Statistical modeling with stochastic processes

Alexandre Bouchard-Côté

Lecture 10, Wednesday March 30

Program for today

- Assignment/logistics
- Applications
  - NLP: language modelling, segmentation, alignment
- Extensions
  - Hierarchies and sequences
  - Pitman-Yor & Beta processes

Assignment/logistics

After class: office hours

Tonight: Solutions to the implementation questions will be posted at the same time as **assignment 2**

Due dates:

- Assignment 2: April 13 (end of the day)
- Assignment 3 and project: April 22 (end of the day)

Important: Recall that if you do a final project, you need to do only 2 assignments. If you do a literature review, do all 3.

Assignment/logistics

Assignment 1:

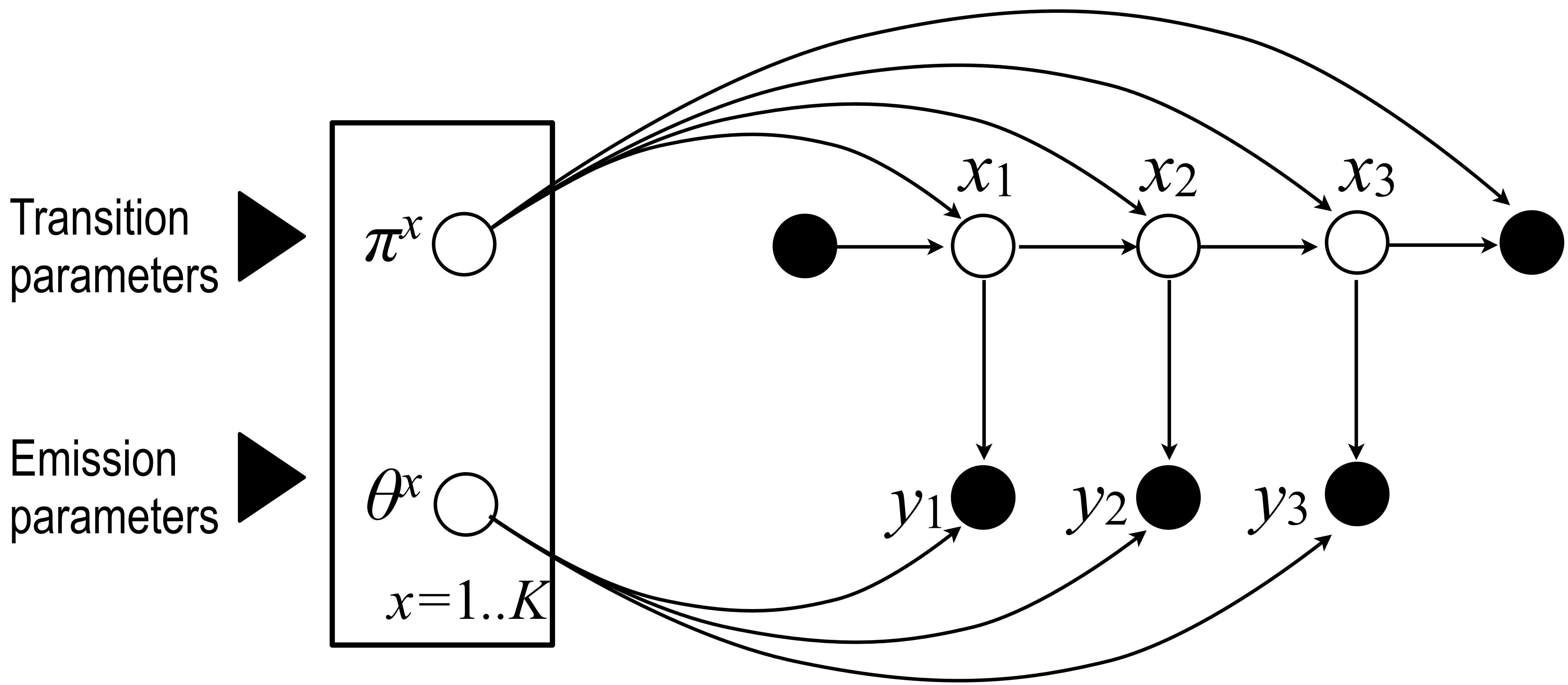
- We will go over some of the solutions for the written questions now, the rest will be posted tomorrow
- You will get back your copy next Monday

Lecture notes:

- Those related to assignment 2 be posted tomorrow as well
- The other ones will follow as soon as I get the latex files from the scribes

