Statistical modeling with stochastic processes

Alexandre Bouchard-Côté

Lecture 2, Wednesday March 2

Plan for today

- Finishing the applications/motivations overview
- Computational issues: overview
- Background
  - Graphical models
  - MCMC
  - Bayesian decision theory

Stochastic processes

> 'A collection of random variables indexed by an arbitrary set S'

Note 1: if S if finite, then back to an 'undergrad' random variable, so we concentrate on S uncountable

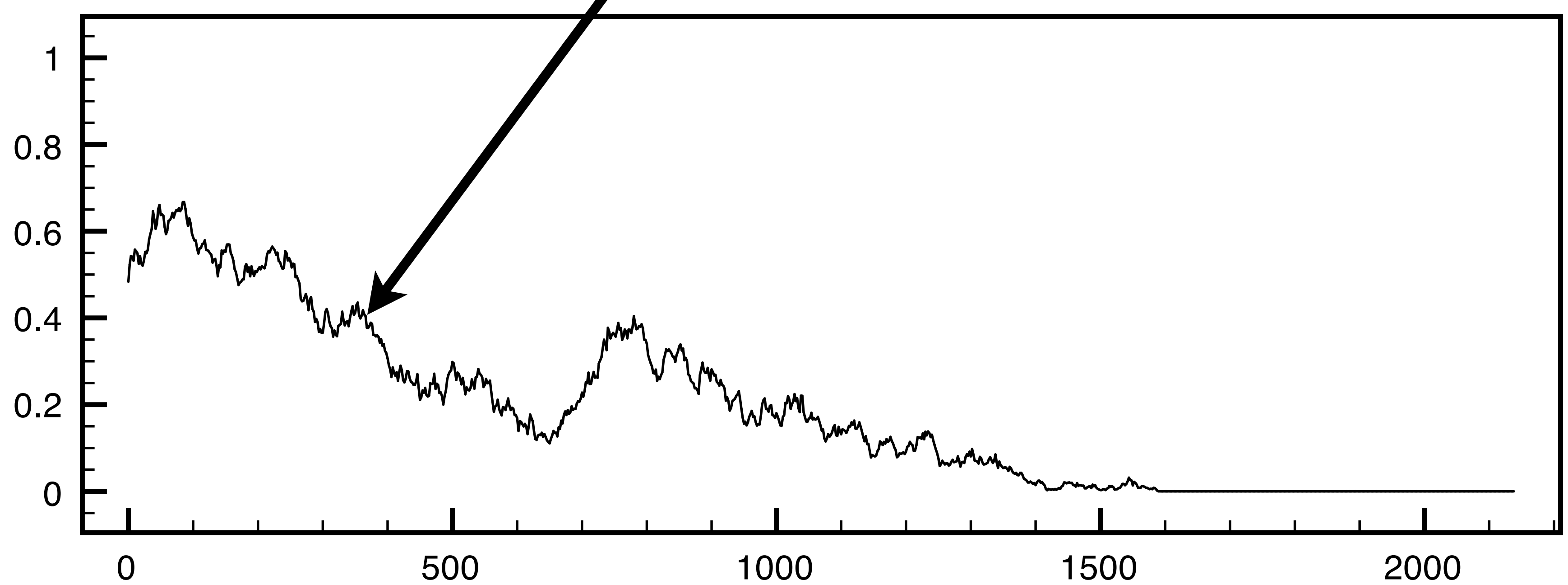
Note 2: S is not necessarily the real line

Example: distribution over functions

Samples: functions f : $\mathbf{R} \to \mathbf{R}$

$(s,\ Y_s(\omega))$

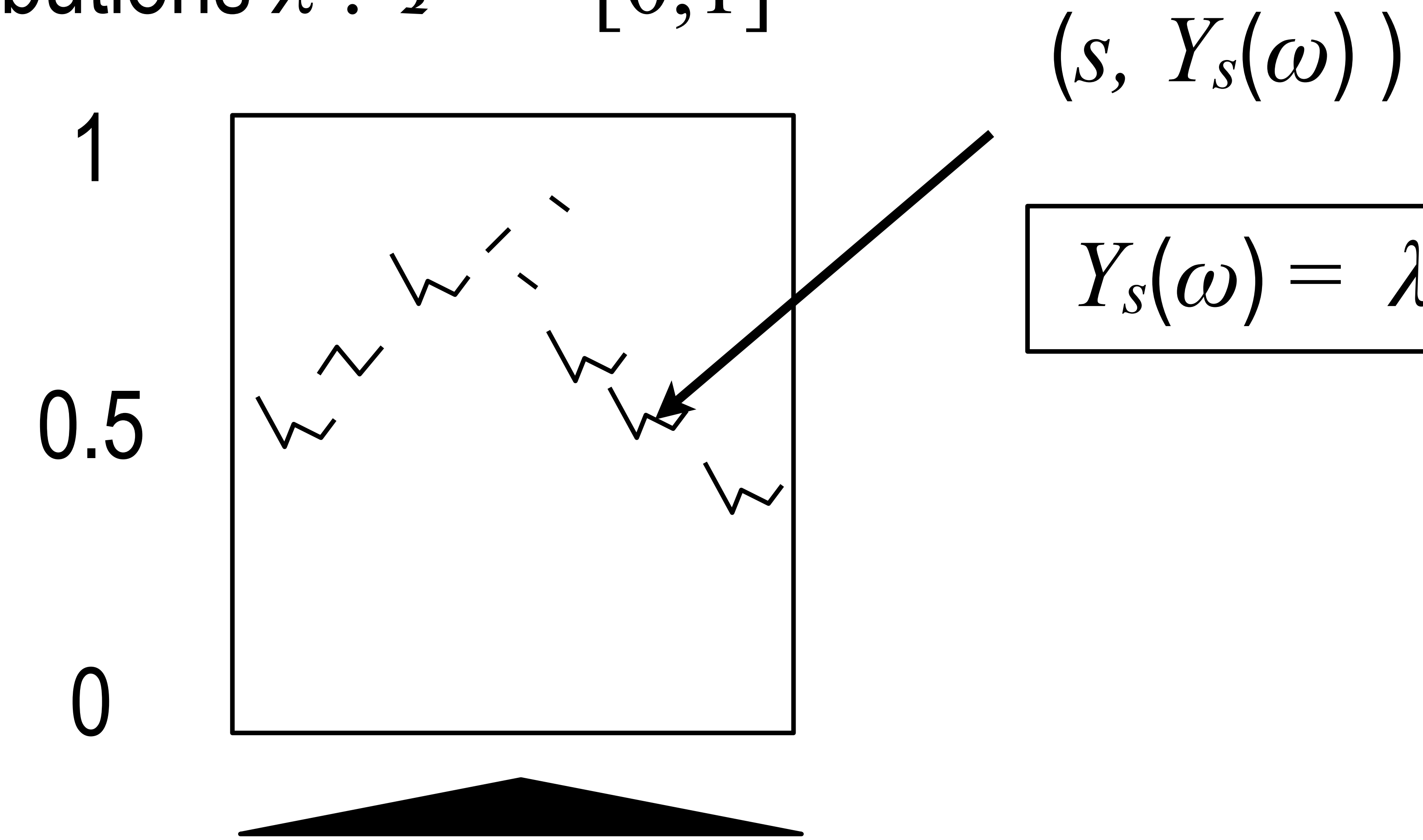
$$\boxed{Y_s(\omega) = f(s)}$$



$S = \mathbf{R}$

Example: distribution over *distributions*

Samples: distributions $\lambda : \mathcal{F} \to [0,1]$



$\boxed{Y_s(\omega) = \lambda(s)}$

$S = \mathcal{F}$, a sigma-algebra (the set of events for λ)

(No topology on this axis this time...)

Why would we need distributions over distributions?

De Finetti theorem: a compelling motivation for priors on parameters...

Suppose: we agree that if our data x_i are reorder, it doesn't matter (exchangeability), e.g.

$$(x_1, x_2, x_3, \ldots) \overset{d}{=} (x_3, x_1, x_2, \ldots)$$

Then: there exists a random variable θ and distributions F_θ such that:

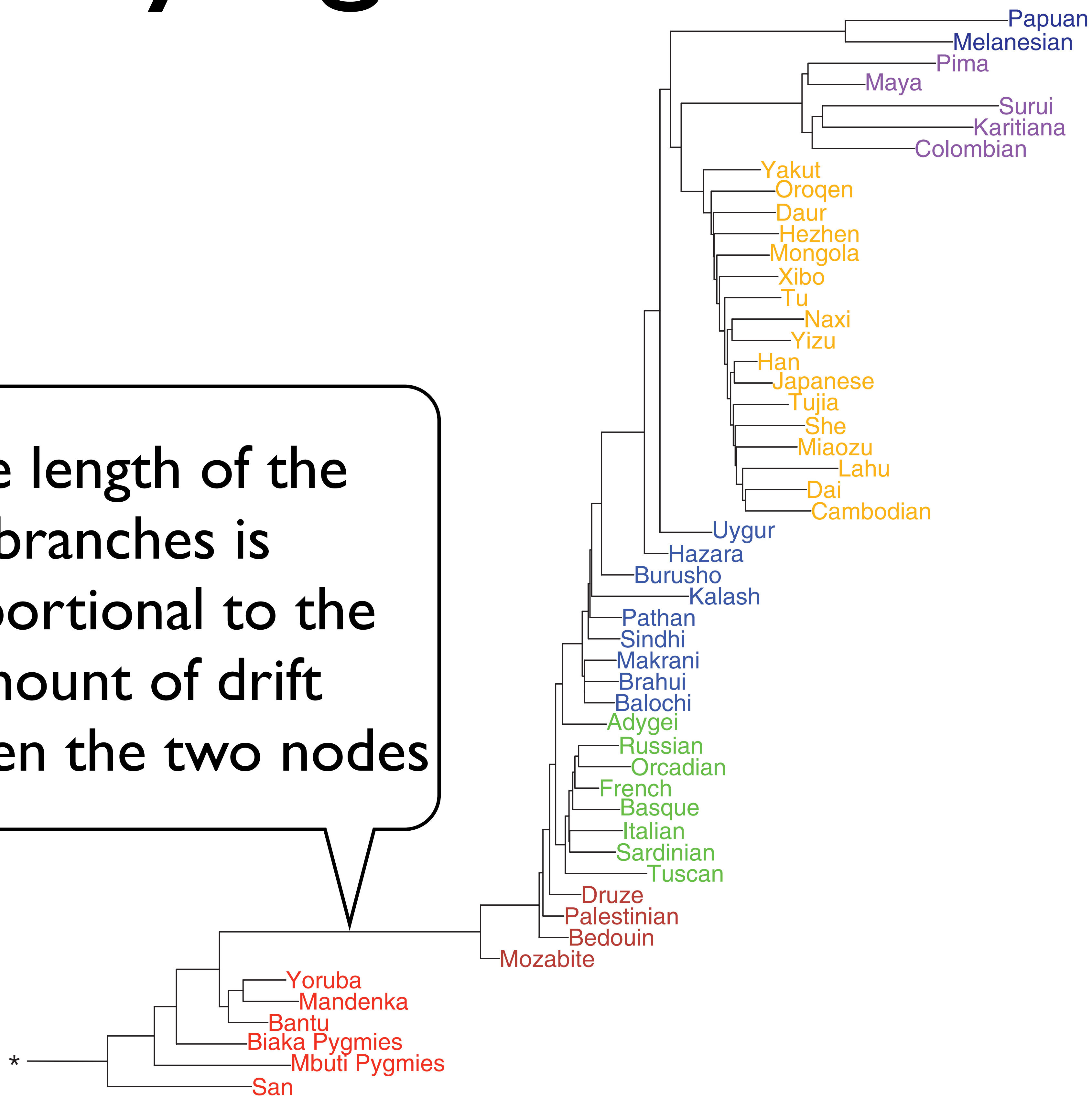
$$x_i | \theta \sim F_\theta$$

Non-Bayesian application: phylogenetic inference

Scientific applications: biology, anthropology, linguistics

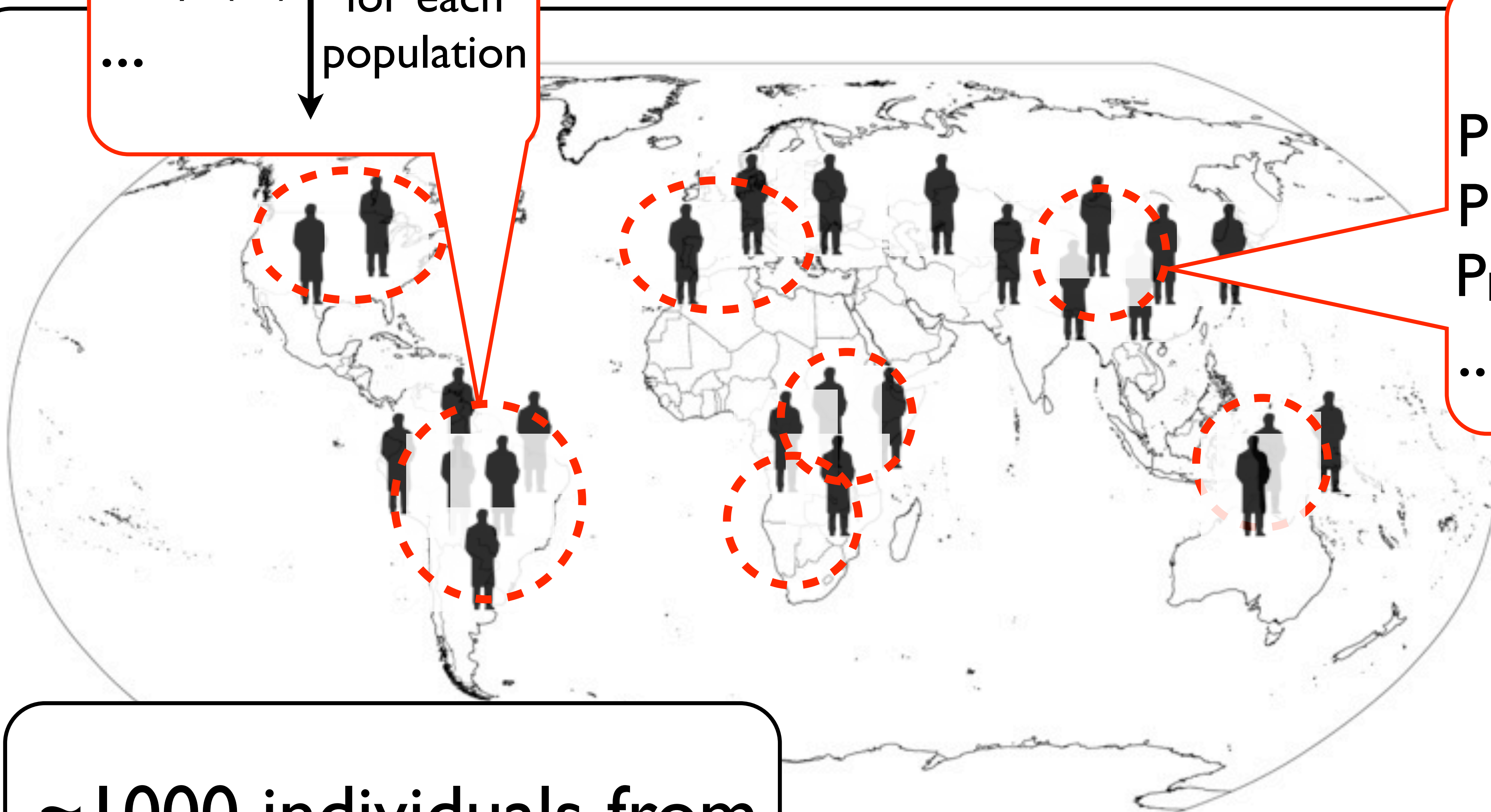
Engineering applications: domain adaptation, multi-task learning

Phylogenetic tree



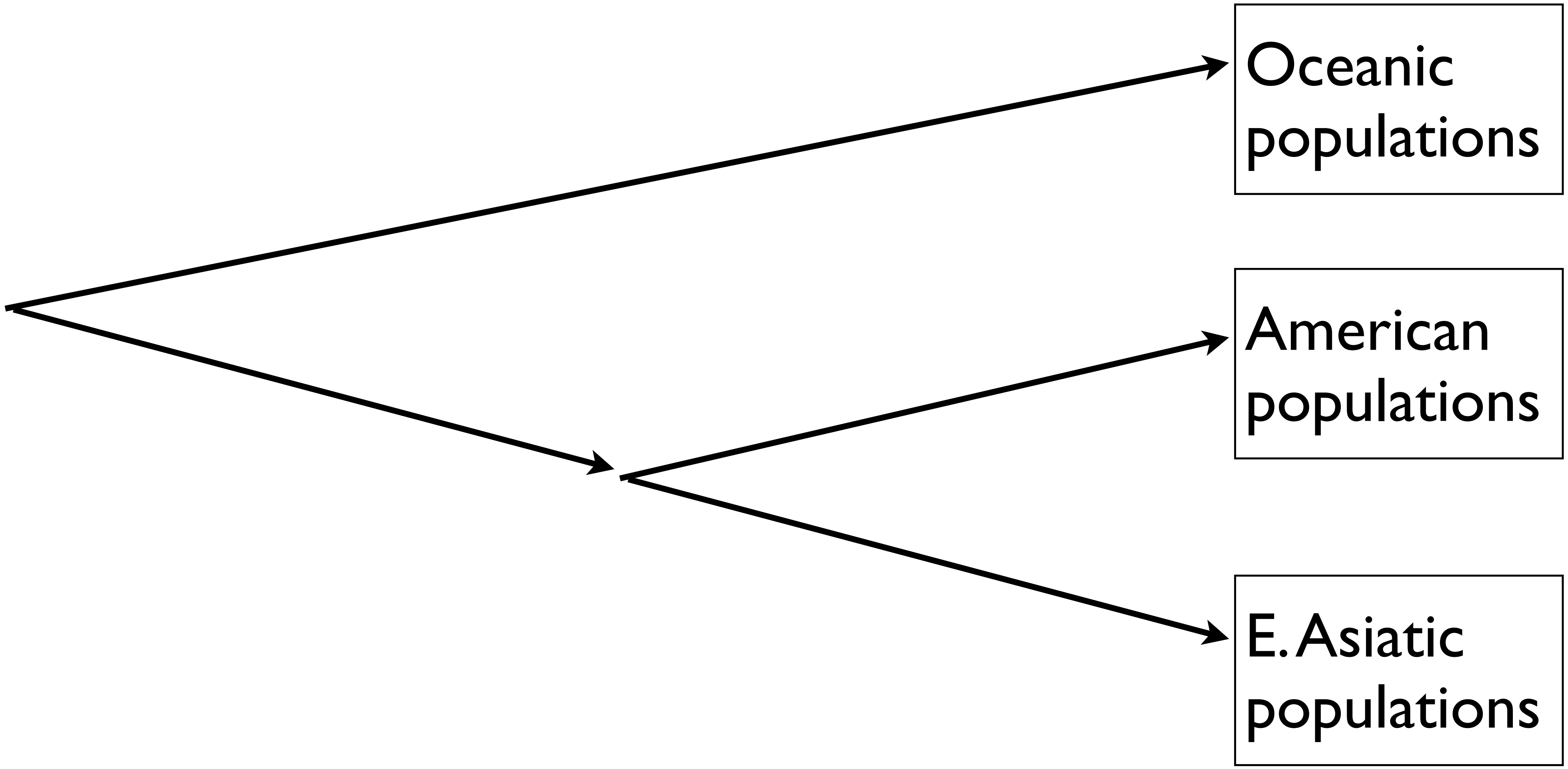
The length of the branches is proportional to the amount of drift between the two nodes

