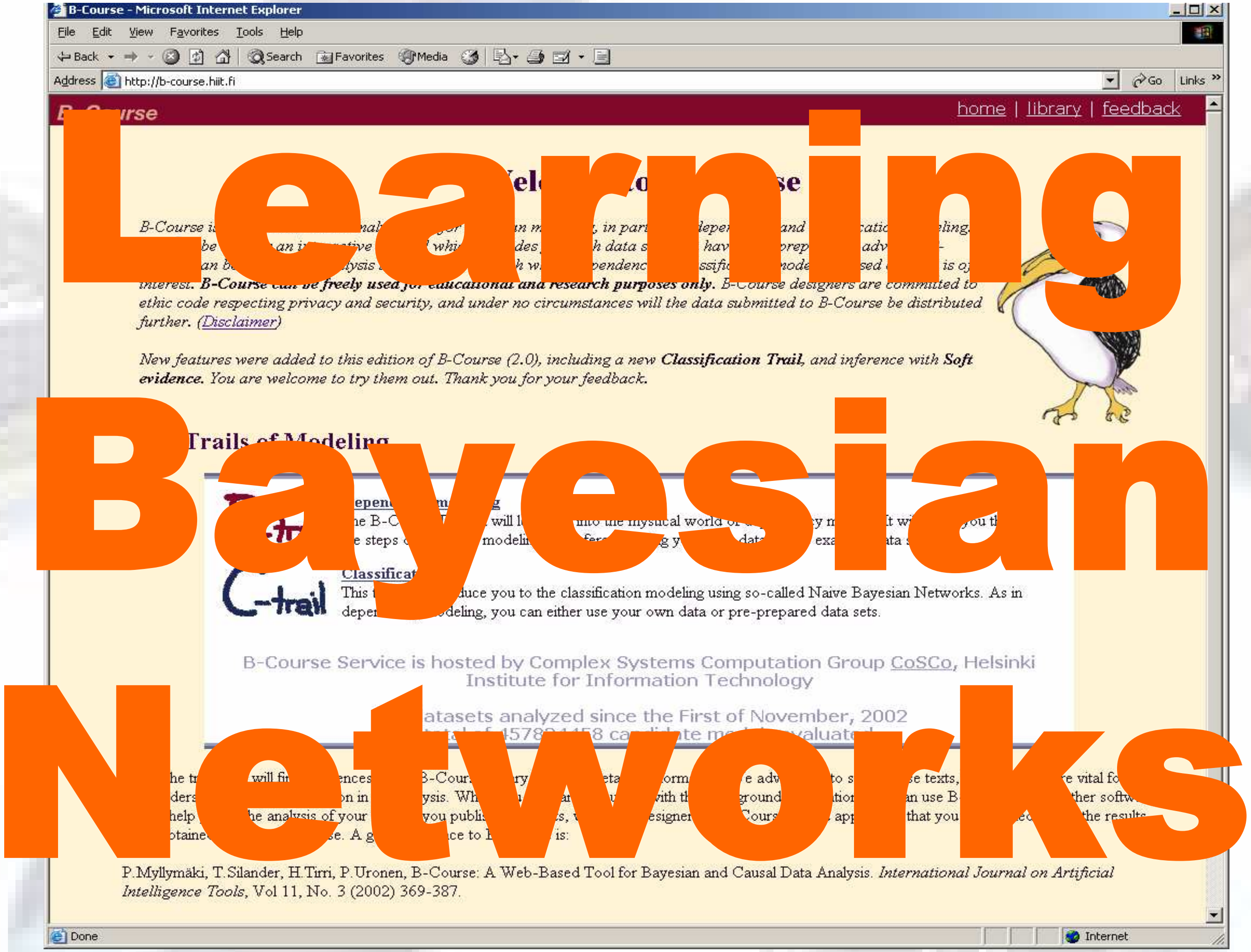
Learning Bayesian Networks


B-Course - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://b-course.hiit.fi
Go Links
B-Course
home | library | feedback
Done
Internet

interest. ***B-Course can be freely used for educational and research purposes only.*** *B-Course designers are committed to ethic code respecting privacy and security, and under no circumstances will the data submitted to B-Course be distributed further. (Disclaimer)*

New features were added to this edition of B-Course (2.0), including a new ***Classification Trail****, and inference with* ***Soft evidence****. You are welcome to try them out. Thank you for your feedback.*

Trails of Modeling

B-Course Service is hosted by Complex Systems Computation Group CoSCo, Helsinki Institute for Information Technology

P.Myllymäki, T.Silander, H.Tirri, P.Uronen, B-Course: A Web-Based Tool for Bayesian and Causal Data Analysis. *International Journal on Artificial Intelligence Tools*, Vol 11, No. 3 (2002) 369-387.

Learning Bayesian Networks

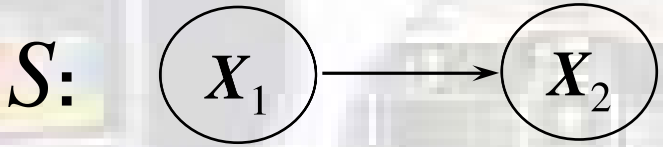
$$p(\mathbf{x}|\theta_s, S^h) = \prod_{i=1}^{n} p(x_i|\mathbf{pa}_i, \theta_i, S^h)$$

joint distribution function

local distribution functions

S^h **is the hypothesis that structure** S **(minimal) encodes the joint distribution function**

Example



$$p(\bar{x}_1, x_2 | \theta_s, S^h) = p(\bar{x}_1 | \theta_1, S^h)\, p(x_2 | \bar{x}_1, \theta_{2|\bar{1}}, S^h)$$

$$= (1 - \theta_1)\, \theta_{2|\bar{1}}$$

$$p(x_1, x_2 | \theta_s, S^h) = p(x_1 | \theta_1, S^h)\, p(x_2 | x_1, \theta_{2|1}, S^h)$$

$$= \theta_1\, \theta_{2|1}$$

$$\theta_s = \{\theta_1, \theta_{2|1}, \theta_{2|\bar{1}}\}$$

Updating or "Learning" Parameters (S^h is certain, θ_s is uncertain)

Let $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be a random sample from $p(\mathbf{x} | \theta_s, S^h)$.

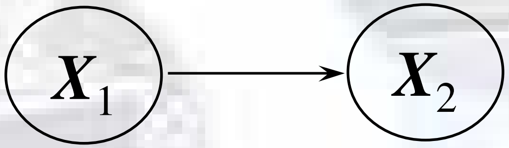
$$p(\theta_s | D, S^h) \propto p(\theta_s | S^h)\ p(D | \theta_s, S^h)$$

$$= p(\theta_s | S^h) \prod_{l=1}^{m} p(\mathbf{x}_l | \theta_s, S^h)$$

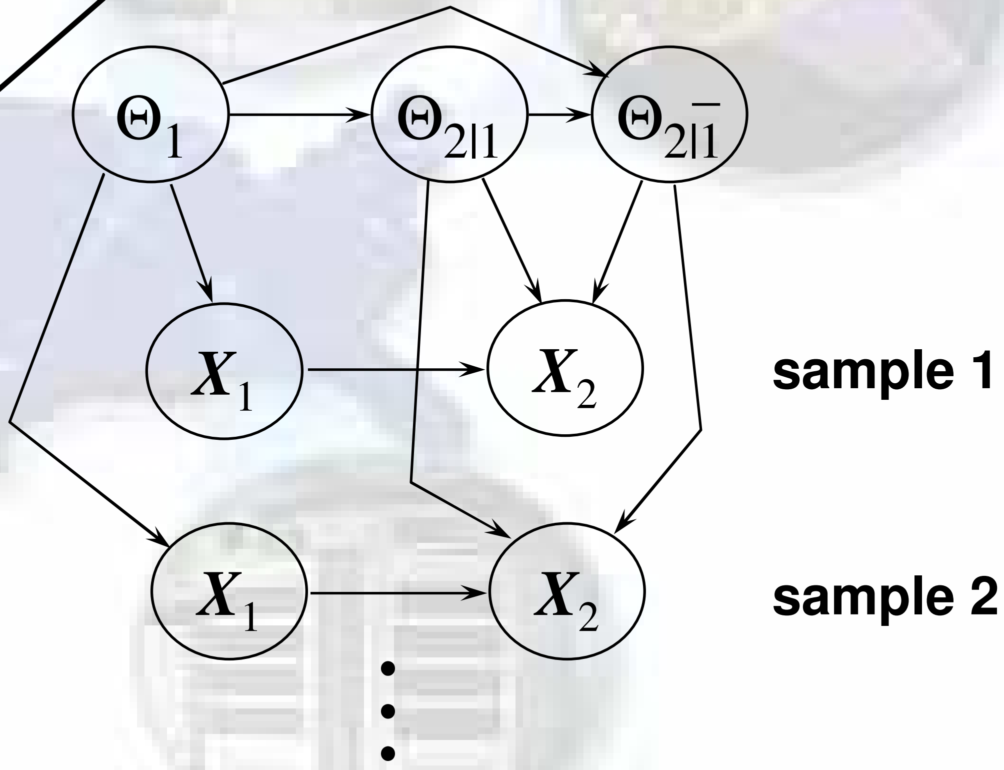
$$P(x_{m+1} \mid D, S^h) = \int P(x_{m+1} \mid \theta_s, D, S^h) P(\theta_s \mid D, S^h) d\theta_s$$

Example

S:



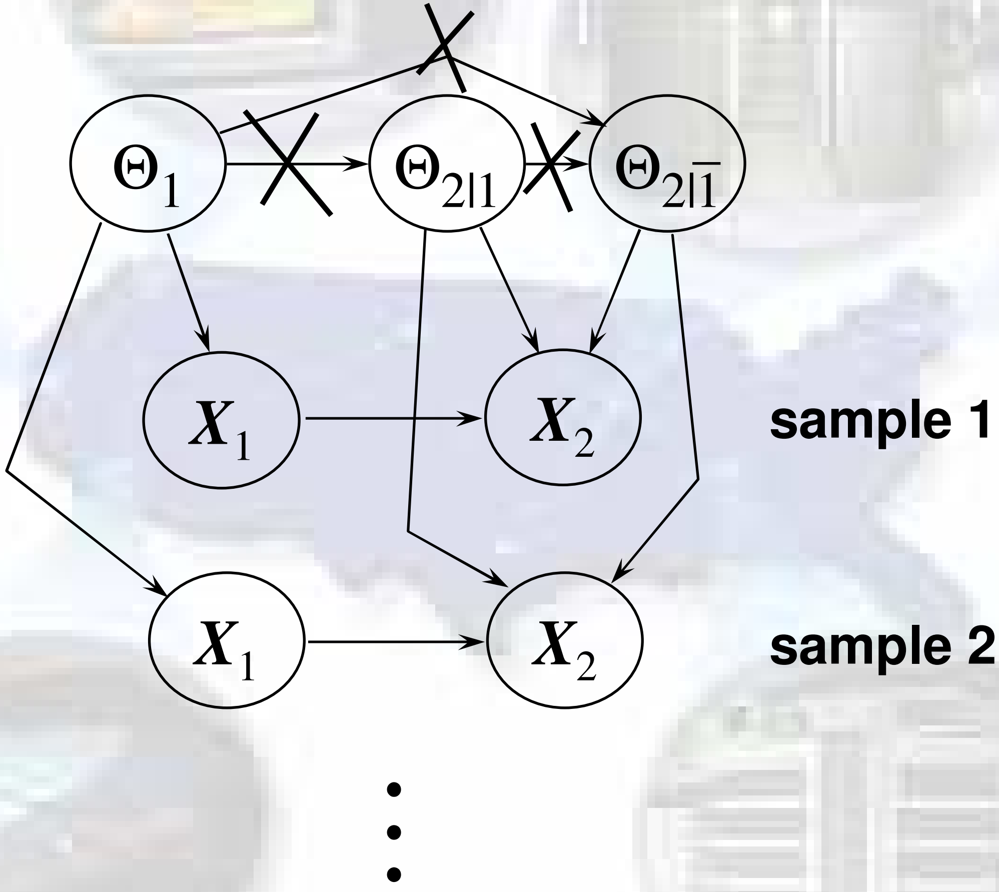
$$\theta_s = \{\theta_1, \theta_{2|1}, \theta_{2|\bar{1}}\}$$



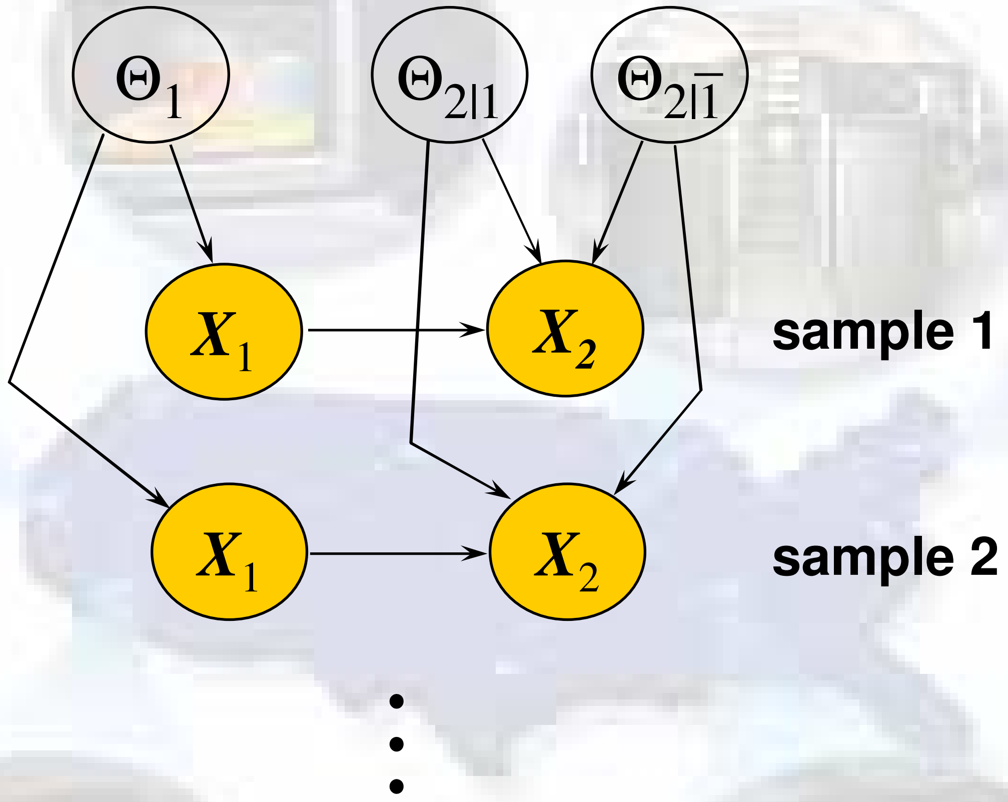
Exact computation of $p(\theta_s|D,S^h)$

- No missing data, no hidden variables
- Independent parameters $$p(\theta_s | S^h) = \prod_{i=1}^{n} p(\theta_i | S^h)$$
- Local distribution functions from the exponential family, conjugate priors

Parameter Independence



No Missing Data



sample 1

sample 2

Each parameter can be updated independently….

$$P(\theta_{2|1} \mid N_{21}, N_{2\bar{1}}, S^h) \propto p(\theta_{2|1})\theta_{2|1}^{N_{21}}(1-\theta_{2|1})^{N_{2\bar{1}}}$$

