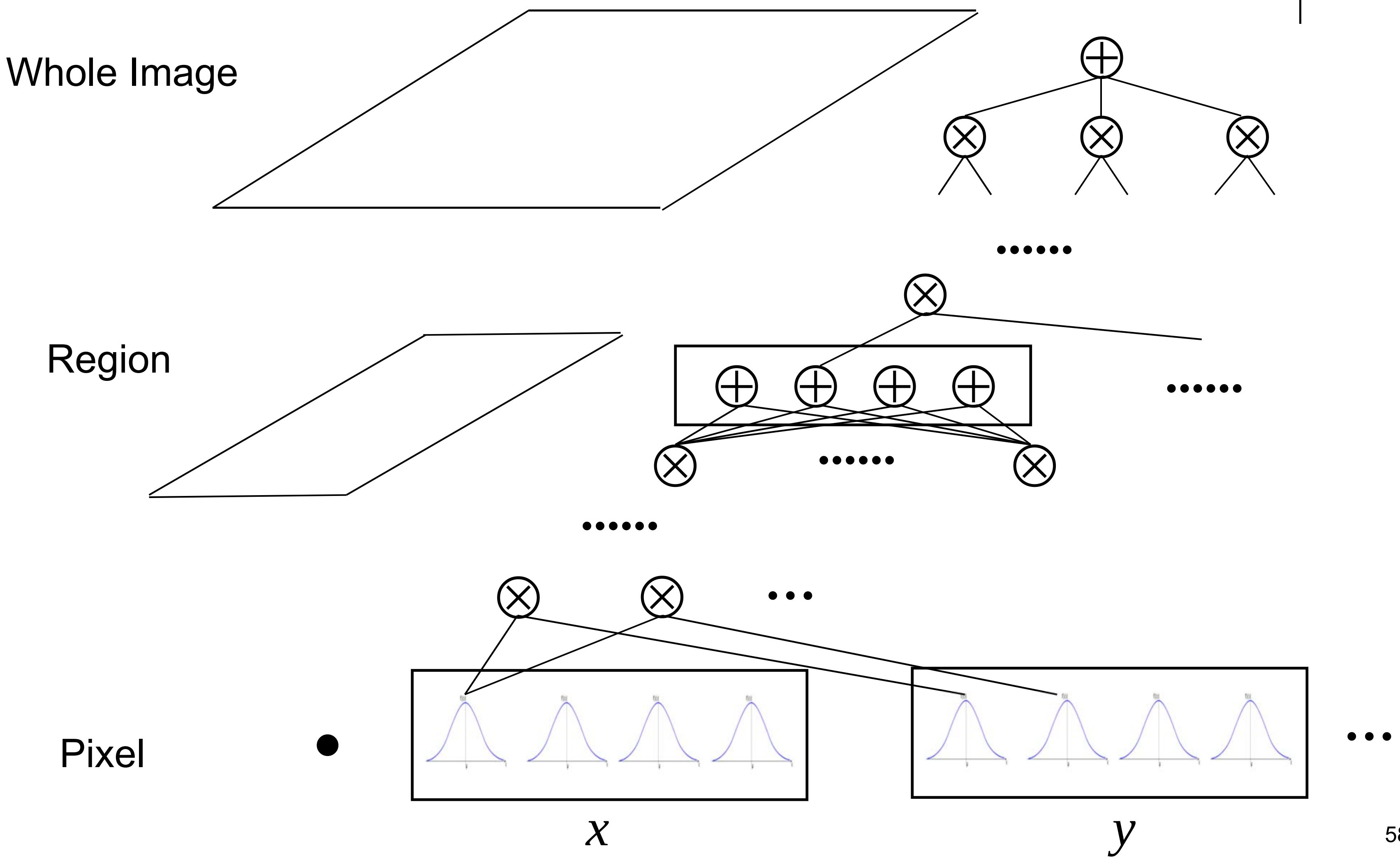
[Poon & D., UAI-11; Best Paper Award]

