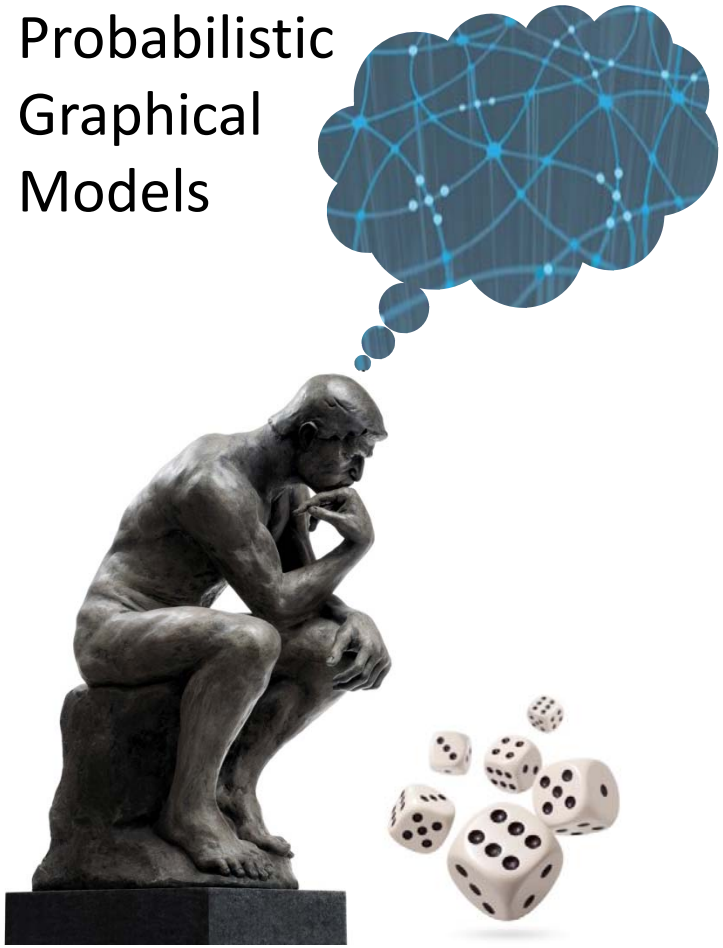
Probabilistic Graphical Models

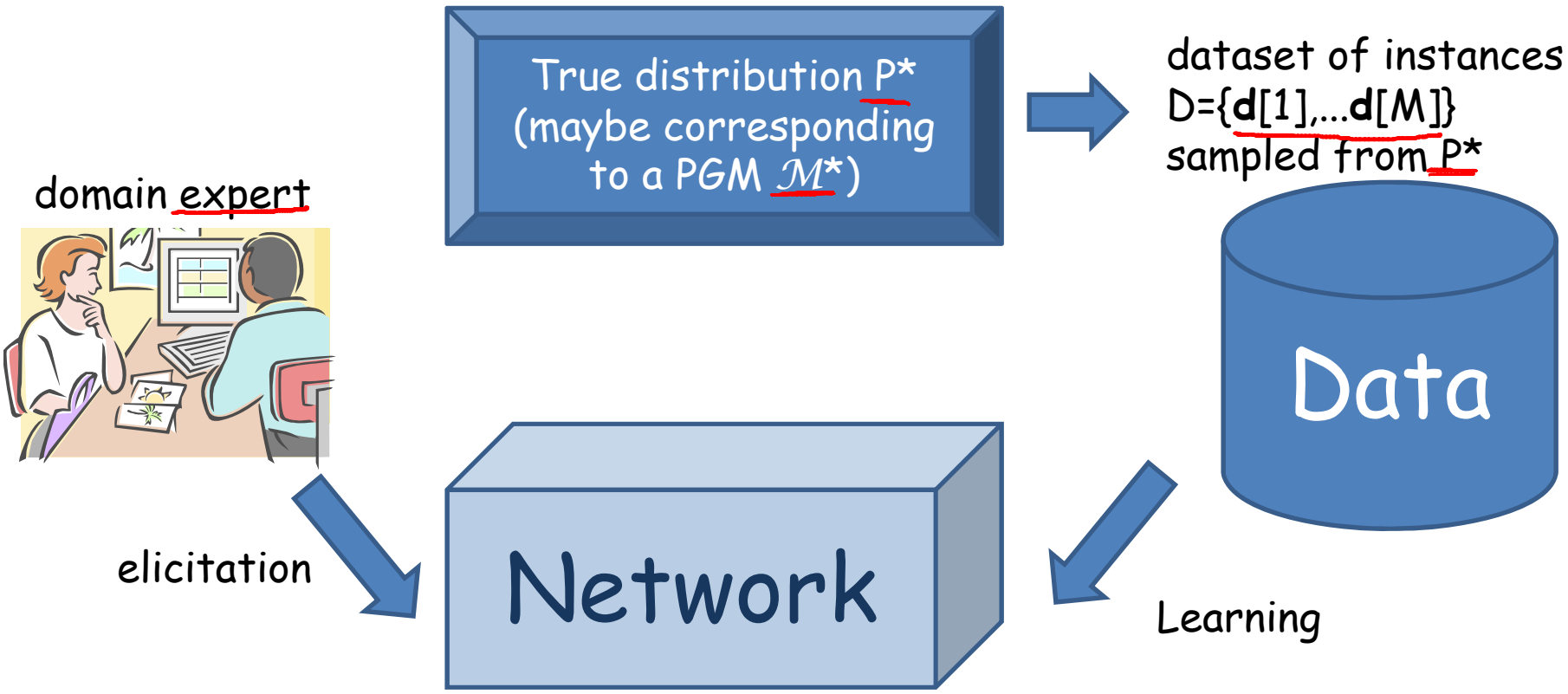


Learning

Overview

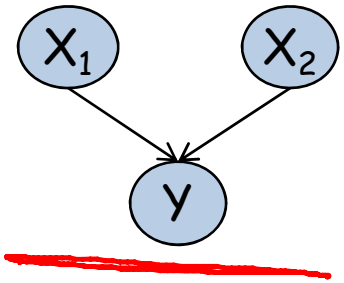
PGM Learning Tasks and Metrics

Learning

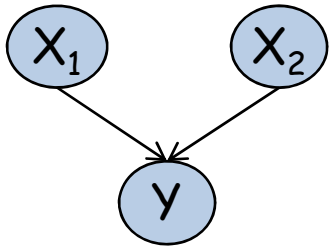

True distribution P*
(maybe corresponding
to a PGM $\mathcal{M}$*)
dataset of instances
D={d[1],...d[M]}
sampled from P*
domain expert
Data
elicitation
Network
Learning

Known Structure, Complete Data

Initial network



Inducer



Input Data

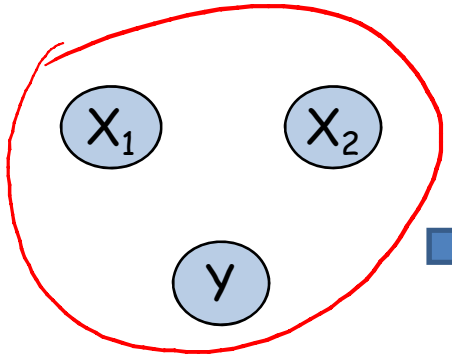
X_1	X_2	Y
x_1^0	x_2^1	y^0
x_1^1	x_2^0	y^0
x_1^0	x_2^1	y^1
x_1^0	x_2^0	y^0
x_1^1	x_2^1	y^1
x_1^0	x_2^1	y^1
x_1^1	x_2^0	y^0

		$P(Y/X_1,X_2)$	