Data

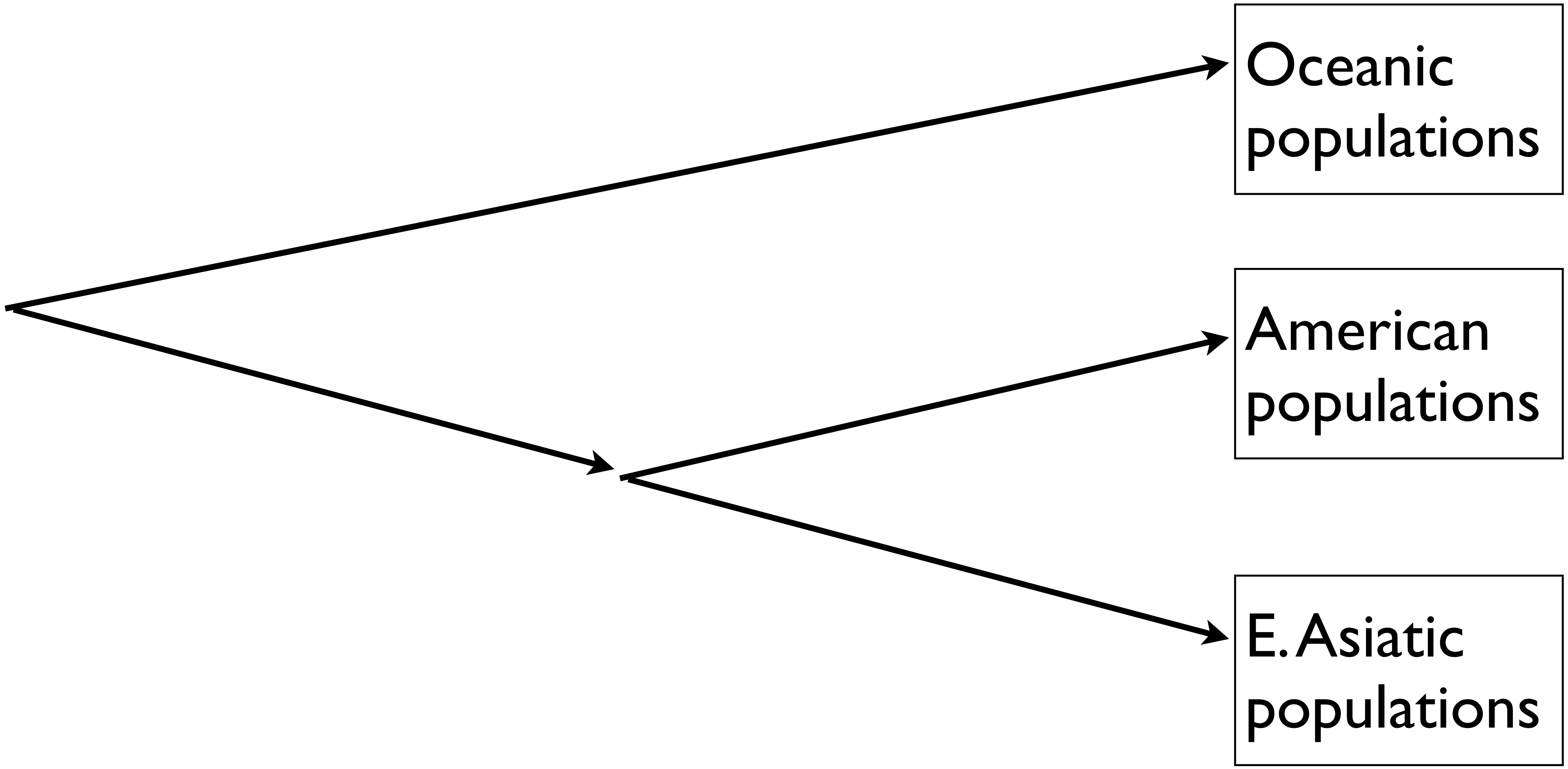


~1000 individuals from
~50 populations

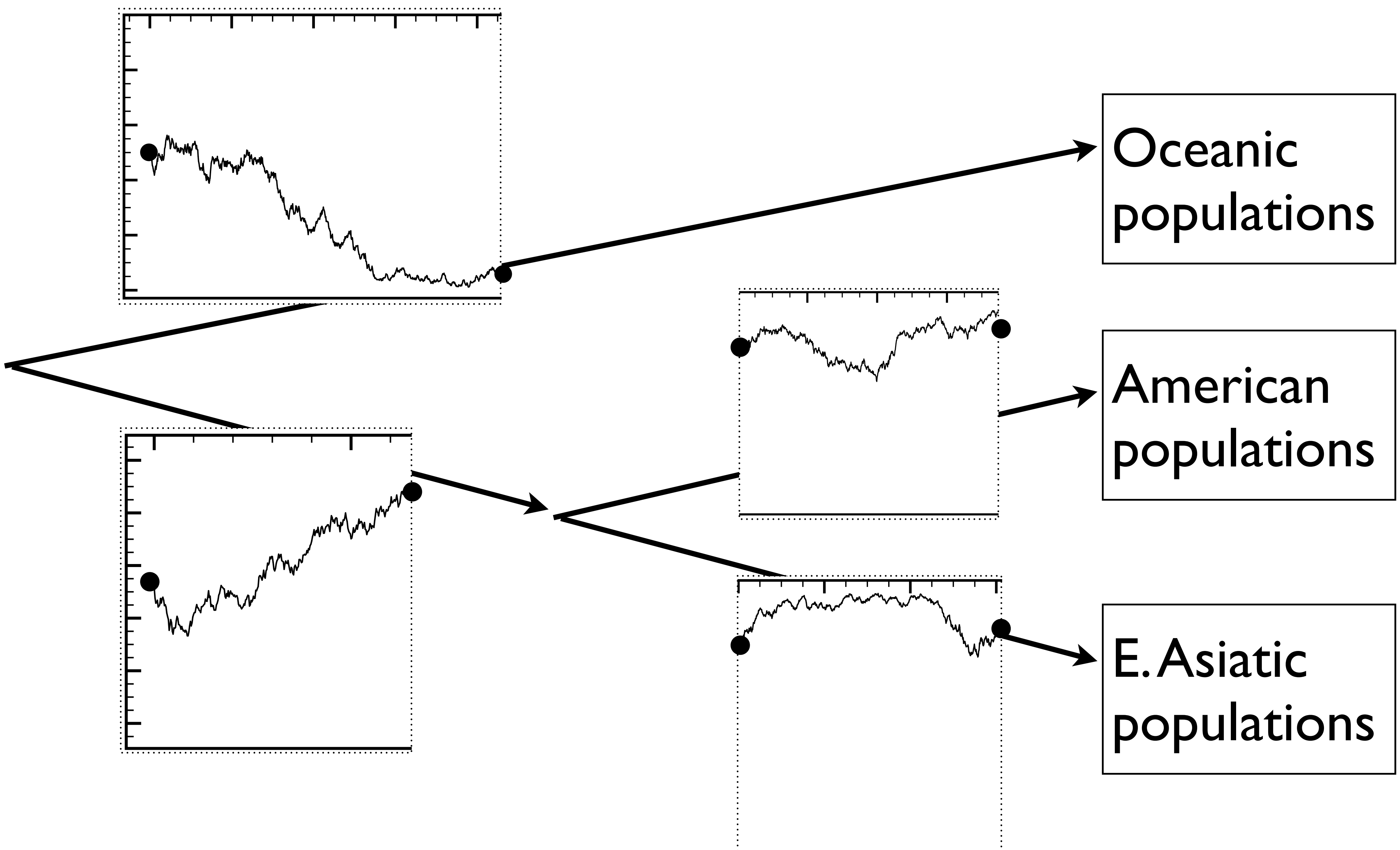
Human Genome Diversity Panel


Oceanic populations
American populations
E. Asiatic populations

Doing the same thing, but with the other tree gives us P(Data | H2)



Doing the same thing, but with the other tree gives us P(Data | H2)



Data: second type

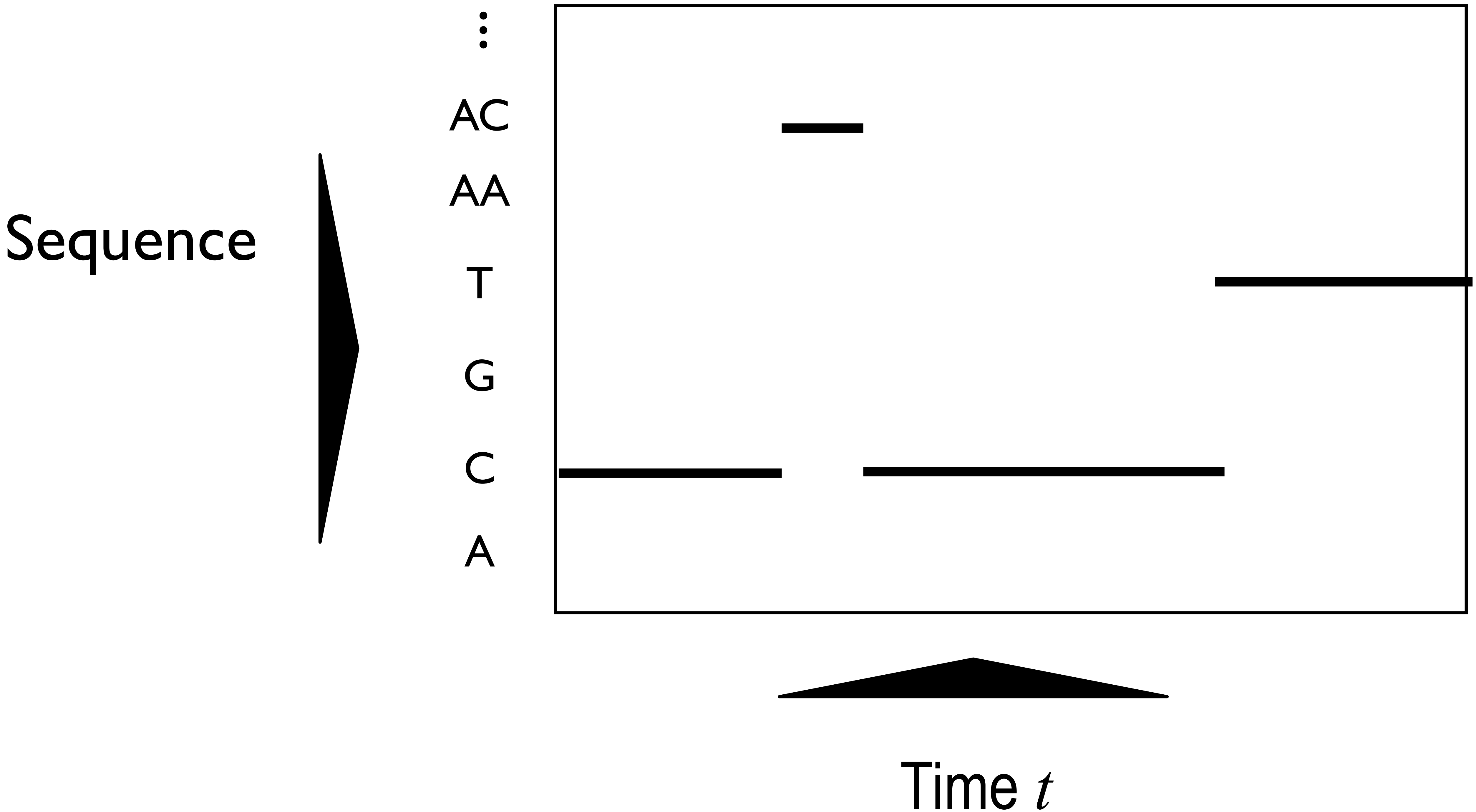
Input: a sequence for each population (taxon)

Taxa		
	a:	C A T A C
	b:	C A G
	c:	A T C C

Output: phylogenetic tree (among other things)

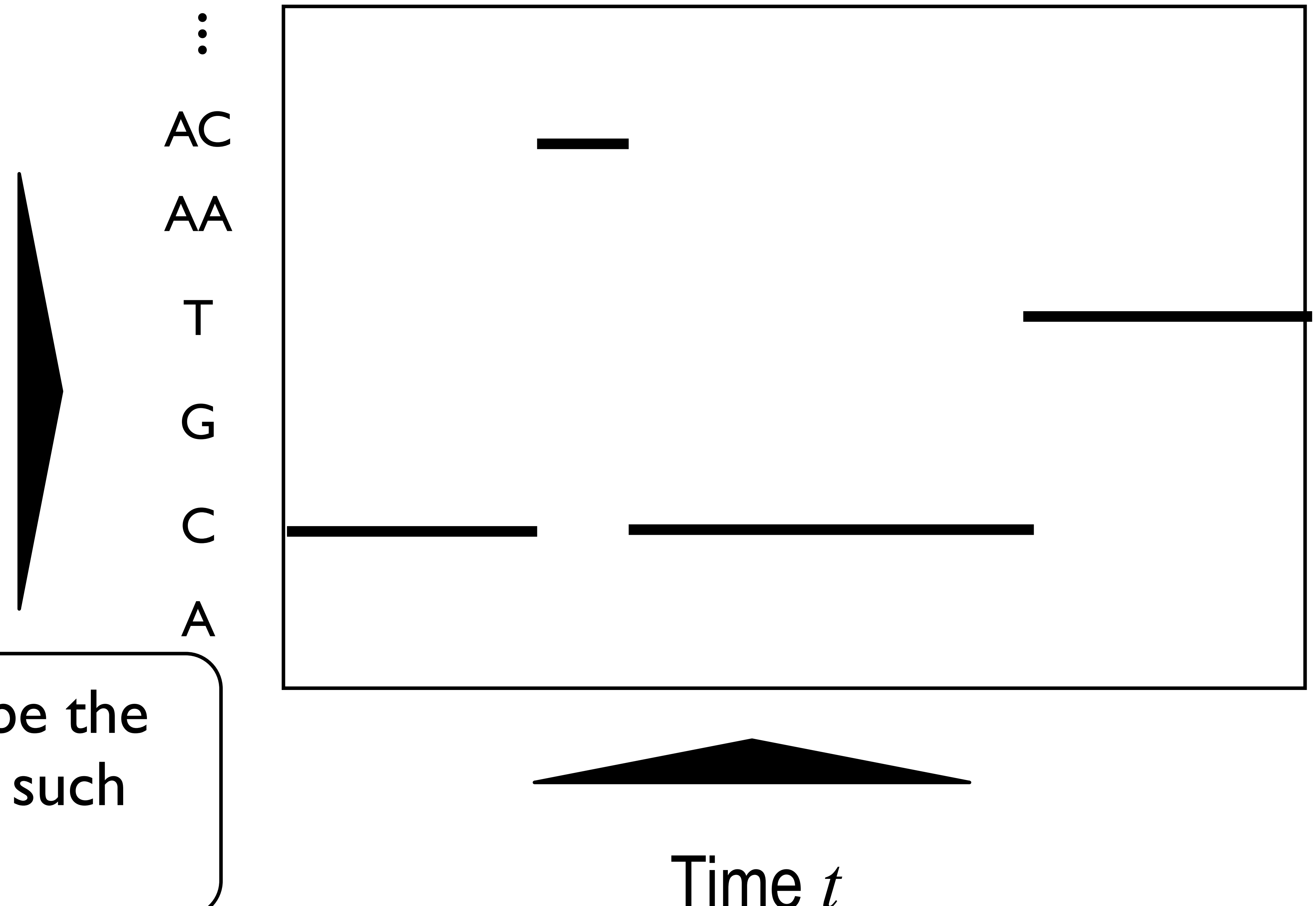
Type of process needed

String-valued stochastic process (instead of real-valued)



Type of process needed

String-valued stochastic process (instead of real-valued)



What could be the marginals of such object?

‘Engineering’/machine learning applications

Domain adaptation: doing the same *task* (for example sentiment analysis) over two *domain* (books, vs. kitchen appliances)

amazon.com

Running with Scissors

Title: Horrible book, horrible.

This book was horrible. I read half, suffering from a headache the entire time, and eventually i lit it on fire. 1 less copy in the world. Don't waste your money. I wish i had the time spent reading this book back. It wasted my life

Avante Deep Fryer; Black

Title: lid does not work well...

I love the way the Tefal deep fryer cooks, however, I am returning my second one due to a defective lid closure. The lid may close initially, but after a few uses it no longer stays closed. I won’t be buying this one again.

