Knowledge Compilation and Machine Learning
A New Frontier

Adnan Darwiche
Computer Science Department
UCLA

with Arthur Choi and Guy Van den Broeck

The Problem

Logic (L)
Knowledge Representation (K)
Probability (P)
Artificial Intelligence (A)

- Must take at least one of Probability or Logic.
- Probability is a prerequisite for AI.
- The prerequisites for KR is either AI or Logic.

L	K	P	A	Students
0	0	1	0	6
0	0	1	1	54
0	1	1	1	10
1	0	0	0	5
1	0	1	0	1
1	0	1	1	0
1	1	0	0	17
1	1	1	0	4
1	1	1	1	3

Table 1: Student enrollment data.

The Problem: Structured Spaces

unstructured

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

course constraints:

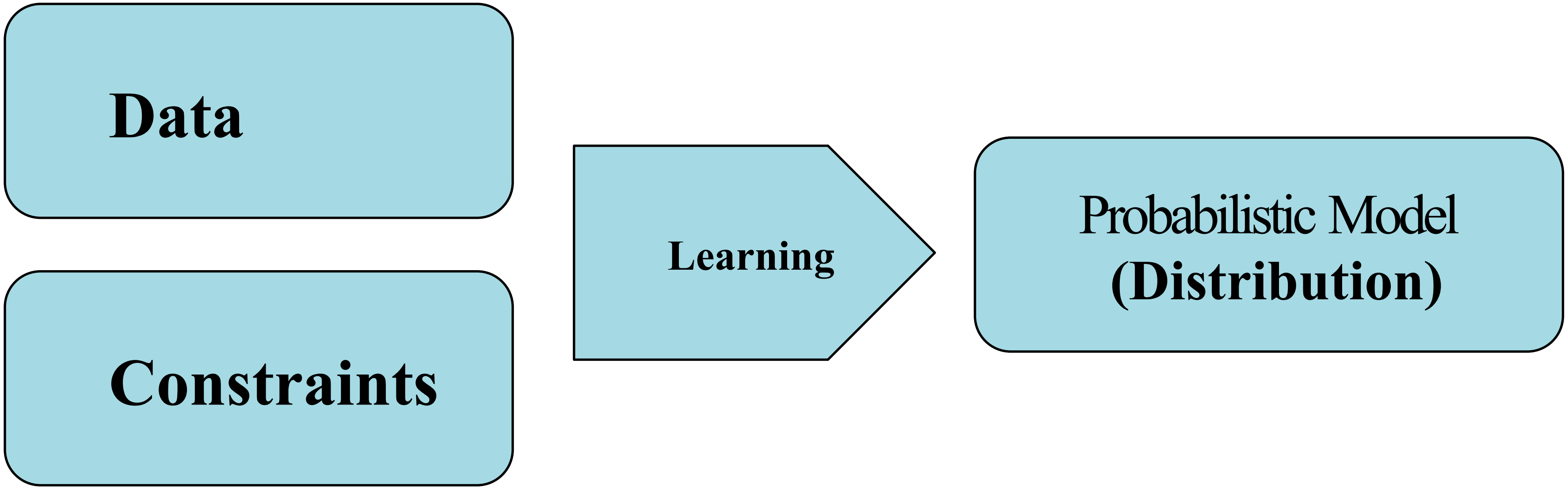
$$P \lor L$$
$$A \Rightarrow P$$
$$K \Rightarrow (P \lor L)$$

9 out of 16 models

structured

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

The Problem



Maximum Likelihood Learning
Learn Pr(.) that maximizes $Pr(d_1)Pr(d_2)....Pr(d_n)$

Constraints: Physics

- Configuration problems (factory constraints):
  - Discover correlations between configured features
  - Predict configurations
- Planning and diagnosis (physics)
- NLP and information extraction (e.g., a sentence must have at least one verb, citations should start with an author)
- Identifying objects in videos (e.g., player cannot be in two places at the same time)

Constraints: Combinatorial Objects

- Rankings, permutations, perfect matchings
- Graphs: DAGs, connected graphs, and spanning trees
- Game traces
- ...

Example: Rankings and Permutations

rank	user 1
1	The Godfather
2	Raiders of the Lost Ark
3	Casablanca
4	The Shawshank Redemption
5	Schindler's List
	...

rank	user 2
1	Star Wars V: The Empire Strikes Back
2	Star Wars IV: A New Hope
3	The Godfather
4	The Shawshank Redemption
5	The Usual Suspects
	...

Learn rankings of movies (permutations)
Predict new movies given preferences

Structured Spaces: Encoding Rankings and Permutations

encode rankings of n items with n^2 Boolean variables A_{ij}

	pos 1	pos 2	pos 3	pos 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

Structured Spaces: Encoding Rankings and Permutations

encode rankings of n items with n^2 Boolean variables A_{ij}

	pos 1	pos 2	pos 3	pos 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

constraint: each item assigned to a unique position (n constraints)

Structured Spaces: Encoding Rankings and Permutations

encode rankings of n items with n^2 Boolean variables A_{ij}

	pos 1	pos 2	pos 3	pos 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

constraint: each item assigned to a unique position (n constraints)

constraint: each position assigned a unique item (n constraints)

Structured Spaces: Encoding Rankings and Permutations

encode rankings of n items with n^2 Boolean variables A_{ij}

	pos 1	pos 2	pos 3	pos 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

total constraints	$2n$
unstructured space	2^{n^2}
structured space	$n!$

Structured Spaces: Encoding Partial Rankings

encode rankings of n items and t tiers with $n \cdot t$ Boolean variables A_{ij}

	tier 1	tier 2	tier 3	tier 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

tier 1	1 item
tiers 1–2	2 items
tiers 1–3	4 items
tiers 1–4	8 items
$\vdots$	

(single elimination tournament)

Structured Spaces: Encoding Partial Rankings

encode rankings of n items and t tiers with $n \cdot t$ Boolean variables A_{ij}

	tier 1	tier 2	tier 3	tier 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

constraint: each item assigned to a unique tier (n constraints)

constraint: each tier assigned m_j items (t constraints)

Structured Spaces: Encoding Partial Rankings

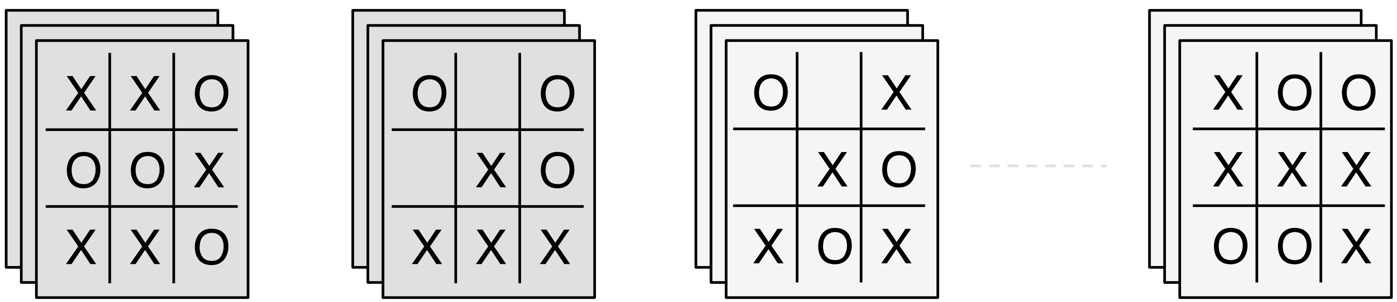
encode rankings of n items and t tiers with $n{\cdot}t$ Boolean variables A_{ij}