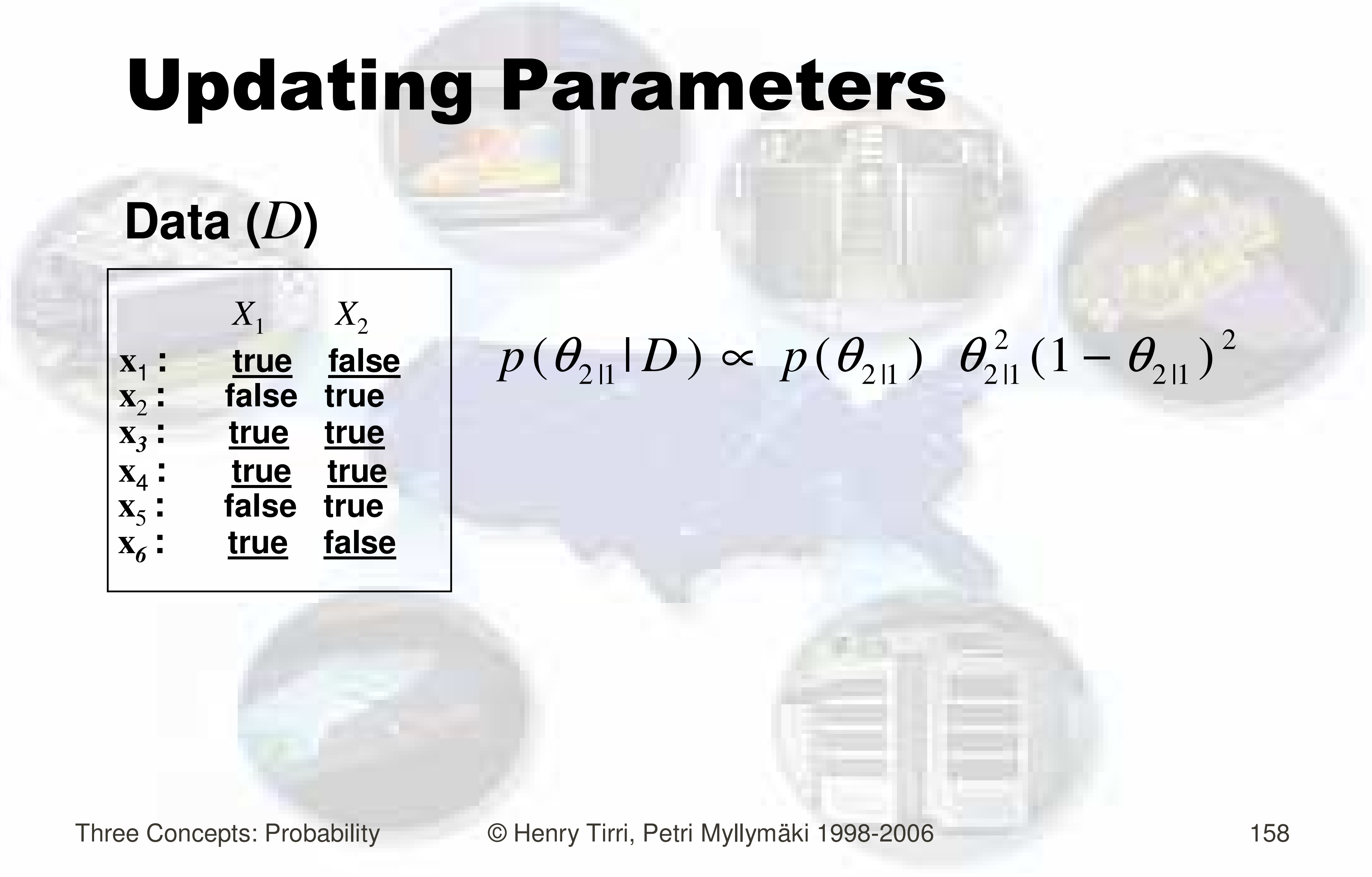
Updating Parameters

Data (D)

	X_1	X_2
$\mathbf{x}_1$:	true	false
$\mathbf{x}_2$:	false	true
$\mathbf{x}_3$:	true	true
$\mathbf{x}_4$:	true	true
$\mathbf{x}_5$:	false	true
$\mathbf{x}_6$:	true	false

$$p(\theta_1 | D) \propto p(\theta_1) \;\; \theta_1^4 (1 - \theta_1 \;)^2$$

Updating Parameters



Data (D)

	X_1	X_2
$\mathbf{x}_1$:	true	false
$\mathbf{x}_2$:	false	true
$\mathbf{x}_3$:	true	true
$\mathbf{x}_4$:	true	true
$\mathbf{x}_5$:	false	true
$\mathbf{x}_6$:	true	false

$$p(\theta_{2|1} \mid D) \propto p(\theta_{2|1}) \;\; \theta_{2|1}^{2} (1 - \theta_{2|1})^{2}$$

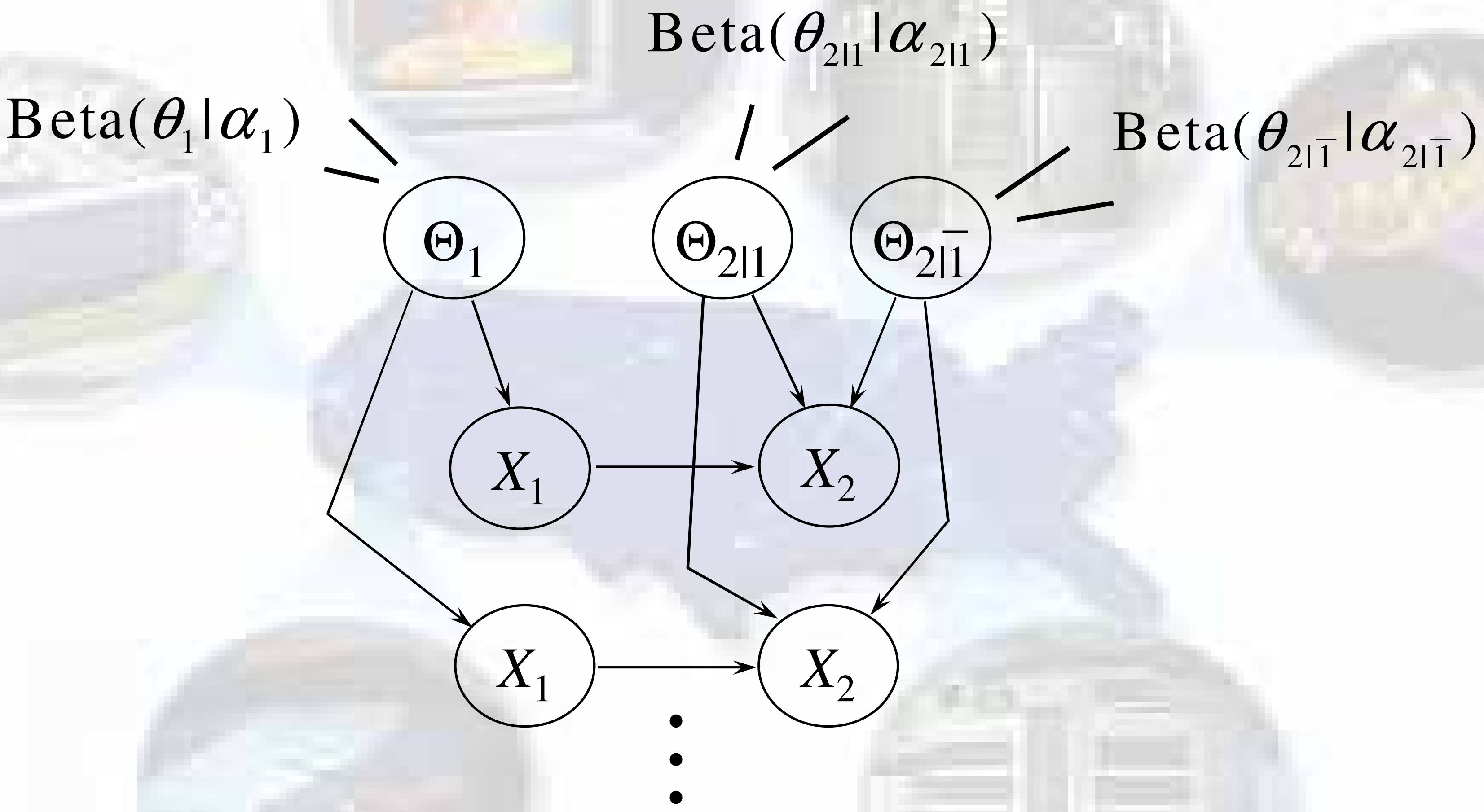
Updating Parameters

Data (D)

	X_1	X_2
$\mathbf{x}_1$:	true	false
$\mathbf{x}_2$:	false	true
$\mathbf{x}_3$:	true	true
$\mathbf{x}_4$:	true	true
$\mathbf{x}_5$:	false	true
$\mathbf{x}_6$:	true	false

$$p(\theta_{2|\bar{1}}|D) \propto p(\theta_{2|\bar{1}})\ \theta_{2|\bar{1}}^{2}(1-\theta_{2|\bar{1}})^{0}$$

Conjugate Priors


Beta($\theta_{2|1}|\alpha_{2|1}$)
Beta($\theta_1|\alpha_1$)
Beta($\theta_{2|\bar{1}}|\alpha_{2|\bar{1}}$)
Θ_1
$\Theta_{2|1}$
$\Theta_{2|\bar{1}}$
X_1
X_2
X_1
X_2

Missing Data

Data

	X_1	X_2
$\mathbf{x}_1$ **:**	**?**	**true**

Beta Beta

$$p(\theta_{2|1} | x_2) = p(\theta_{2|1} | x_1, x_2)\, p(x_1 | x_2) + p(\theta_{2|1} | \overline{x}_1, x_2)\, p(\overline{x}_1 | x_2)$$

Mixing coefficients

For real problems, exact calculations are intractable.

Approximate computation of $p(\theta_s|D,S^h)$

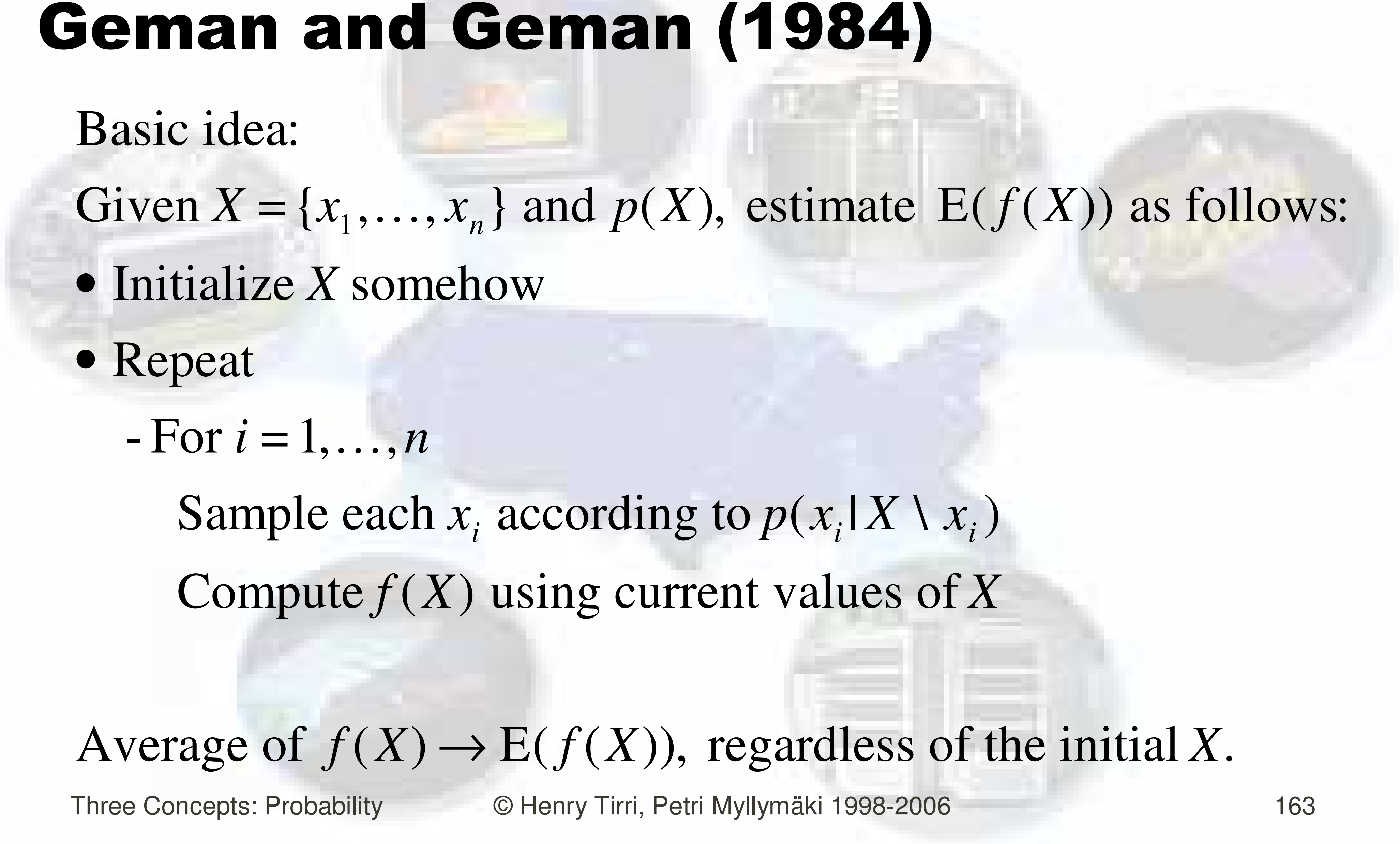
- Monte Carlo: Gibbs sampling, importance sampling (e.g., Neal 93)

- Gaussian approximation (e.g., Kass et al. 88)

$$p(\theta_s \mid S, S^h) \xrightarrow{m\to\infty} c\,p(D \mid \tilde{\theta}_s, S^h)\,e^{-\frac{1}{2}(\theta_s-\tilde{\theta}_s)^t A(\theta_s-\tilde{\theta}_s)}$$

- MAP: $\tilde{\theta}_s$

Gibbs Sampling
Geman and Geman (1984)



Basic idea:

Given $X = \{x_1, \ldots, x_n\}$ and $p(X)$, estimate $\mathrm{E}(f(X))$ as follows:

- Initialize X somehow
- Repeat

```
- For $i = 1, \ldots, n$
    Sample each $x_i$ according to $p(x_i | X \setminus x_i)$
    Compute $f(X)$ using current values of $X$
```

Average of $f(X) \rightarrow \mathrm{E}(f(X))$, regardless of the initial X.