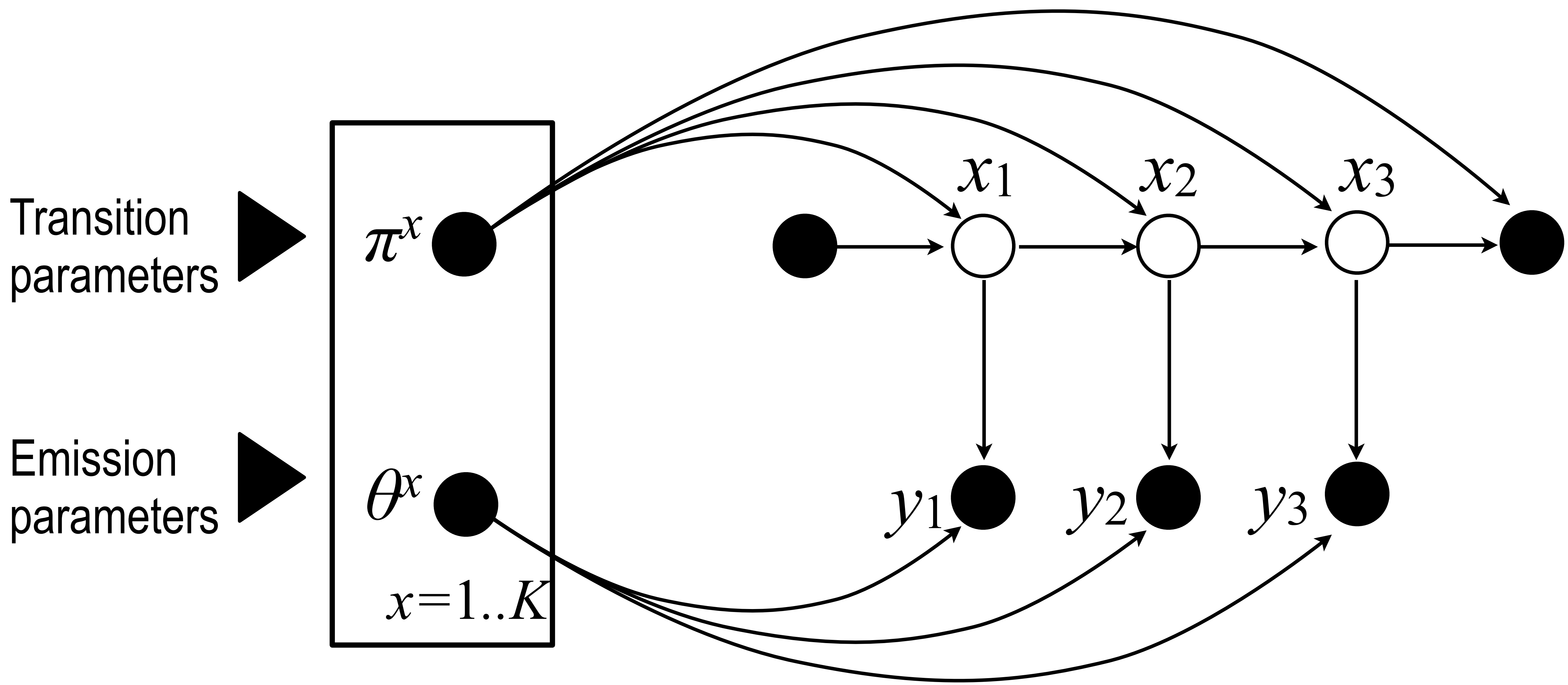
Question 4.1.A

Consider the graphical model we used in the previous question, and assume that there is a Dirichlet prior on the parameters. Describe two MCMC moves: one that samples all the sentences at once conditioning on the parameters, and one that samples a single word but collapses the parameters.