	tier 1	tier 2	tier 3	tier 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

$n+t$ total constraints

each constraint is an exactly-k-variables-true constraint

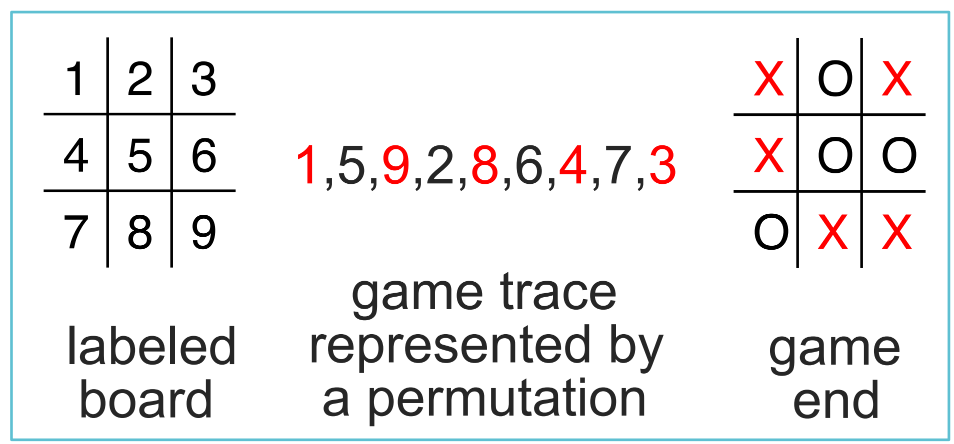
Example:
Game Traces



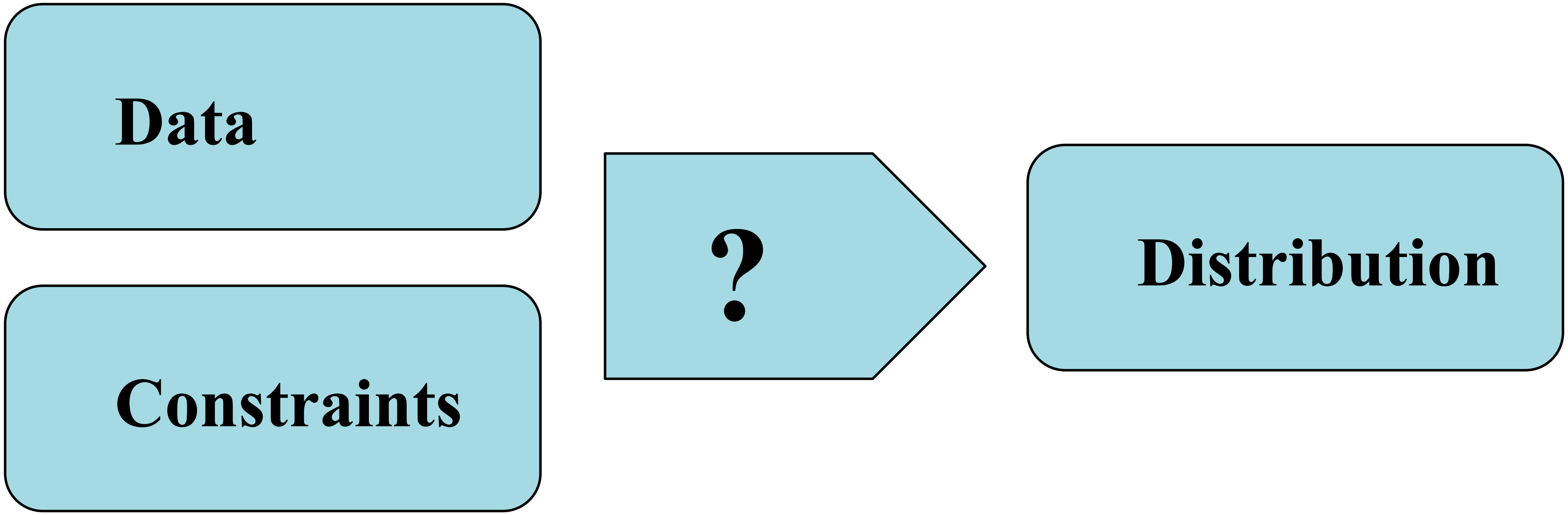
Game 1 Game 2 Game 3 Game n

Learn from game plays:

- play styles
- strategies
- skill levels



A Solution



What do people do now?

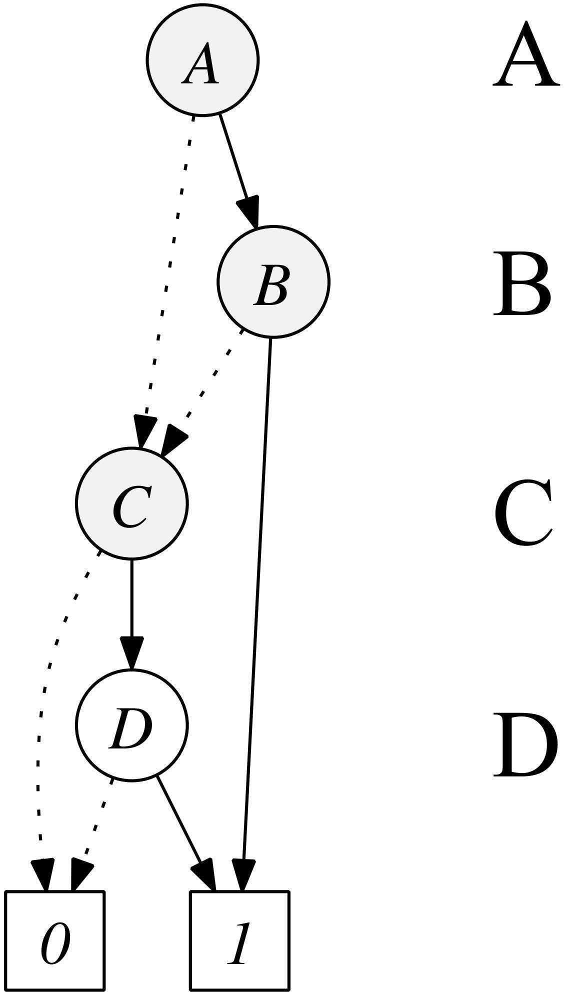
- Physics:
  - Ignore constraints; or
  - Handcraft constraints into models
    (e.g., Kevin Murphy, Dan Roth)
- Combinatorial objects:
  - Develop specialized distributions
    (e.g., Mallows model)

Tractable Representations of Constraints

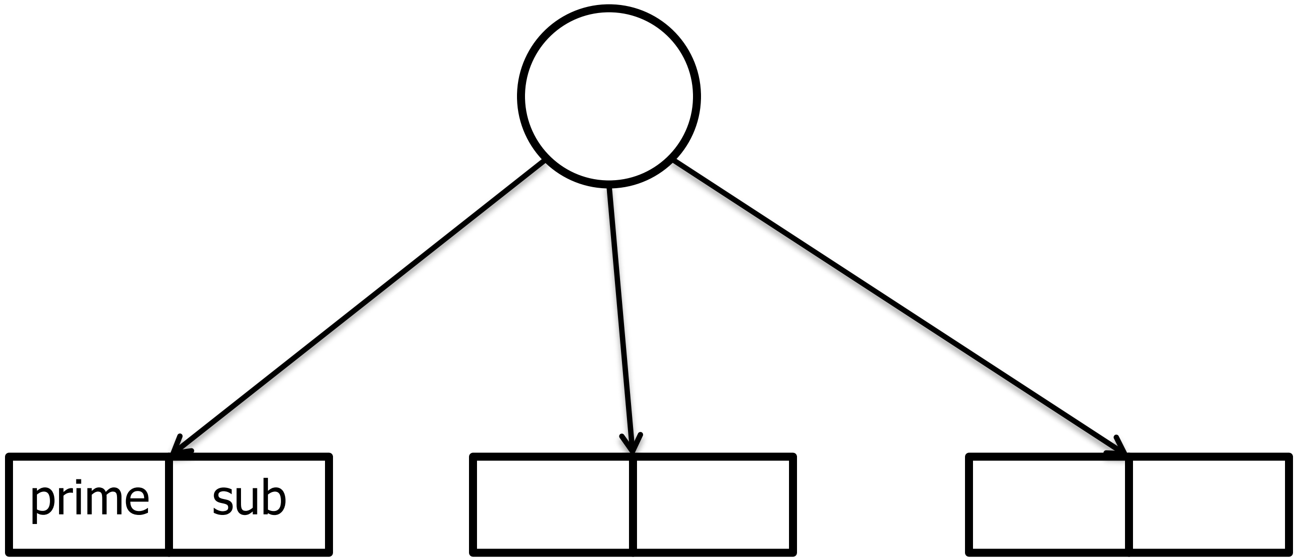
- **Knowledge Compilation** (classical)**:**
  - What representation? What tractable queries?
  - SDD: Sentential Decision Diagrams