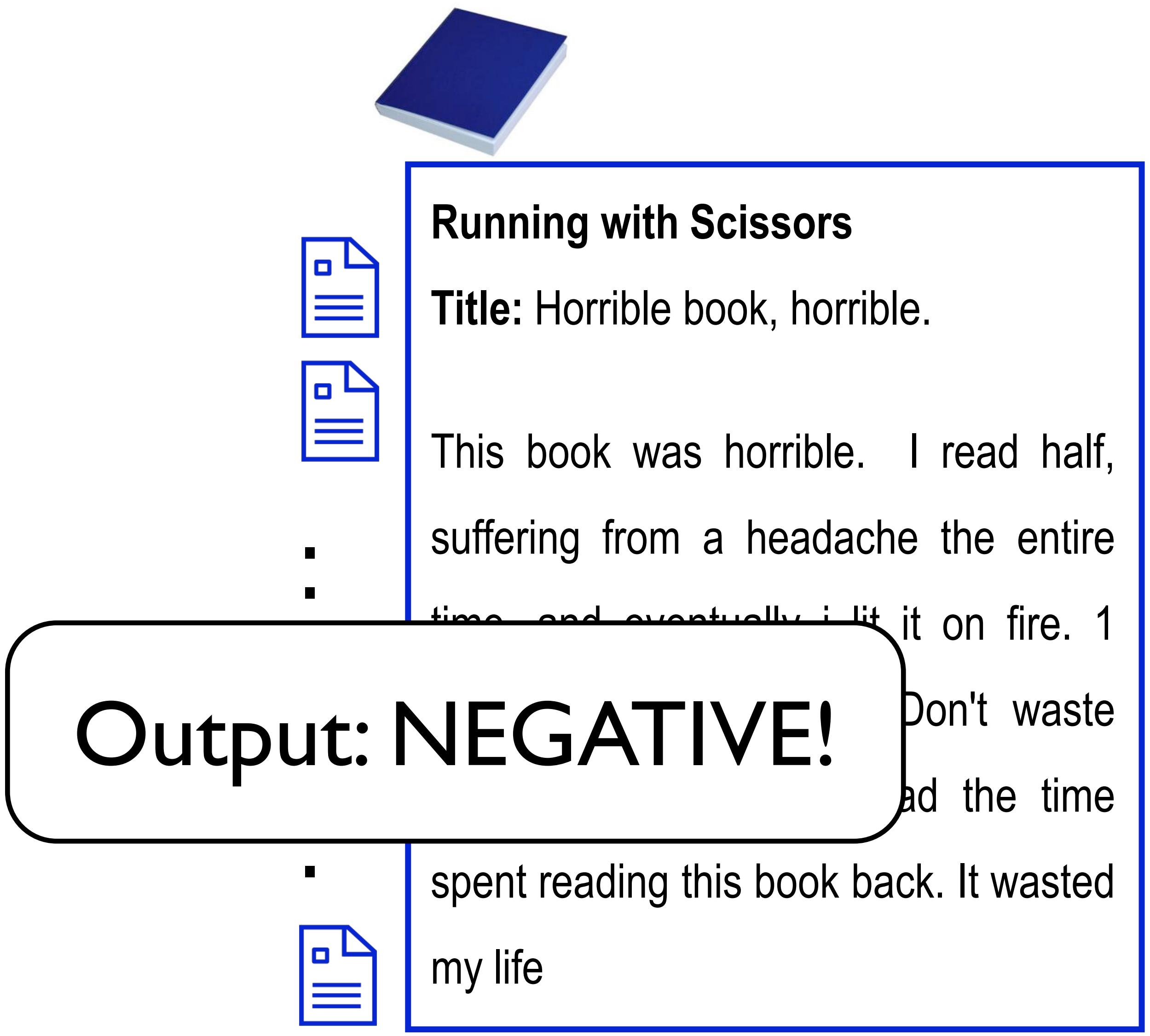
NEGATIVE!

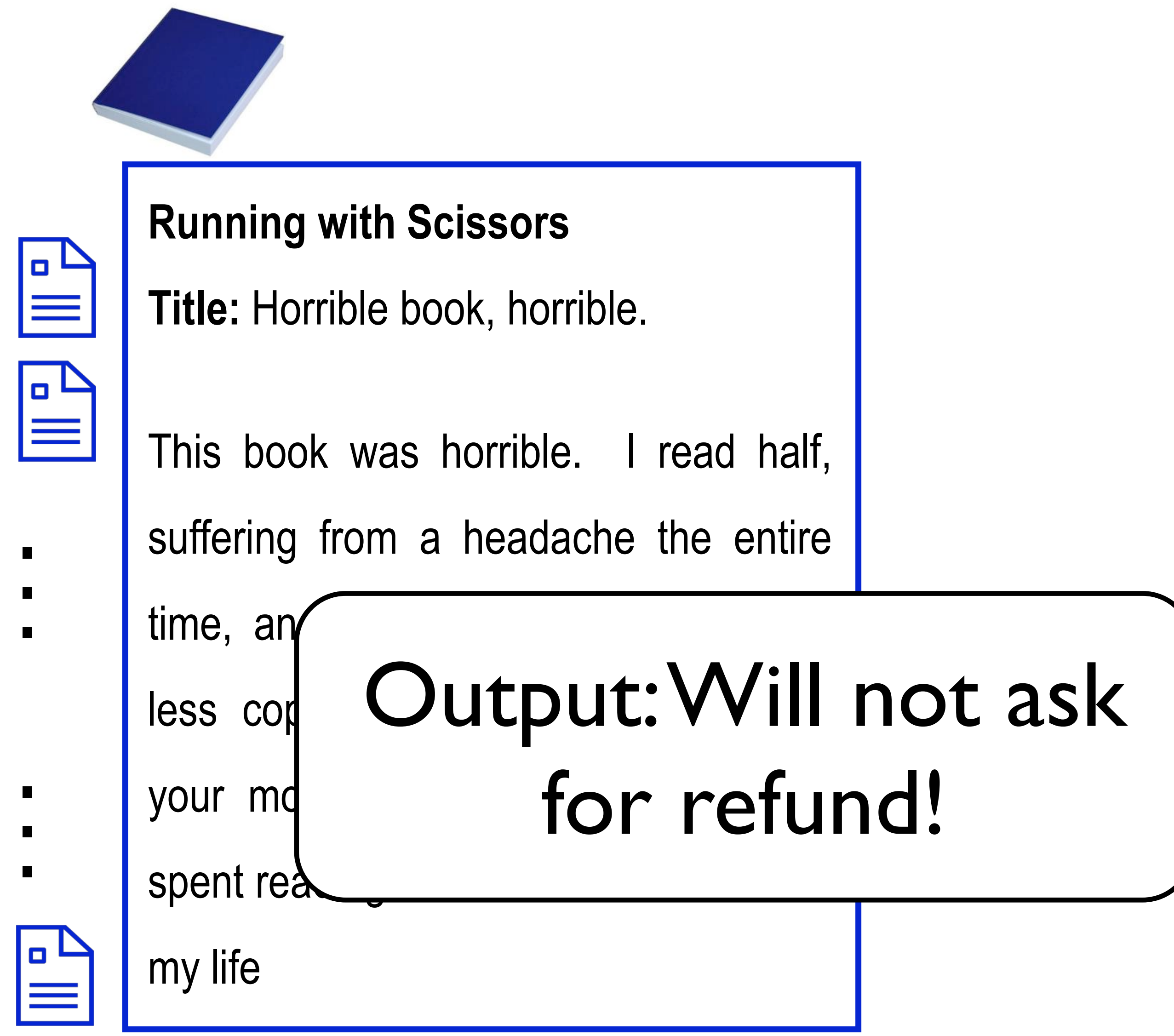
Output: NEGATIVE!

Slide from John Blitzer

‘Engineering’/machine learning applications

Multi-task learning: doing two *tasks* (sentiment analysis vs. predicting if a customer will ask for refund) over one *domain*





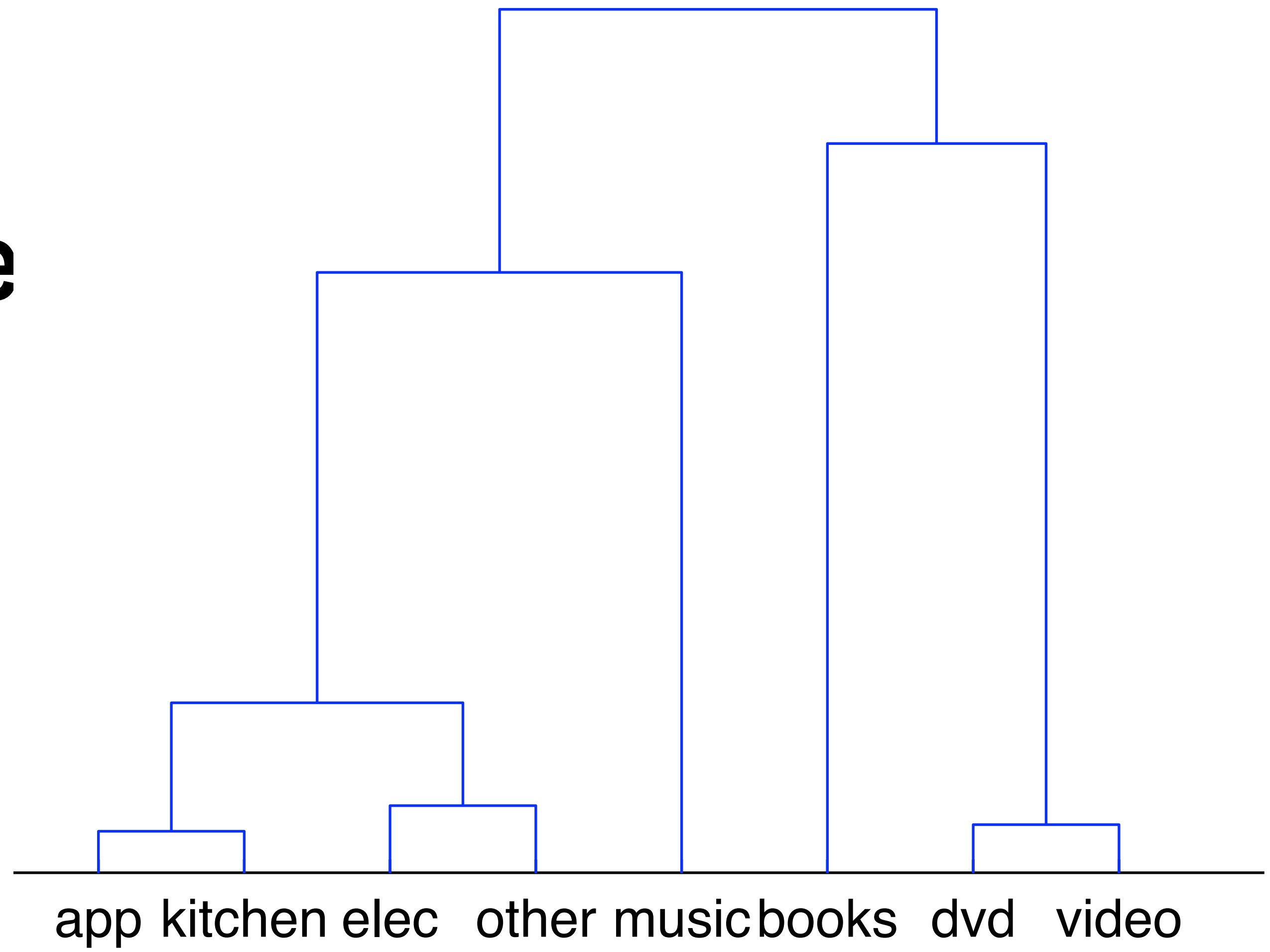
Slide from John Blitzer

Phylogenetic tree application

Task: doing classification over more than two domains or tasks

Latent variable: a tree mirroring how closely related tasks/domain are

Domain adaptation example



Computational Issues

Lazy computation

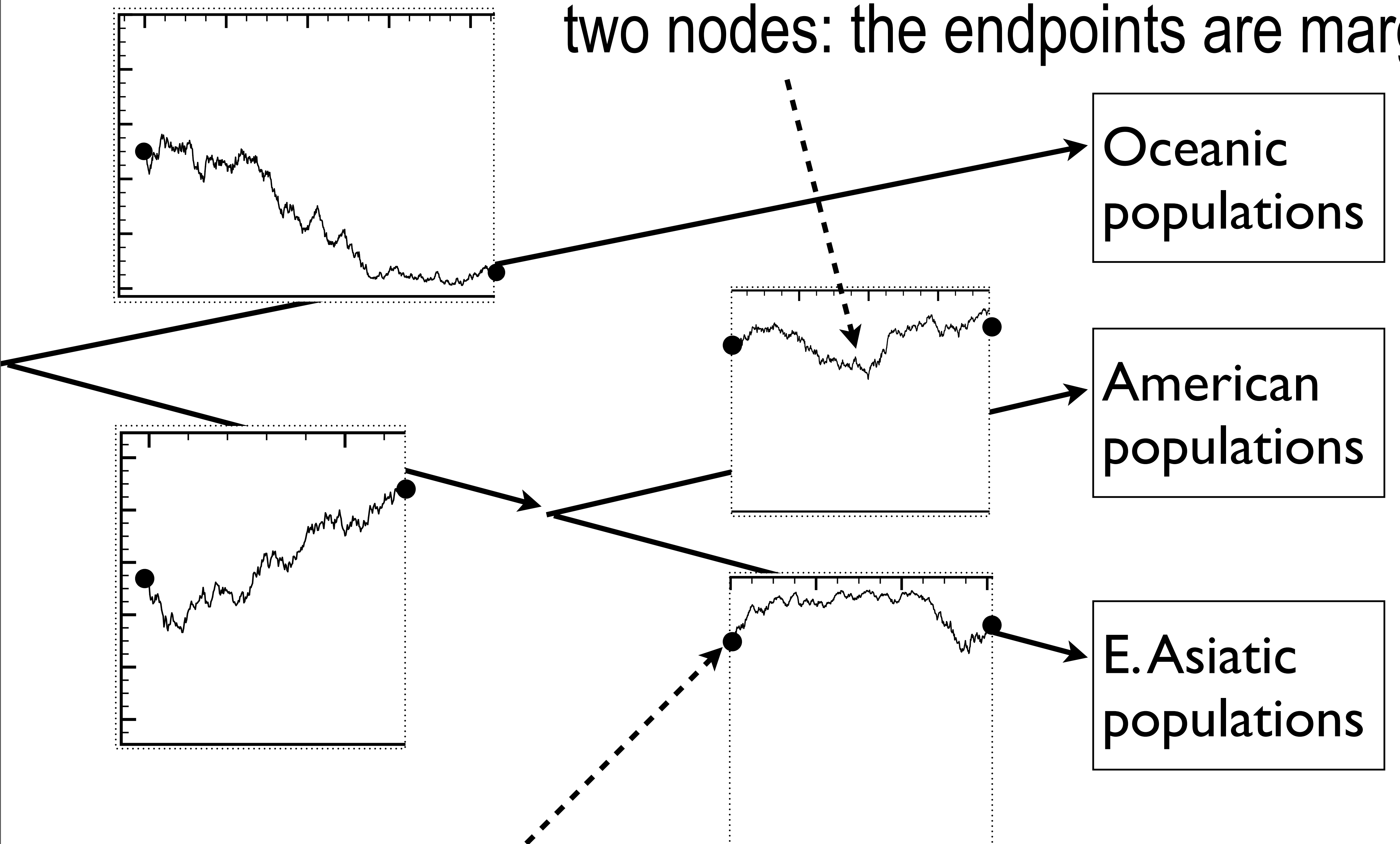
We have introduced prior over infinite support distributions, transition matrices, feature vectors, etc.
If we cannot even represent a single *sample,* how are we going to be able to do inference?

General principle: lazy computation. Represent some parts of the samples implicitly. If we can show that a part of the sample will not affect the answer, don’t store it in memory!

Does *not* mean we can replace these priors by finite support equivalents: we don’t know a priori which part of the sample we will be able to ignore.

Easy example

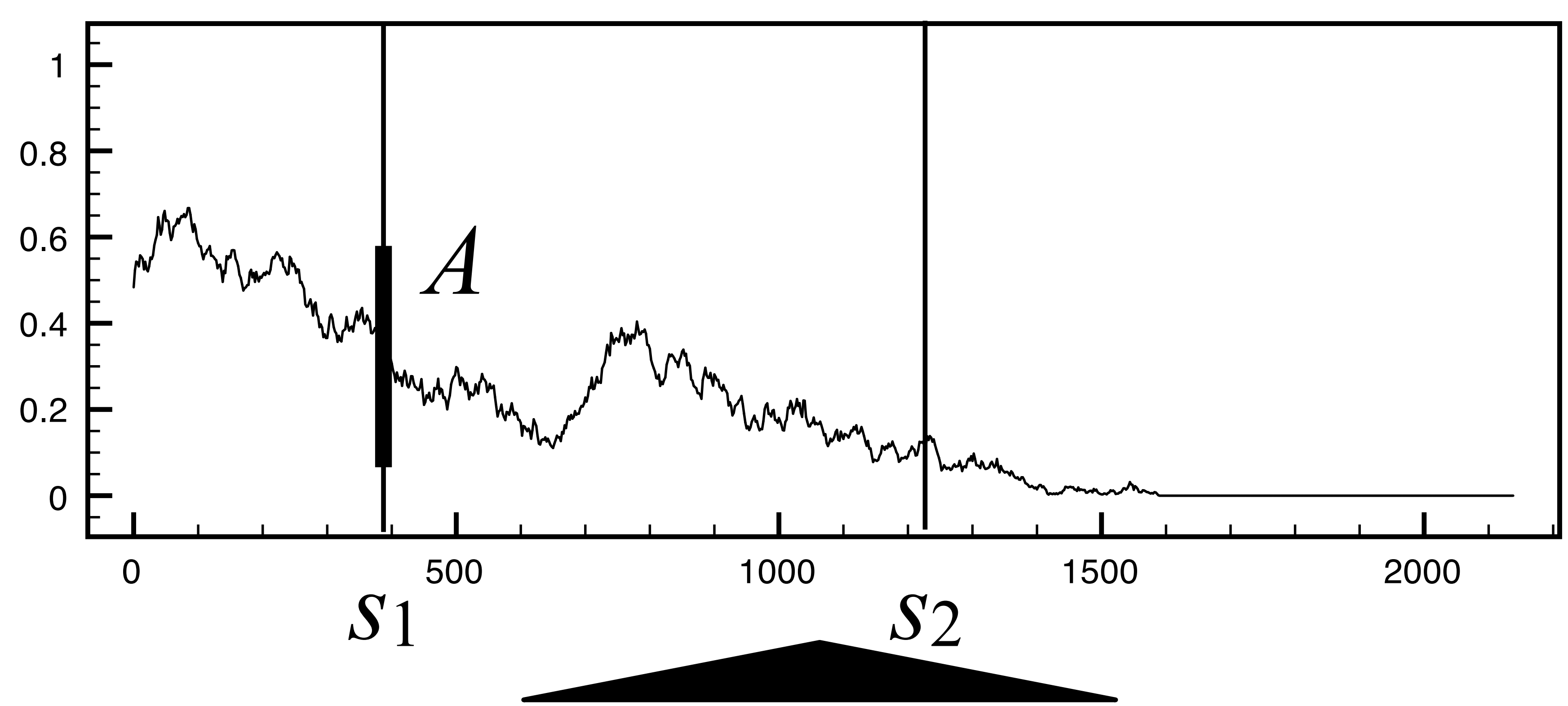
Don't need to keep track of the values between two nodes: the endpoints are marginally normal



Only marginal at the internal nodes need to be maintained; **Note:** tree unknown, so we don't know a priori what are the internal nodes

Why do we know the marginals? By definition!