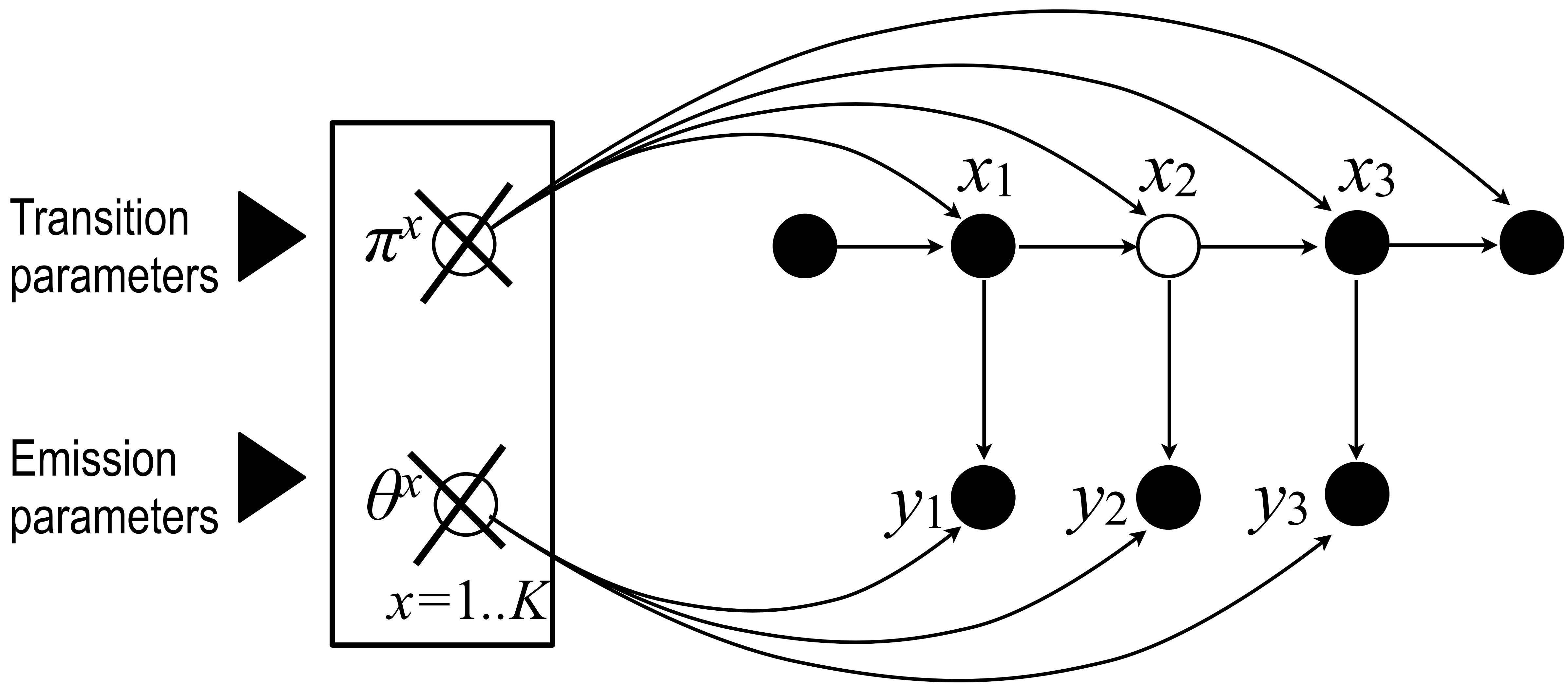
Question 4.1.A

Sampling sentence at once: direct from Q.1.1



Question 4.1.A

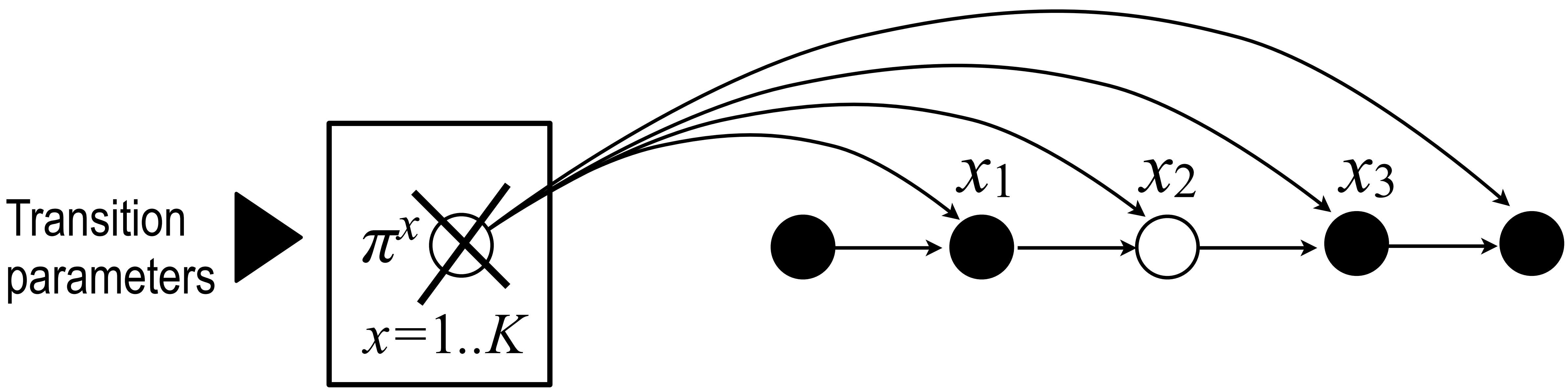
Collapsing/marginalizing parameters: two methods...



Question 4.1.A

Collapsing/marginalizing parameters: two methods...

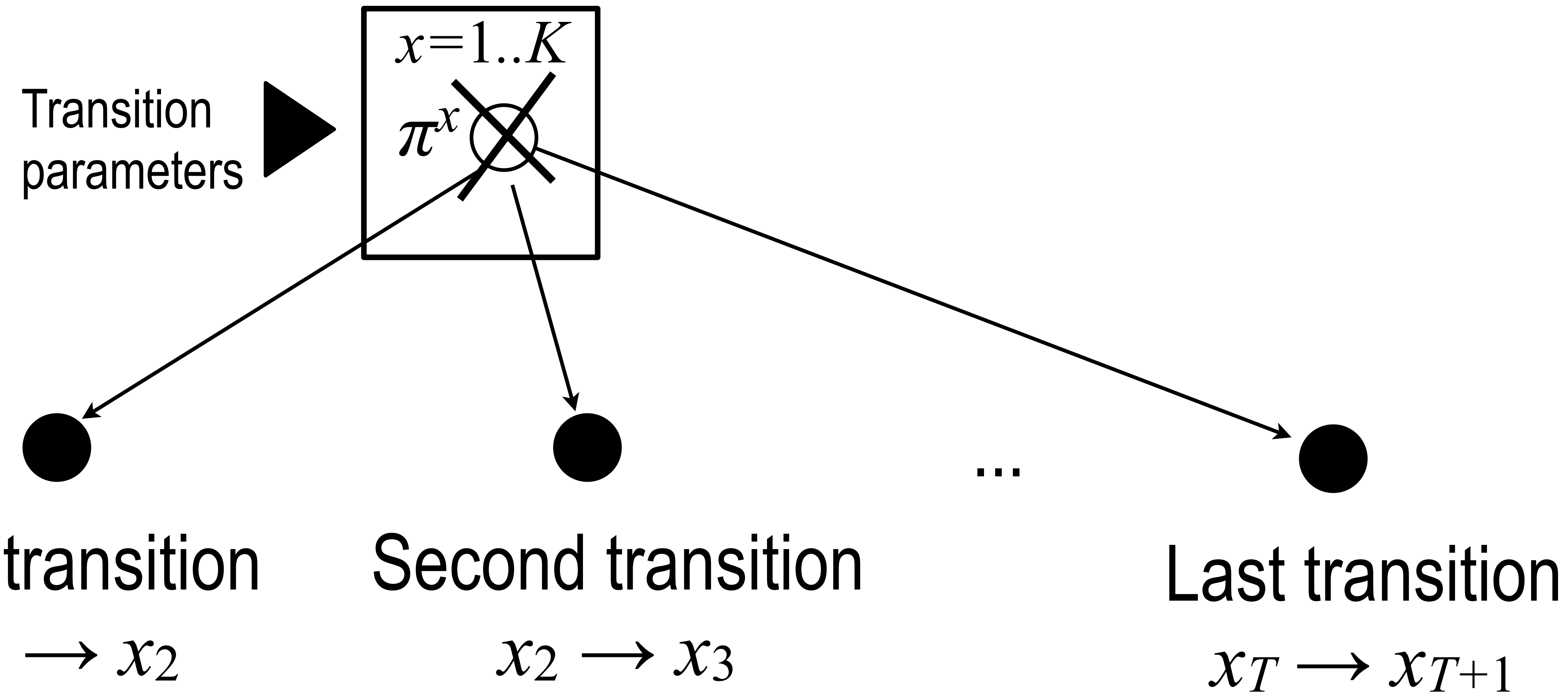
Let's forget about the observations for simplicity