- **Machine Learning** (new)**:**
  - Parameterize SDDs to induce distributions
  - Learn SDD parameters from data

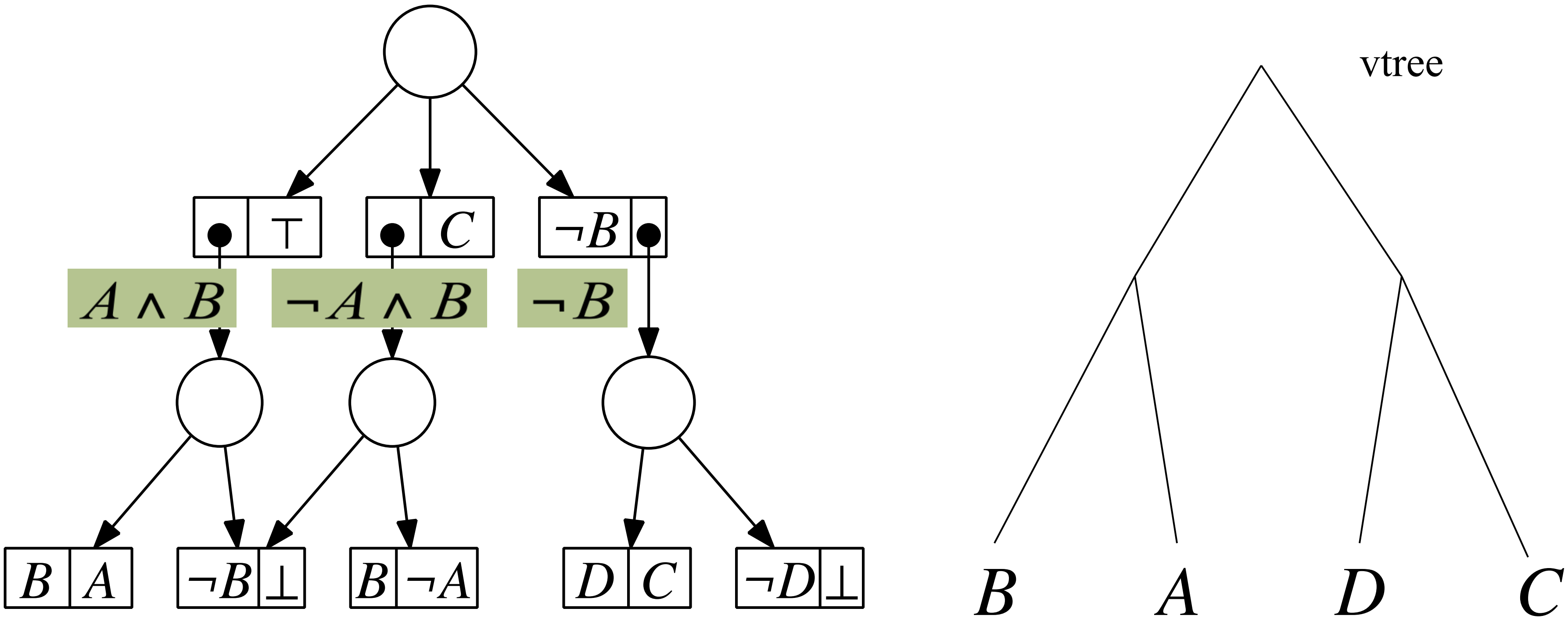
Binary Decision Diagram (OBDD)



Sentential Decision Diagram (SDD)



Sentential Decision Diagram (SDD)

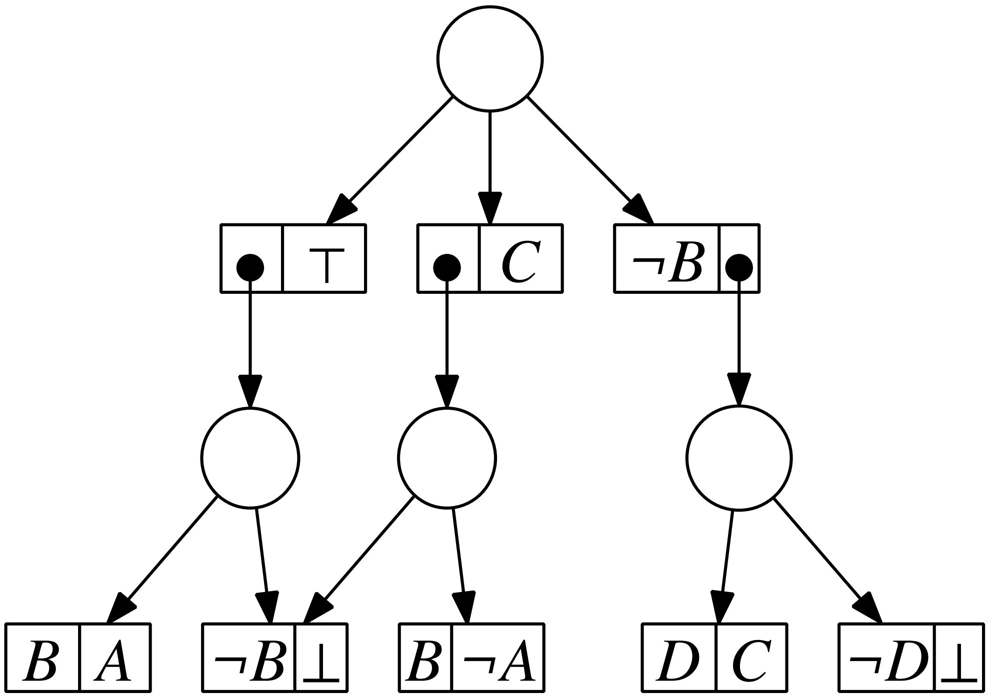


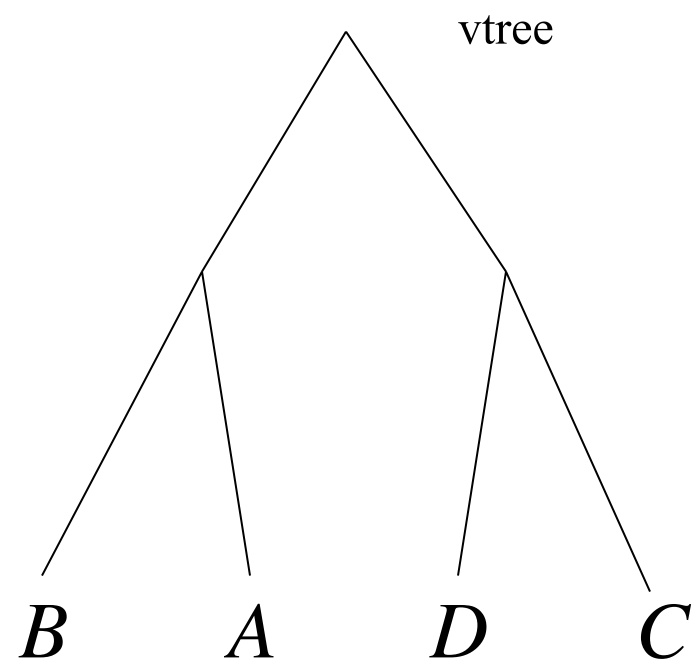
$$f = (A \wedge B) \vee (B \wedge C) \vee (C \wedge D)$$

Sentential Decision Diagram (SDD)