What are the bare minimum conditions for λ to be marginals of Y_s ?



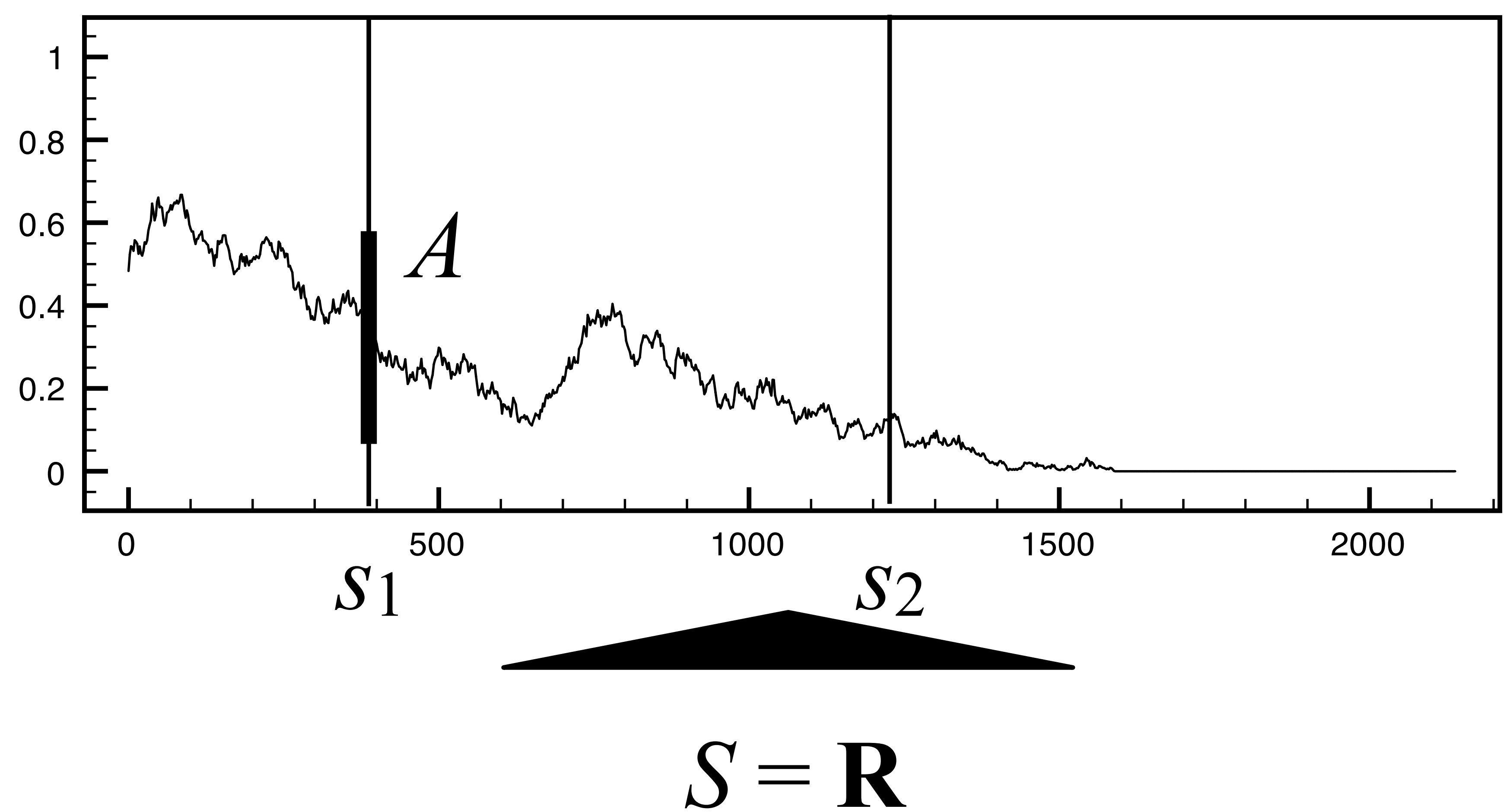
$S = \mathbf{R}$

Why do we know the marginals? By definition!

What are the bare minimum conditions for λ to be marginals of Y_s ?

$$\lambda_{s_1}(A) = \lambda_{s_1,s_2}(A, \mathbf{R}) \quad \text{[marginalization]}$$

$$\lambda_{s_1,s_2}(A_1, A_2) = \lambda_{s_2,s_1}(A_2, A_1) \quad \text{[perm]}$$



Why do we know the marginals? By definition!

What are the bare minimum conditions for λ to be marginals of Y_s ?

$$\boxed{\begin{array}{ll} \lambda_{s_1}(A) = \lambda_{s_1,s_2}(A, \mathbf{R}) & \text{[marginalization]} \\ \lambda_{s_1,s_2}(A_1, A_2) = \lambda_{s_2,s_1}(A_2, A_1) & \text{[perm]} \end{array}}$$

Kolmogorov: if these *consistency* conditions hold for any finite number of variables (not just a pair), then there is a stochastic process with these marginals.

Brownian motion: take λs_i to be multivariate normal distributions with sparse covariance depending on $\{s_i\}$

Less obvious cases

In other cases, the original process definition might not be amenable to efficient inference.

Fortunately, many equivalent representation often exist

Slide from Kurt Miller

Image from http://www.nature.com/nsmb/journal/v7/n6/fig_tab/nsb0600_443_F1.html

Less obvious cases

In other cases, the original process definition might not be amenable to efficient inference.

Fortunately, many equivalent representation often exist

- Stick breaking
- Levy construction
- Chinese Restaurant
- Polya urn

Slide from Kurt Miller

Approximations are often needed

- Monte Carlo
  - MCMC (Markov Chain) and SMC (Sequential)
  - Slice and other auxiliary variables, split-merge, type-level and collapsed samplers

- Variational
  - Legendre-Fenchel transformation
  - Standard relaxations

Background: back to the game

Distribution identity

If $X,\ Y$ are independent Gamma's with the same scale parameter, what is the distribution of $X\,/\,(X+Y)$

A Uniform

B Beta

What a Bayesian would do if...

They would optimize...

Y : Observations
X : Latent
L : Loss function
(strictly convex say)

A $$\mathrm{argmin}_x L(x, \mathbb{E}(X|Y))$$

B $$\mathrm{argmin}_x \mathbb{E}(L(x, X)|Y)$$

Explanation of the question

Task: given an observed random variable Y, what value should we guess for a related random variable X which is unobserved?

Example: Y are observed UBC students heights, assumed to be iid, and normally distributed with unknown mean X

Criterion: if we make a guess x and the real value is x^*, we pay a cost of $L(x,x^*)$ --- this is called a *loss function*.

The Bayesian choice

Task: given an observed random variable Y, what value should we guess for a related random variable X which is unobserved?

Criterion: if we make a guess x and the real value is x^*, we pay a cost of $L(x,x^*)$ --- this is called a *loss function*.

In the Bayesian framework: you should answer

$$\boxed{\operatorname{argmin}_x \mathbb{E}(L(x, X)|Y)}$$

Argument for and against using a Bayes estimator

Pros:

- Easy to create ‘good’ estimators handling missing data, prior knowledge
- Automatic framework for shrinkage and regularization
- Certain optimality guarantees when the model is correct (consistency, admissibility)--more on that later

Cons:

- Can lack robustness to model misspecification
- Often needs to be approximated, so sometimes it might be possible to exactly compute a statistically suboptimal estimator and get a better end result in practice

Pro and con:

- For large amount of data, prior is washed out.

The Bayesian choice: examples

Example 1: Suppose X is discrete, i.e. $X \dot{\in} \{1, 2, \ldots N\}$

Computing the Bayes estimator:

$$\mathbb{E}(L(1,X)|Y) = \sum_{x=1}^{N} L(1,x)\mathbb{P}(X = x|Y)$$

$$\mathbb{E}(L(2,X)|Y) = \sum_{x=1}^{N} L(2,x)\mathbb{P}(X = x|Y)$$

$$\vdots$$

$$\mathbb{E}(L(N,X)|Y) = \sum_{x=1}^{N} L(N,x)\mathbb{P}(X = x|Y)$$

Return the index of the minimum of these numbers

The Bayesian choice: examples

Model: Y are observed UBC students heights, assumed to be iid, and normally distributed with unknown mean X

Example 2: Suppose $L(x,x^*) = (x - x^*)^2$

Computations

Discrete case: When X is discrete the posterior,

$$P(X = x|Y)$$

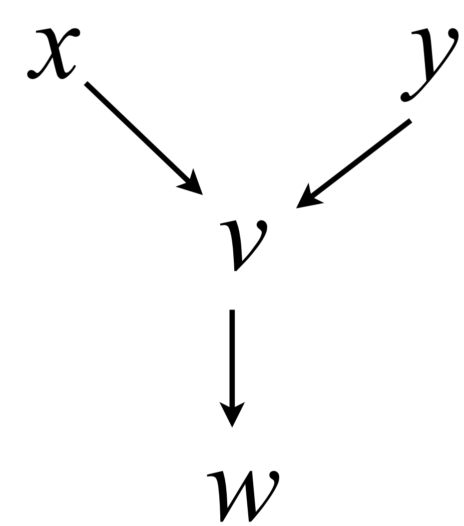
is often (but not always) the computational bottleneck when dealing with Bayes estimators.

Continuous case: When X is continuous and conjugate, computing the posterior can often (but not always) be done by computing the parameters of the posterior.

In both cases, computing the posterior can be intractable.
What's next: how to compute and approximate posteriors

Graphical models

Consider the following graphical model and conditional independence statement:



'Given w, x is indep. of y'

A The statement is always true

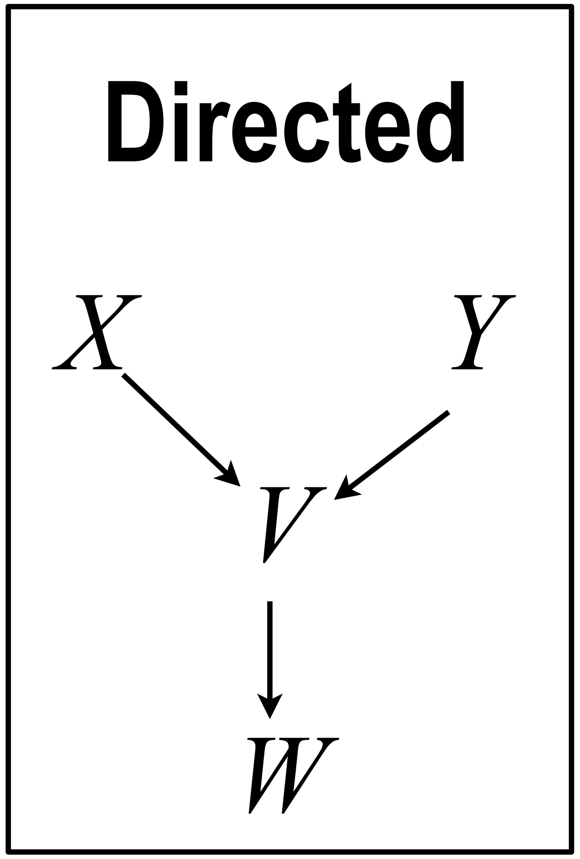
B The statement is not necessarily true

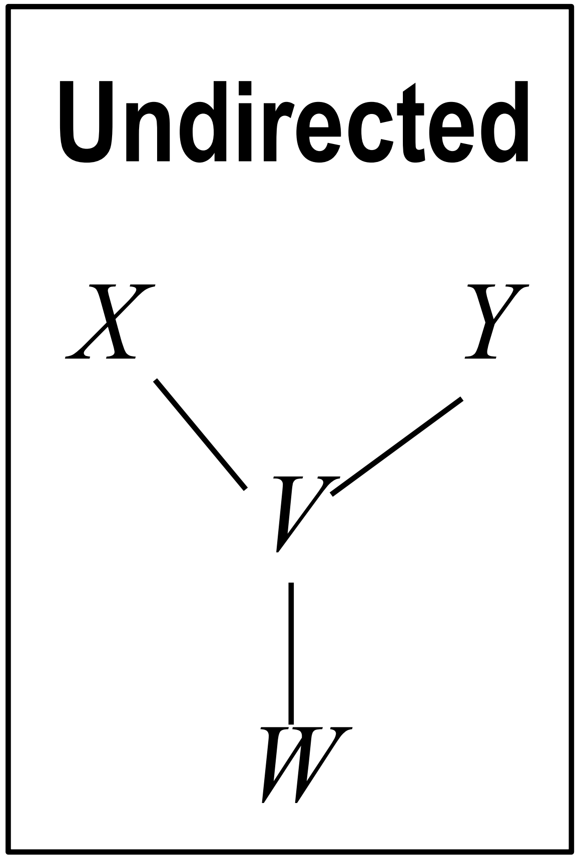
Review: graphical models

What they are: Graphs where nodes are random variables.

What is their use: A language for expressing conditional independence statements. Formally: a graphical model corresponds to a family of probability distributions.

Two types:





Directed Graphical Models

Basic fact: any joint density can be written as a product of conditional densities, one for each random variable.

Example: $p(x,y,z) = p_1(x)\ \ p_2(y|x)\ \ p_3(z|x,y)$

Sometimes: Some of the conditionals can be simplified

Example: $p_3(z|x,y) = p'_3(z|y)$ i.e. $X \perp Z \mid Y$

Directed graphical model: for each conditional, add an edge between each variable we condition on into the current variable.

Example: $X \longrightarrow Y \longrightarrow Z$

Directed Graphical Models

Example: $X \longrightarrow Y \longrightarrow Z$

Interpretation: the collection of all distributions that can be factorized as

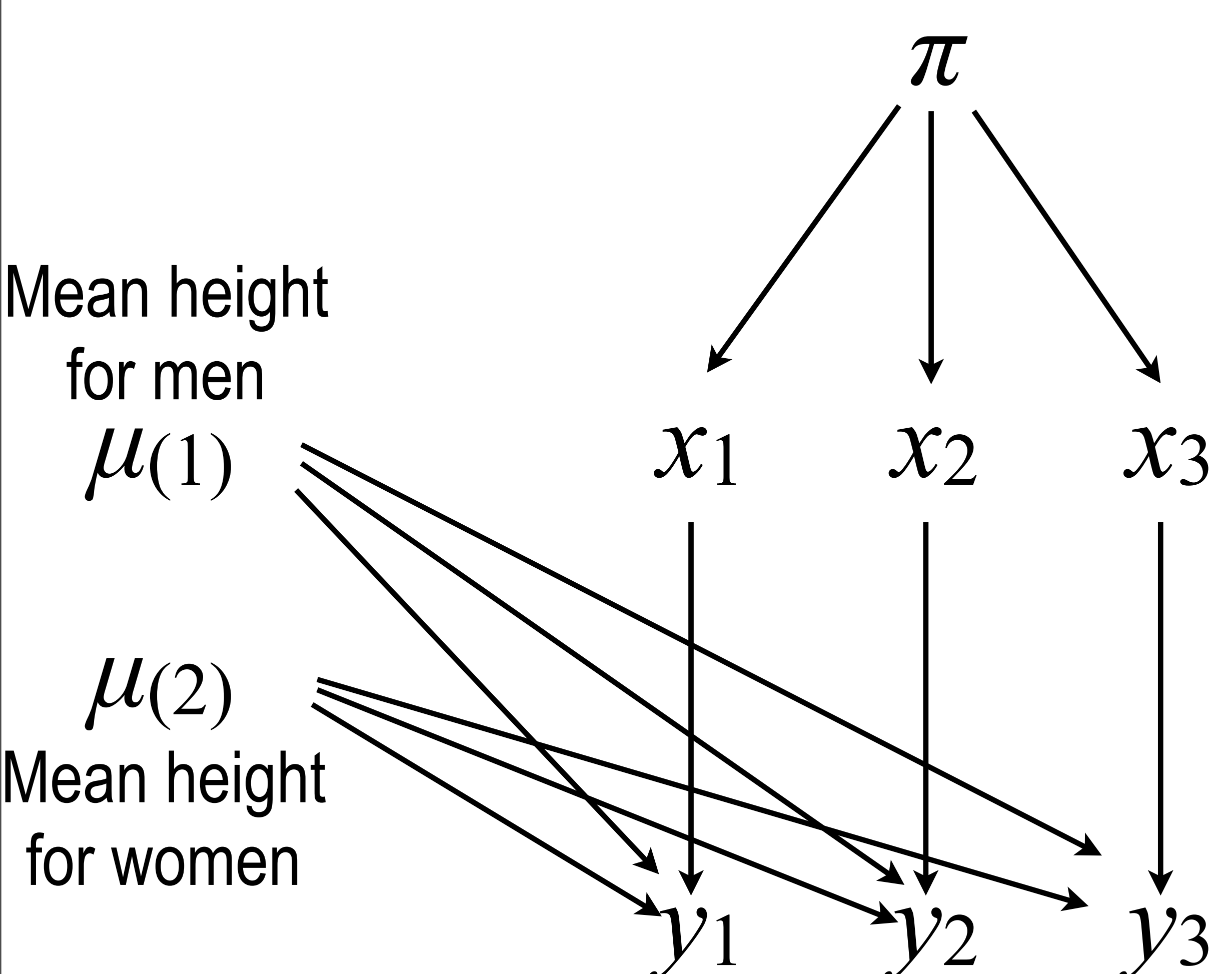
$$p(x,y,z) = p_1(x)\ \ p_2(y|x)\ \ p_3(z|y)$$

for some non-negative p_i s such that for each w:

$$\int\ p_i(v|w)\ m(\mathrm{d}v) = 1$$

Directed graphical models: important examples

Mixture model: (UBC student height with 2 components)
say we have only 3 observations