- 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

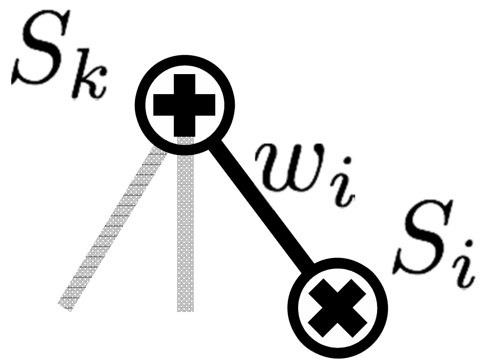
Architecture



Example Completions

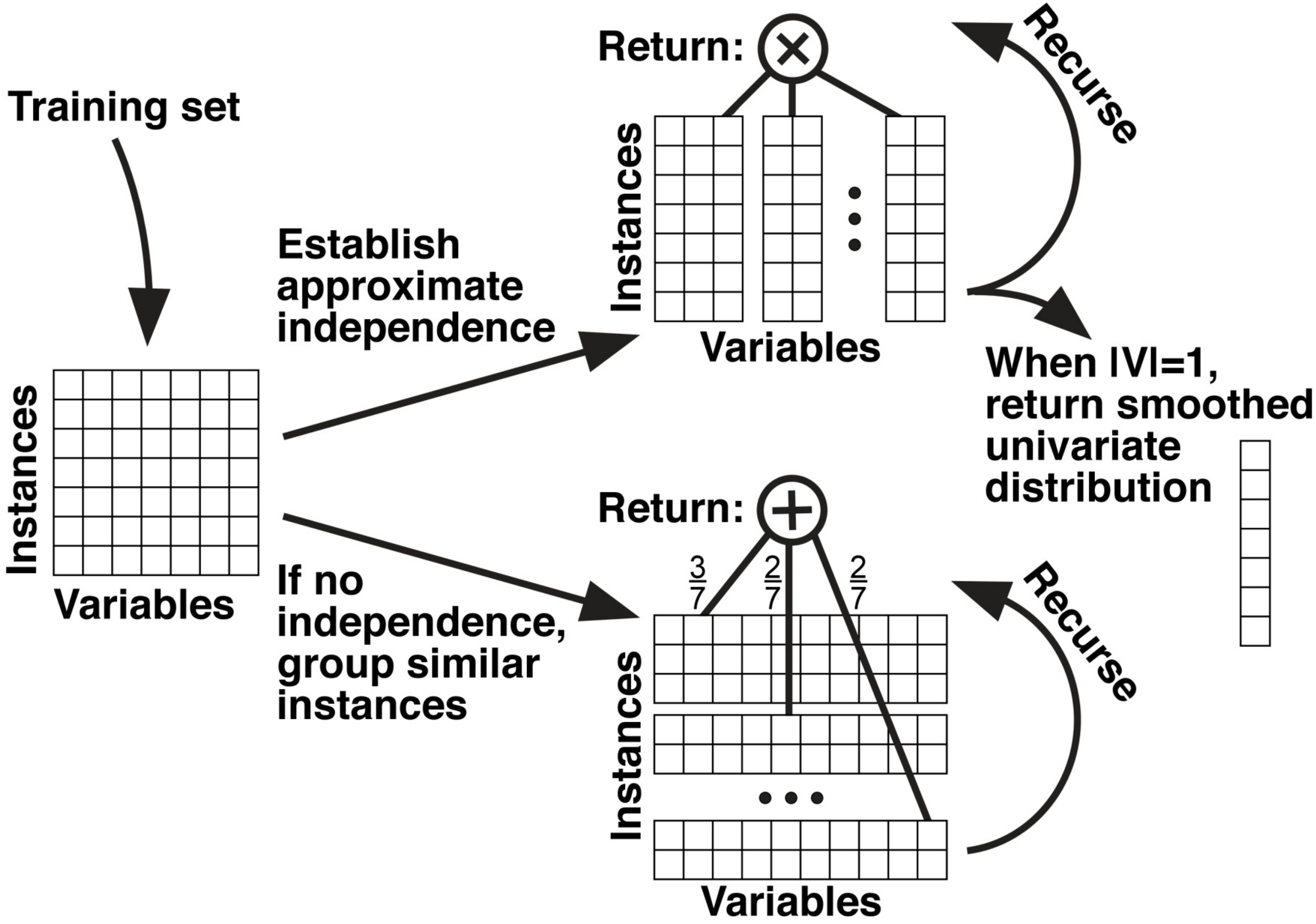
Original
SPN
DBM
DBN
PCA
Nearest Neighbor