SDD package: reasoning.cs.ucla.edu/sdd

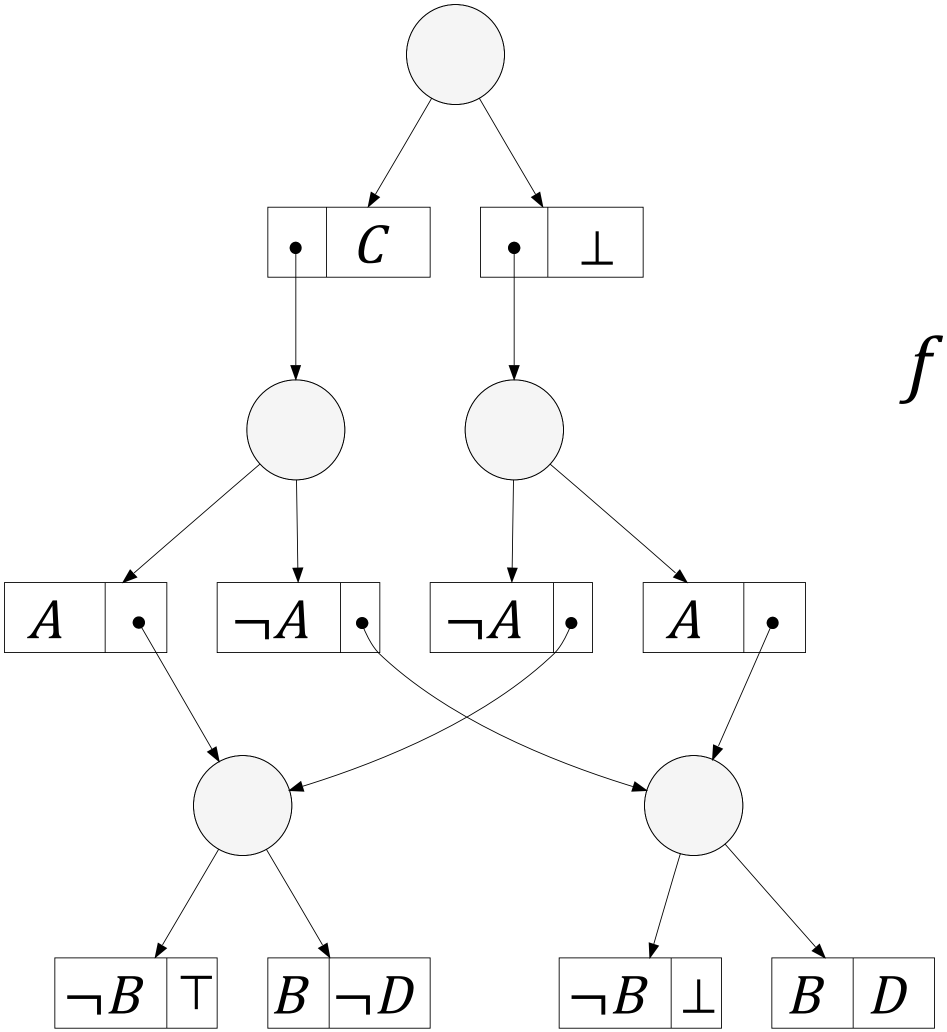
Constraints → SDD of minimal size





$$f = (A \wedge B) \vee (B \wedge C) \vee (C \wedge D)$$

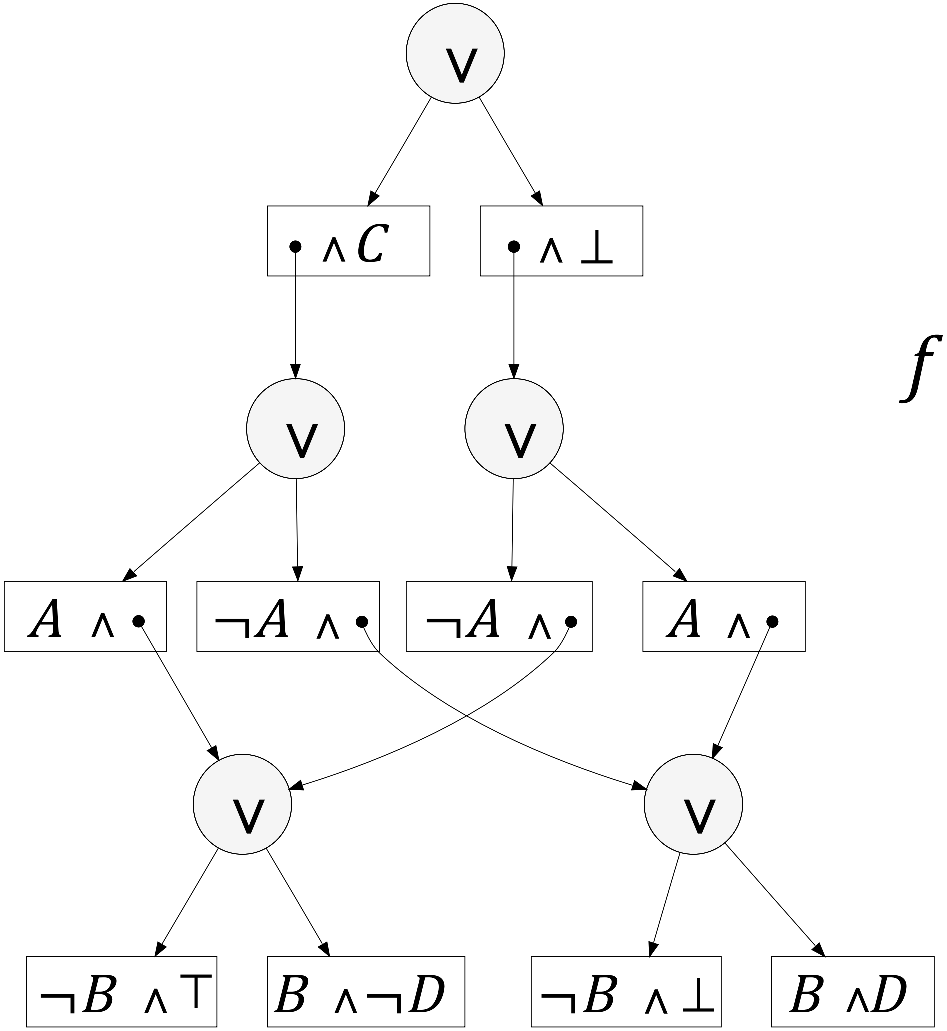
Representing Structured Spaces: Sentential Decision Diagrams (SDDs)



$$f(A, B, C, D) = (A \oplus (B \wedge D)) \wedge C$$

SDD package: reasoning.cs.ucla.edu/sdd

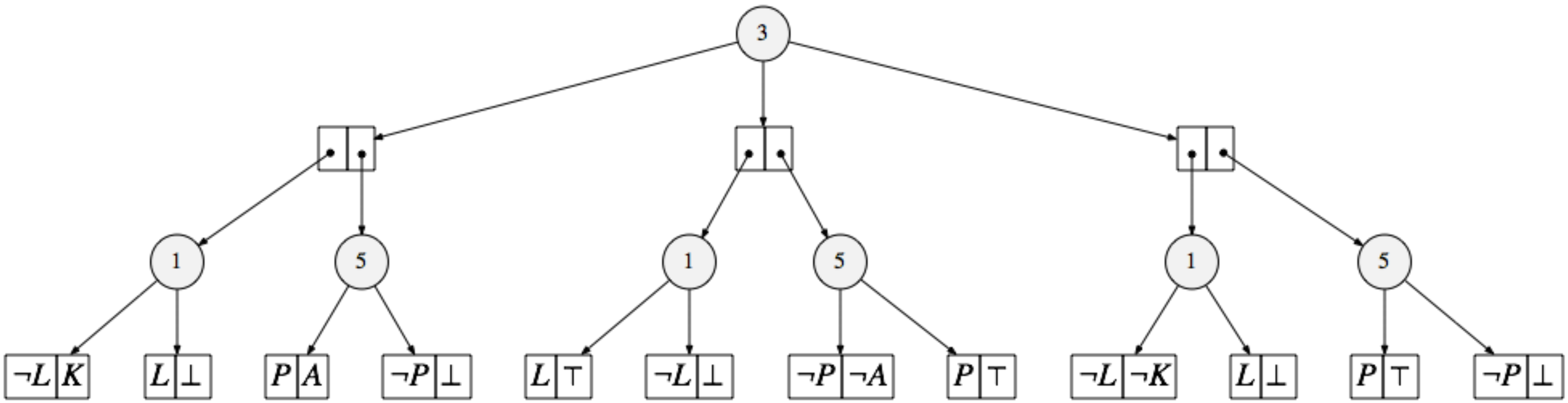
Representing Structured Spaces: Sentential Decision Diagrams (SDDs)