X_1	X_2	y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

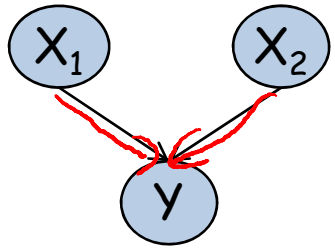
Daphne Koller

Unknown Structure, Complete Data

Initial network



Inducer



Input Data

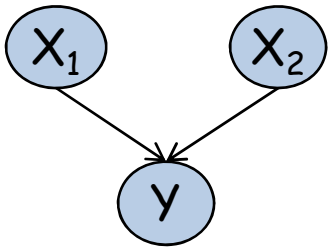
X_1	X_2	Y
x_1^0	x_2^1	y^0
x_1^1	x_2^0	y^0
x_1^0	x_2^1	y^1
x_1^0	x_2^0	y^0
x_1^1	x_2^1	y^1
x_1^0	x_2^1	y^1
x_1^1	x_2^0	y^0

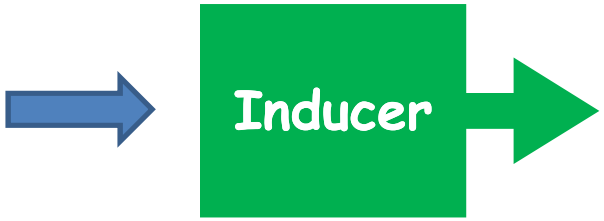
		$P(Y/X_1,X_2)$	
X_1	X_2	y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

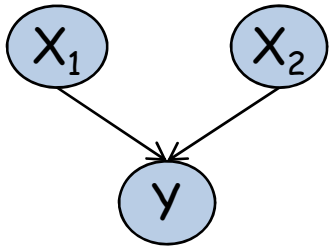
Daphne Koller

Known Structure, Incomplete Data

Initial network







Input Data

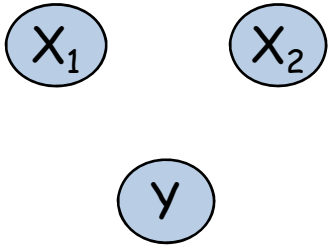
X_1	X_2	Y
?	x_2^1	y^0
x_1^1	?	y^0
?	x_2^1	?
x_1^0	x_2^0	y^0
?	x_2^1	y^1
x_1^0	x_2^1	?
x_1^1	?	y^0