Weight Learning: Summary



Update	**Soft Inference** (Marginals)	**Hard Inference** (MAP States)
Generative EM	$\Delta w_i \propto w_i \frac{\partial S}{\partial S_k}$	$\Delta w_i = c_i$
Generative Gradient	$\Delta w_i = \eta \frac{\partial S}{\partial S_k} S_i$	$\Delta w_i = \eta \frac{c_i}{w_i}$
Discriminative Gradient	$\Delta w_i = \eta\left(\overbrace{\frac{S_i}{S}\frac{\partial S}{\partial S_k}}^{\text{true label}} - \overbrace{\frac{S_i}{S}\frac{\partial S}{\partial S_k}}^{\text{exp. label}}\right)$	$\Delta w_i = \frac{\eta}{w_i}\left(\overbrace{c_i}^{\text{true}} - \overbrace{c_i}^{\text{test}}\right)$

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:** `Is(Family,SocialUnit)`
    `Is(Smiths,Family)`
  - **Subpart:** `Has(Family,Adult,2)`
    `Has(Smiths,Anna,Adult1)`
  - **Relation:** `Parent(Family,Adult,Child)`
    `Married(Anna,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

Tvc.Qbsujujpo!Gvodujpo

$$Z(X,C) = \left(\sum_S e^{w_s} Z(X,S)\right) \times \left(\prod_P Z(P(X),C_P)^{n_P}\right) \times \left(\prod_R (1+e^{w_R})\right)$$

Pckfdu Dmbtt

Tvcdmbttft

Tvcqbsut

Sfmbujpot

$$Z(KB) = Z(TopObject, TopClass)$$

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.

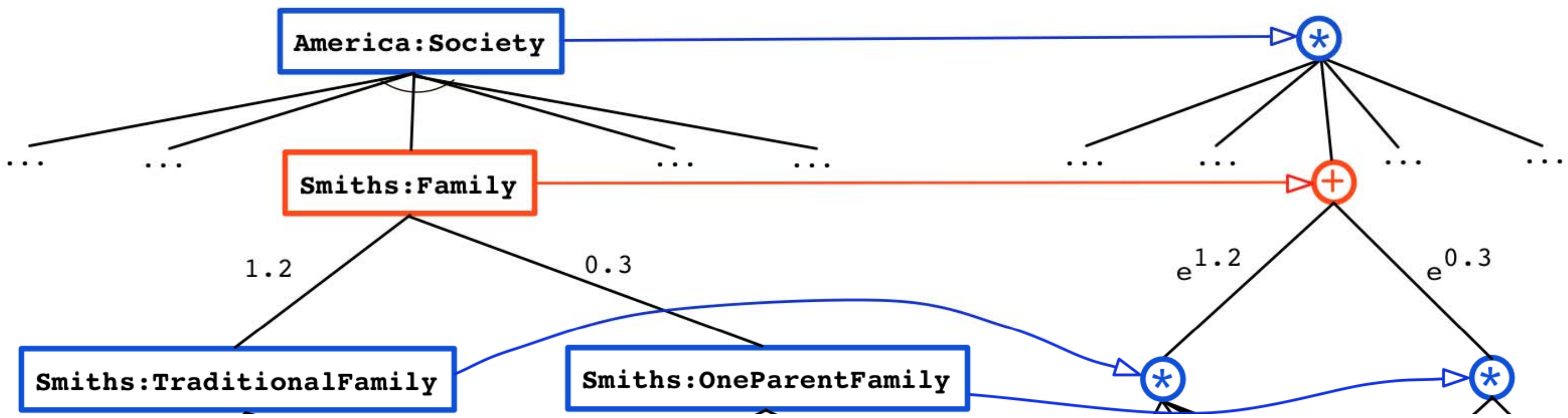
Time = Space = O(#Rules X #Objects)

