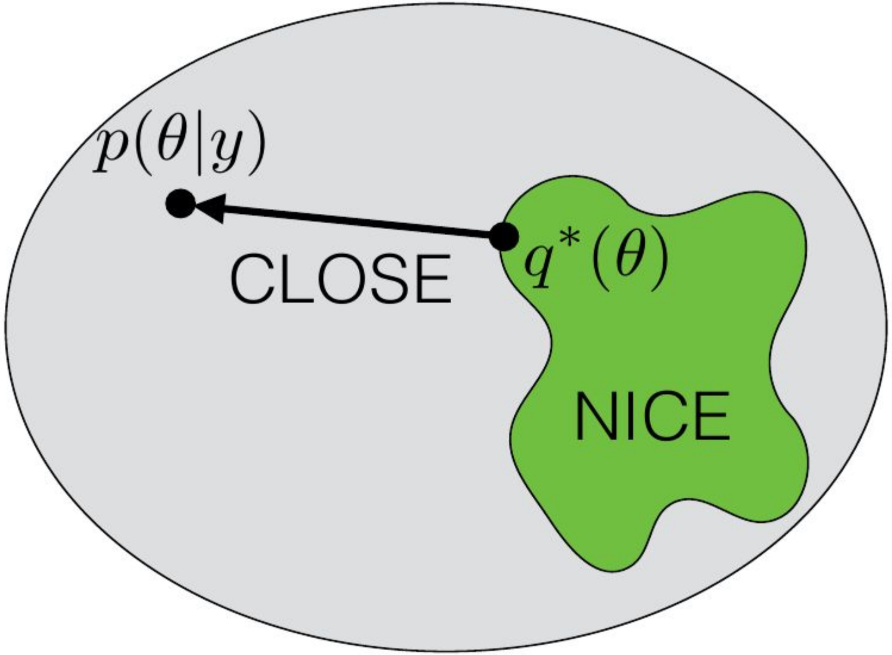
Learning the structure of Probabilistic Graphical Models - introduction

Tomasz Kajdanowicz

The presentation in based on the D. Koller and N. Friedman “Probabilistic graphical model”, chapter 16 and slides material

Approximate Bayesian Inference



Instead: an optimization approach

- Approximate posterior with *q**

$$q^* = \mathrm{argmin}_{q \in Q} f(q(\cdot), p(\cdot|y))$$

- Variational Bayes (VB): *f* is Kullback-Leibler divergence

$$KL(q(\cdot)||p(\cdot|y))$$

q^* - what is its form?

- Selection of exponential distributions
- Mean-field variational Bayes

$$Q_{MFVB} := \left\{ q : q(\theta) = \prod_{j=1}^{J} q_j(\theta_j) \right\}$$

Approximate Bayesian inference

Use q^* to approximate $p(\cdot|y)$

Optimization

$$q^* = \operatorname{argmin}_{q \in Q} f(q(\cdot), p(\cdot|y))$$

Variational Bayes

$$q^* = \operatorname{argmin}_{q \in Q} KL(q(\cdot)||p(\cdot|y))$$

Mean-field variational Bayes

$$q^* = \operatorname{argmin}_{q \in Q_{\mathrm{MFVB}}} KL(q(\cdot)||p(\cdot|y))$$

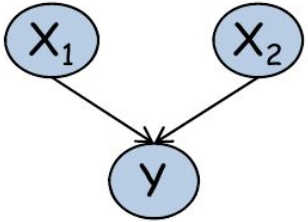
- Coordinate descent
- Stochastic variational inference (SVI) [Hoffman et al 2013]
- Automatic differentiation variational inference (ADVI) [Kucukelbir et al 2015, 2017]

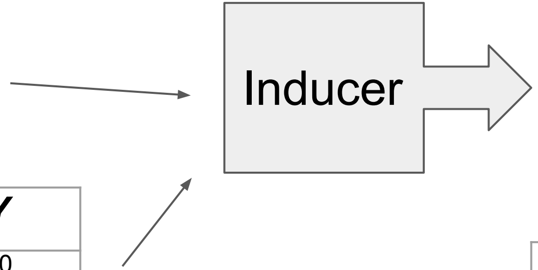
How to build underlying structure of the model?

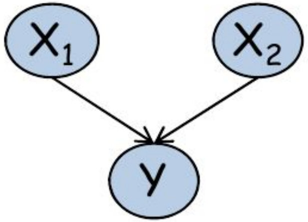
- Known Structure, Complete Data
- Unknown Structure, Complete Data
- Known Structure, Incomplete Data
- Unknown Structure, Incomplete Data
- Latent Variables, Incomplete Data

Known Structure, Complete Data

Initial network







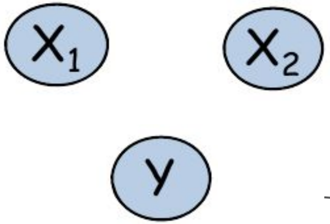
Input data

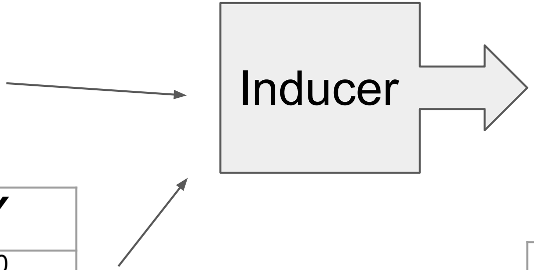
X_1	X_2	Y
x_1^0	x_2^1	y^0
x_1^1	x_2^0	y^0
x_1^0	x_2^1	y^1
x_1^0	x_2^0	y^0
x_1^1	x_2^1	y^1
x_1^0	x_2^1	y^1
x_1^1	x_2^0	y^0