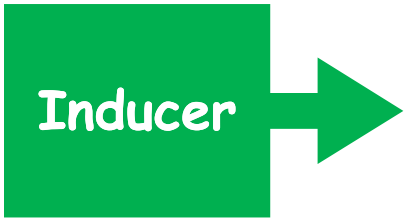
X_1	X_2	$P(Y/X_1,X_2)$ y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

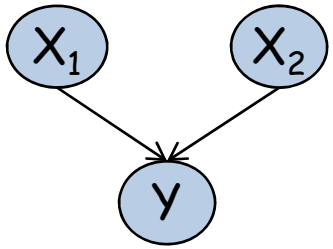
Daphne Koller

Unknown Structure, Incomplete Data

Initial network







Input Data

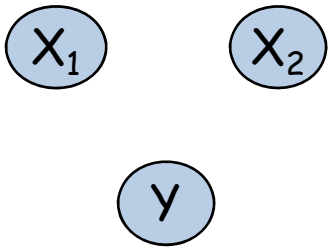
X_1	X_2	Y
?	x_2^1	y^0
x_1^1	?	y^0
?	x_2^1	?
x_1^0	x_2^0	y^0
?	x_2^1	y^1
x_1^0	x_2^1	?
x_1^1	?	y^0

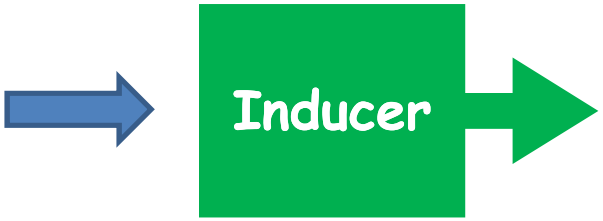
X_1	X_2	$P(Y/X_1,X_2)$	
		y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

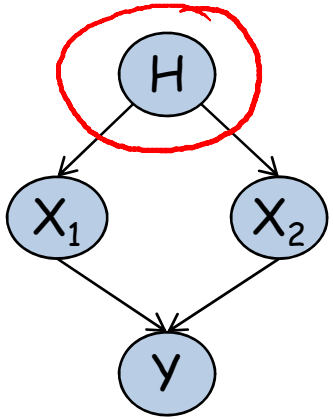
Daphne Koller

Latent Variables, Incomplete Data

Initial network







Input Data

X_1	X_2	Y
?	x_2^1	y^0
x_1^1	?	y^0
?	x_2^1	?
x_1^0	x_2^0	y^0
?	x_2^1	y^1
x_1^0	x_2^1	?
x_1^1	?	y^0

X_1	X_2	$P(Y/X_1,X_2)$ y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

Daphne Koller

PGM Learning Tasks I

- Goal: Answer general probabilistic queries about new instances
- Simple metric: Training set likelihood
  - $P(D : \mathcal{M}) = \Pi_m P(\mathbf{d}[m] : \mathcal{M})$ (IID)
- But we really care about new data
  - Evaluate on test set likelihood – $P(D' : \mathcal{M})$

PGM Learning Tasks II

- Goal: Specific prediction task on new instances
  - Predict target variables **y** from observed variables **x**
  - E.g., image segmentation, speech recognition
- Often care about specialized objective
  - E.g., pixel-level segmentation accuracy
- Often convenient to select model to optimize
  - likelihood Π_m P(**d**[m] : $\mathcal{M}$) or
  - conditional likelihood Π_m P(**y**[m] | **x**[m] : $\mathcal{M}$)
- Model evaluated on "true" objective over test data

PGM Learning Tasks III

- Goal: Knowledge discovery of $\mathcal{M}^*$
  - Distinguish direct vs indirect dependencies
  - Possibly directionality of edges
  - Presence and location of hidden variables
- Often train using likelihood
  - Poor surrogate for structural accuracy
- Evaluate by comparing to prior knowledge

Daphne Koller

Avoiding Overfitting

- Selecting $\mathcal{M}$ to optimize training set likelihood overfits to statistical noise
- Parameter overfitting
  - Parameters fit random noise in training data
  - Use regularization / parameter priors
- Structure overfitting
  - Training likelihood always increases for more complex structures
  - Bound or penalize model complexity

Selecting Hyperparameters

- Regularization for overfitting involves hyperparameters:
  - Parameter priors (regularization)
  - Complexity penalty
- Choice of hyperparameters makes a big difference to performance
- Must be selected on validation set (cross-validation)

Why PGM Learning

- Predictions of structured objects (sequences, graphs, trees)
  - Exploit correlations between several predicted variables
- Can incorporate prior knowledge into model
- Learning single model for multiple tasks
- Framework for knowledge discovery