Using Gibbs Sampling to Approximate $p(\theta|D,S^h)$

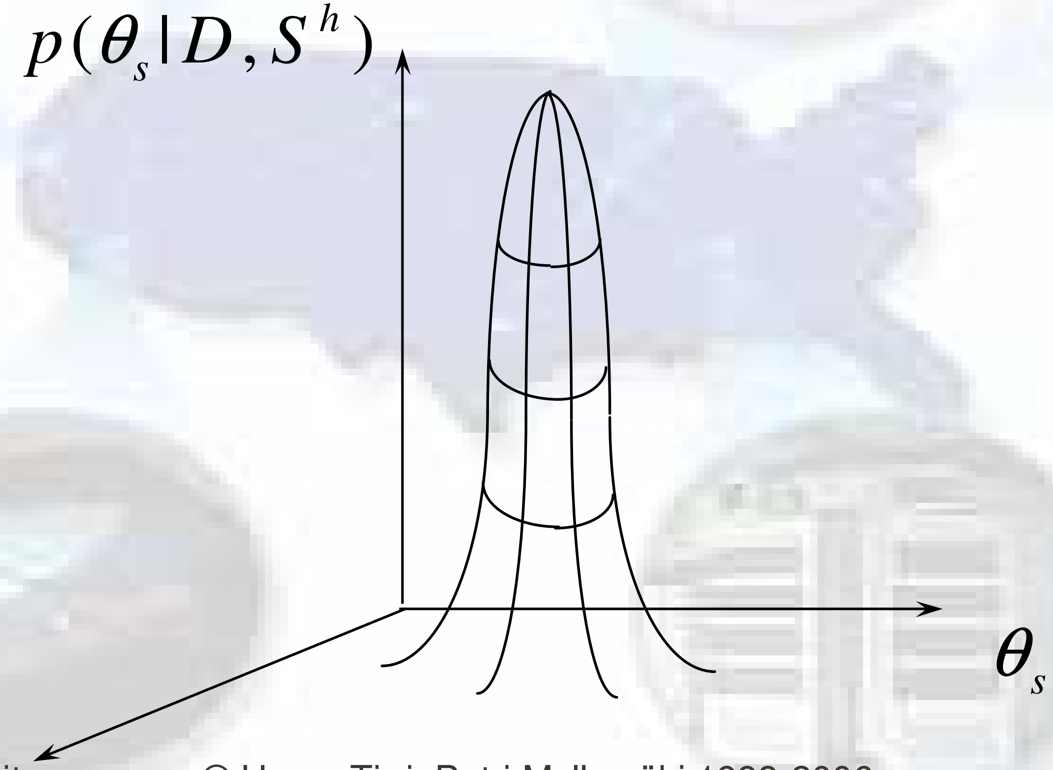
- Initialize θ_s
- Repeat
  - Sample missing data (given θ_s and D_l) producing D_l^c
  - Compute $p(\theta_s | D_l^c, S^h)$
  - Sample θ_s from $p(\theta_s | D_l^c, S^h)$

Average of $p(\theta_s | D_l^c, S^h) \rightarrow p(\theta_s | D_l, S^h)$

Gaussian Approximation
Tierney and Kadane (1986), Kass et al. (1988)

Basic idea:

for large data sets, $p(\theta_s | D, S^h)$ will often be ~Gaussian



Gaussian Approximation

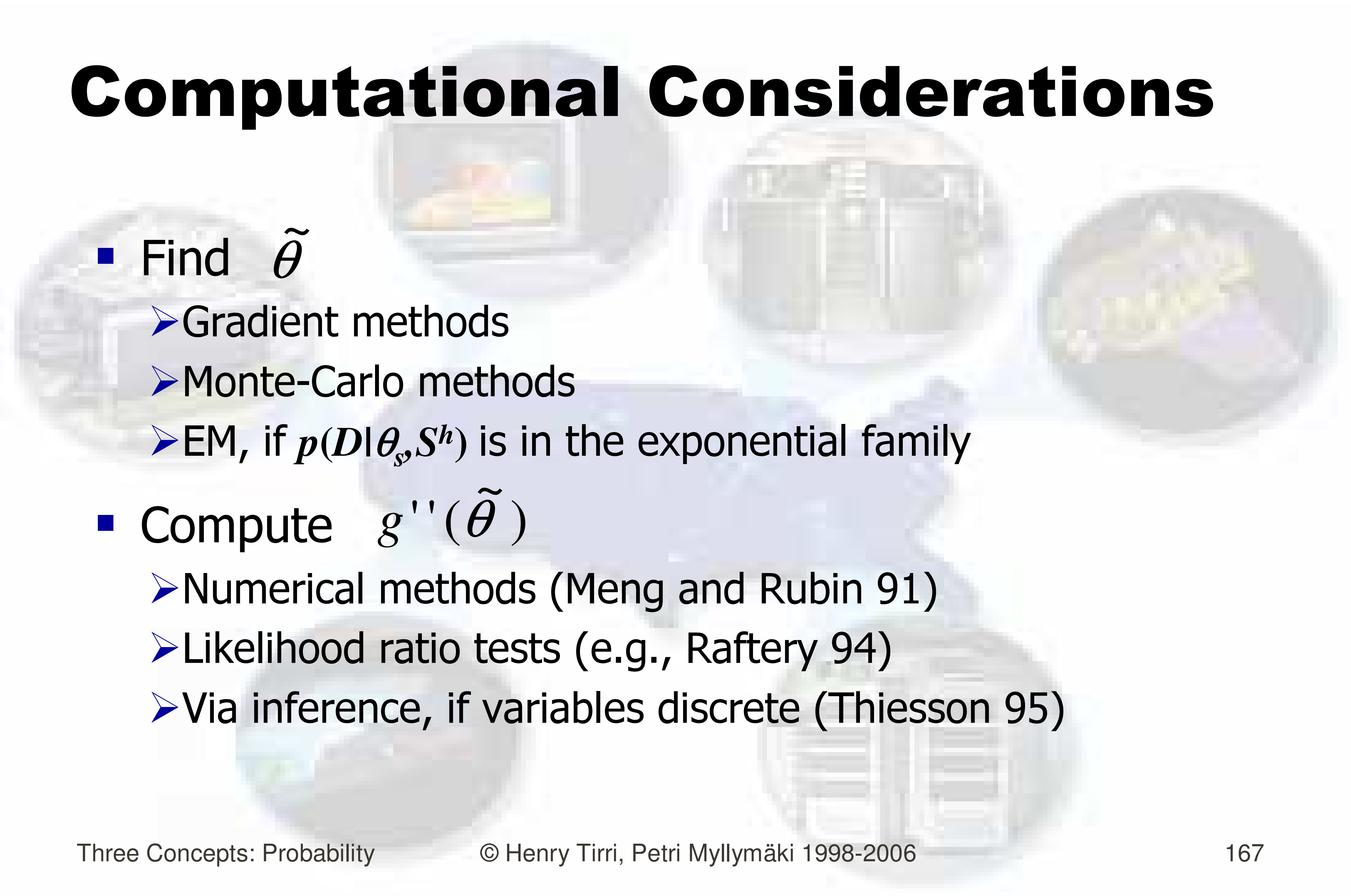
$$g(\theta) \equiv \log p(D|\theta)\, p(\theta)$$

Expand $g(\theta)$ about its maximum $g(\tilde{\theta})$:

$$g(\theta) \approx g(\tilde{\theta}) + -\frac{1}{2}(\theta - \tilde{\theta})^T A(\theta - \tilde{\theta}) \quad (A = -g''(\tilde{\theta}))$$

$$p(D|\theta)\, p(\theta) \approx p(D|\tilde{\theta})\; p(\tilde{\theta})\; e^{-\frac{1}{2}(\theta-\tilde{\theta})^T A(\theta-\tilde{\theta})}$$

Computational Considerations



- Find $\tilde{\theta}$
  - Gradient methods
  - Monte-Carlo methods
  - EM, if $p(D|\theta_s,S^h)$ is in the exponential family
- Compute $g''(\tilde{\theta})$
  - Numerical methods (Meng and Rubin 91)
  - Likelihood ratio tests (e.g., Raftery 94)
  - Via inference, if variables discrete (Thiesson 95)

Expectation-Maximization (EM) Algorithm (Dempster et al. (1977))

- Initialize parameters
- Expectation step: compute the expected sufficient statistics

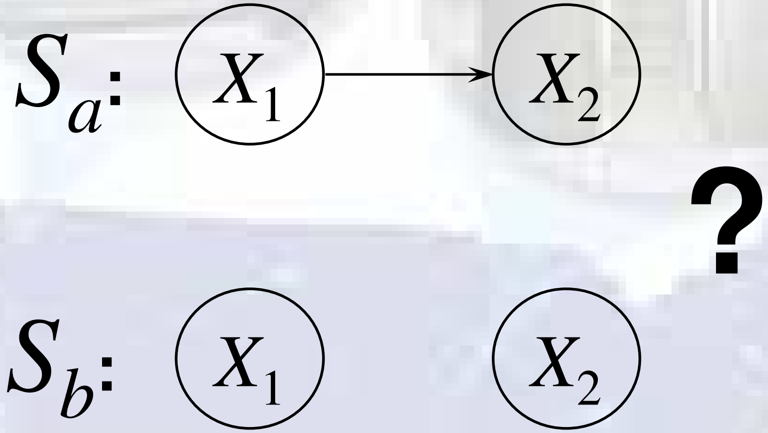
$$E(N_{12}|\theta_s) = \sum_{l=1}^{N} p(x_1, x_2|\mathbf{x}_l, \theta_s)$$

- Maximization step: choose parameters so as to maximize their posterior probability given the expected sufficient statistics

$$\theta_{2|1} := \frac{E(N_{21}|\theta_s) + \alpha_{21} - 1}{E(N_{21}|\theta_s) + \alpha_{21} - 1 + E(N_{2\bar{1}}|\theta_s) + \alpha_{2\bar{1}} - 1}$$

- Iterate. Parameters will converge to a local MAP value

Learning Structure (both S^h and θ_s are uncertain)



$$p(x_{m+1} \mid D) = p(x_{m+1} \mid S_a^h, D)\, p(S_a^h \mid D) + p(x_{m+1} \mid S_b^h, D)\, p(S_b^h \mid D)$$

model averaging

Learning Structure (both S^h and θ_s are uncertain)

$$p(S^h|D) \propto p(S^h)\ p(D|S^h)$$

marginal likelihood

$$= p(S^h) \int p(D|\theta_s, S^h)\ p(\theta_s|S^h)\ d\theta_s$$

Parameters updated as before

Model Selection

- The number of possible structures for a given domain is more than exponential in the number of variables
- Solution: Use only one or a handful of models
- Necessary components:
  - Search method
  - Scoring method

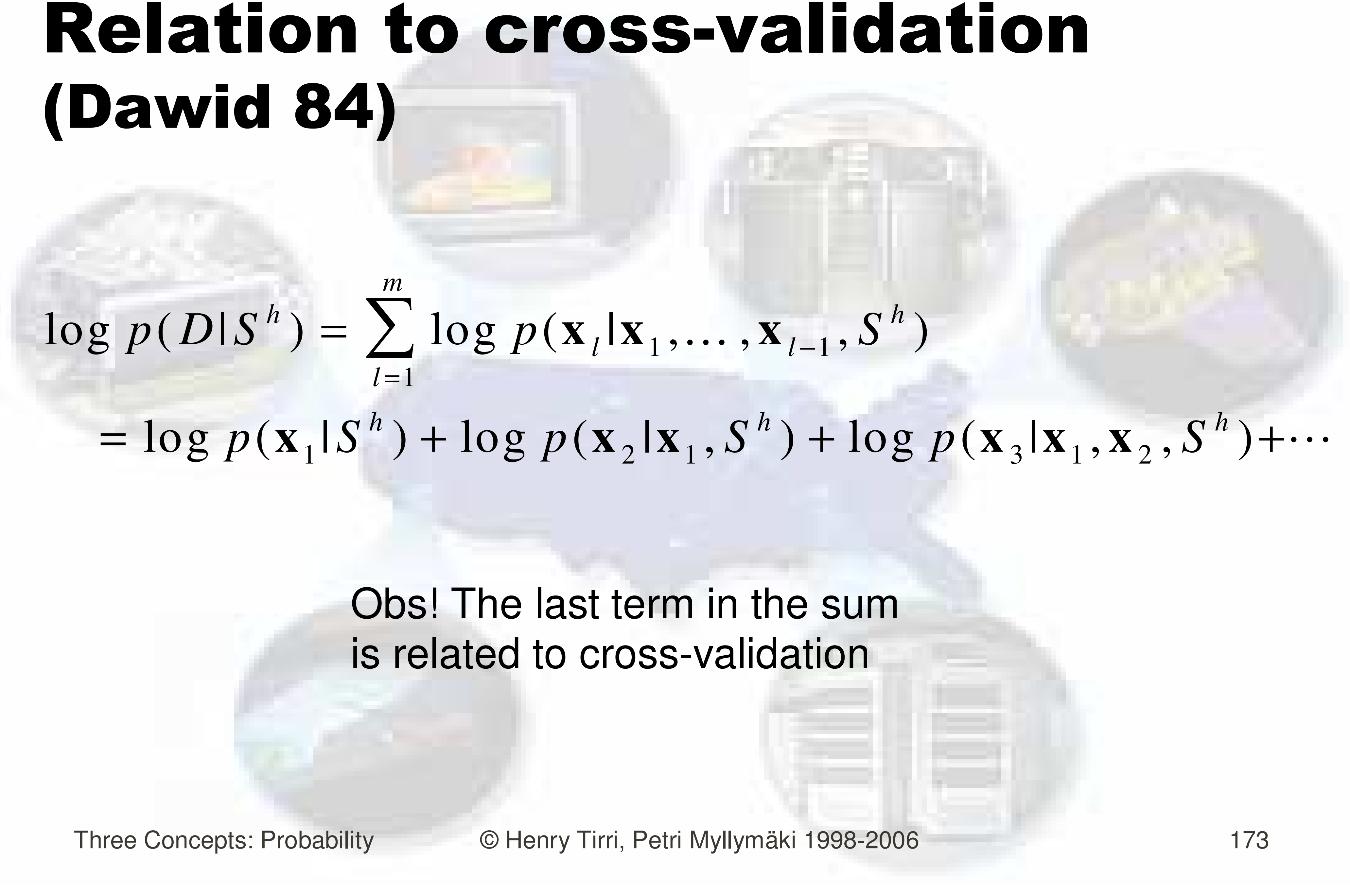
One Reasonable Score: Posterior Probability of a Structure

$$p(S^h|D) \propto p(S^h)\; p(D|S^h)$$

$$= p(S^h) \int p(D|\theta_s, S^h)\; p(\theta_s|S^h)\, d\theta_s$$

structure prior **likelihood** **parameter prior**

Relation to cross-validation (Dawid 84)

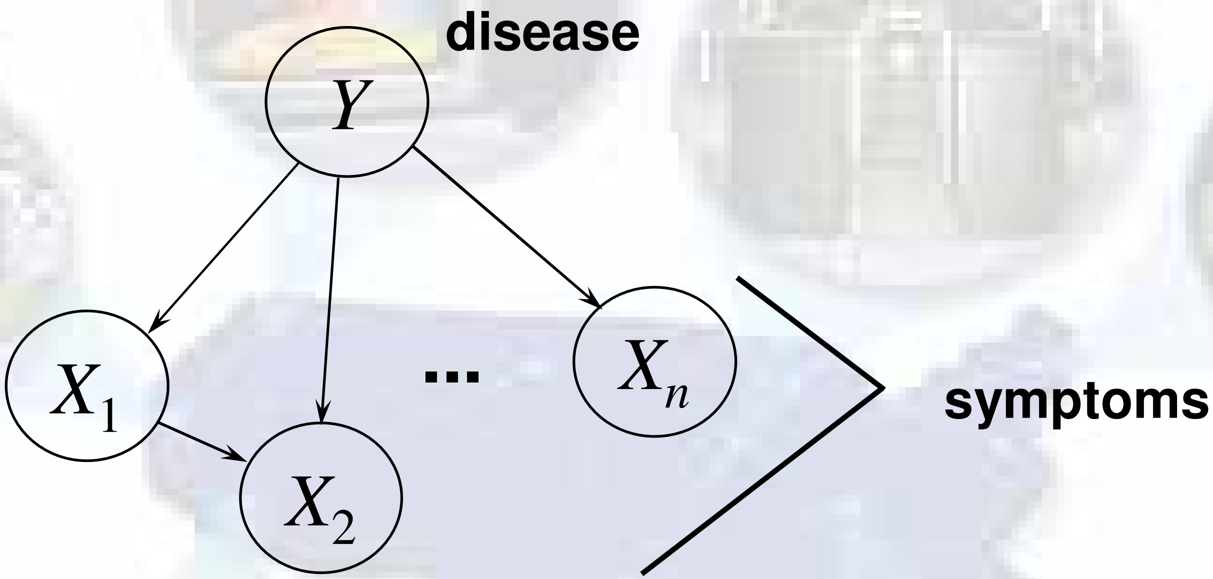


$$\log p(D|S^h) = \sum_{l=1}^{m} \log p(\mathbf{x}_l|\mathbf{x}_1,\ldots,\mathbf{x}_{l-1},S^h)$$

$$= \log p(\mathbf{x}_1|S^h) + \log p(\mathbf{x}_2|\mathbf{x}_1,S^h) + \log p(\mathbf{x}_3|\mathbf{x}_1,\mathbf{x}_2,S^h) + \cdots$$