1- Generate a male/female relative frequence

$$\pi \sim \text{Beta}(\text{male prior pseudo counts, female P.C})$$

2- Generate the sex of each student for each i

$$x_i \mid \pi \sim \text{Mult}(\pi)$$

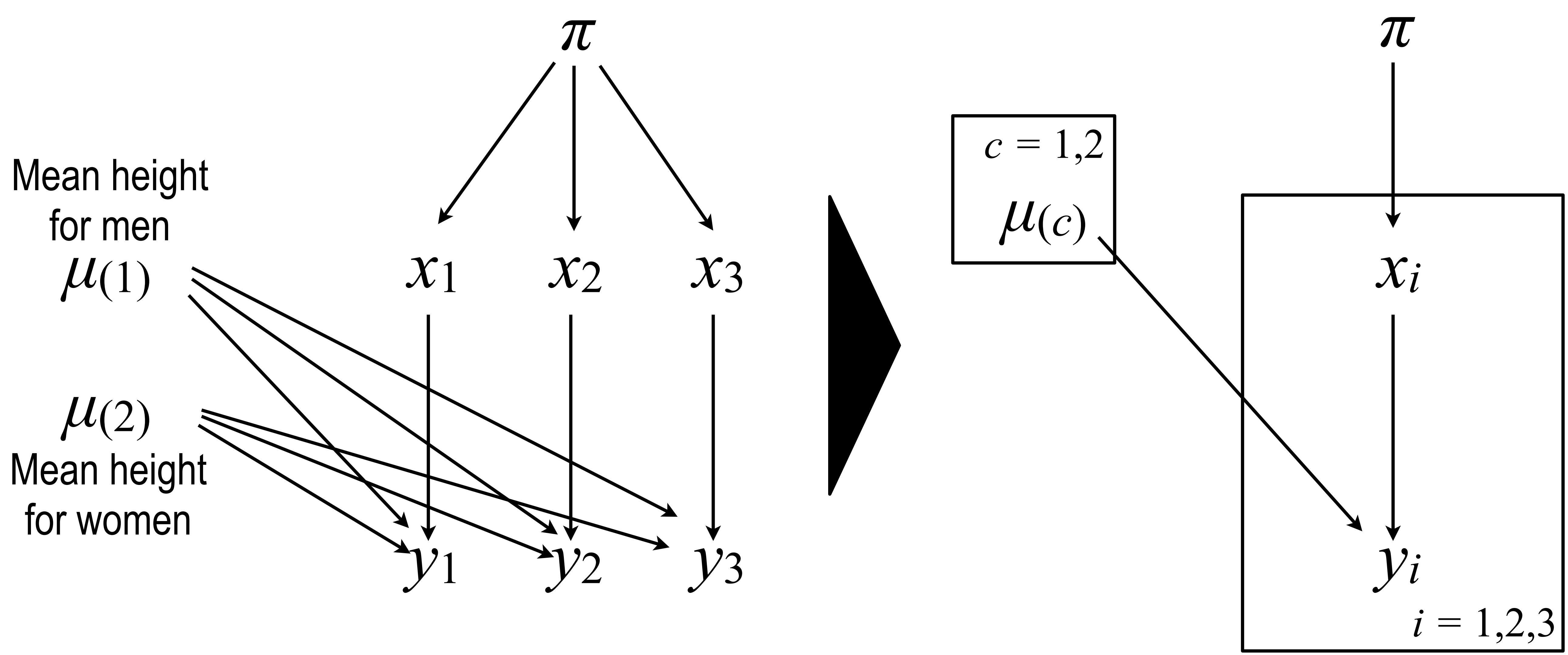
3- Generate the mean height of each cluster c

$$\mu_{(c)} \sim \text{N}(\text{prior height, how confident prior})$$

4- Generate student heights for each i

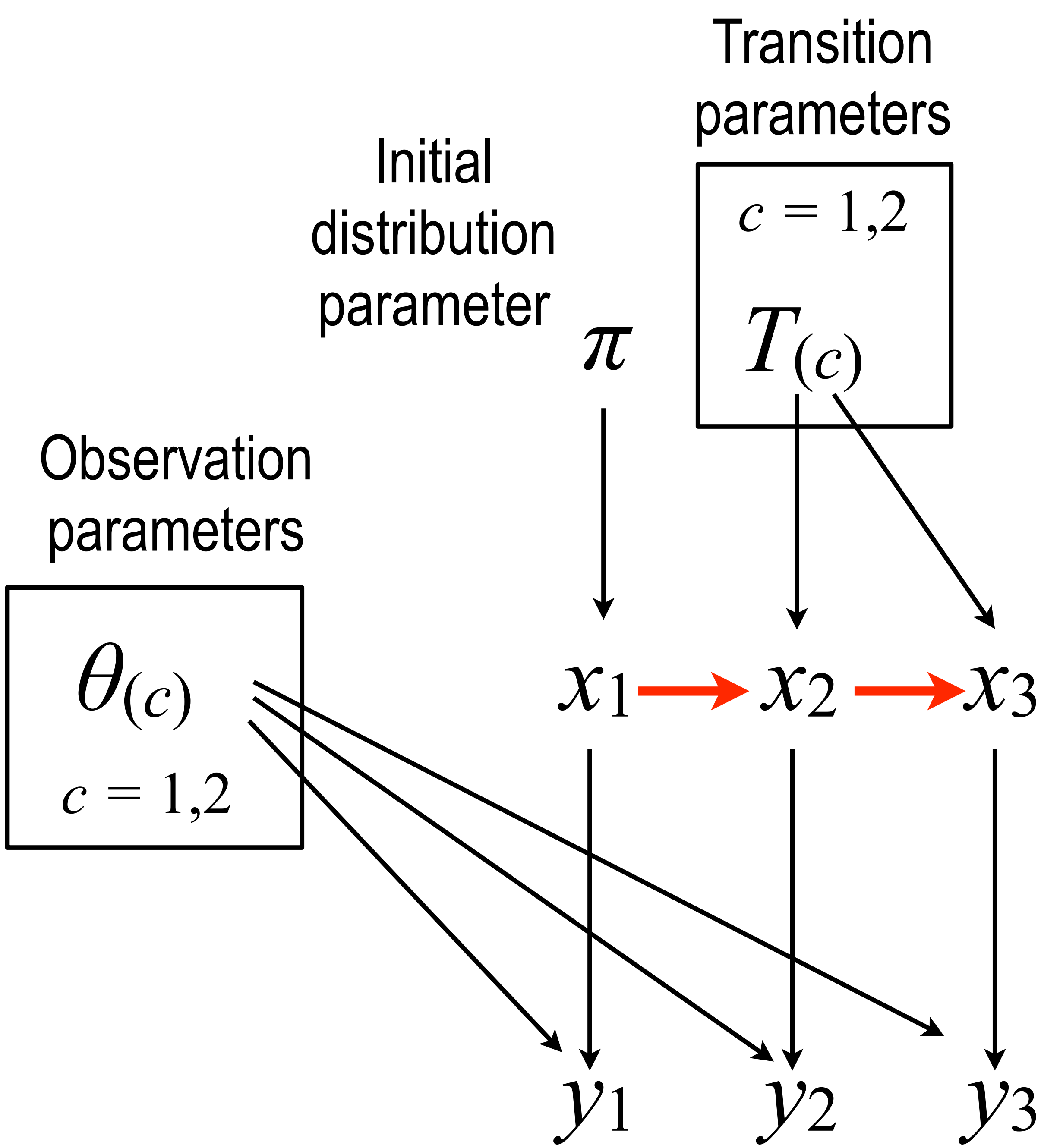
$$y_i \mid x_i, \mu_{(1)}, \mu_{(2)} \sim \text{N}(\mu_{(x_i)}\text{ ,variance})$$

Plate notation



Directed graphical models: important examples

Hidden Markov Model (HMM) (two hidden states, discrete time)



1- Generate an initial distribution parameter

$$\pi \sim \text{Beta}(\text{first cluster's P.C., other's P.C})$$

2- Generate transition param.: the distribution over next hidden state for each hidden state c

$$T_c \sim \text{Beta}(\text{first cluster's P.C., other's P.C})$$

3- Generate the hidden states at each time i

$$x_i \mid \pi, x_{i-1} \sim \text{Mult}(T(x_{i-1}))$$

4- Generate the observation parameter: distribution over observations for each cluster c

$$\theta_{(c)} \sim \text{Beta}(\text{first observation's P.C., other's P.C})$$

5- Generate observation at each time i

$$y_i \mid x_i, \theta_{(c)} \sim \text{Mult}(\theta(x_i))$$

Directed graphical models

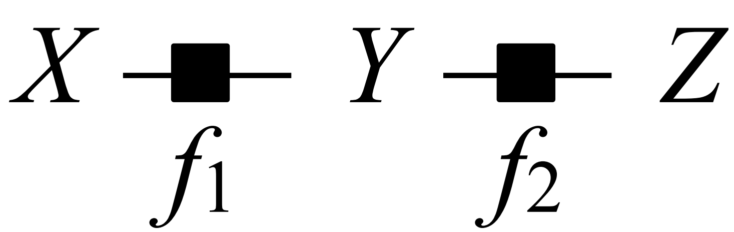
Summary: directed graphical models are convenient to describe a model (a 'generative story')

Caveat: it takes more work to find what are the conditional independence statements implied by directed graphical models..

Undirected Graphical Models

As in directed graphical models, we start by factorizing the joint density, but this time, the factors are *not* required to be conditional or marginal distributions.
Example: $p(x,y,z) = f_1(x,y)\ f_2(y,z)$

Undirected graphical model: for each factor, add a square connecting the variables appearing in this factor
Example:



Undirected Graphical Models

Example: $X - \blacksquare - Y - \blacksquare - Z$

Interpretation: the collection of all distributions such that their density that can be factorized as

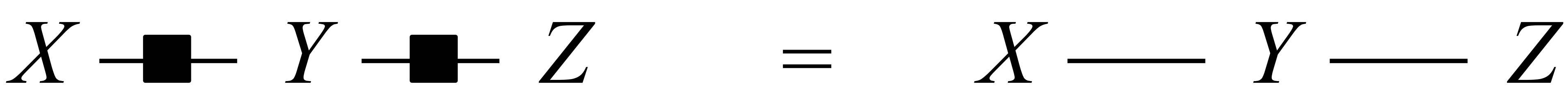
$$p(x,y,z) = f_1(x,y)\ f_2(y,z)$$

for some non-negative f_i

Undirected Graphical Models

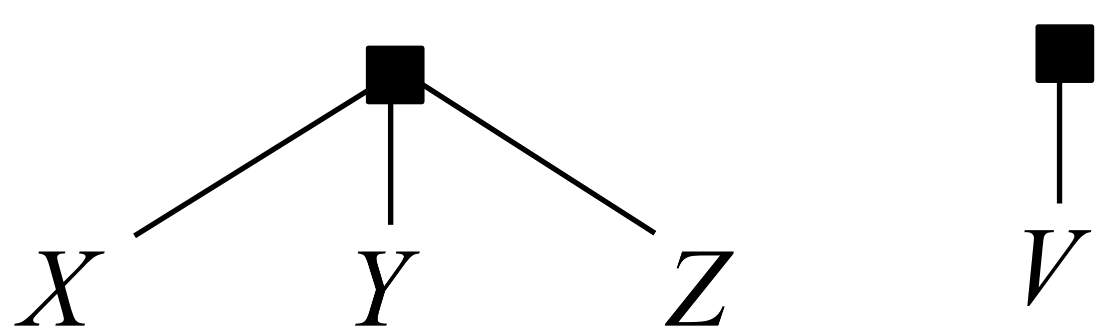
Notation: when a factor links only two nodes, we will not bother drawing it:

Example: $p(x,y,z) = f_1(x,y)\ f_2(y,z)$



Other times, the square will be useful:

Example: $p(v,x,y,z) = f_1(x,y,z)\ f_2(v)$



Undirected Graphical Models

Finding conditional independence statement: easy in undirected graphical models

Example: do we have $X \perp Z \mid V, W$ for all distributions in the collection corresponding to the graphical model below?

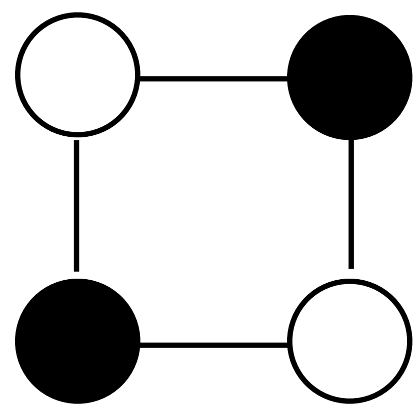
$$\begin{array}{ccc} X & — & W \\ | & & | \\ V & — & Z \end{array}$$

Undirected Graphical Models

Example: do we have $X \perp Z \mid V, W$ for all distributions in the collection corresponding to the graphical model below?

First step: shade the node we are conditioning on

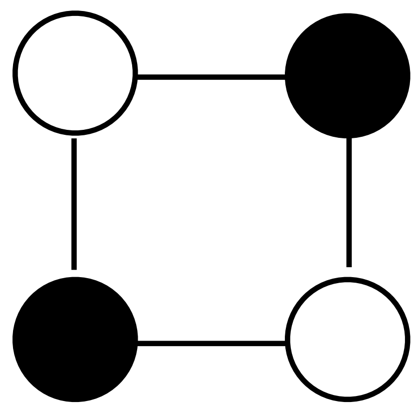
$$\begin{tikzcd} X \arrow[r, dash] \arrow[d, dash] & W \arrow[d, dash] \\ V \arrow[r, dash] & Z \end{tikzcd}$$



Second step: check if there is a path between the two query nodes (X and Z) that does not go a shaded node

Undirected Graphical Models

Example: do we have $X \perp Z \mid V, W$ for all distributions in the collection corresponding to the graphical model below?



First step: shade the node we are conditioning on

Second step: check if there is a path between the two query nodes (X and Z) that does not go a shaded node

If there are no such path: $X \perp Z \mid V, W$ for all distributions in the collection corresponding to the graphical model below

If there is such a path: there *could be* dependence

Undirected graphical models

Summary: undirected graphical models take a bit more work to construct, but they are more useful at inference time (finding independence statement simplifies sums/ integrals)

Connection between directed and undirected

Note: if you have a decomposition for directed models, you can use it to define an undirected model, but the undirected model will have more edges!

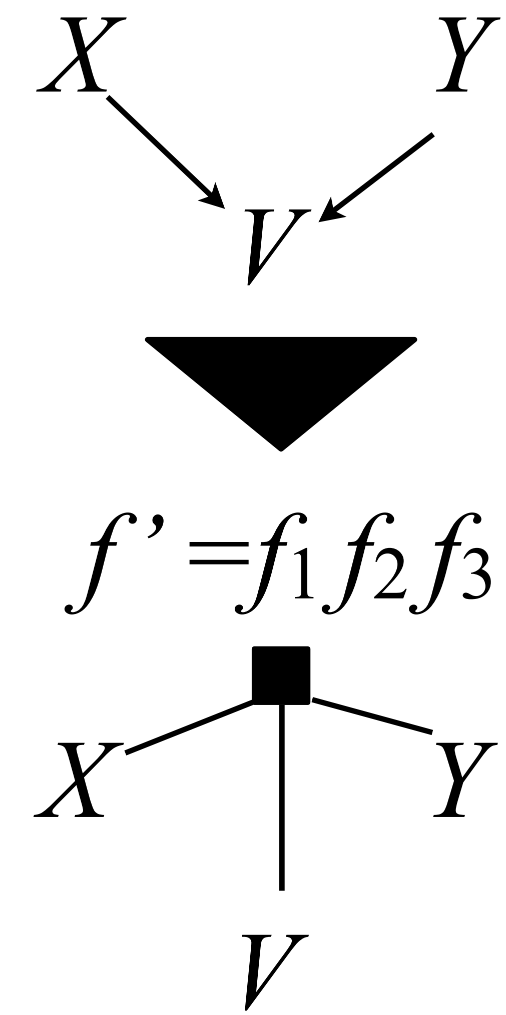
Example:

$$p(x,y,v) = p_1(x)\ \ p_2(y)\ \ p_3(z|x,y)$$

Can be viewed as:

$$f'(x,y,z) = f_1(x)\ \ f_2(y)\ \ f_3(x,y,z)$$

('Moralization')



Where we are headed

Goal: computing the posterior distributions needed for the Bayes estimator

Often (but not always) they correspond to computing the posterior over a single node or a pair of nodes connected by an edge in a graphical model

Example:

Where we are headed

Goal: computing the posterior distributions needed for the Bayes estimator

For now: assume that all the random variables are discrete (will relax this later)

Two cases: If the undirected graphical model...