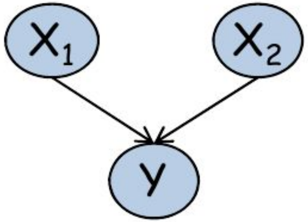
X_1	X_2	$P(Y\|X_1,X_2)$	
		y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

Unknown Structure, Complete Data

Initial network







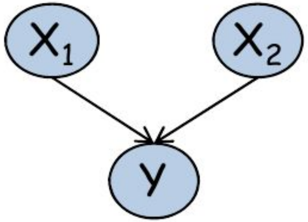
Input data

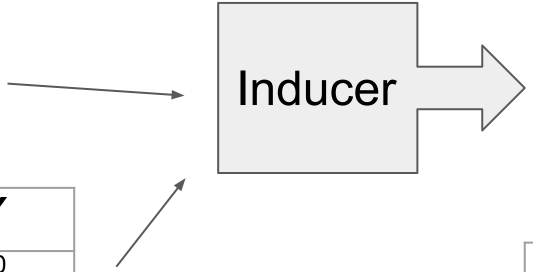
X_1	***X_2***	***Y***
x_1^0	x_2^1	y^0
x_1^1	x_2^0	y^0
x_1^0	x_2^1	y^1
x_1^0	x_2^0	y^0
x_1^1	x_2^1	y^1
x_1^0	x_2^1	y^1
x_1^1	x_2^0	y^0

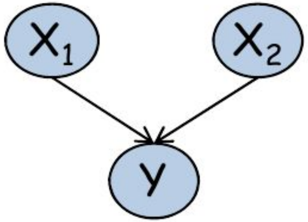
X_1	***X_2***	***$P(Y\|X_1,X_2)$***	
		y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

Known Structure, Incomplete Data

Initial network







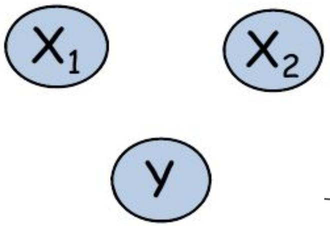
Input data

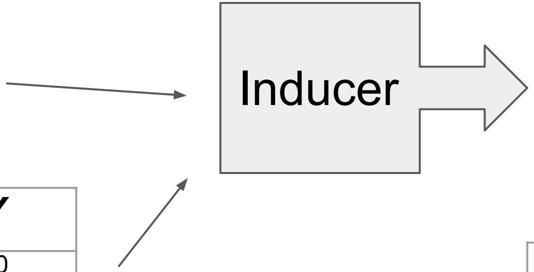
X_1	X_2	Y
?	x_2^1	y^0
x_1^1	?	y^0
?	x_2^1	?
x_1^0	x_2^0	y^0
?	x_2^1	y^1
x_1^0	x_2^1	?
x_1^1	?	y^0

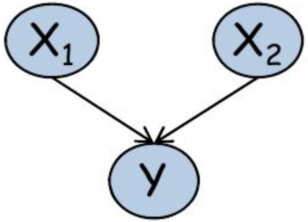
X_1	X_2	$P(Y\|X_1,X_2)$	
		y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

Unknown Structure, Incomplete Data

Initial network







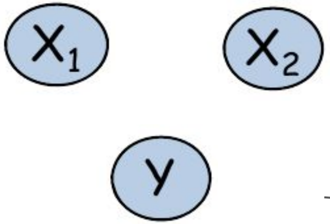
Input data

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

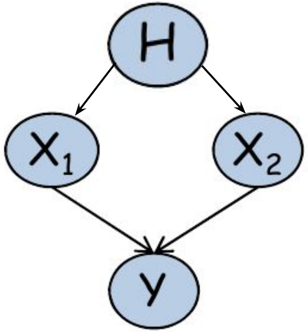
X_1	X_2	$P(Y \mid X_1,X_2)$	
		y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

Latent Variables, Incomplete Data

Initial network



Inducer



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	$P(Y \mid X_1,X_2)$	
		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

Why do we learn the models at all? (I)

- Goal: Answer general probabilistic queries about new instances
- Simple metric: Training set likelihood on model *M*
  - $P(D \mid M) = \Pi_m P(d[m] \mid M)$
- But we really care about new data?
  - Evaluate on test set likelihood $P(D'|M)$

Why do we learn the models at all? (II)

- Goal: Specific prediction task on new instances
  - Predict target variables **y** from observed variables **x**
  - care about specialized objective
  - often convenient to select model *M* to optimize
    - likelihood Π_m P(d[m] | *M*)
    - conditional likelihood Π_m P(y[m] | x[m] , *M*)
- Model evaluated on “true” objective over test data

Why do we learn the models at all? (III)

- Goal: Knowledge discovery of 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

Overfitting

- Selecting 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

Regularization

- Bayesian learning uses of a prior probability
  - lowers probability to more complex models
- model selection techniques
  - Akaike information criterion (AIC)
  - minimum description length (MDL)
  - Bayesian information criterion(BIC)
- Alternative of controlling overfitting without regularization: cross-validation.

sample size

AIC

k - the number of estimated parameters in the model

$\hat{L}$ - the maximum value of the likelihood function for the model

$$AIC = 2k - 2ln(\hat{L})$$

BIC

k - the number of estimated parameters in the model

$\hat{L}$ - the maximum value of the likelihood function for the model

n - sample size

$$BIC = \ln(n)k - 2\ln(\hat{L})$$

Selecting Hyperparameters

- Regularization for overfitting involves hyperparameters
  - parameter priors
  - complexity penalty
- Choice of hyperparameters makes a big difference to performance
- Must be selected on validation set