First method: direct marginalization

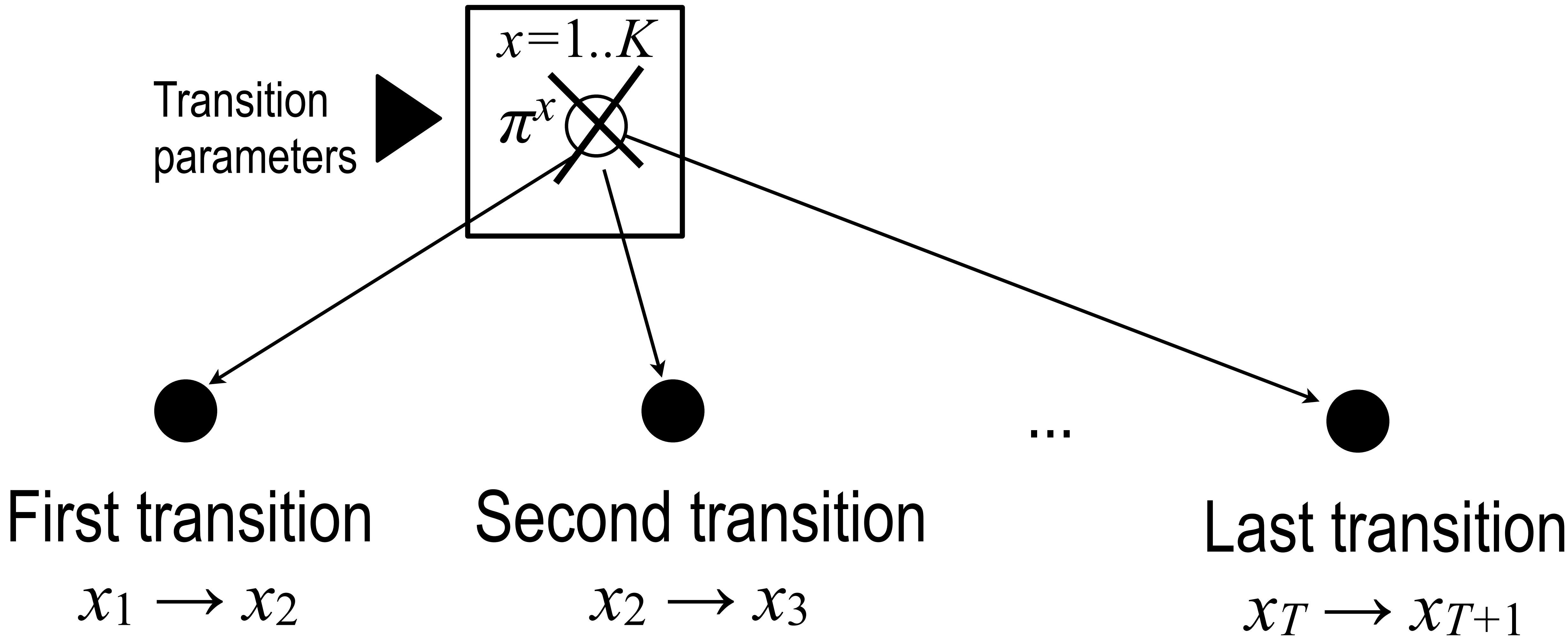
Question 4.1.A: Exchangeability trick

Idea: the *states* visited are not exchangeable (they are Markovian), but the *transitions* are exchangeable



(modulo a base measure that is equal to one or zero)