Obs! The last term in the sum is related to cross-validation

Predictive Score
Spiegelhalter et al. (1993)

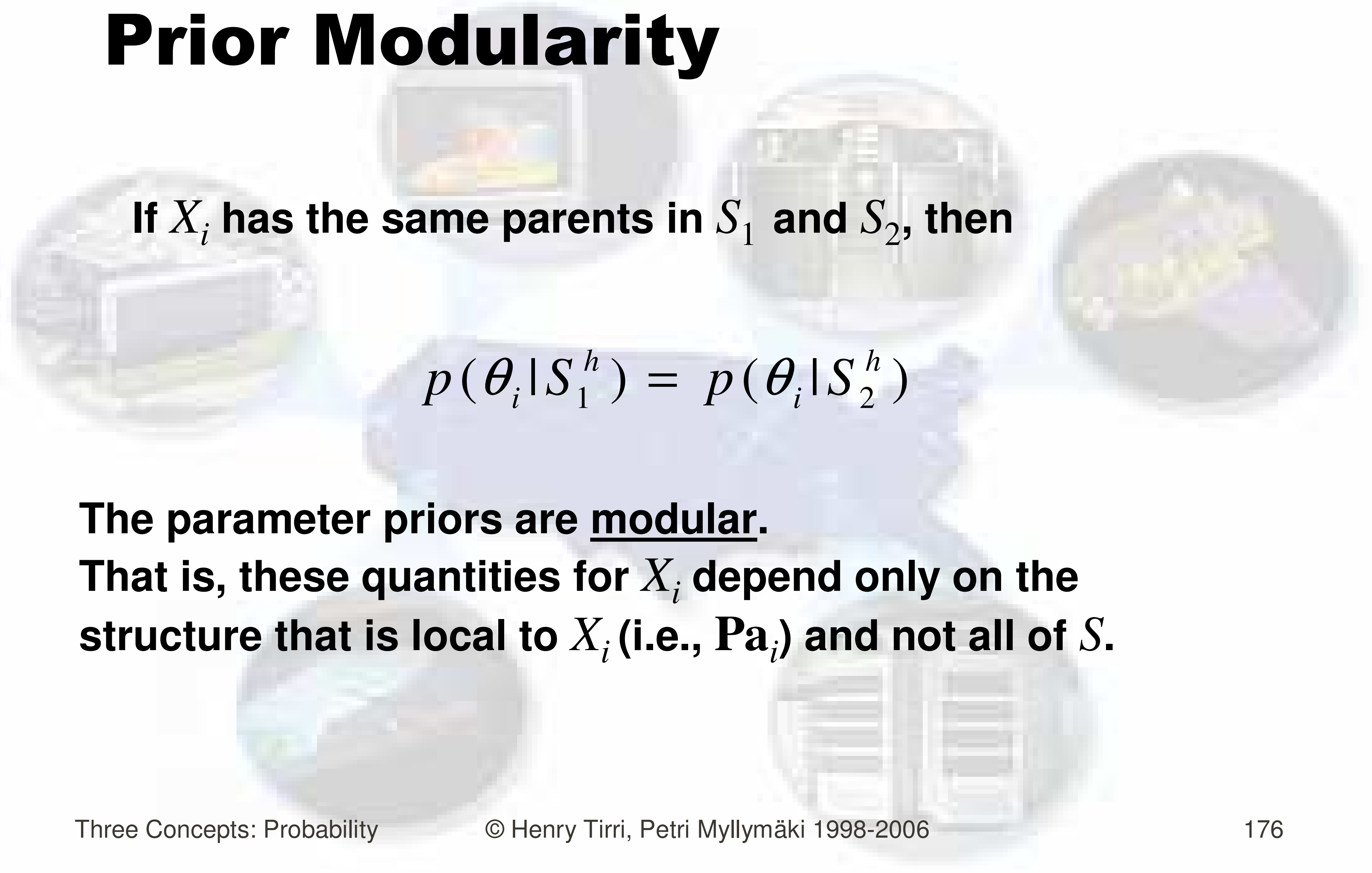


$$\mathrm{pred}(S^h) = \sum_{l=1}^{m} \log p(y_l | \mathbf{x}_l, \{y_1, \mathbf{x}_1\}, \ldots, \{y_{l-1}, \mathbf{x}_{l-1}\}, S^h)$$

Exact computation of $p(D|S^h)$

- No missing data, no hidden variables
- Independent parameters $$p(\theta_s|S^h) = \prod_{i=1}^{n} p(\theta_i|S^h)$$
- Local distribution functions from the exponential family, conjugate priors
- (Prior modularity)

Prior Modularity

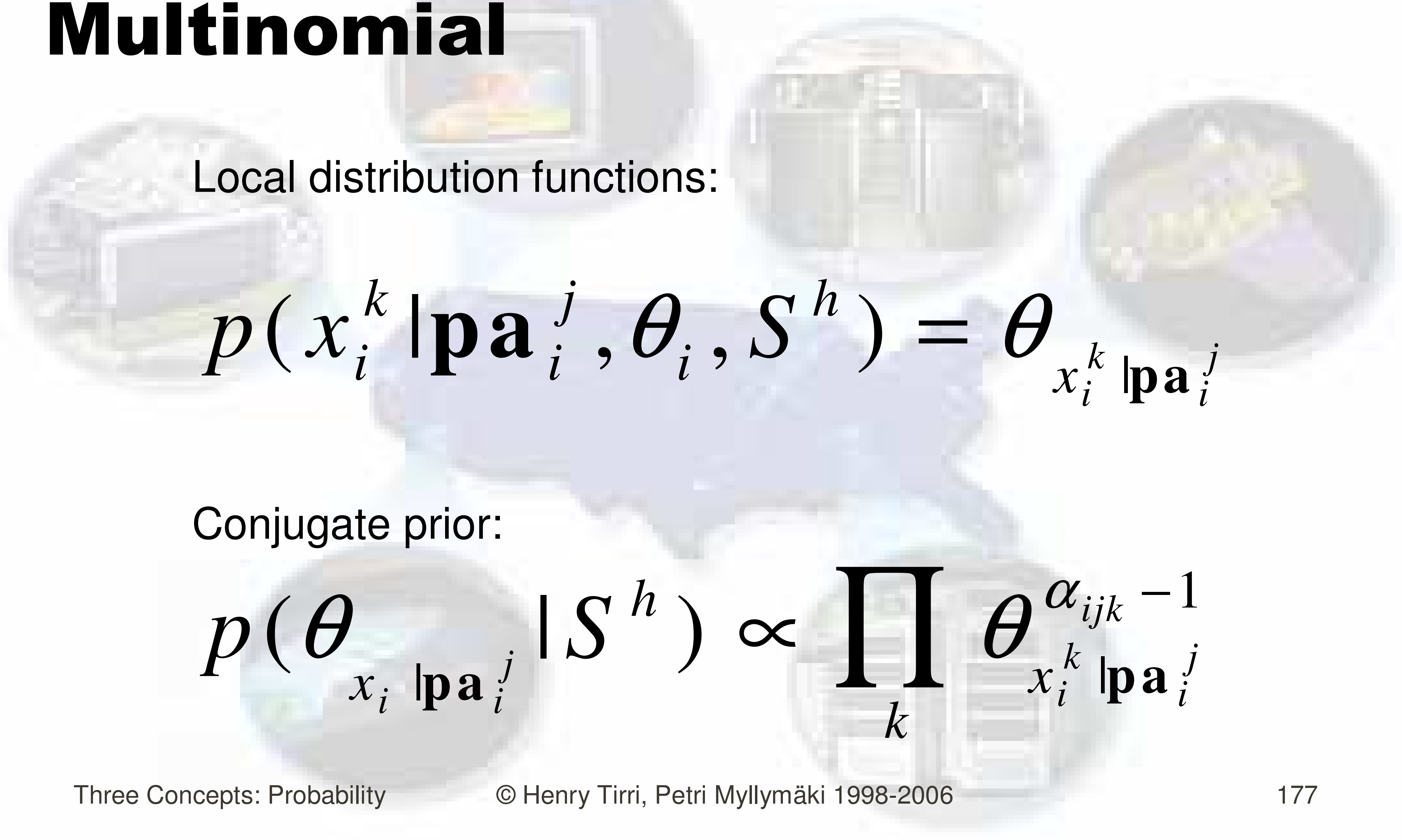


If X_i has the same parents in S_1 and S_2, then

$$p(\theta_i | S_1^h) = p(\theta_i | S_2^h)$$

The parameter priors are modular.
That is, these quantities for X_i depend only on the structure that is local to X_i (i.e., $\mathbf{Pa}_i$) and not all of S.

Example: All Local Distributions Multinomial



Local distribution functions:

$$p(x_i^k \mid \mathbf{pa}_i^j, \theta_i, S^h) = \theta_{x_i^k \mid \mathbf{pa}_i^j}$$

Conjugate prior:

$$p(\theta_{x_i \mid \mathbf{pa}_i^j} \mid S^h) \propto \prod_k \theta_{x_i^k \mid \mathbf{pa}_i^j}^{\alpha_{ijk}-1}$$

Bayesian Dirichlet Score
Cooper and Herskovits (1991)

$$p(D|S^h) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

N_{ijk}: # cases where $X_i = x_{\mathrm{i}}^{\mathrm{k}}$ and $\mathbf{Pa}_i = \mathbf{pa}_i^j$

r_i: number of states of X_i

q_i: number of instances of parents of X_i

$$\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk} \qquad N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$$

Approximations for $p(D|S^h)$

- Monte Carlo
- Gaussian approximation

$$\log p(D|S^h) = \log p(D|S^h, \tilde{\theta}_s) + \log p(\tilde{\theta}_s|S^h)$$
$$+ (d/2)\log(2\pi) - (1/2)\log|A| + O(m^{-1})$$

- Bayesian Information Criterion (BIC)

$$\log p(D|S^h) = \log p(D|S^h, \hat{\theta}_s) - (d/2)\log(m) + O(1)$$

Bayesian Information Criterion (BIC) Schwarz (1978)

$$\log p(D|S^h) = \log p(D|S^h, \hat{\theta}_s) - \frac{d}{2} \log m$$

- No priors needed
- BIC = - *approximation* of stochastic complexity (Rissanen 87)

Another Approximation
Cheeseman & Stutz (1995)

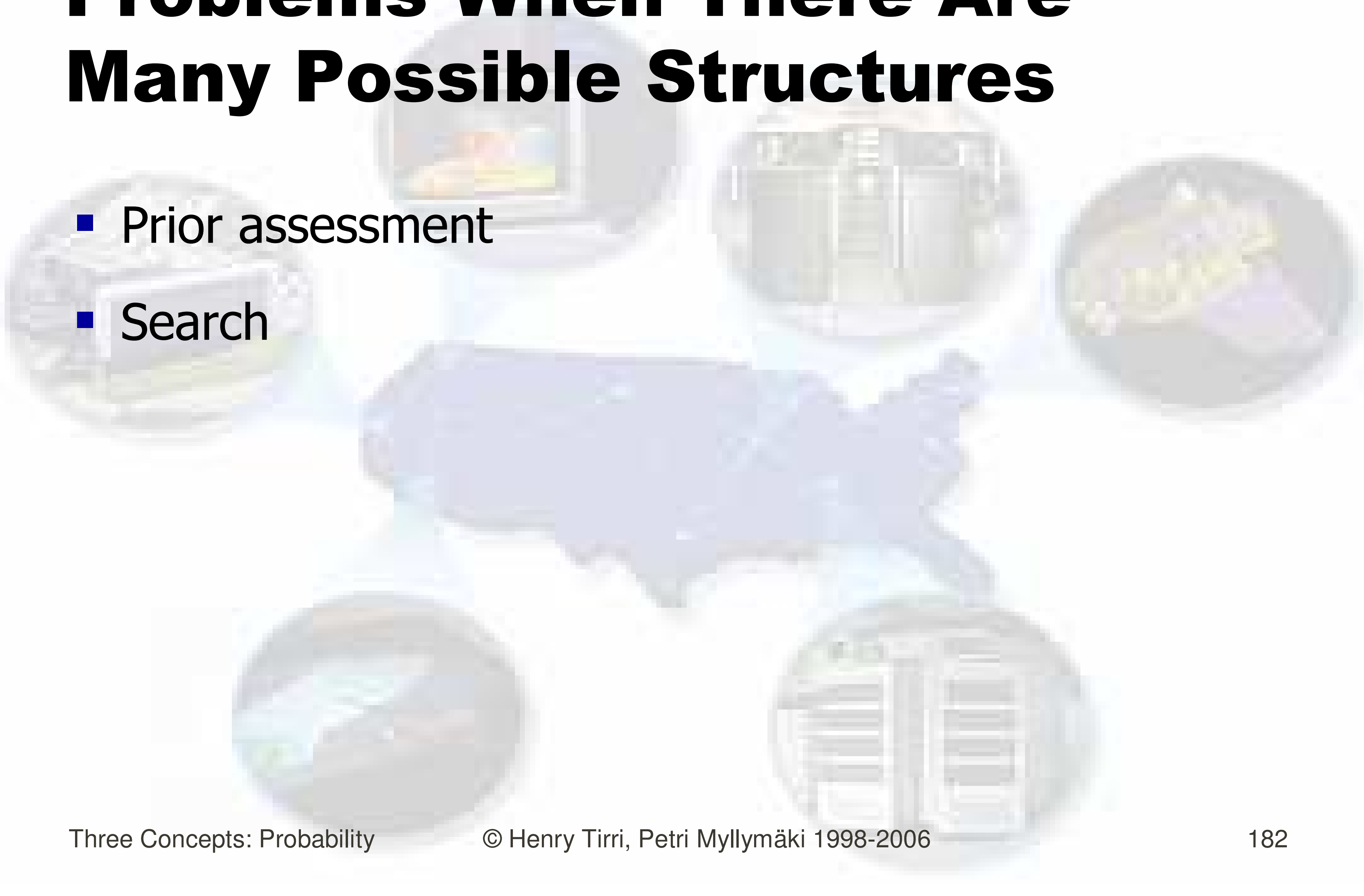
$$\log p(D|S^h) \approx \log p(D_{\mathrm{EM}}|S^h)$$

$$p(D_{EM}|S^h) \approx \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + E(N_{ij}))} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + E(N_{ijk}))}{\Gamma(\alpha_{ijk})}$$