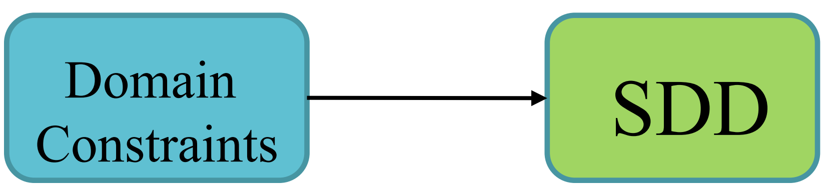


$$f(A, B, C, D) = (A \oplus (B \wedge D)) \wedge C$$

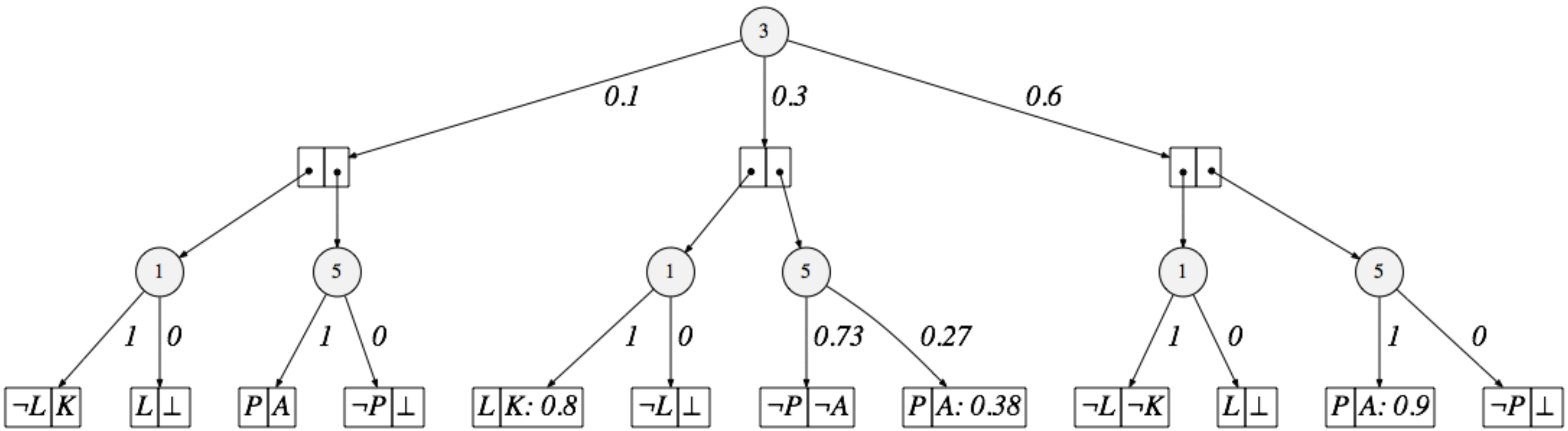
SDD package: reasoning.cs.ucla.edu/sdd

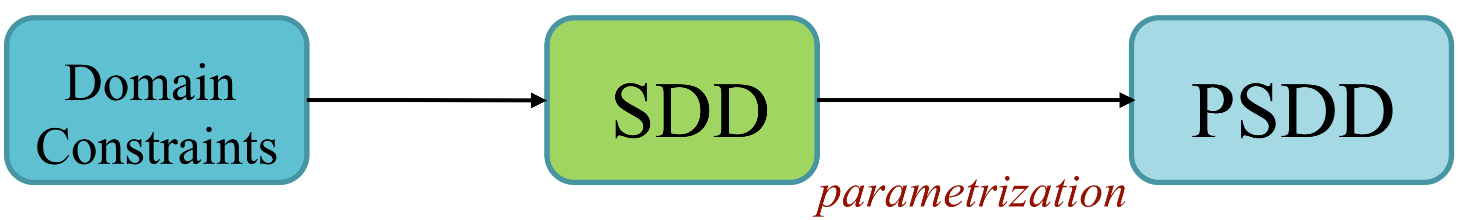
Structured Spaces: SDDs



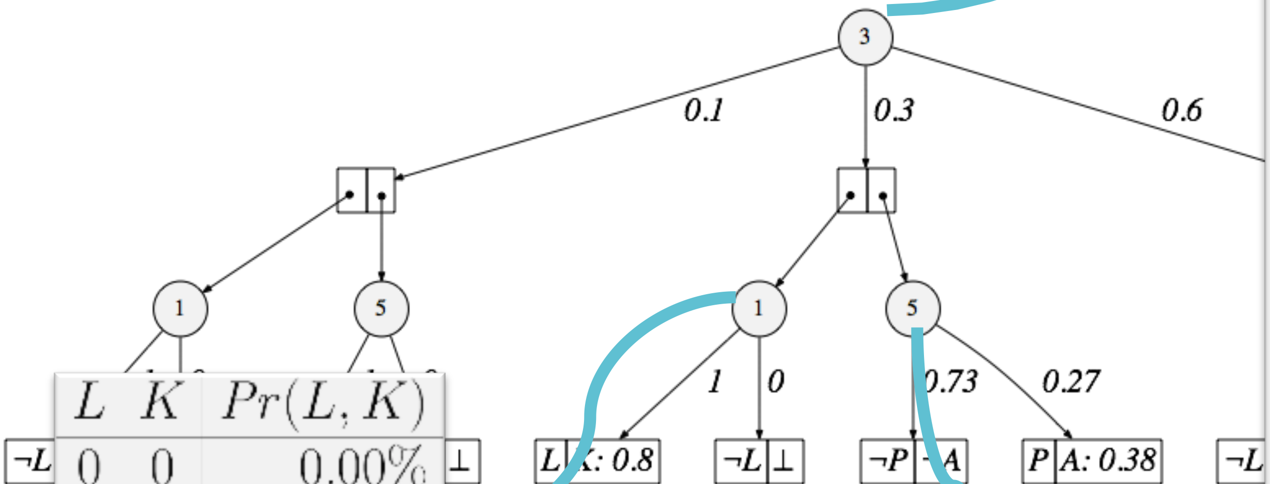


Distributions over Structured Spaces: PSDDs





Distribution



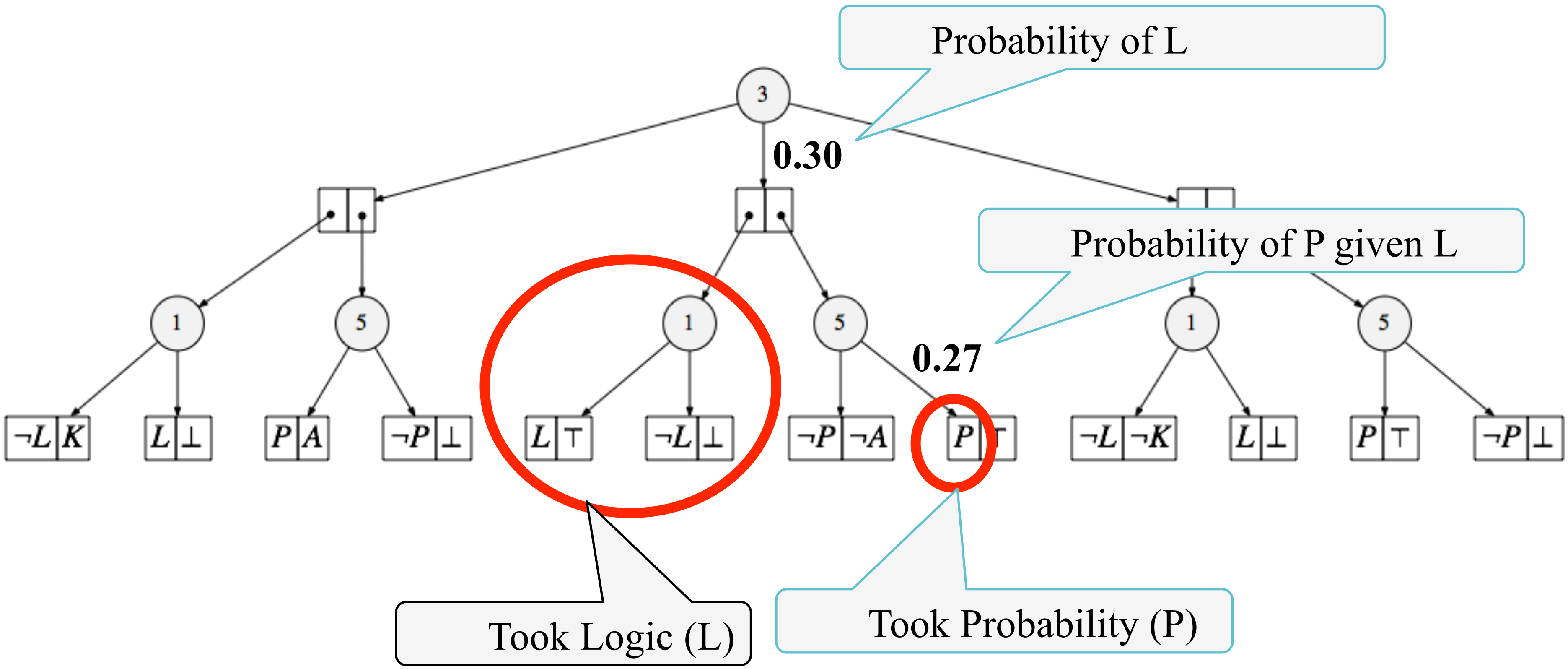
L	K	$Pr(L, K)$
0	0	0.00%
0	1	0.00%
1	0	20.00%
1	1	80.00%

P	A	$Pr(P, A)$
0	0	73.33%
0	1	0.00%
1	0	16.67%
1	1	10.00%

L	K	P	A	$Pr(L, K, P, A)$
0	0	0	0	0.00%
0	0	0	1	0.00%
0	0	1	0	6.00%
0	0	1	1	54.00%
0	1	0	0	0.00%
0	1	0	1	0.00%
0	1	1	0	0.00%
0	1	1	1	10.00%
1	0	0	0	4.40%
1	0	0	1	0.00%
1	0	1	0	1.00%