Question 4.1.A: Exchangeability trick



Resampling one state will change at most two of these variables

Pretend they are the last two ones

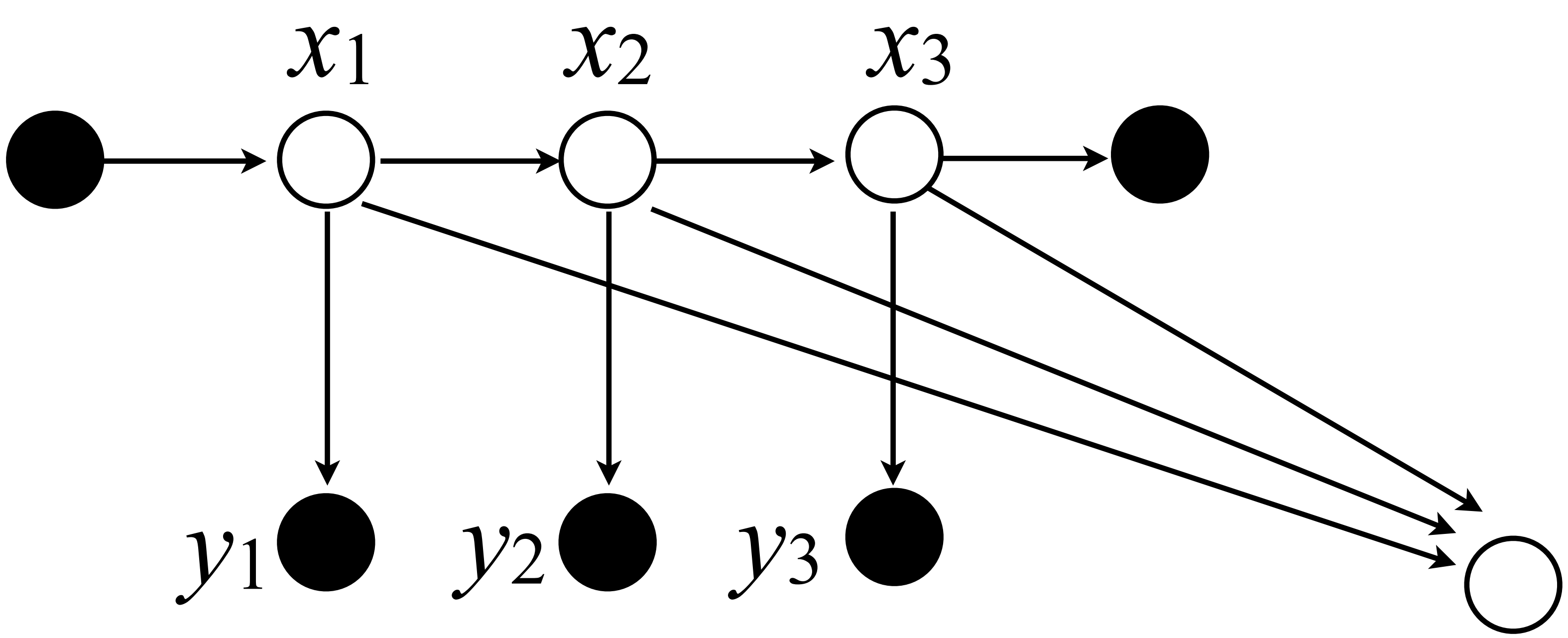
Question 4.1.A

Consider a different prediction problem for part D of the previous question: finding the number of distinct contiguous alpha-beta blocks. For example, in the sequence:

"NNYYNYYYNYYYYYYYNNNN",

the correct answer would be 3. Suppose the loss is the absolute value between the prediction and the truth. How would you approximate the Bayes estimator in this case?

Question 4.1.A



Deterministic auxiliary variable:
number of contiguous ‘Y’ blocks in the current state

Hierarchical models: review and big picture

Language models

Shannon's game: guess the next word...

I have lived in San ______

I am not going to go ______

there or their?

Application: finding which sentence is more likely

Example: Speech recognition

Problem...

Prior for prefix 1

Distribution over what follows after the prefix
Fix ___

Guess	Pr
a	0.92
...	...
...	...

Prior for prefix 2

Distribution over what follows after the prefix
a ___

Guess	Pr
certain	0.46
text	0.46
...	...

...

...

Some prefixes are rare. Is that a problem?

Solution: hierarchical model

Hyper-prior over words---not specific to a prefix

Distribution over words in text dataset

Guess	Pr
the	0.04
a	0.02
...	...

Prior for prefix 1

Distribution over what follows after the prefix
Fix ___

Guess	Pr
a	0.92
...	...
...	...

Prior for prefix 2

Distribution over what follows after the prefix
a ___

Guess	Pr
certain	0.46
text	0.46
...	...