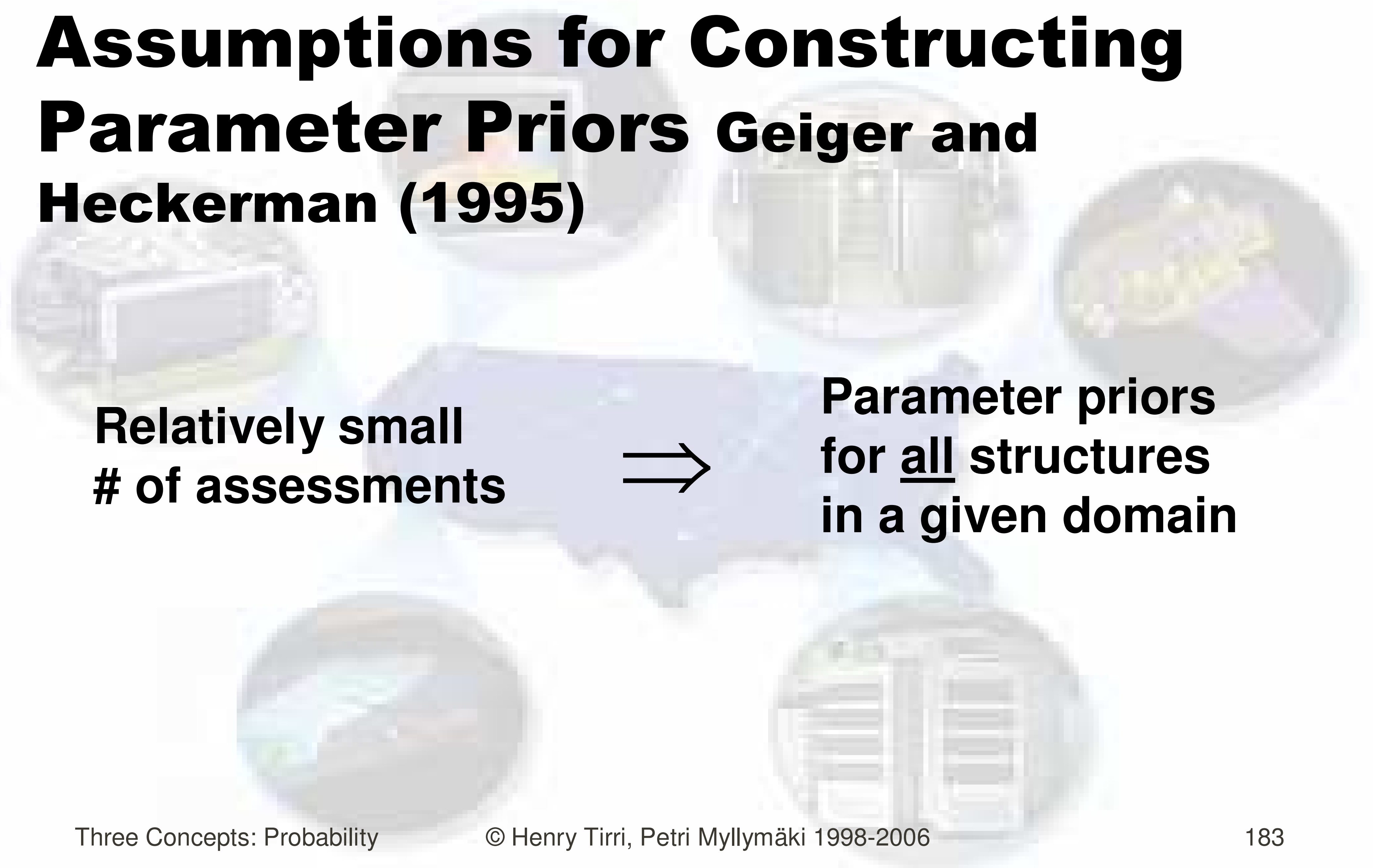
- Accuracy (CH96): Gaussian > CS >> BIC
- As efficient as BIC

Problems When There Are Many Possible Structures

- Prior assessment
- Search



Assumptions for Constructing Parameter Priors Geiger and Heckerman (1995)



Relatively small # of assessments ⇒ **Parameter priors for all structures in a given domain**

Assumptions That Simplify Assessment of Priors

- Parameter independence
- Prior modularity
- (Conjugate priors)
- Marginal likelihood equivalence

Distribution Equivalence

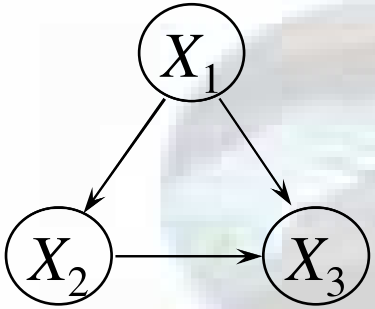
Suppose the local likelihoods are restricted to some family F (e.g., discrete, linear regression).

Two network structures for X are distribution equivalent wrt F if they encode the same distributions on X.

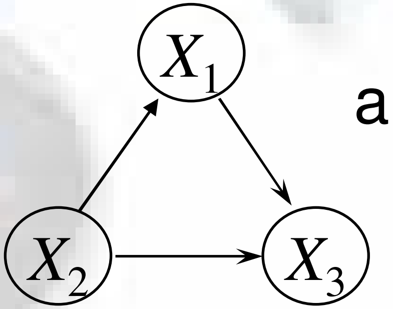
Independence and Distribution Equivalence

S_1, S_2 distribution eqv wrt some F $\Rightarrow$ S_1, S_2 independence eqv but the converse does not hold for all F. E.g.:

$$p(x_i | \mathbf{pa}_i^j, \theta_i, S^h) = \frac{1}{1 + \exp\left\{a_i + \sum_{x_j \in \mathbf{pa}_i} b_{ij} x_j\right\}}$$



and



are not distribution equivalent

Assumption: Covered-Arc-Reversal Equivalence

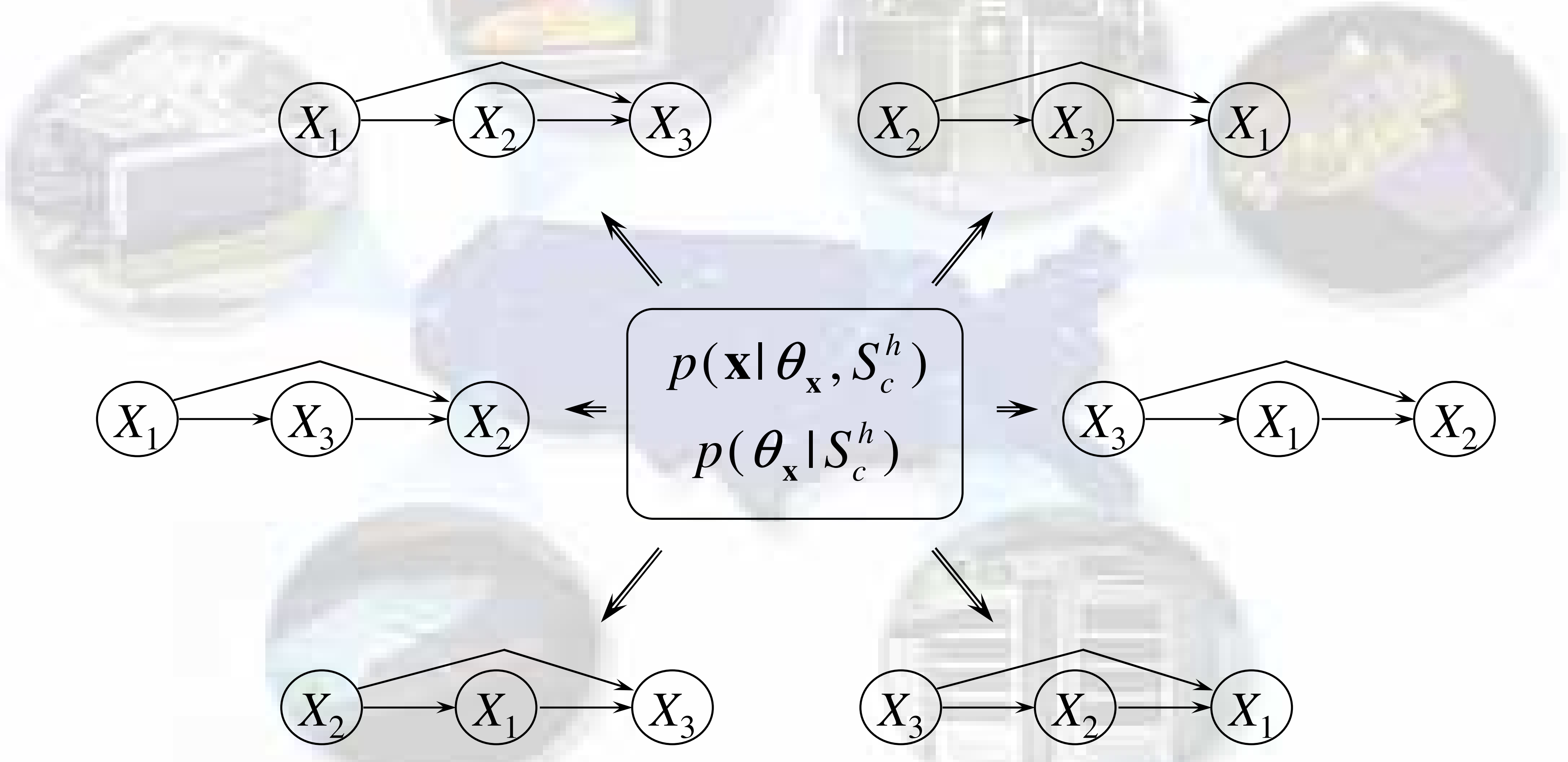
Given local likelihoods restricted to F, any two structures for X that differ only by a covered arc reversal are distribution equivalent.

<u>Examples:</u>

- unrestricted discrete
- linear regression (e.g., Shachter and Kenley)

S_1, S_2 **independence eqv** $\Rightarrow S_1, S_2$ **distribution eqv wrt F**

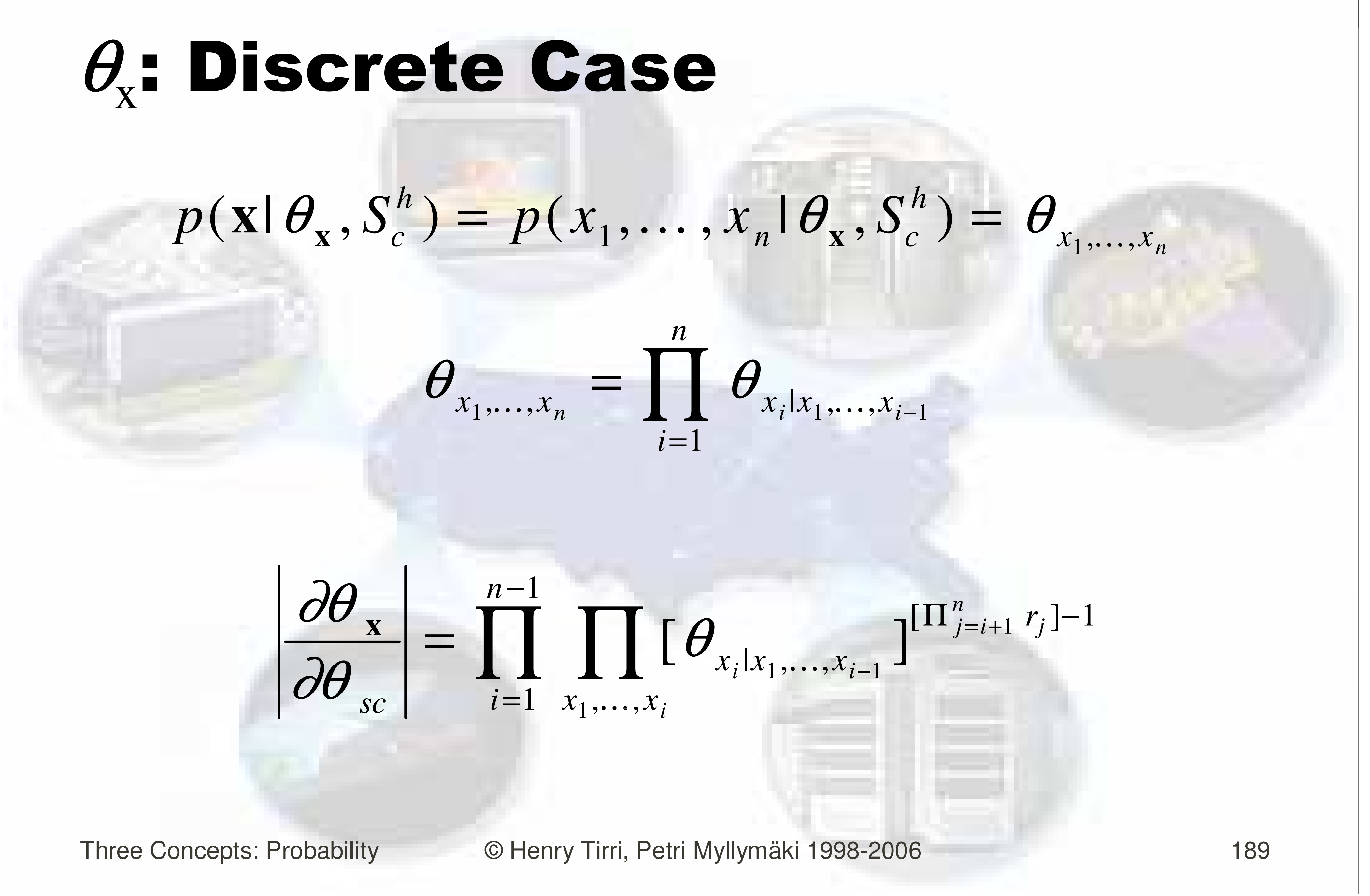
θ_{x}: Parameters for the Joint Likelihood



$$p(\mathbf{x}|\theta_{\mathbf{x}}, S_c^h)$$

$$p(\theta_{\mathbf{x}}|S_c^h)$$

θ_{x}: Discrete Case

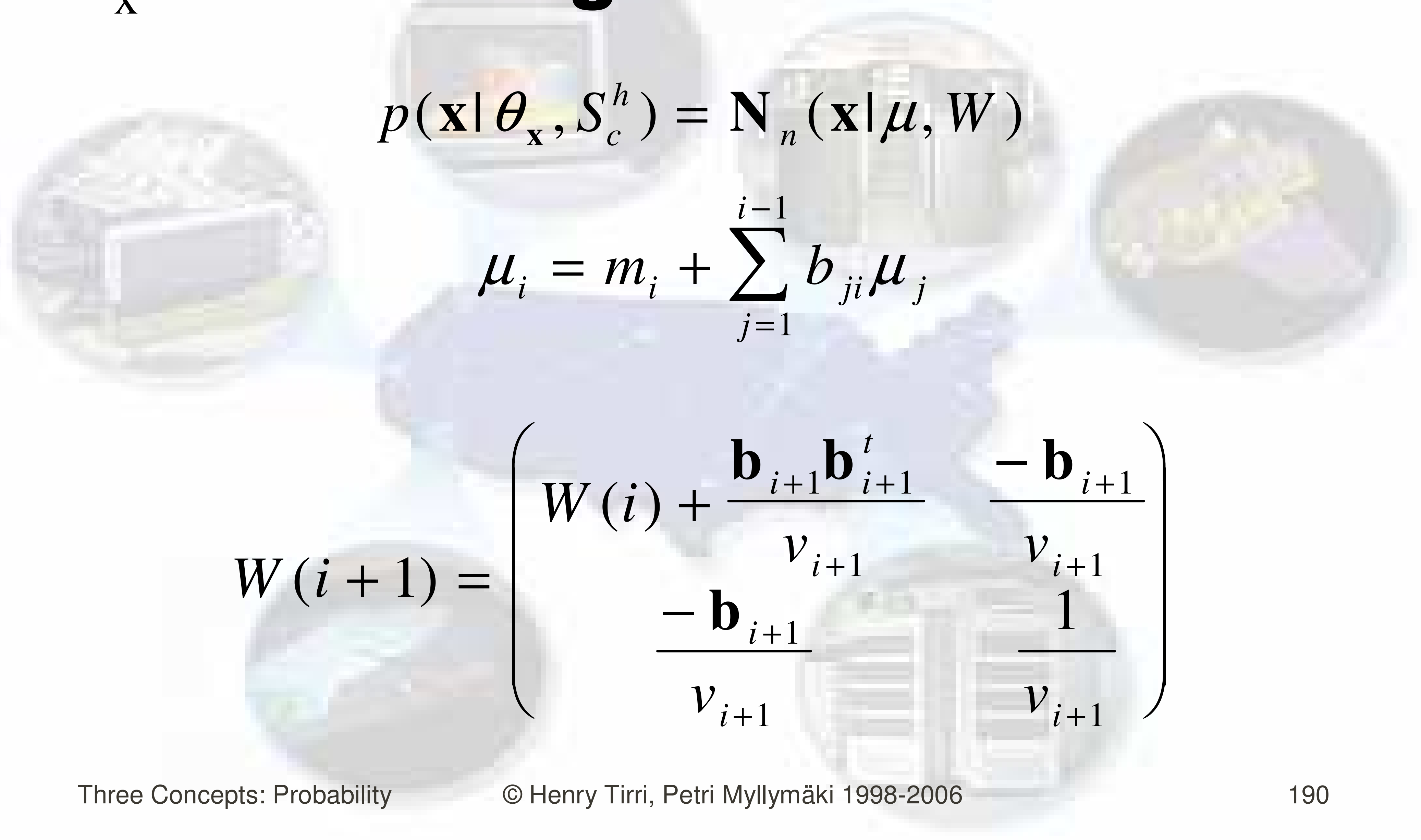


$$p(\mathbf{x}|\theta_{\mathbf{x}}, S_c^h) = p(x_1, \ldots, x_n | \theta_{\mathbf{x}}, S_c^h) = \theta_{x_1,\ldots,x_n}$$