Why TML Is Tractable

KB structure is isomorphic to Z computation:

- Parts = Products
- Classes = Sums

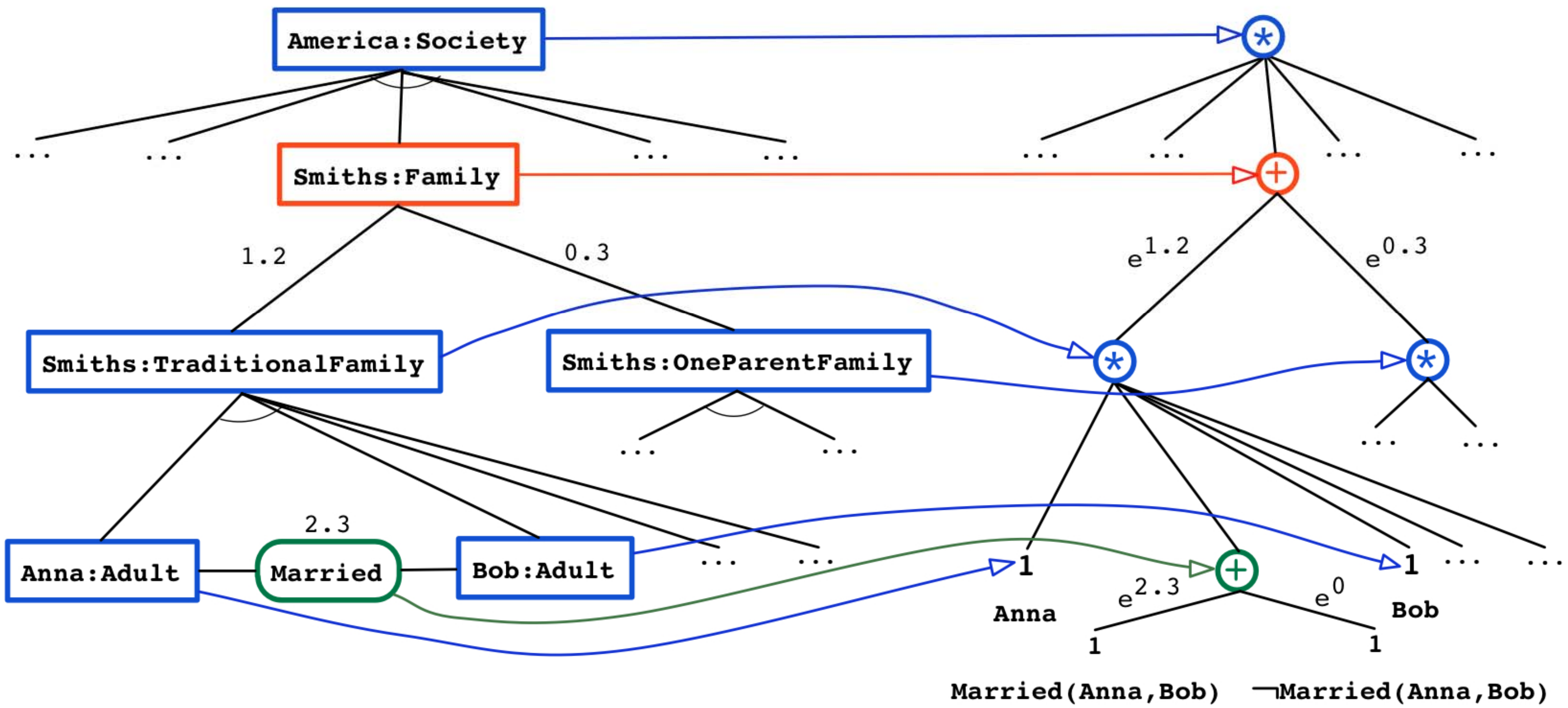
Why TML Is Tractable



KB structure is isomorphic to Z computation:

- Parts = Products
- Classes = Sums

Why TML Is Tractable



KB structure is isomorphic to 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**

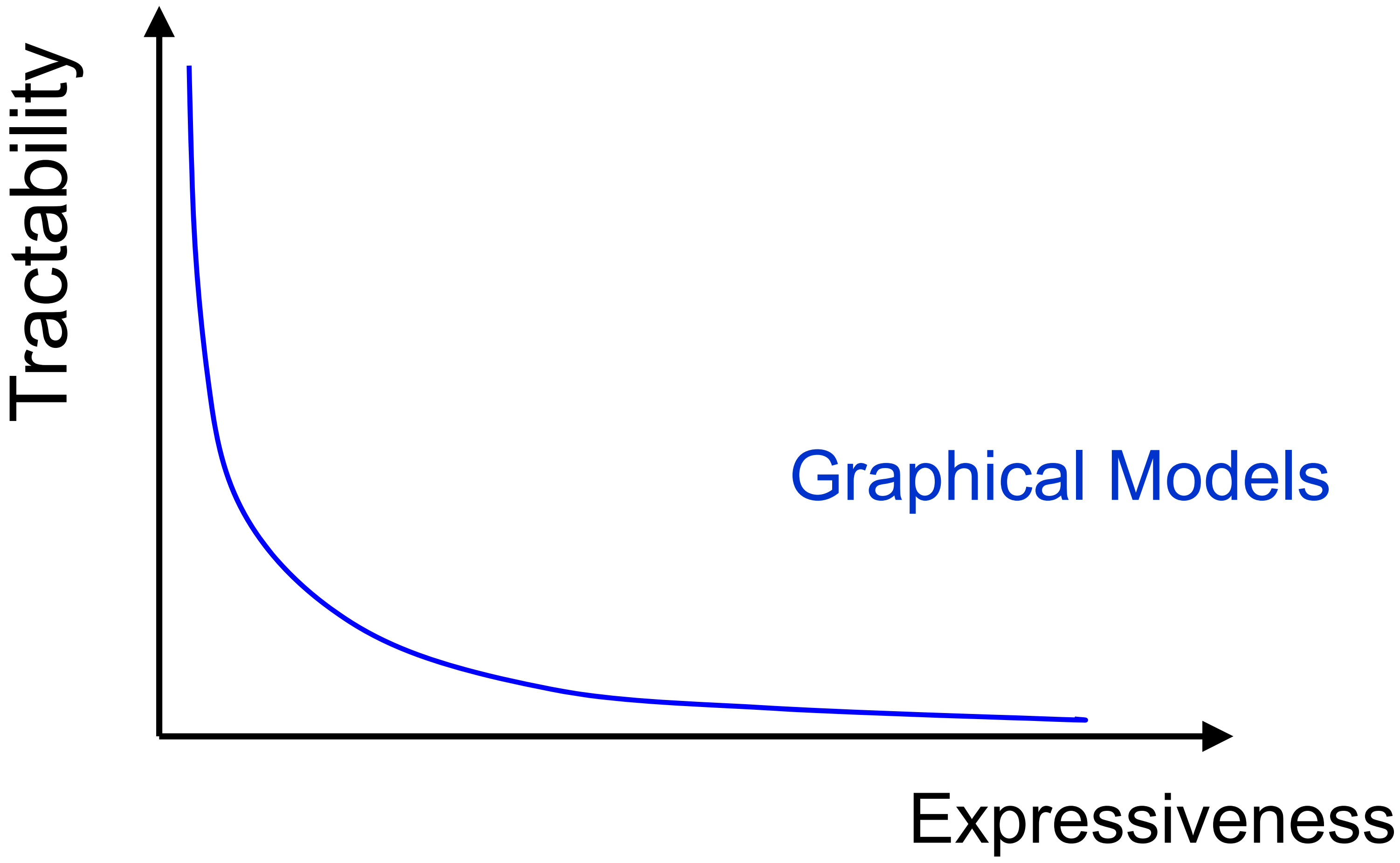
Other Tractable Models