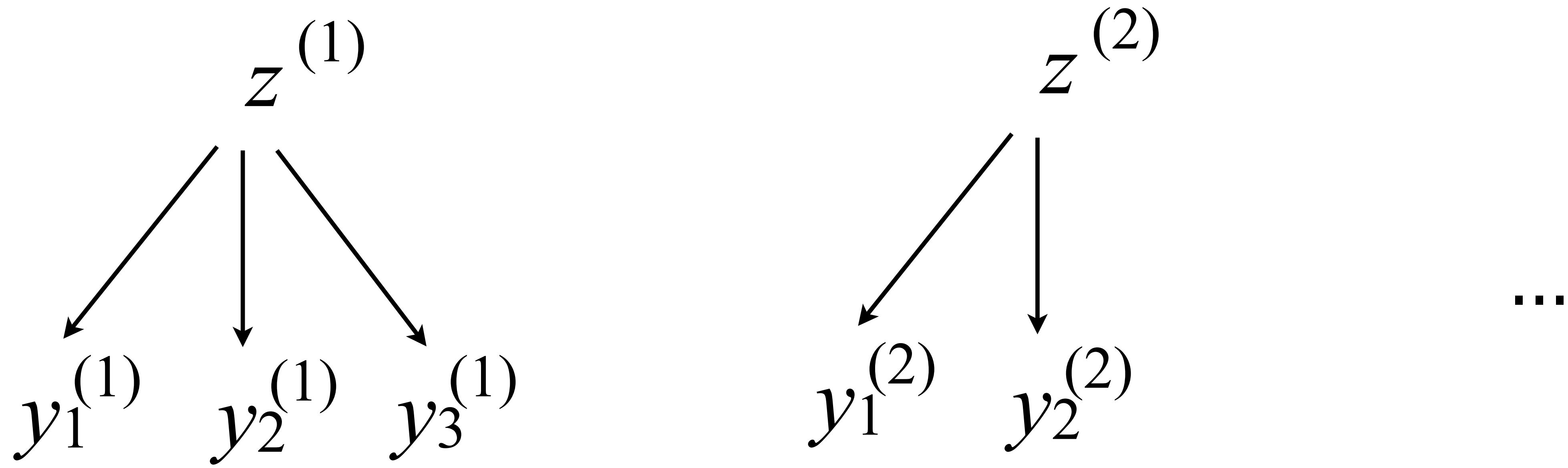
...

...

Important idea: hierarchical Bayesian models

Applies: whenever we are doing estimation on related (or not so related) sub-problems.

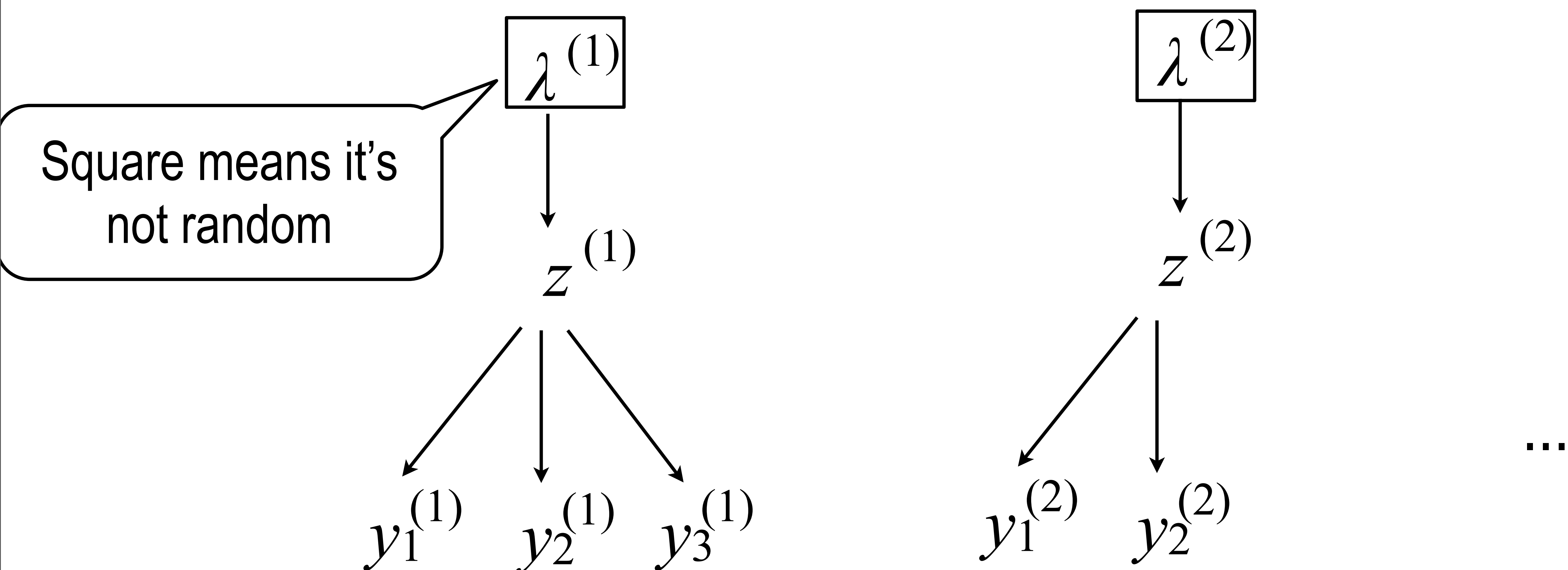
For today: assume we know the hierarchy



	<u>(Sub) Problem 1</u>	<u>(Sub) Problem 2</u>
Eg:	Dist. of word after 'Fix __'	Dist. of word after 'a __'
	Progression of HIV in patient 1	Progression of HIV in patient 2