$$\theta_{x_1,\ldots,x_n} = \prod_{i=1}^{n} \theta_{x_i|x_1,\ldots,x_{i-1}}$$

$$\left|\frac{\partial\theta_{\mathbf{x}}}{\partial\theta_{sc}}\right| = \prod_{i=1}^{n-1} \prod_{x_1,\ldots,x_i} [\theta_{x_i|x_1,\ldots,x_{i-1}}]^{[\Pi_{j=i+1}^{n} r_j]-1}$$

θ_{x}: Linear-Regression Case

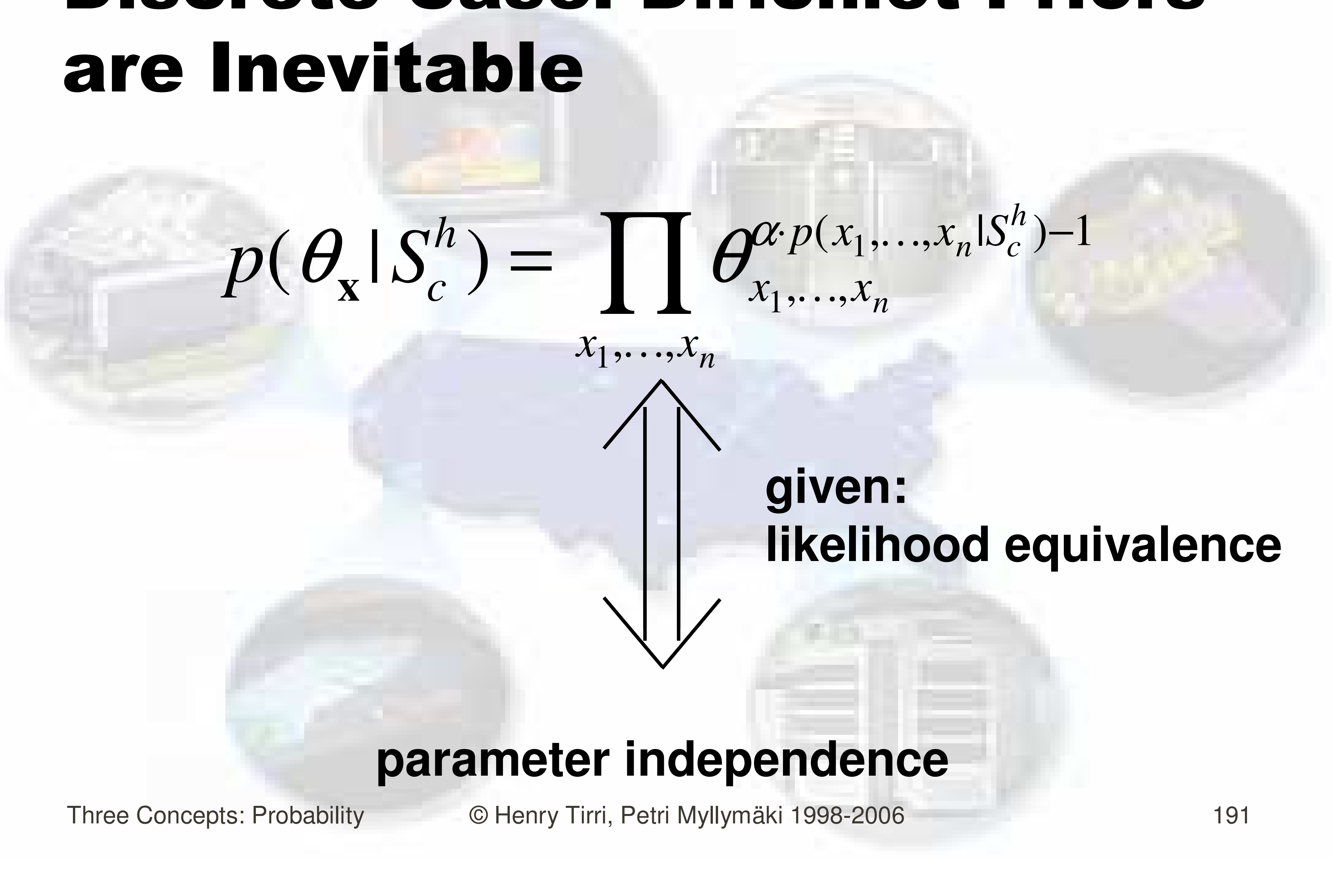


$$p(\mathbf{x}|\theta_{\mathbf{x}},S_c^h)=\mathbf{N}_n(\mathbf{x}|\mu,W)$$

$$\mu_i = m_i + \sum_{j=1}^{i-1} b_{ji}\mu_j$$

$$W(i+1)=\begin{pmatrix} W(i)+\dfrac{\mathbf{b}_{i+1}\mathbf{b}_{i+1}^t}{v_{i+1}} & \dfrac{-\mathbf{b}_{i+1}}{v_{i+1}} \\ \dfrac{-\mathbf{b}_{i+1}}{v_{i+1}} & \dfrac{1}{v_{i+1}}\end{pmatrix}$$

Discrete Case: Dirichlet Priors are Inevitable



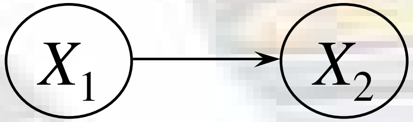
$$p(\theta_{\mathbf{x}} | S_c^h) = \prod_{x_1,\ldots,x_n} \theta_{x_1,\ldots,x_n}^{\alpha\cdot p(x_1,\ldots,x_n|S_c^h)-1}$$

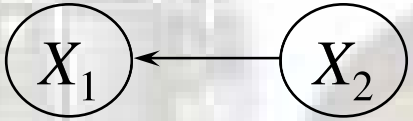
$$\Updownarrow$$

given:
likelihood equivalence

parameter independence

Discrete Case

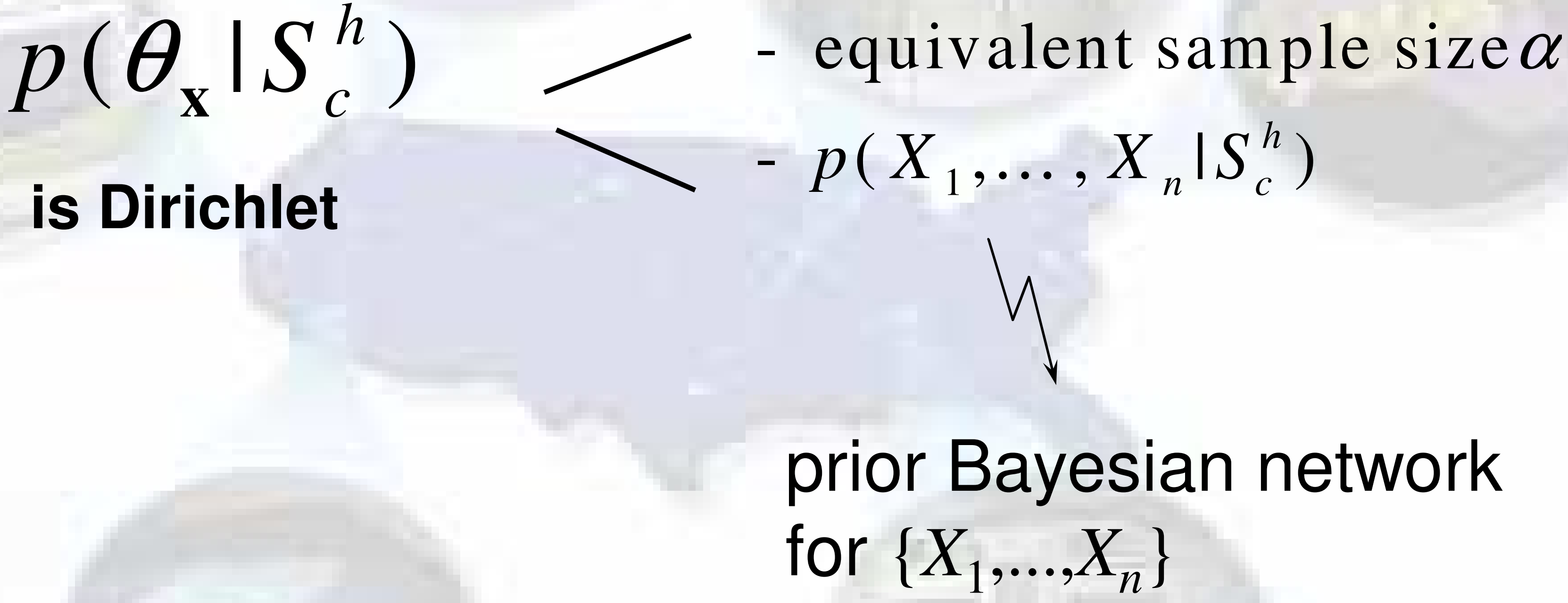




⇓ **parameter independence**
likelihood equivalence

$$\frac{f_1(\theta_1)\, f_{2|1}(\theta_{y|x})\, f_{2|\bar{1}}(\theta_{2|\bar{1}})}{\theta_1(1-\theta_1)} = \frac{f_2(\theta_2)\, f_{1|2}(\theta_{1|2})\, f_{1|\bar{2}}(\theta_{1|\bar{2}})}{\theta_2(1-\theta_2)}$$

Assessments for the BDe Score



$p(\theta_{\mathbf{x}} | S_c^h)$

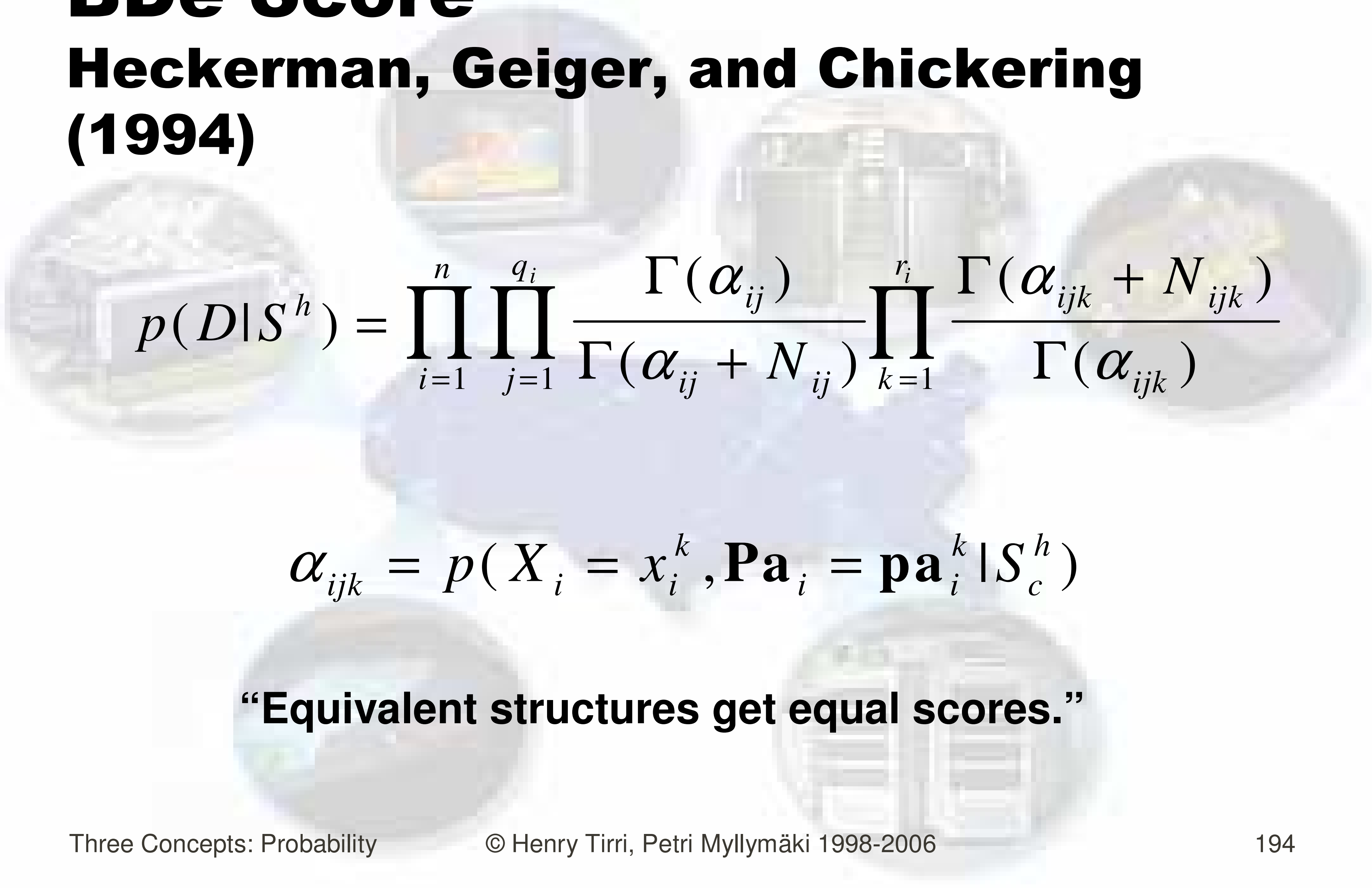
is Dirichlet

- equivalent sample size α

- $p(X_1, \ldots, X_n | S_c^h)$

prior Bayesian network for $\{X_1,\ldots,X_n\}$

BDe Score
Heckerman, Geiger, and Chickering (1994)



$$p(D|S^h) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

$$\alpha_{ijk} = p(X_i = x_i^k, \mathbf{Pa}_i = \mathbf{pa}_i^k | S_c^h)$$

“Equivalent structures get equal scores.”