1. ... is a tree, the posterior can be computed exactly in polynomial time
2. ... is *not* a tree, the posterior usually needs to be approximated using a MC or variational technique

Exact inference and dynamic programming

Example: predicting part of speech (POS)

???	???	???	???
Alex	likes	big	houses

Is ‘houses’ a NOUN or a VERB?

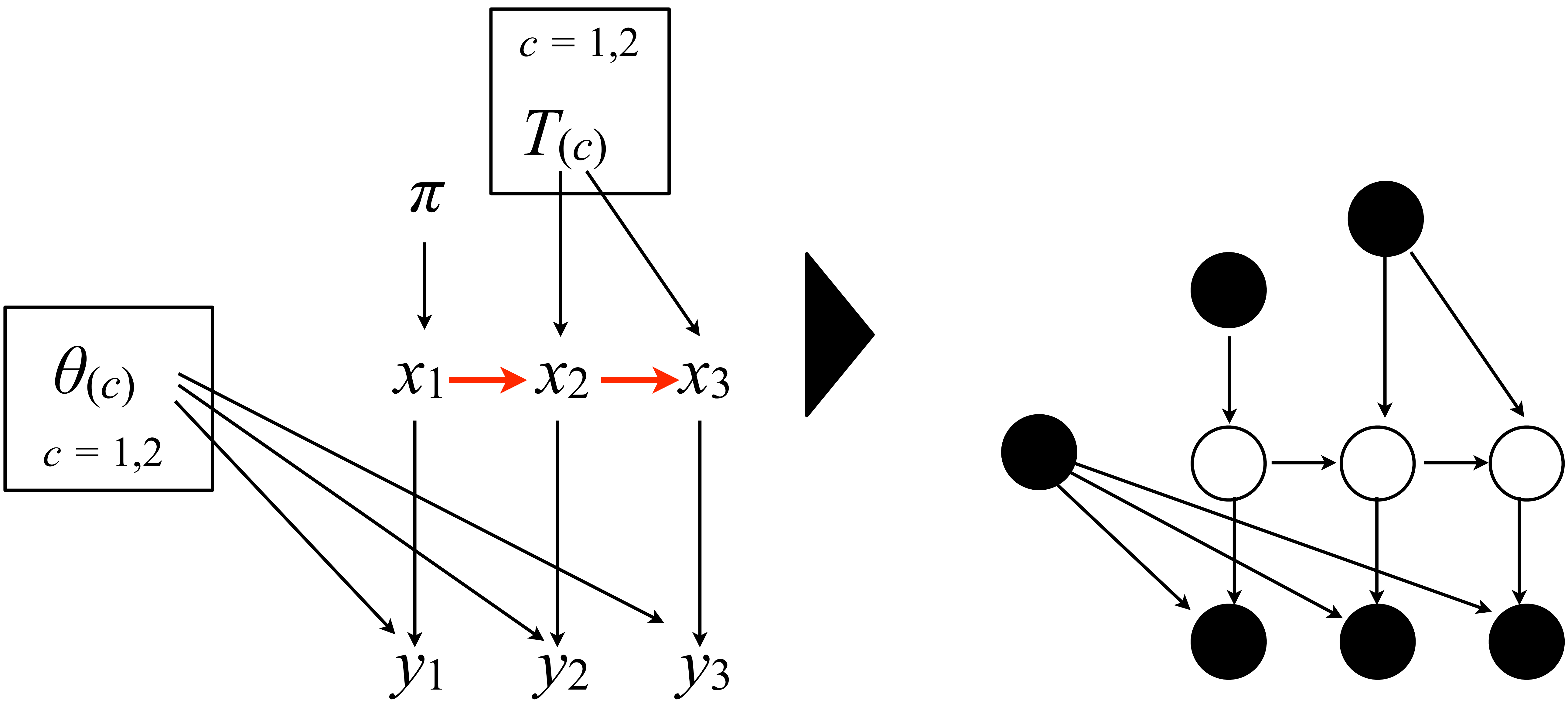
What we want to leverage:

(1) some POS sequences (ngrams) are much more common than others (ADJ NOUN vs. ADJ VERB)

(2) each POS has a different distribution over associated words

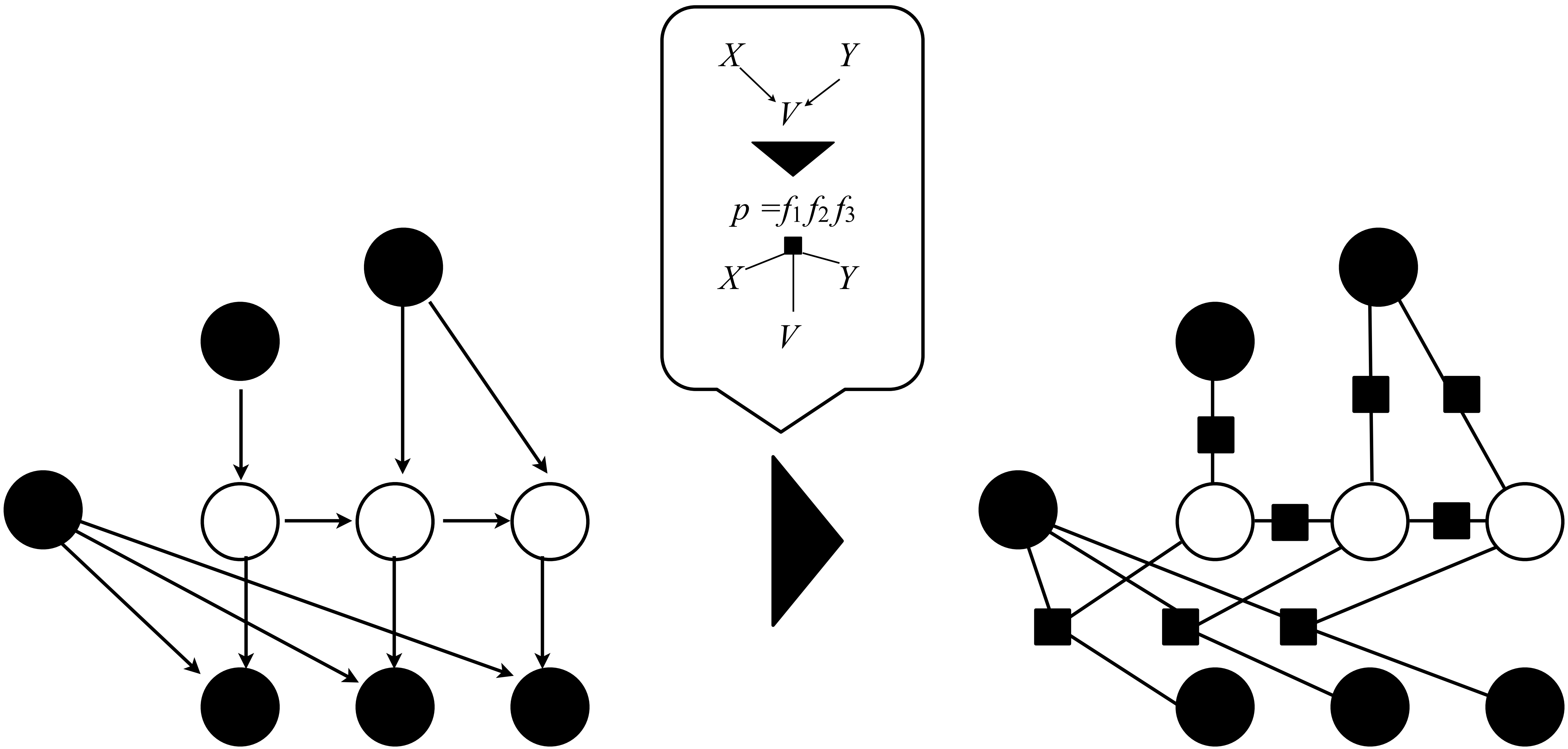
Exact inference and dynamic programming

Suppose: parameters are known, so we condition on them



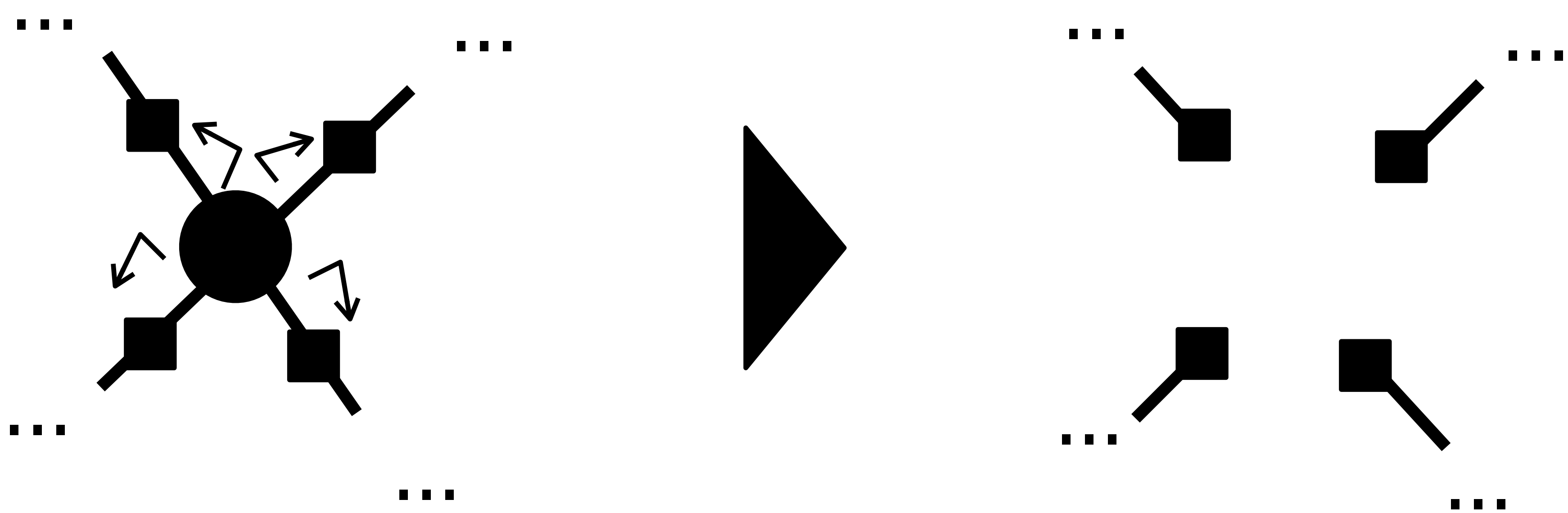
Exact inference and dynamic programming

Next step: turning the directed model into an undirected one



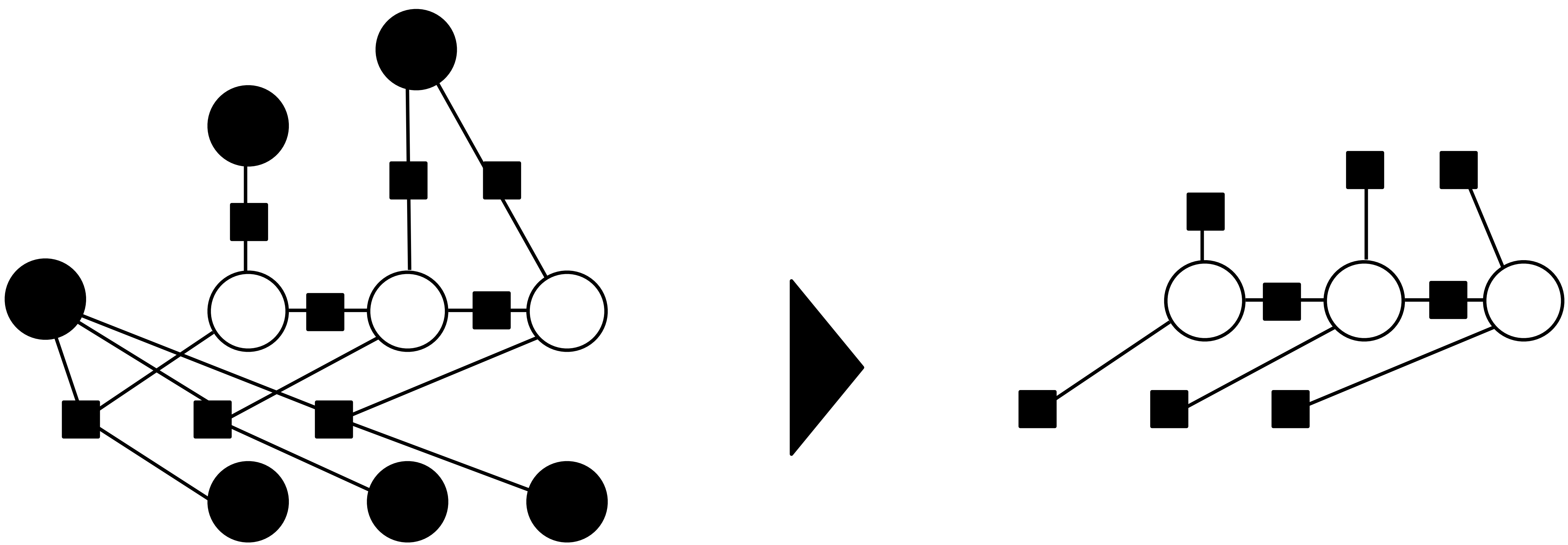
Exact inference and dynamic programming

Simplifying undirected models:



Exact inference and dynamic programming

Simplifications:



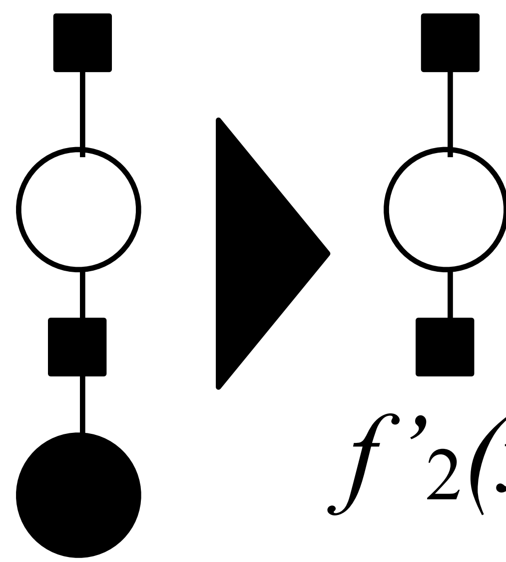
Exact inference and dynamic programming

Consequence of simplification: renormalization needed

Example:

$f_1(x) = p_1(x)$

$f_2(y|x) = p_2(y|x)$



$f_1(x)$

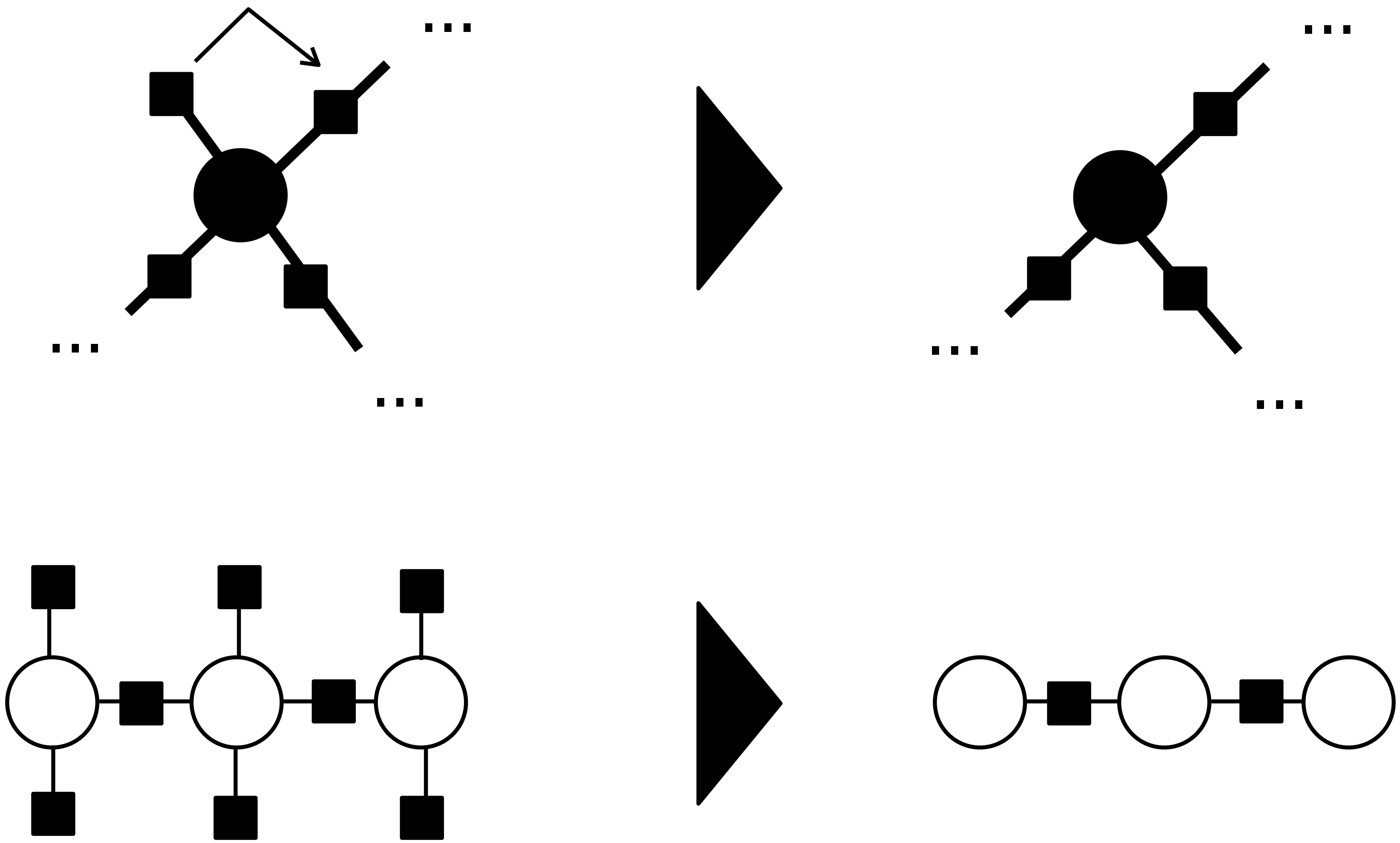
$f'_2(x) = p_2(y_0|x)$

$$P(X = x|Y = y_0) = \frac{f_1(x) f'_2(x)}{\sum_{x'} f_1(x') f'_2(x')}$$

$$= \frac{f_1(x) f'_2(x)}{Z}$$

Bayes rule: can interpret Z as $P(Y = y_0)$

Further simplification



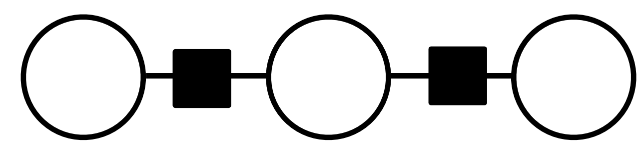
Renormalization

$$P(X_1 = x_1, X_2 = x_2, X_3 = x_3 | \text{params, obs})$$

$$= \frac{f_1(x_1, x_2) f_2(x_2, x_3)}{\sum_{x_1'} \sum_{x_2'} \sum_{x_3'} f_1(x_1', x_2') f_2(x_2', x_3')}$$