1	0	1	1	0.60%
1	1	0	0	17.6%
1	1	0	1	0.00%
1	1	1	0	4.00%
1	1	1	1	2.40%

Semantics of Parameters

p = Pr(prime | context)



Learning from Data

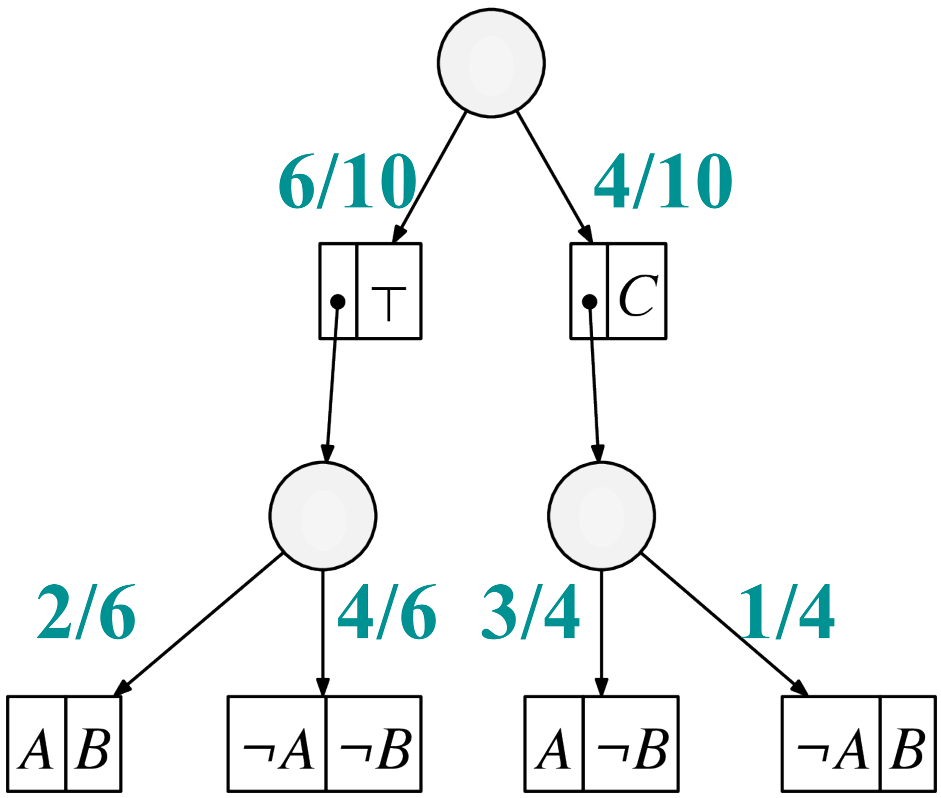
complete dataset

id	X	Y	Z
1	x_1	y_2	z_1
2	x_2	y_1	z_2
3	x_2	y_1	z_2
4	x_1	y_1	z_1
5	x_1	y_2	z_2

incomplete dataset

id	X	Y	Z
1	x_1	y_2	?
2	x_2	y_1	?
3	?	?	z_2
4	?	y_1	z_1
5	x_1	y_2	z_2

Learning PSDDs from **Complete** Data



closed form estimates:

$$\theta_i^{ml} = \frac{\mathbf{D}\#(p_i, \gamma_n)}{\mathbf{D}\#(\gamma_n)}$$

A	B	C	
1	1	1	1
1	1	0	1
1	0	1	3
0	1	1	1
0	0	1	2
0	0	0	2

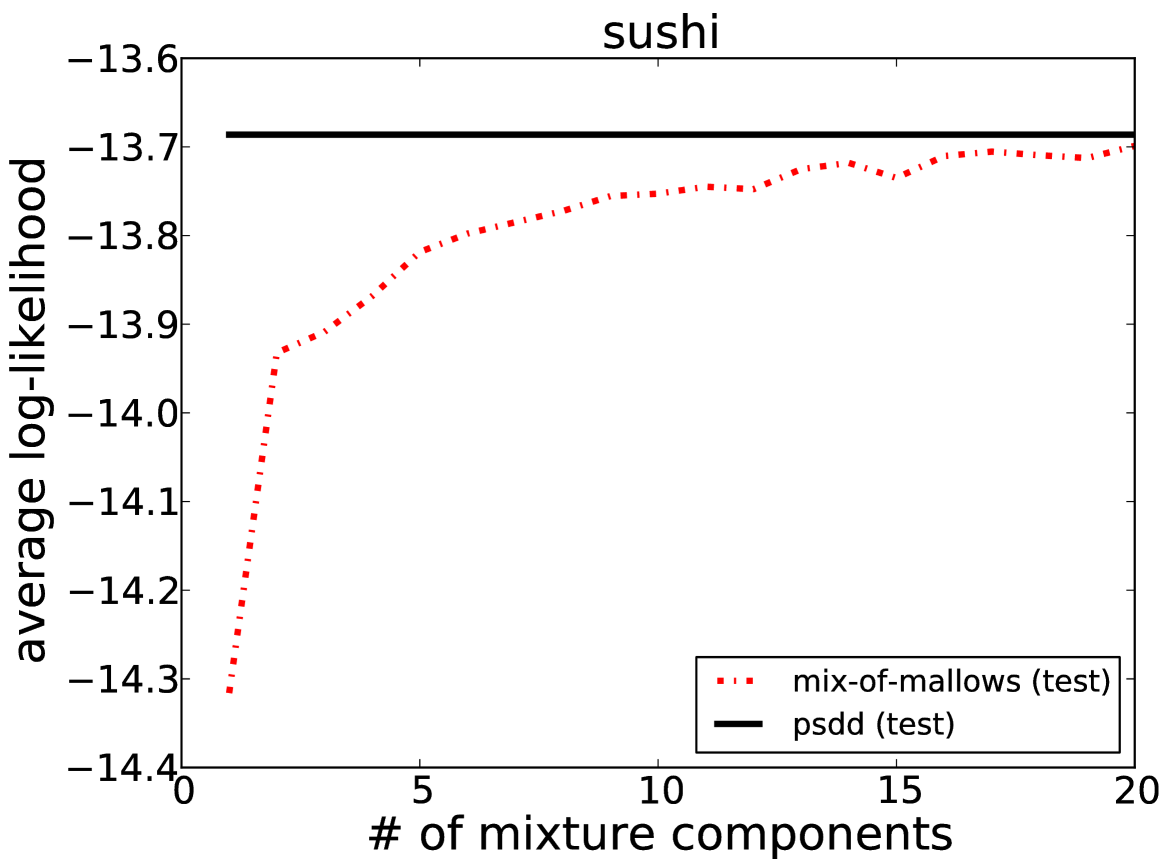
Learning PSDDs from Complete Data Example: Preference Distributions