Important idea: hierarchical Bayesian models

Assumption: each model has shared hyper-parameters λ (parameters of the distribution of the priors z)



(Sub) Problem 1 | (Sub) Problem 2

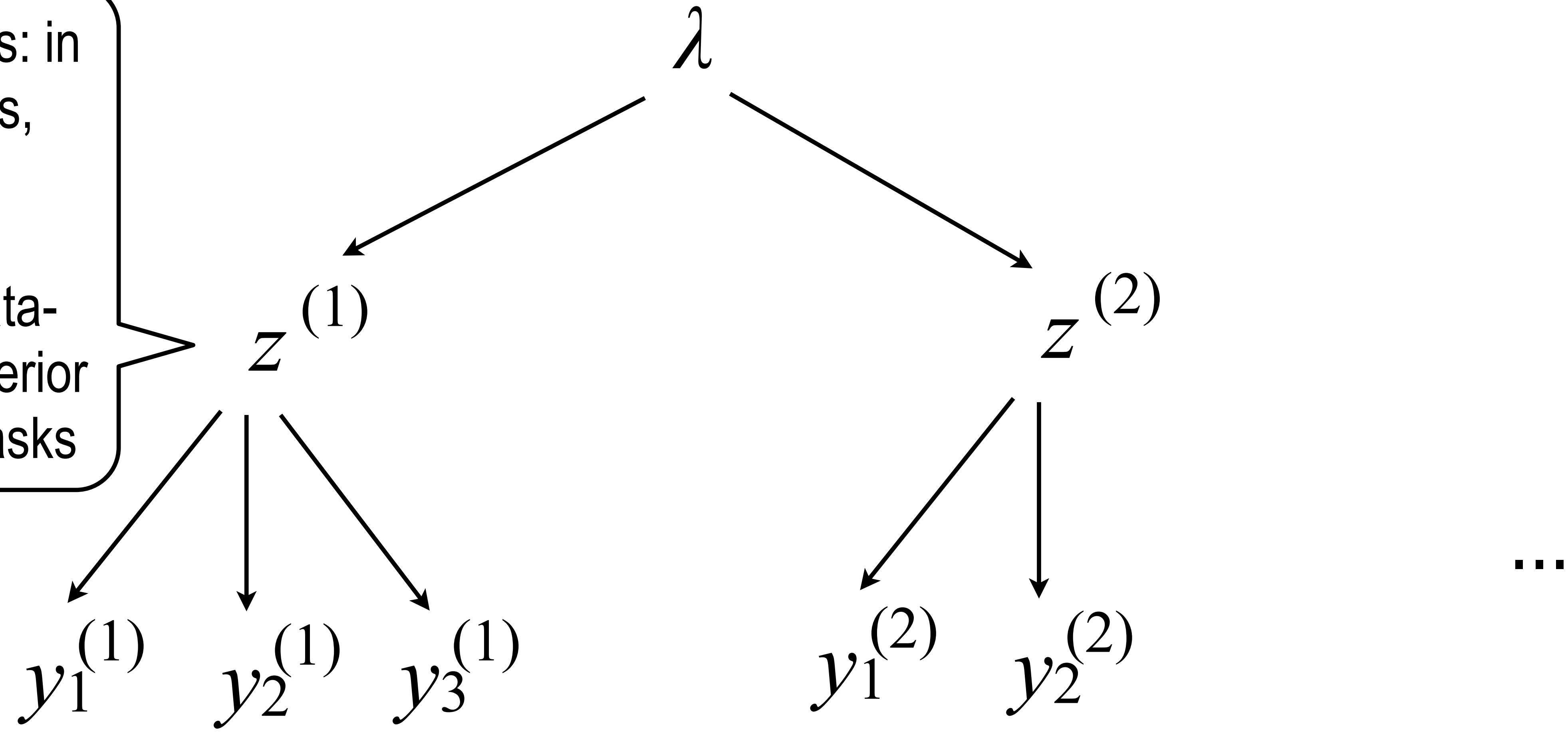
Eg: Dist. of word after 'Fix __' | Dist. of word after 'a __'

Progression of HIV in patient 1 | Progression of HIV in patient 2

Important idea: hierarchical Bayesian models

Ideas: make the hyper-parameters λ random (1) and shared by all tasks (2). This binds all the tasks/subproblems.

Intuition why it works: in data-rich problems, posterior mostly determined by observations; in data-poor problems, posterior informed by other tasks