$$= \frac{f_1(x_1, x_2) f_2(x_2, x_3)}{Z}$$

$$\propto f_1(x_1, x_2) f_2(x_2, x_3)$$

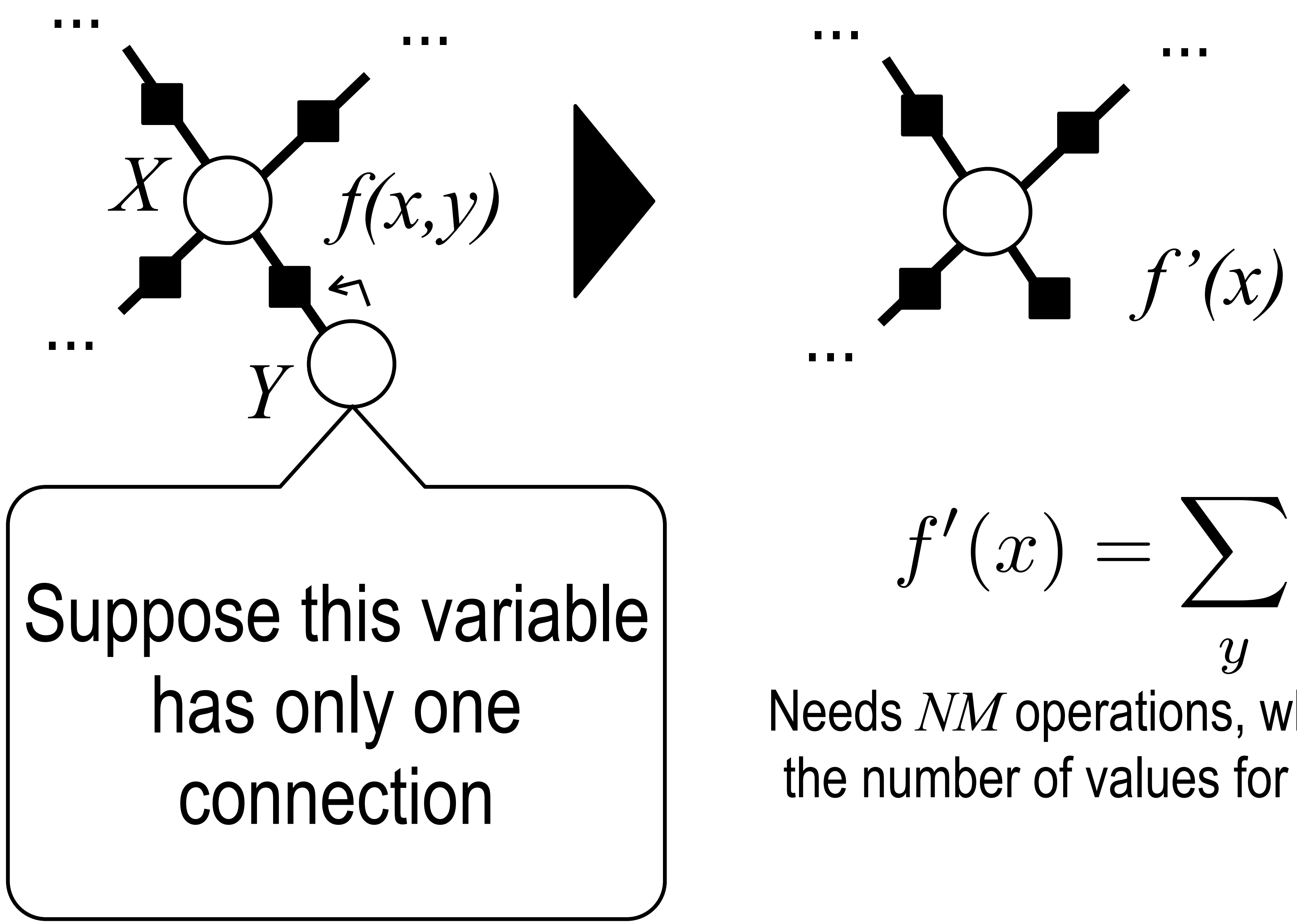


NOUN	NOUN	NOUN
NOUN	NOUN	VERB
NOUN	VERB	ADJ
ADV	VERB	VERB

...

Note: Naive enumeration is expensive!
There are 4 hidden possible POS in the three hidden states, so 4 x 4 x 4 = 64

Another simplification/transformation



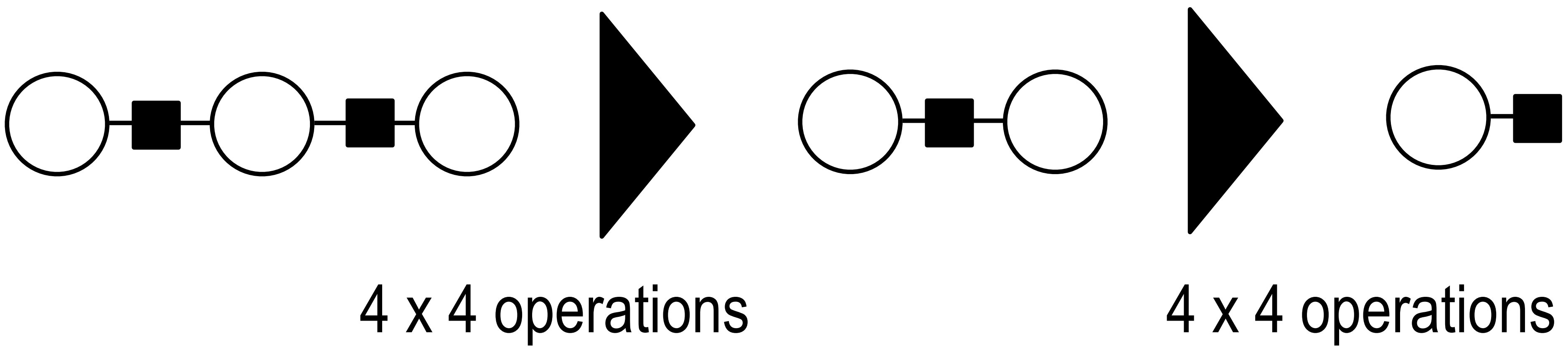
$$f'(x) = \sum_y f(x, y)$$

Needs NM operations, where N, M are the number of values for each variable

Note: still preserves Z and marginals

Efficient inference: elimination algorithm

Consequence: for chains, efficient computation of Z and one-node or two-nodes marginals for tree-shaped undirected graphical models



Less operations than naive enumeration!

In general: if a *chain* has length T and N states, computing Z takes $T\,N^2$ operations instead of N^T

For tree-shaped models: same story!

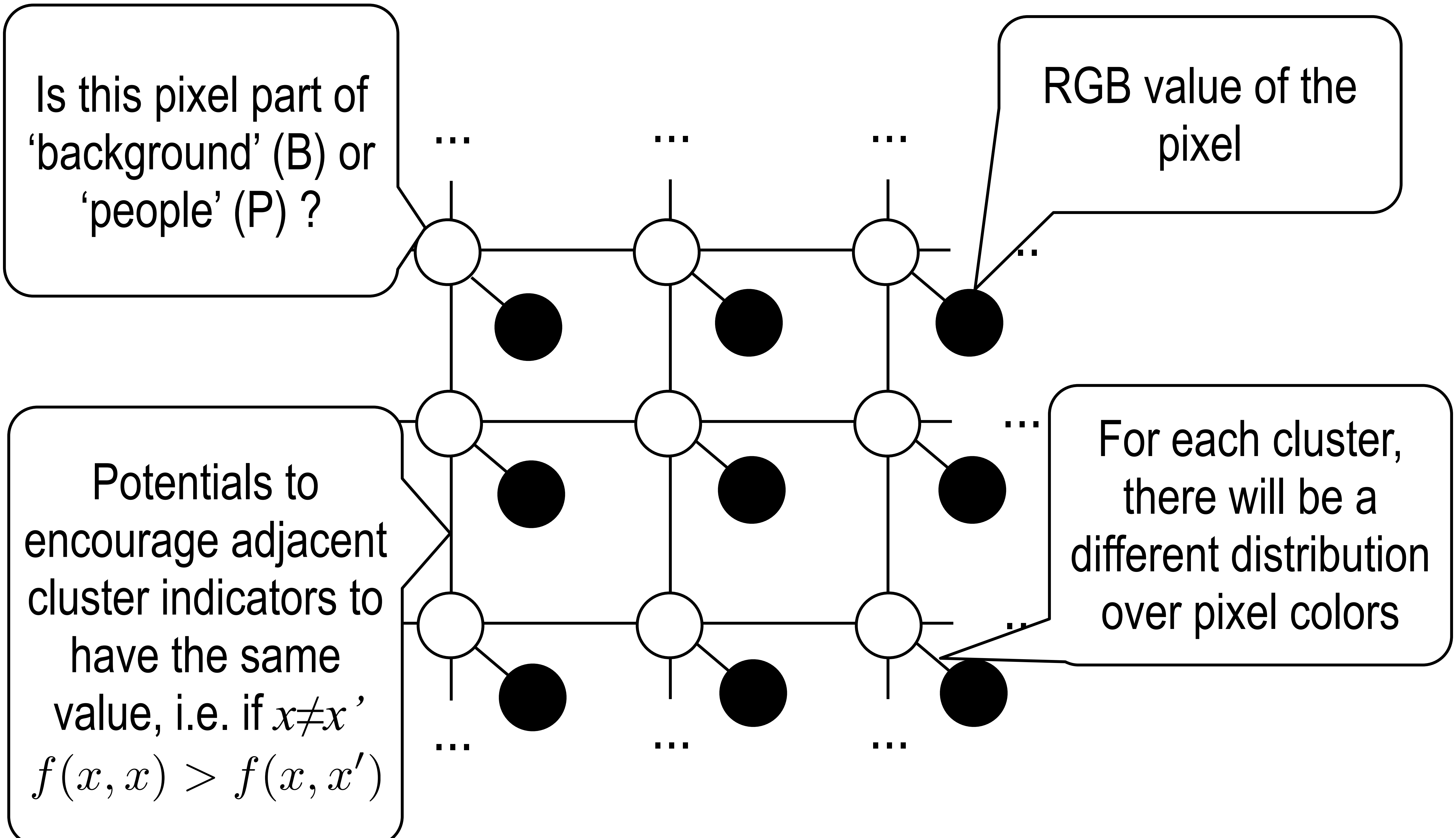
For non-tree models: we need to figure out something else...

Example of a non-tree model

Task: given some images (a 2D array of pixels), segment it into clusters of pixels

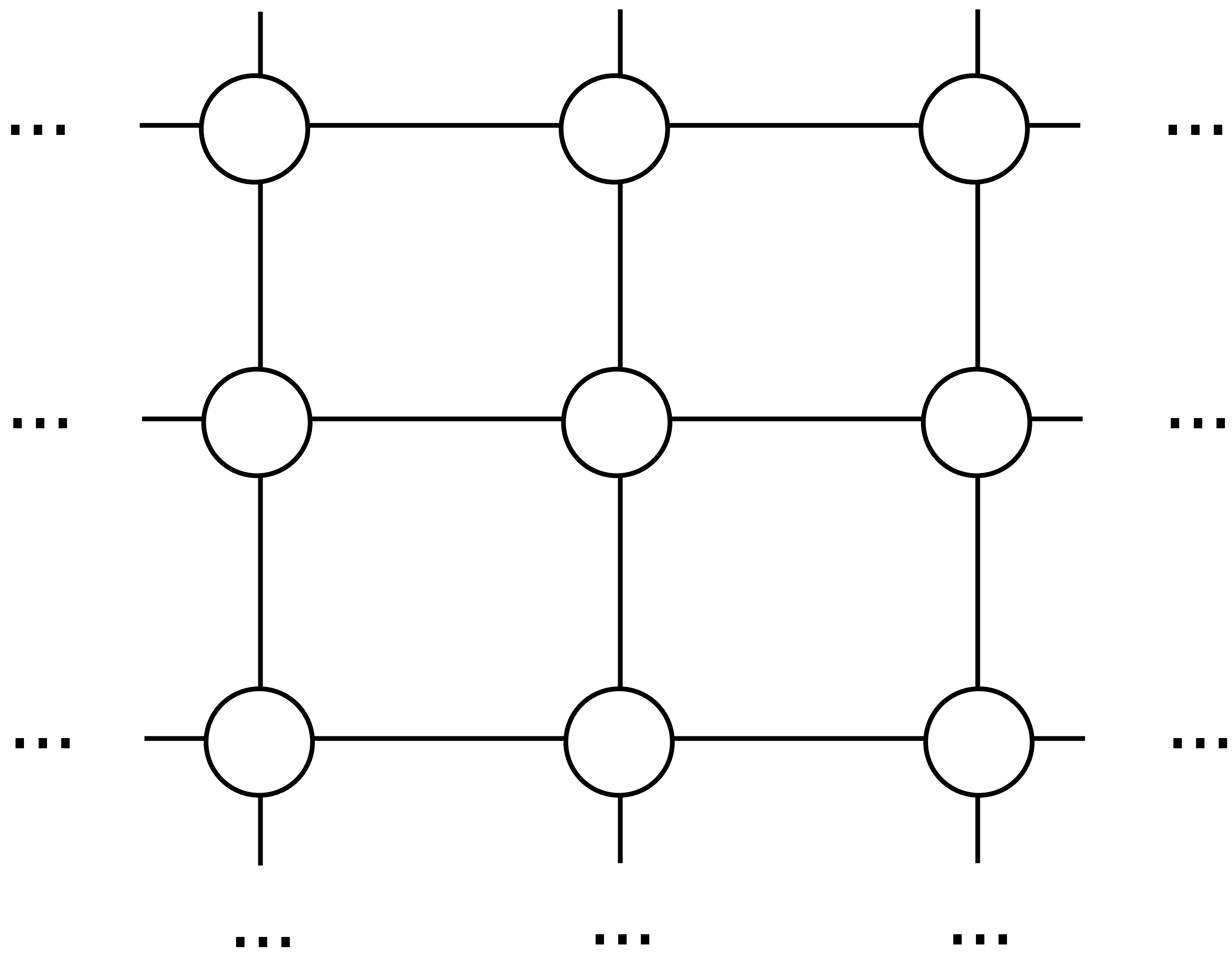
In general, there is an unknown number of clusters, so we will apply nonparametric priors, but for now, assume there are only 'background' and 'people' clusters

Model for image segmentation



Note: we can also define models without bothering to normalize

After simplification

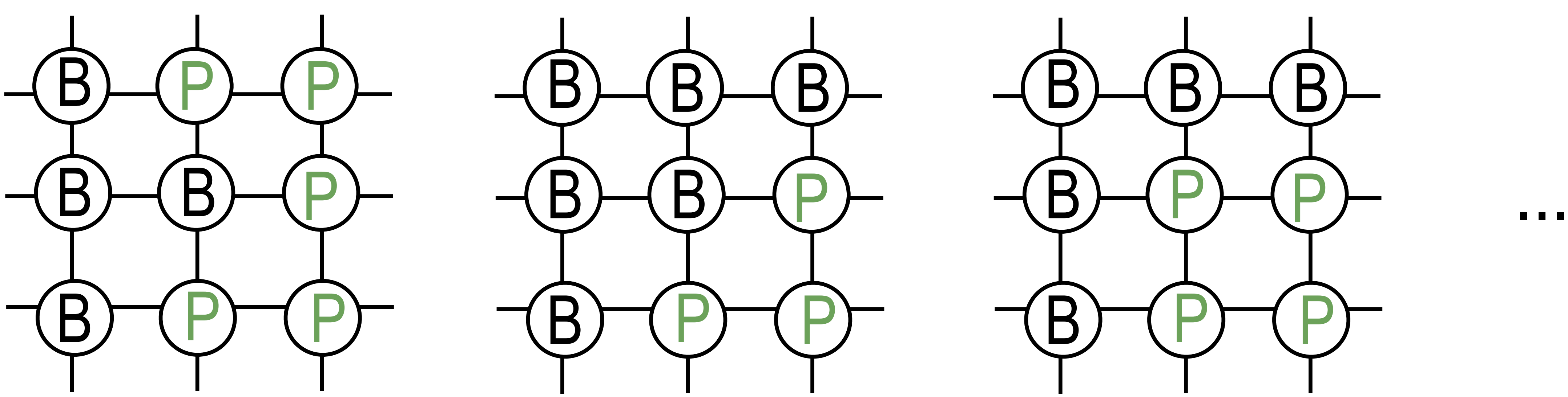


MCMC methods

What it does: Same as the elimination algorithm (normalization and posterior), but not limited to trees.