- Sushi Dataset (Kamishima 2003):
  - 10 types of sushi
  - 5k total rankings (complete dataset)

- PSDD for total rankings
  - 4,097 parameters
  - $Pr(A_{ij})$ is probability that sushi i is rank j

Learning PSDDs from Complete Data Example: Preference Distributions

- training set (3,500) testing set (1,500)
- Mixture-of-Mallows
  - # of components from 1 to 20
  - EM with 10 random seeds
  - implementation of Lu & Boutilier



Learning PSDDs from Complete Data Example: Preference Distributions

Tractable Queries:

- most likely ranking (MPE)

$$\mathbf{x}^\star = \operatorname*{argmax}_{\mathbf{x}} Pr(\mathbf{x})$$

- expected ranking

$$\mathrm{E}[j] = \sum_j j \cdot Pr(A_{ij})$$

most likely ranking

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

expected rankings

exp. rank	sushi
3.21	fatty tuna
4.56	tuna
4.99	shrimp
5.09	salmon roe
5.23	see eel
5.43	sea urchin
5.91	tuna roll
5.91	squid
6.80	egg
7.87	cucumber roll

Learning PSDDs from **Incomplete** Data

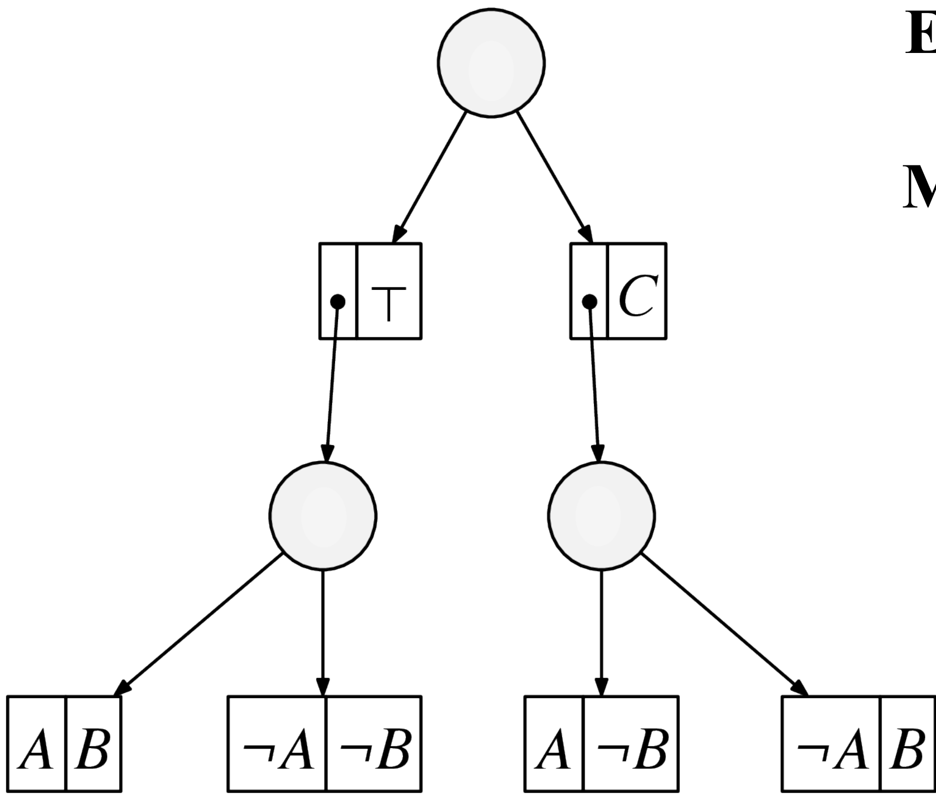
incomplete dataset

id	X	Y	Z
1	x_1	y_2	?
2	x_2	y_1	?
3	?	?	z_2
4	?	y_1	z_1
5	x_1	y_2	z_2

complete dataset using $Pr_i(X,Y,Z)$

x_1	y_2	z_1	.11
x_1	y_2	z_2	.89

Learning PSDDs from **Incomplete** Data



EM Algorithm:

E-Step: complete the dataset (soft completion)

M-Step: learn using completed dataset (in closed form)

closed form EM updates:

$$\theta_i^t = \frac{\sum_{j=1}^{N} Pr_{\theta^{t-1}}(p_i, \gamma_n \mid \mathbf{d}_j)}{\sum_{j=1}^{N} Pr_{\theta^{t-1}}(\gamma_n \mid \mathbf{d}_j)}$$

Structured Datasets

a classical complete dataset

id	X	Y	Z
1	x_1	y_2	z_1
2	x_2	y_1	z_2
3	x_2	y_1	z_2
4	x_1	y_1	z_1
5	x_1	y_2	z_2

a classical incomplete dataset

id	X	Y	Z
1	x_1	y_2	?
2	x_2	y_1	?
3	?	?	z_2
4	?	y_1	z_1
5	x_1	y_2	z_2

a new type of incomplete dataset

id	X	Y	Z
1	$X \equiv Z$		
2	x_2 and (y_2 or z_2)		
3	$x_2 \Rightarrow y_1$		
4	$X \oplus Y \oplus Z \equiv 1$		
5	x_1 and y_2 and z_2		

-example
-partial description of an example

Structured Datasets

a classical **complete** dataset
(e.g., total rankings)

id	1st sushi	2nd sushi	3rd sushi	…
1	fatty tuna	sea urchin	salmon roe	…
2	fatty tuna	tuna	shrimp	…
3	tuna	tuna roll	sea eel	…
4	fatty tuna	salmon roe	tuna	…
5	egg	squid	shrimp	…

a classical **incomplete** dataset
(e.g., top-*k* rankings)

id	1st sushi	2nd sushi	3rd sushi	…
1	fatty tuna	sea urchin	?	…
2	fatty tuna	?	?	…
3	tuna	tuna roll	?	…
4	fatty tuna	salmon roe	?	…
5	egg	?	?	…

Structured Datasets

a classical **complete** dataset
(e.g., total rankings)

id	1st sushi	2nd sushi	3rd sushi	…
1	fatty tuna	sea urchin	salmon roe	…
2	fatty tuna	tuna	shrimp	…
3	tuna	tuna roll	sea eel	…
4	fatty tuna	salmon roe	tuna	…
5	egg	squid	shrimp	…

a new type of **incomplete** dataset
(e.g., **partial** rankings)

id	1st sushi	2nd sushi	3rd sushi	…
1	(fatty tuna > sea urchin) and (tuna > sea eel)			…
2	(fatty tuna is 1st) and (salmon roe > egg)			…
3	tuna > squid			…
4	egg is last			…
5	egg > squid > shrimp			…

Key Result

PSDD n representing distribution Pr_n

SDD m representing Boolean constraint α

Theorem. If both n and m respect vtree v, then $Pr_n(\alpha)$ can be computed in polytime.