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

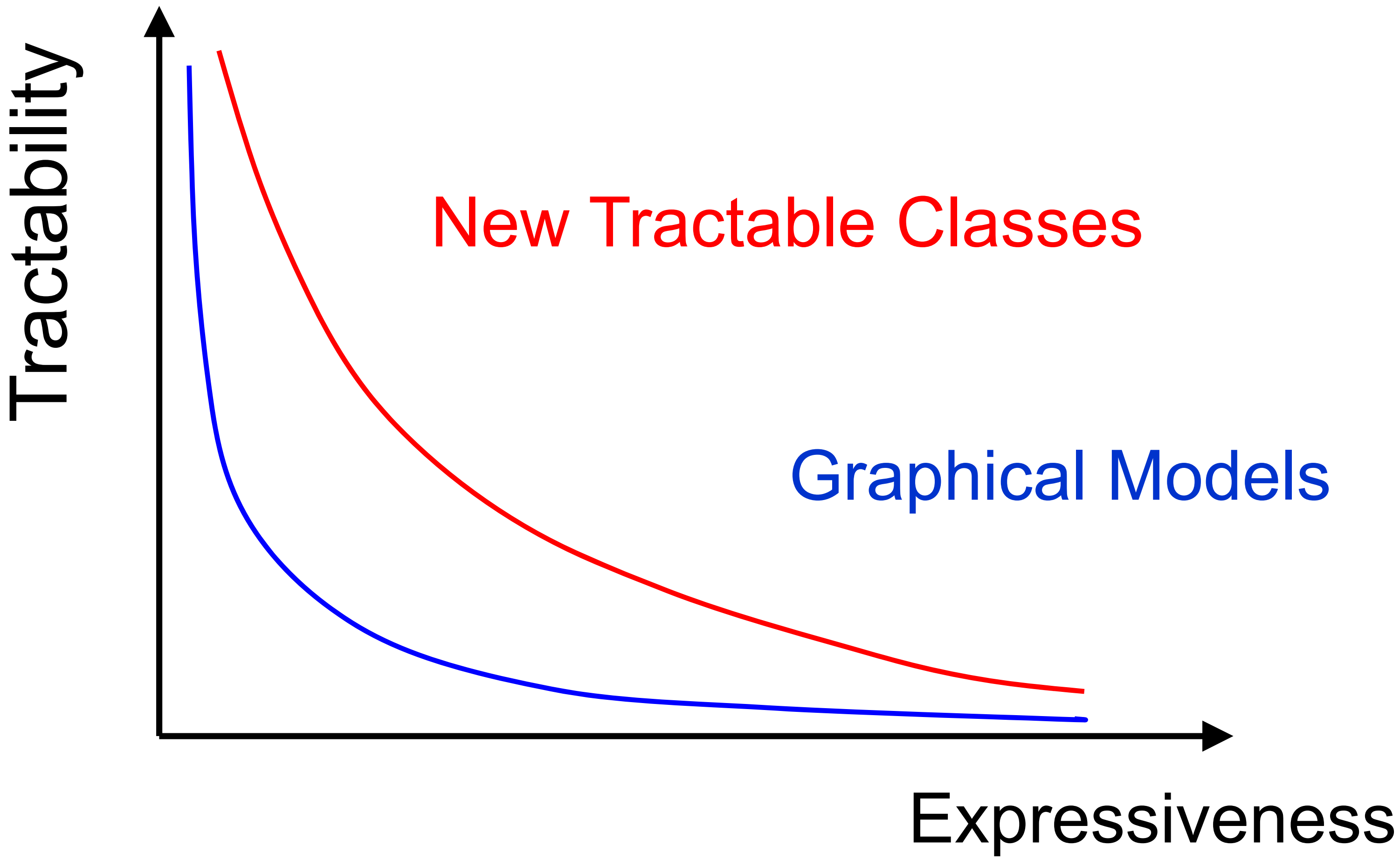
Summary

- 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



Summary


Tractability
New Tractable Classes
Graphical Models
Expressiveness