Linear-Regression Case: Normal-Wishart Prior is Inevitable

$$p(\theta_{\mathbf{x}} | S_c^h) = \mathbf{Nw}_n(\theta_{\mathbf{x}} | \mu_0, \alpha_\mu, W_0, \alpha_W)$$

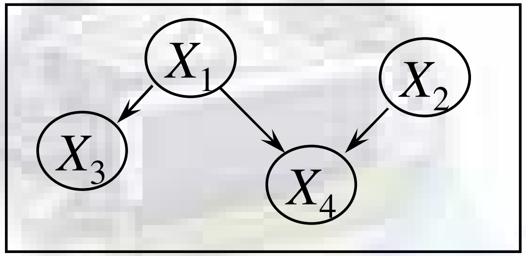
$$\Updownarrow$$

given: likelihood equivalence

parameter independence

Combine Domain Knowledge and Data

Prior Network



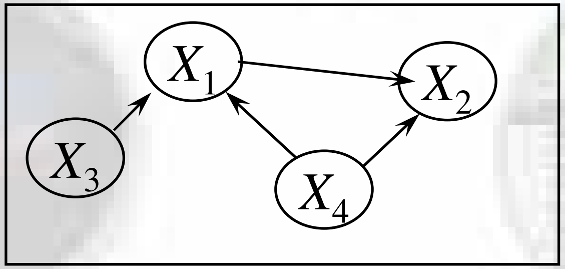
Equivalent Sample Size

$$\boxed{\alpha}$$

Data

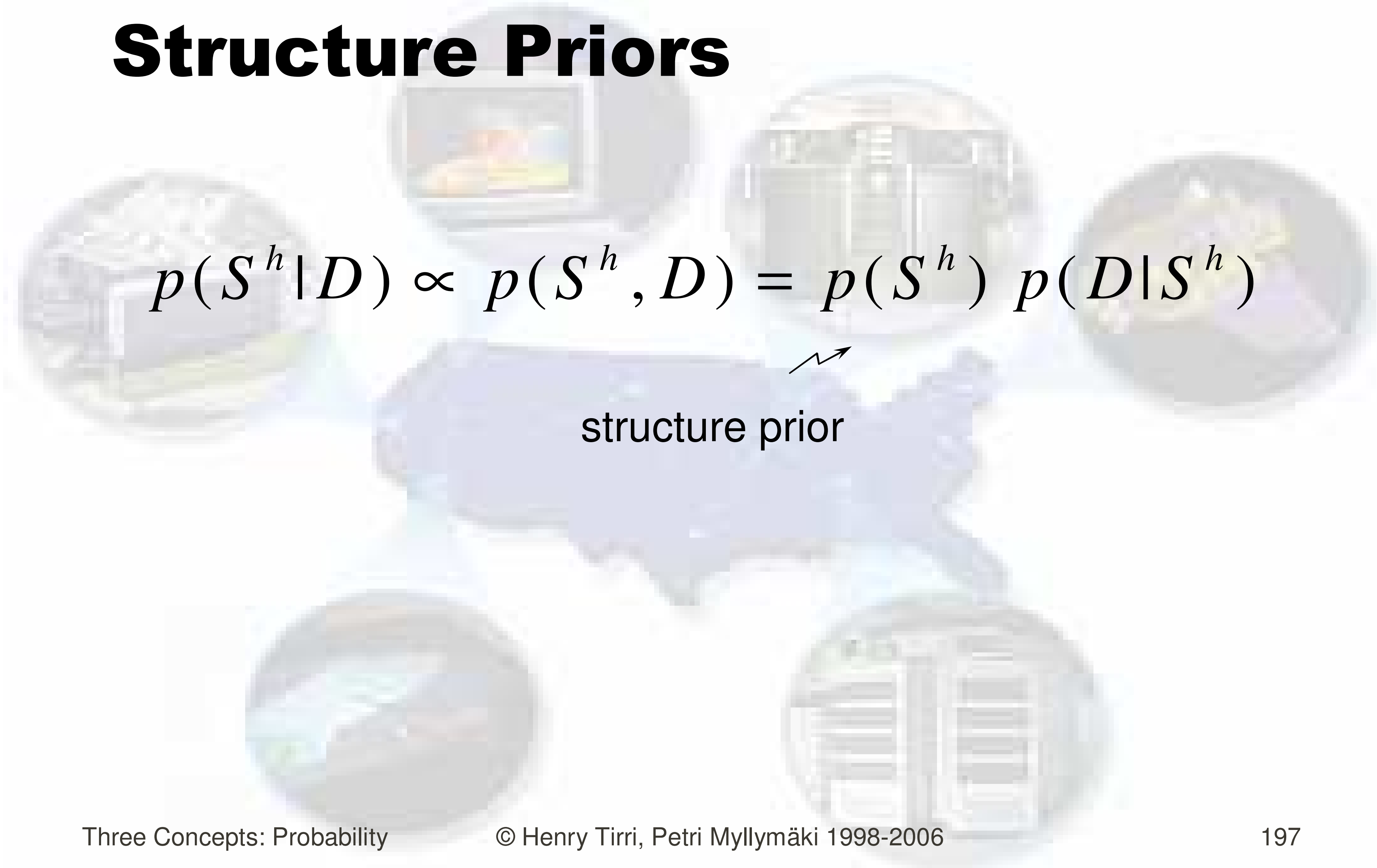
	X_1	X_2	X_3	. . .	X_n
$\mathbf{x}_1$:	true	false	true	. . .	true
$\mathbf{x}_2$:	false	true	true	. . .	false
		⋮			
$\mathbf{x}_N$:	true	true	false	. . .	true

Priors for all structures



Learned structure

Structure Priors



$$p(S^h|D) \propto p(S^h,D) = \underbrace{p(S^h)}_{\text{structure prior}}\ p(D|S^h)$$

Structure Prior

$$p(S^h) \propto \kappa^{\delta}, \quad 0 < \kappa < 1$$

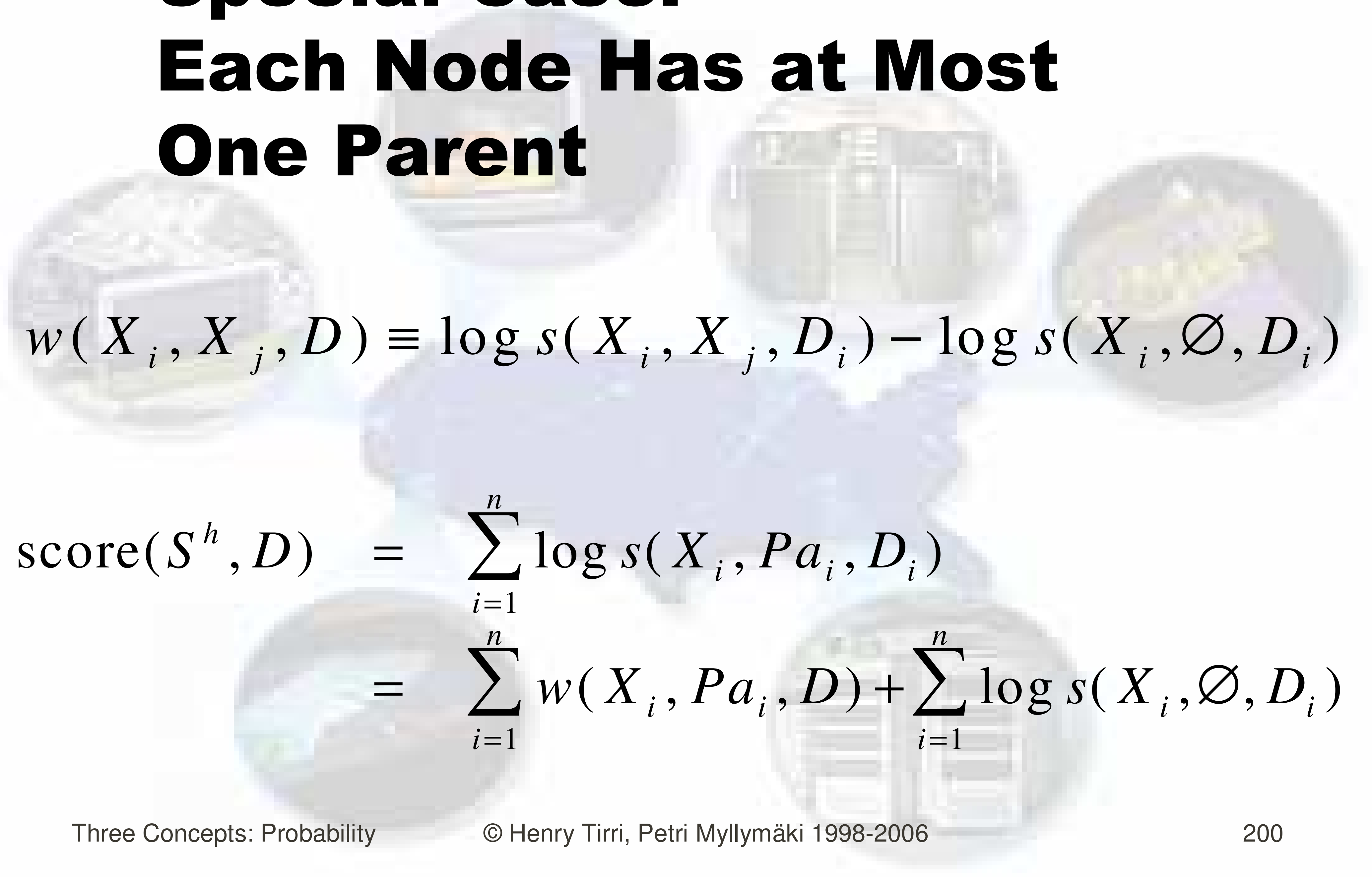
δ is the number of arcs that differ between S^h and the prior network

Search Methods

- Finding the structure with the highest score among those structures with at most k parents is NP hard for k>1 (Chickering 95)
- Heuristic methods
  - *Greedy (+/- restarts)
  - Best-first search
  - Monte-Carlo search methods
- Search space: ADGs vs. equivalence classes
  - Spirtes and Meek (1995), Chickering (1996)
- Complete vs. incomplete data

$$\mathrm{score}(S, D) = \prod_{i=1}^{n} s(X_i, \mathbf{Pa}_i, D_i)$$

Special Case: Each Node Has at Most One Parent



$$w(X_i, X_j, D) \equiv \log s(X_i, X_j, D_i) - \log s(X_i, \emptyset, D_i)$$

$$\begin{aligned}\text{score}(S^h, D) &= \sum_{i=1}^{n} \log s(X_i, Pa_i, D_i) \\ &= \sum_{i=1}^{n} w(X_i, Pa_i, D) + \sum_{i=1}^{n} \log s(X_i, \emptyset, D_i)\end{aligned}$$

Each Node Has at Most One Parent

The network with the highest probability is the one for which $\Sigma\, w(X_i\,, Pa_i, D)$ is a maximum

- **Scores without likelihood+prior equivalence:**
  Maximum branchings (Edmonds)

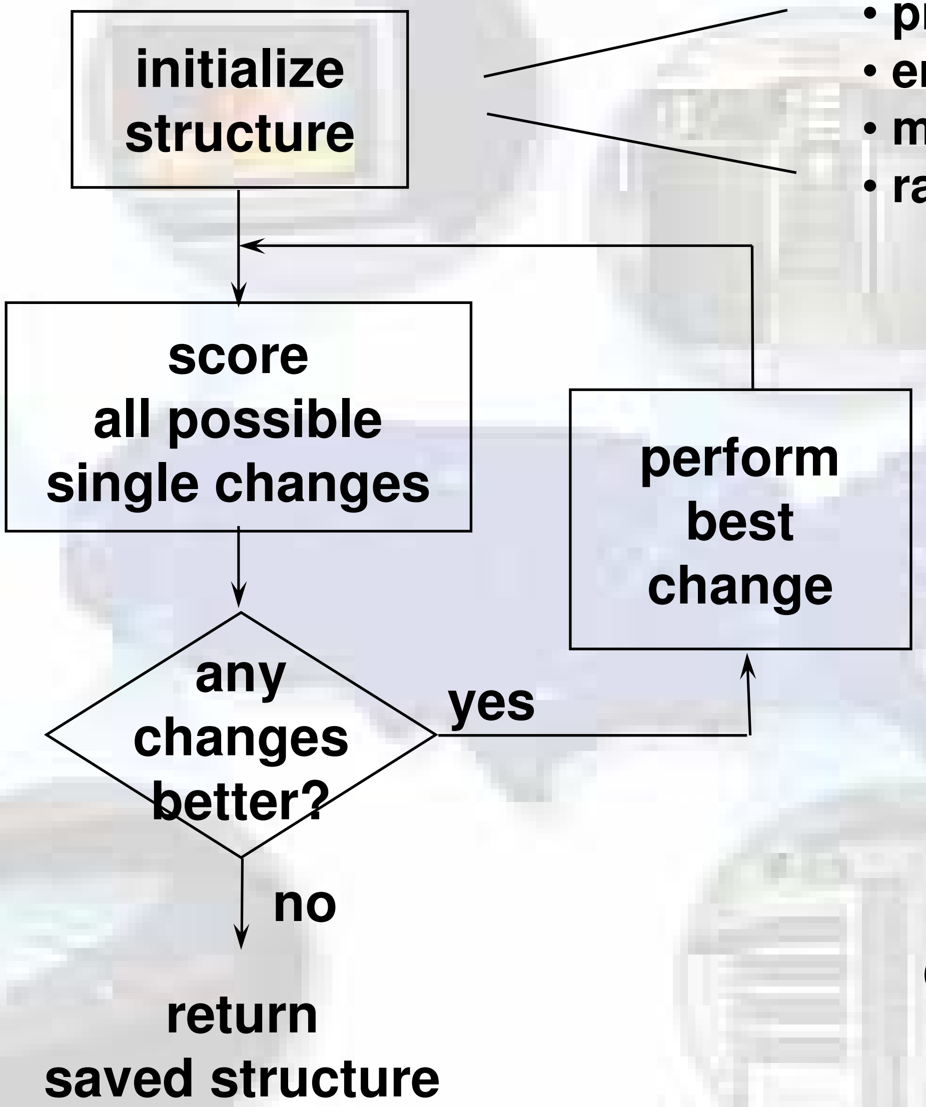
- **Scores with likelihood+prior equivalence:**
  Maximum spanning tree

General Case: Each Node Has at Most *k* Parents

Finding best structure is NP-hard for k>1

- Greedy (+/- restarts)
- Best-first search
- Simulating annealing
- Other Monte-Carlo approaches