Learning PSDDs from Incomplete Data Example: Preference Distributions

- Movielens Dataset:
  - 3,900 movies, 6,040 users, 1m ratings
  - take ratings from 64 most rated movies
  - ratings 1-5 converted to pairwise prefs.
- PSDD for **partial** rankings
  - 4 tiers
  - 18,711 parameters

movies by expected tier

rank	movie
1	The Godfather
2	The Usual Suspects
3	Casablanca
4	The Shawshank Redemption
5	Schindler’s List
6	One Flew Over the Cuckoo’s Nest
7	The Godfather: Part II
8	Monty Python and the Holy Grail
9	Raiders of the Lost Ark
10	Star Wars IV: A New Hope

Structured Spaces: SDDs

items	tier size	Size		
n	k	SDD	Structured Space	Unstructured Space
8	2	443	840	$1.84 \cdot 10^{19}$
27	3	4,114	$1.18 \cdot 10^{9}$	$2.82 \cdot 10^{219}$
64	4	23,497	$3.56 \cdot 10^{18}$	$1.04 \cdot 10^{1233}$
125	5	94,616	$3.45 \cdot 10^{31}$	$3.92 \cdot 10^{4703}$
216	6	297,295	$1.57 \cdot 10^{48}$	$7.16 \cdot 10^{14044}$
343	7	781,918	$4.57 \cdot 10^{68}$	$7.55 \cdot 10^{35415}$

Structured Queries
Example: Preference Distributions

rank	movie
1	Star Wars V: The Empire Strikes Back
2	Star Wars IV: A New Hope
3	The Godfather
4	The Shawshank Redemption
5	The Usual Suspects

Structured Queries Example: Preference Distributions

rank	movie
1	Star Wars V: The Empire Strikes Back
2	Star Wars IV: A New Hope
3	The Godfather
4	The Shawshank Redemption
5	The Usual Suspects

2nd ranked movie:
not a good recommendation!

Structured Queries
Example: Preference Distributions

rank	movie
1	Star Wars V: The Empire Strikes Back
2	Star Wars IV: A New Hope
3	The Godfather
4	The Shawshank Redemption
5	The Usual Suspects

- no other Star Wars movie in top-5
- at least one comedy in top-5

rank	movie
1	Star Wars V: The Empire Strikes Back
2	American Beauty
3	The Godfather
4	The Usual Suspects
5	The Shawshank Redemption

diversified recommendations via
logical constraints

Roles for Knowledge Compilation in ML

- **Define the structured space of a probabilistic model**
  (Learn with physics, or over combinatorial objects)

- **Describe examples of a dataset**
  (Beyond classical, conjunctive, datasets)

- **Formulate/answer rich queries**
  (Beyond classical, conjunctive, queries)

Pending Questions

- **Learning PSDD structure**
  (Which SDD/vtree for a given set of constraints?)

- **What makes SDDs amenable to this?**
  (Parameterization, learning, etc)

- **What other representations can play a similar role?**
  (Expand Knowledge Compilation Map)

- **Where else can we use this?**
  (Further applications)

Conclusion
Enabled by Knowledge Compilation

- Domain-independent framework for inducing distributions over structured spaces
- Learning parameters with complete and incomplete data
- Closed-form learning with complete data
- Generalized form for incomplete data (structured data)
- Tractable Inference
- Structured queries (query and condition on arbitrary formulas).

References

Probabilistic Sentential Decision Diagrams
Doga Kisa and Guy Van den Broeck and Arthur Choi and Adnan Darwiche
KR, 2014.

Learning with Massive Logical Constraints
Doga Kisa and Guy Van den Broeck and Arthur Choi and Adnan Darwiche
Workshop on Learning Tractable Probabilistic Models (LTPM), ICML 2014.

Tractable Learning for Structured Probability Spaces
Arthur Choi and Guy Van den Broeck and Adnan Darwiche
IJCAI, 2015.

Related Work

- Arithmetic Circuits (ACs) (Darwiche, JACM 2003)

- Learning ACs (Lowd &Domingos, UAI 2008-Thesis 2010)

- SPNs (Poon & Domingos, UAI 2011)
  SPNs equivalent to ACs (Rooshenas & Lowd, ICML 2014)

- PSDDs (Kisa et al., KR 2014):
  Next-generation ACs: circuits over structured spaces

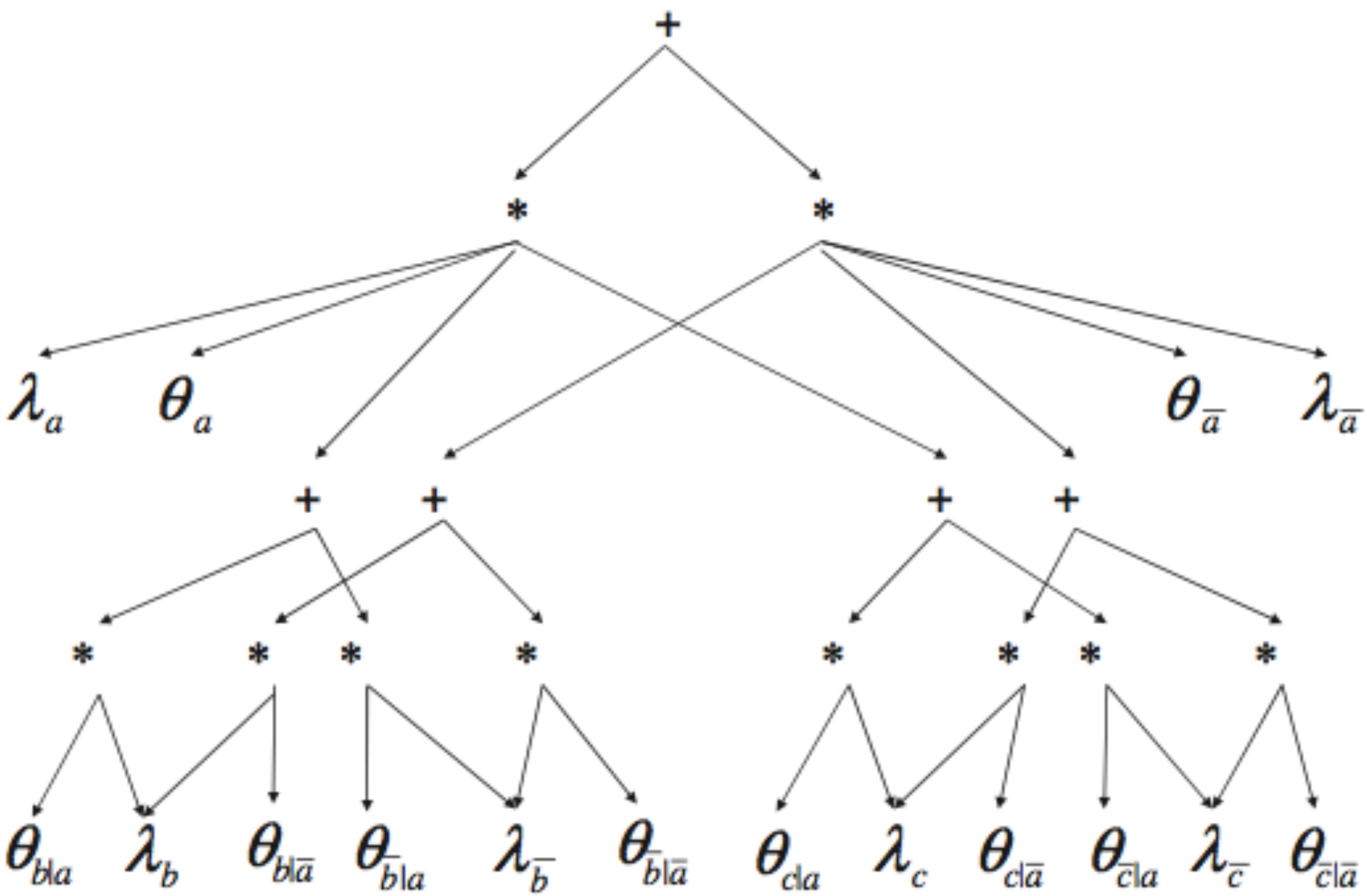


Figure 12.2: An arithmetic circuit for the Bayesian network in Figure 12.1.