Con: approximate instead of exact

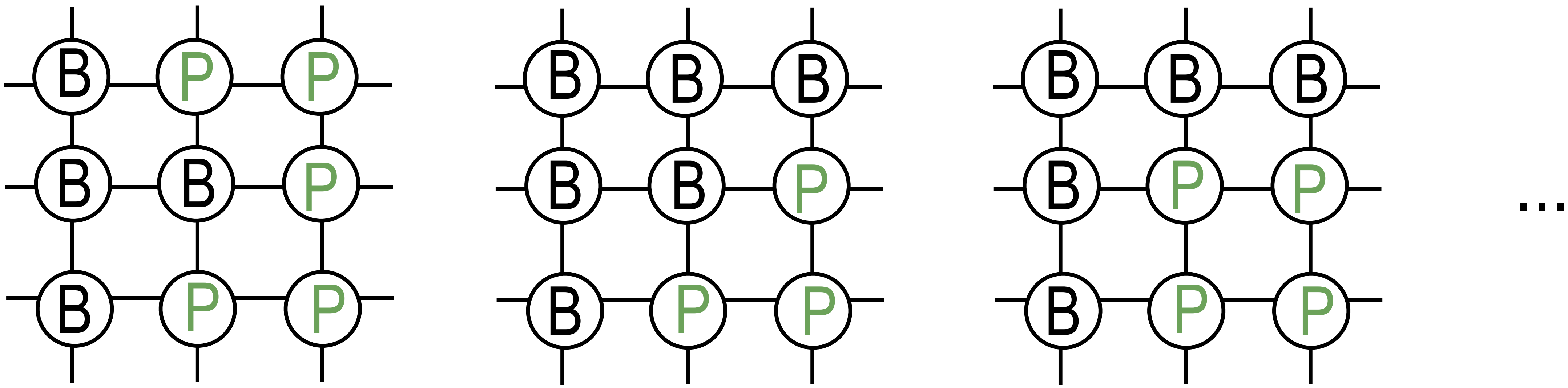
Output: a list of samples, i.e. the model with values for the hidden nodes filled in (imputed)



MCMC methods: how does it work?

Things to discuss:

- How to compute posterior expectations from these samples (e.g. Bayes estimator)
- How to create the samples so that they are approximately distributed according to the posterior?
- How to compute Z from these samples

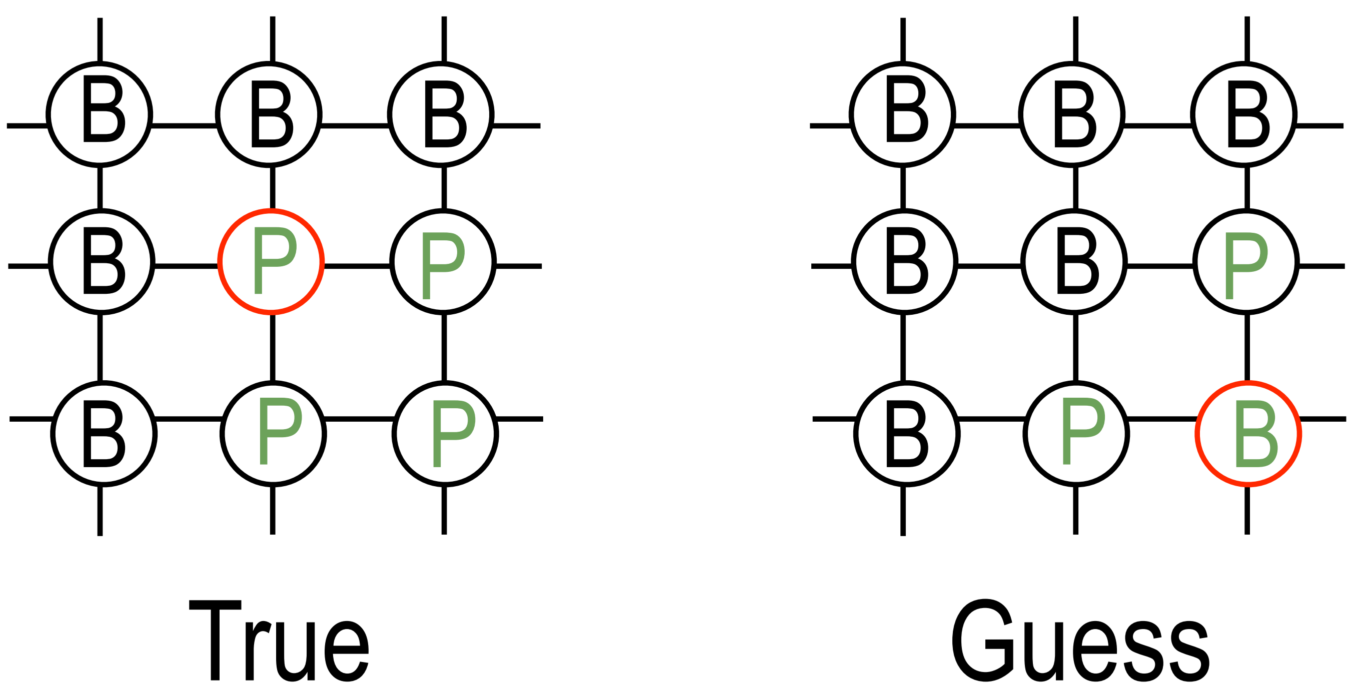


First item: Using the samples to compute posterior expectations

Task: given some images (a 2D array of pixels), segment it into clusters of pixels (‘background’ or ‘people’)

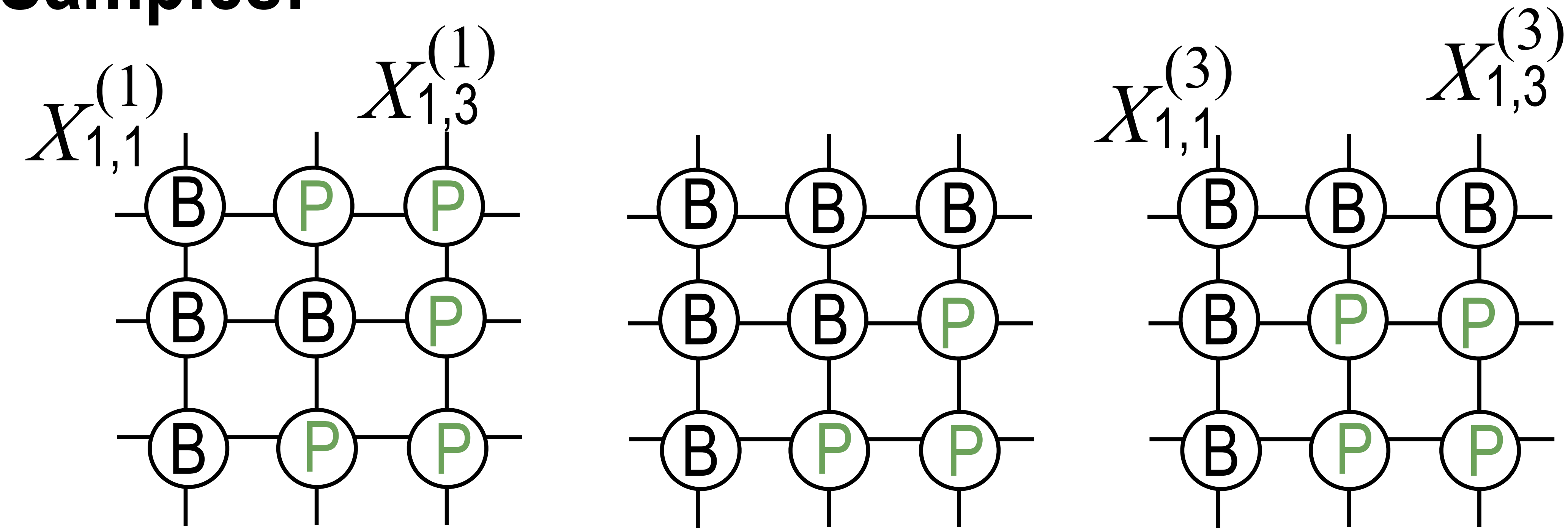
Loss function: Number of misclassified pixels

Example:



Computing the posterior

Samples:



Monte Carlo estimator: for S samples

$$\mathbb{E}f(X) \approx \frac{1}{S}\sum_{i=1}^{S} f(X^{(i)})$$

Second item: generating samples approximately distributed according to posterior

What is the Metropolis hasting acceptance ratio?

A $$\frac{p(x')q(x' \rightarrow x)}{p(x)q(x \rightarrow x')}$$

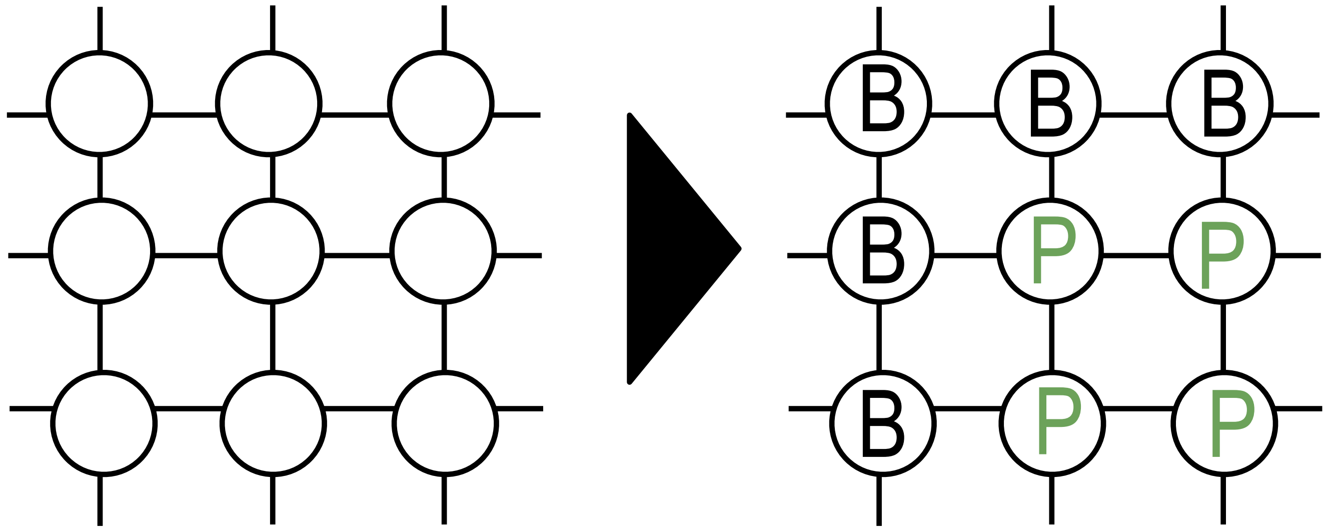
B $$\frac{p(x)q(x \rightarrow x')}{p(x')q(x' \rightarrow x)}$$

x' : Proposed
x : Current
p : Joint density
q(v --> w) density of proposing w from v

Let's start by an easy special case: 'Naive' Gibbs sampling

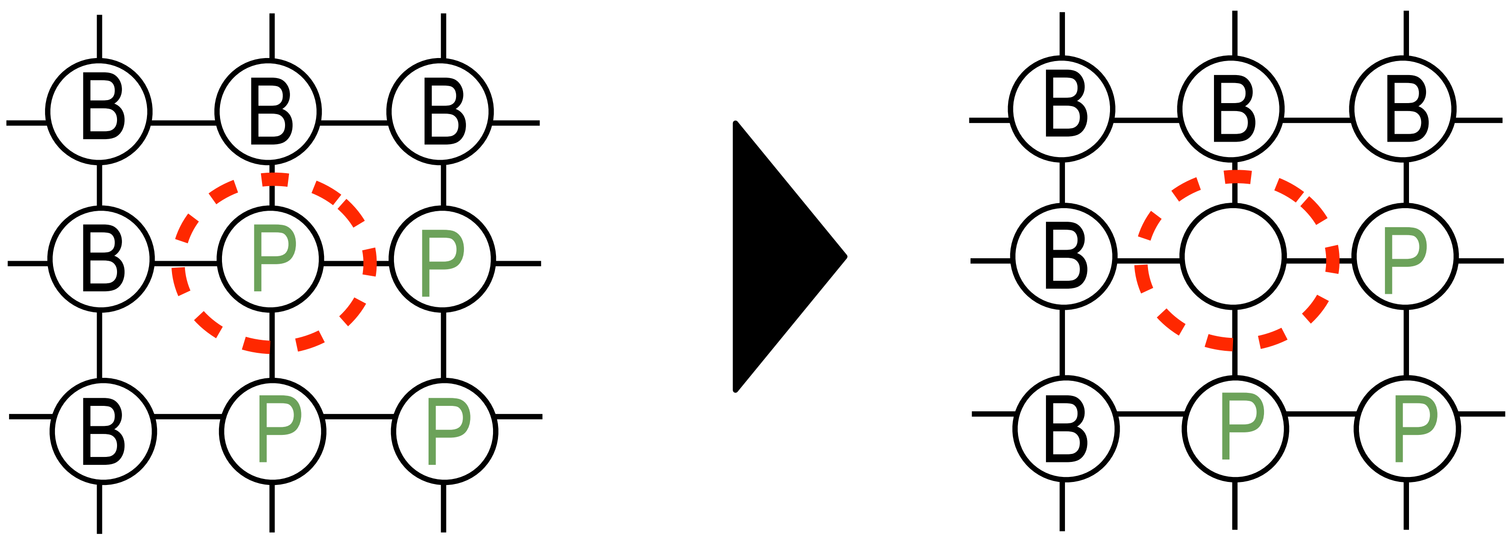
Idea: at each iteration, maintain a guess for all the hidden nodes

Init.: guess arbitrary values for the hidden nodes

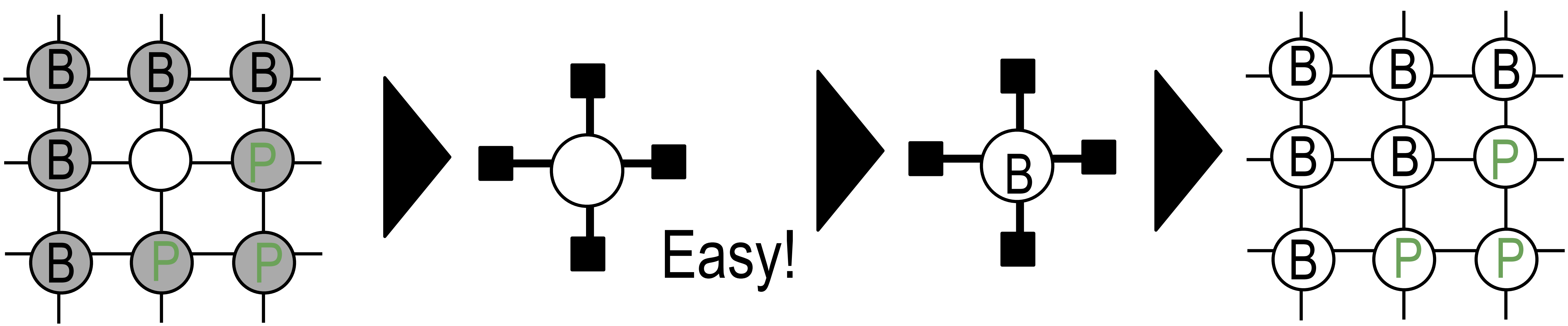


Let's start by an easy special case: 'Naive' Gibbs sampling

Loop: pick one node *(i,j)* at random, erase the contents of the guessed values in *(i,j)*, freeze the value of the other nodes

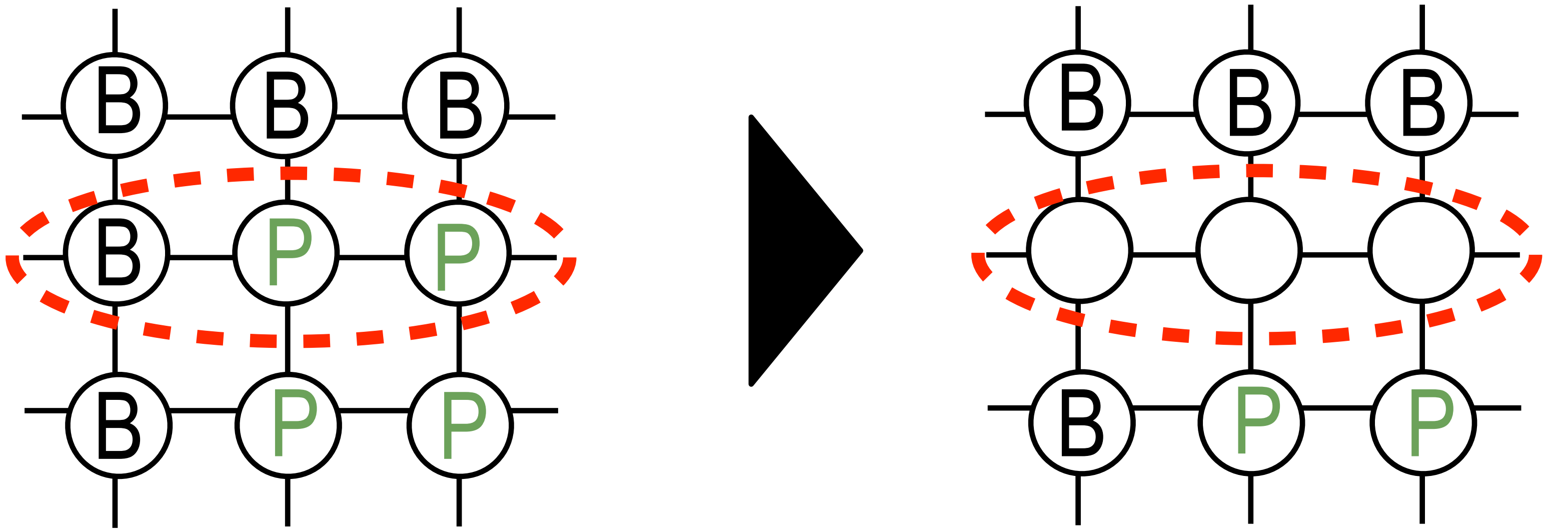


Then: resample a value for the node *(i,j)* conditioning on all the others, and write this to the current state at *(i,j)*



Better Gibbs samplers

Loop: pick a subset of nodes N at random, erase the contents of the guessed values in N, freeze the value of the nodes not in N



Then: resample a value for the nodes in N conditioning on all the others, and write this to the current state at N