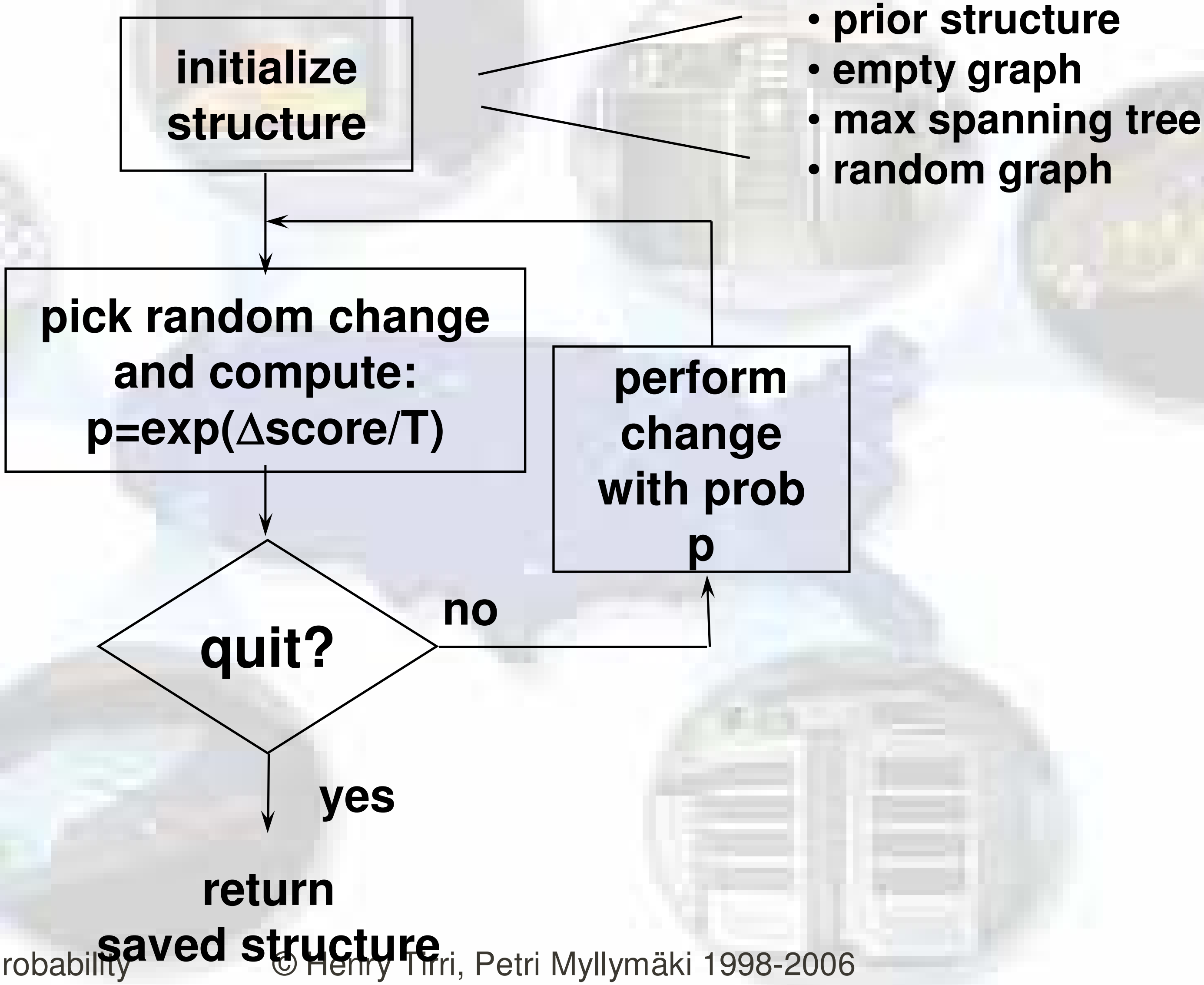
Local Search



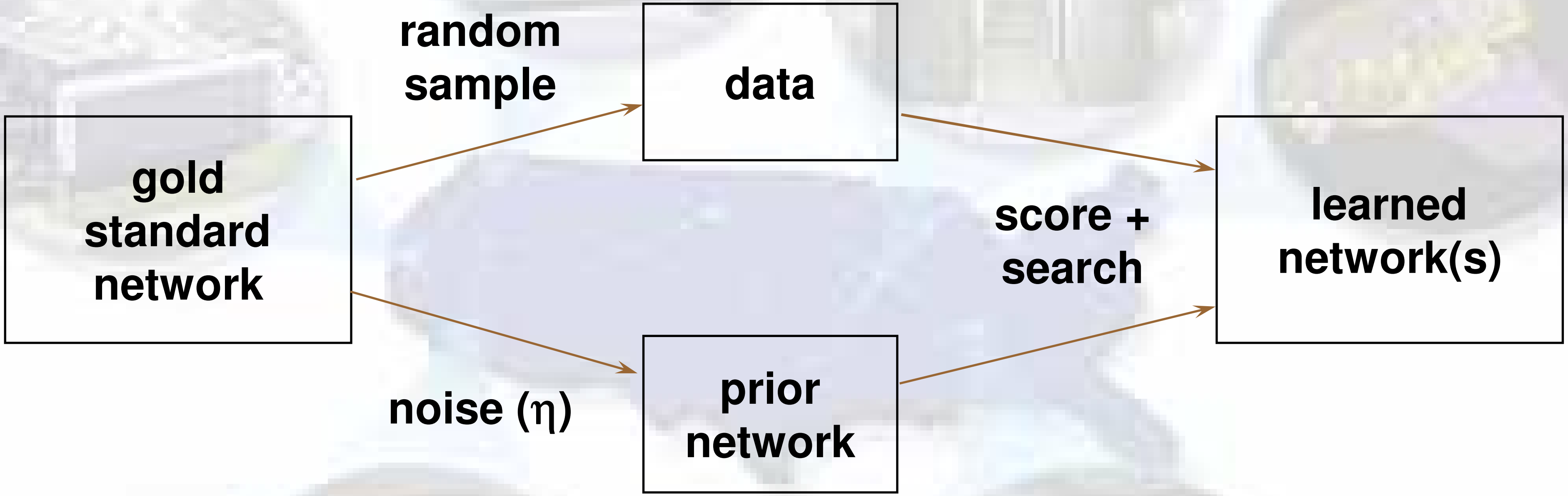
- prior structure
- empty graph
- max spanning tree
- random graph

extension: multiple restarts

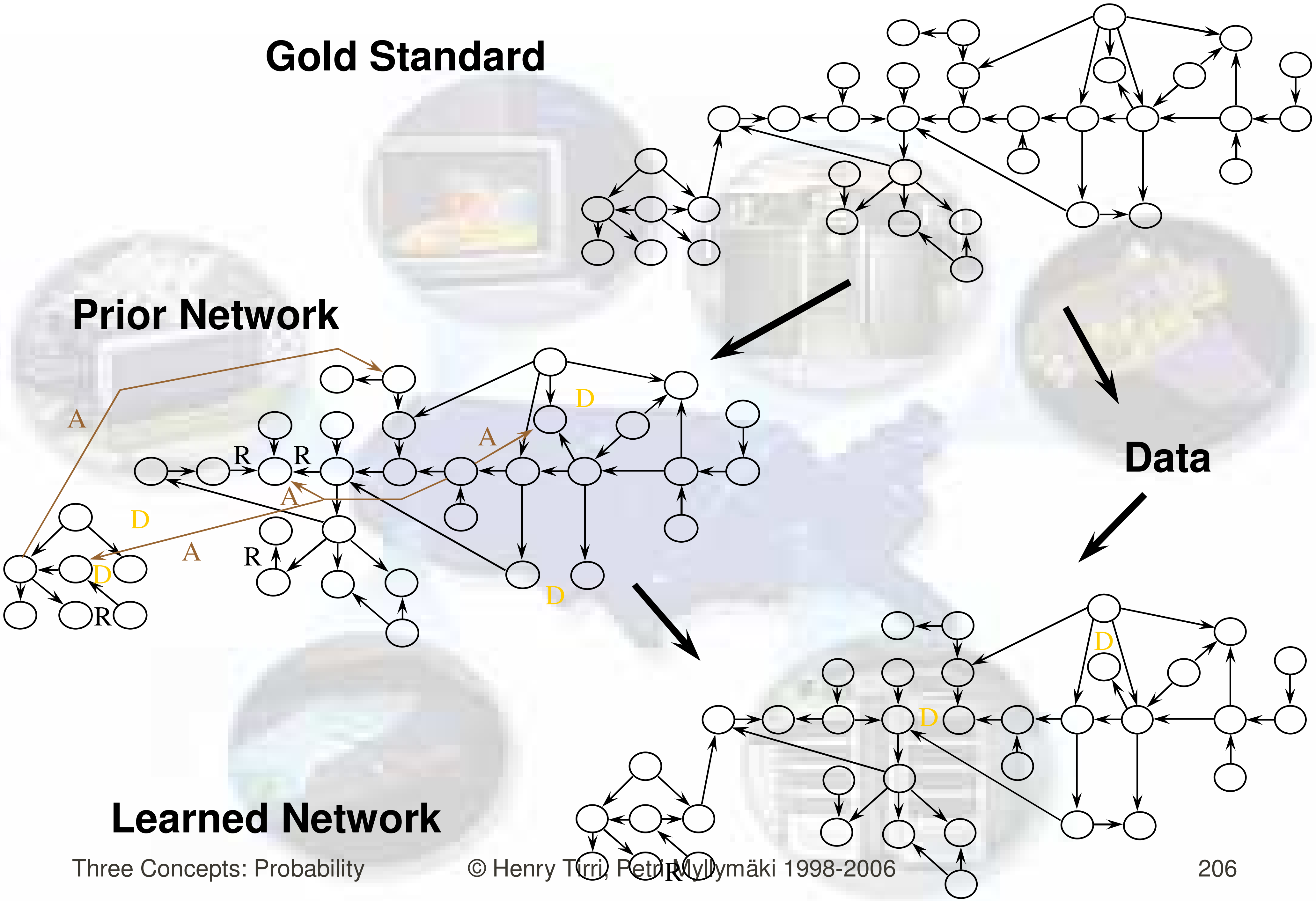
Simulated Annealing (Metropolis)



Evaluation Methodology



Two measures of utility of learned network:
- Cross Entropy (gold standard network, learned network)
- Structural difference



Gold Standard

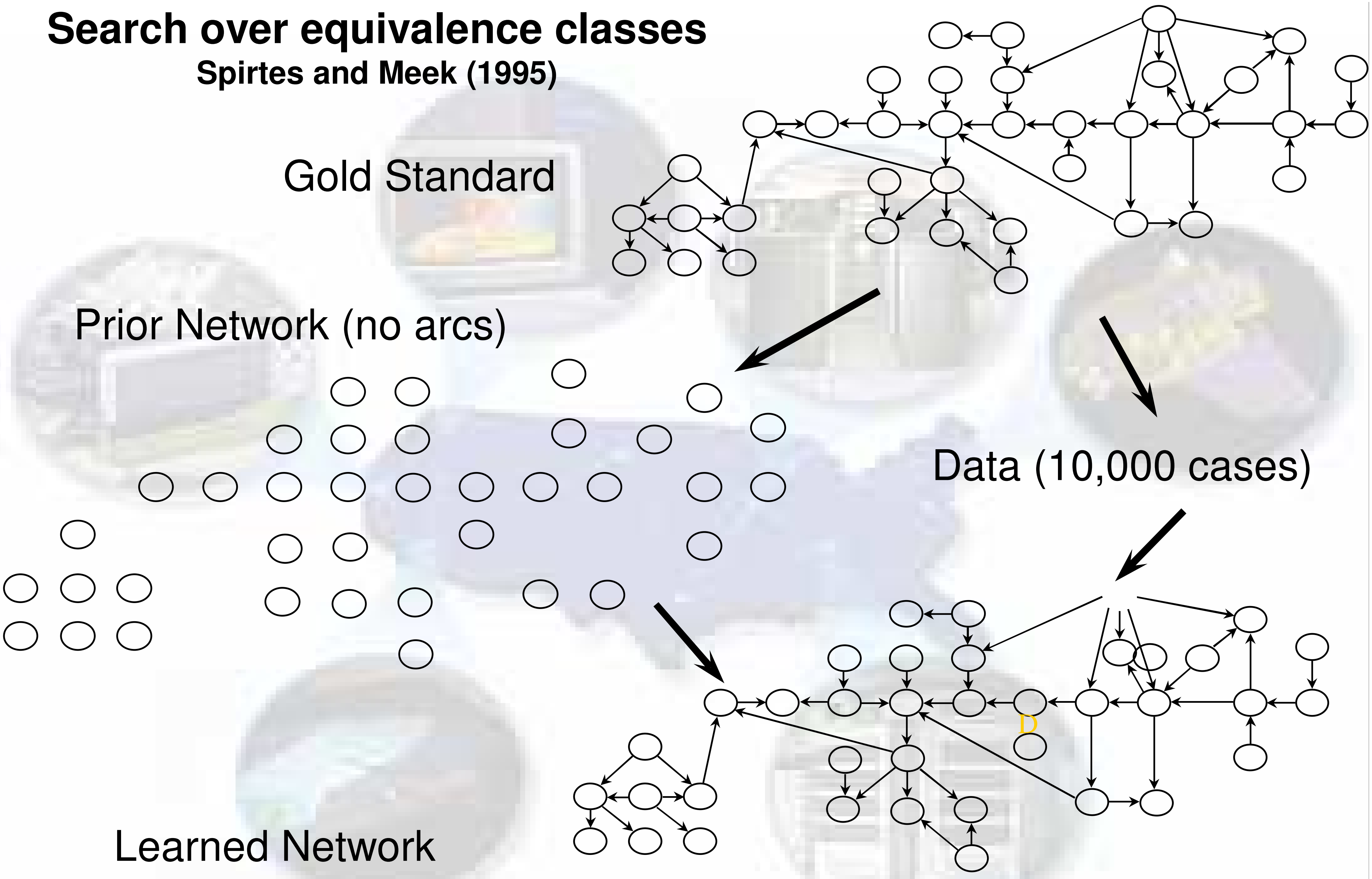
Prior Network

Data

Learned Network

Search over equivalence classes

Spirtes and Meek (1995)



Gold Standard

Prior Network (no arcs)

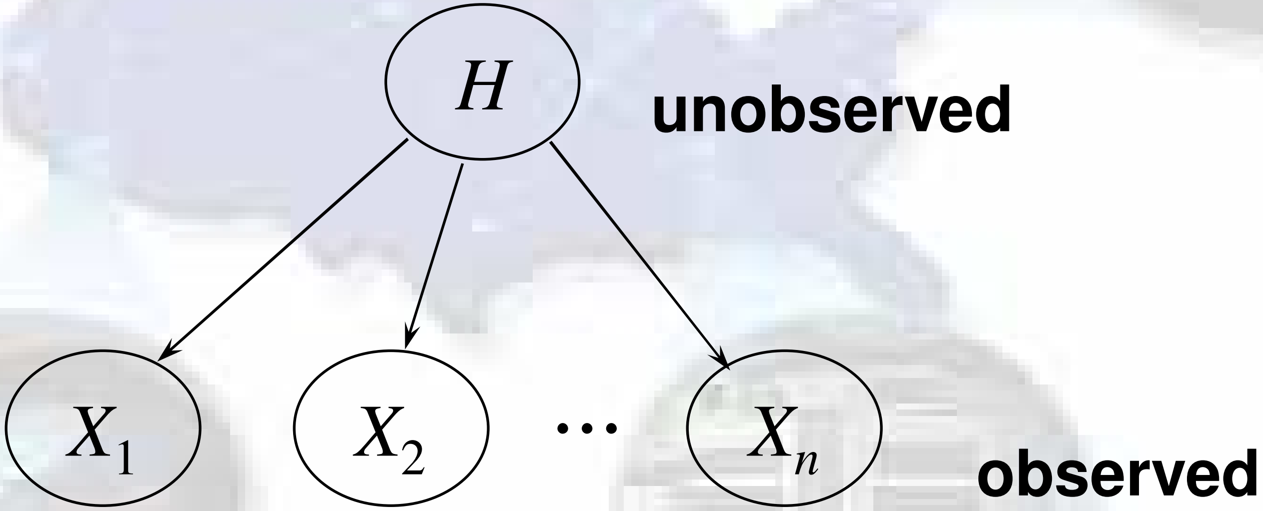
Data (10,000 cases)

Learned Network

Learning with Hidden Variables

Special case of learning with missing data

E.g.: AutoClass (Cheeseman)



Identifying Hidden Variables

- Prior knowledge (e.g., AutoClass)
- Dependency cliques