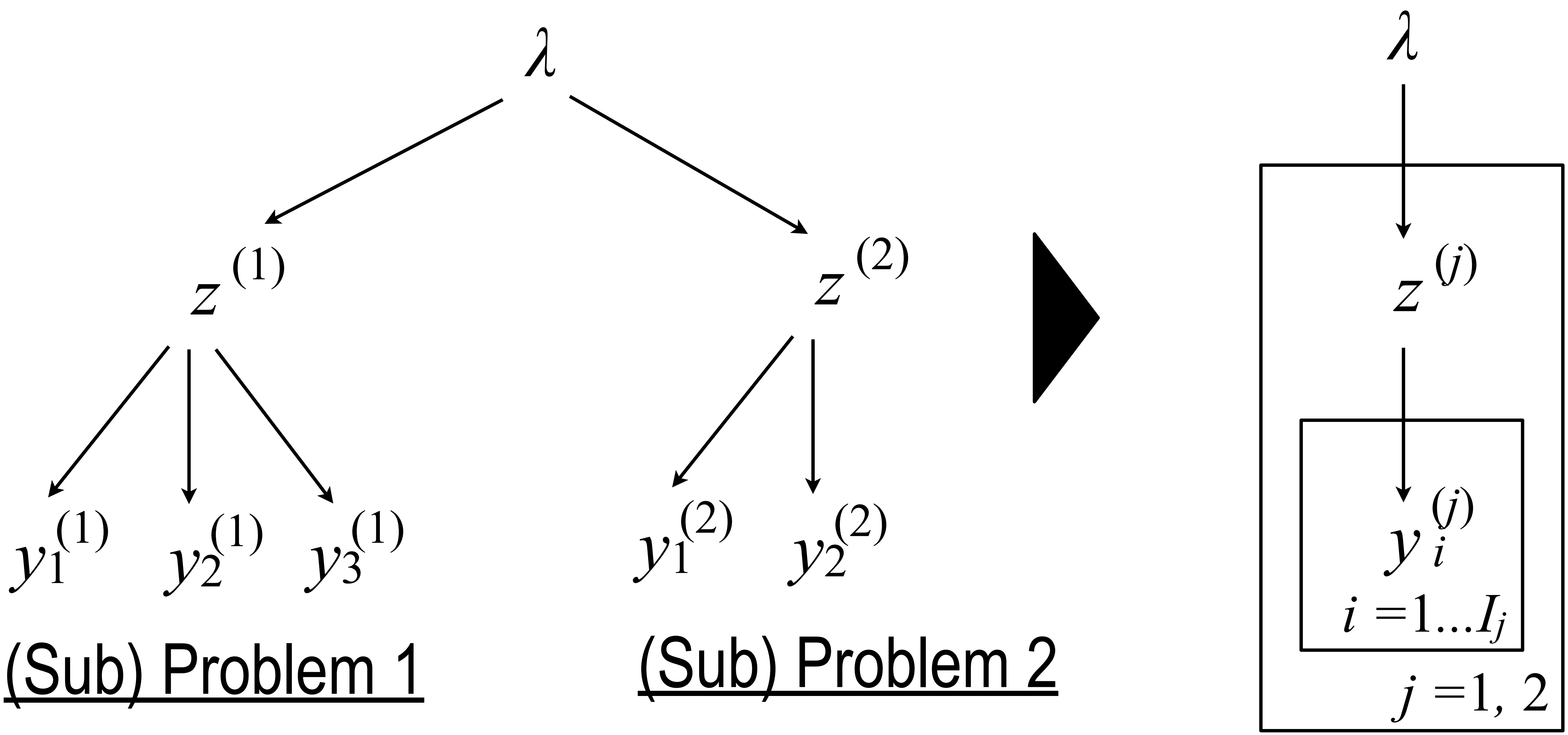


(Sub) Problem 1 — (Sub) Problem 2

Eg: Dist. of word after 'Fix __' — Dist. of word after 'a __'

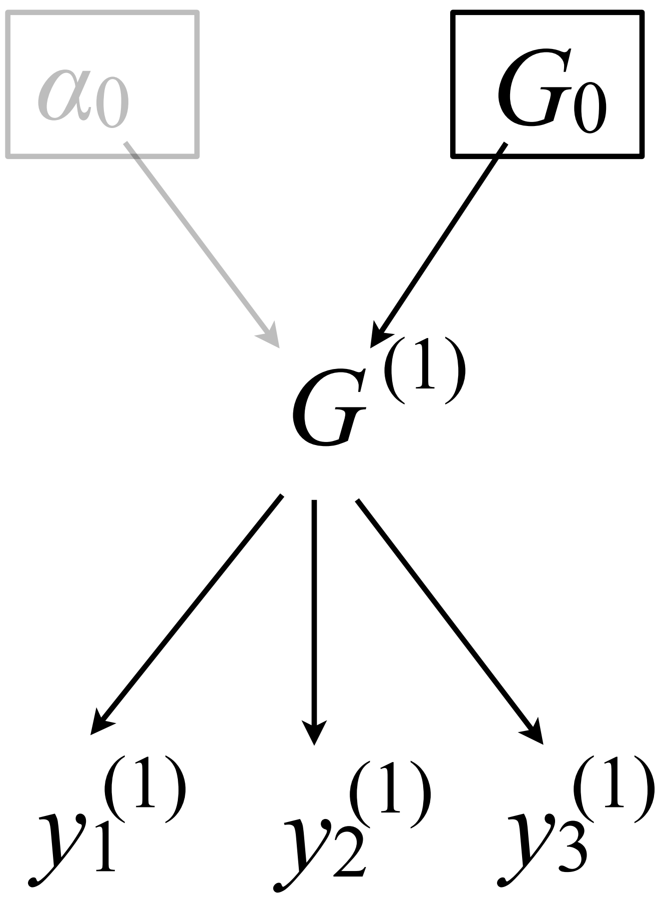
Progression of HIV in patient 1 — Progression of HIV in patient 2

Important idea: hierarchical Bayesian models

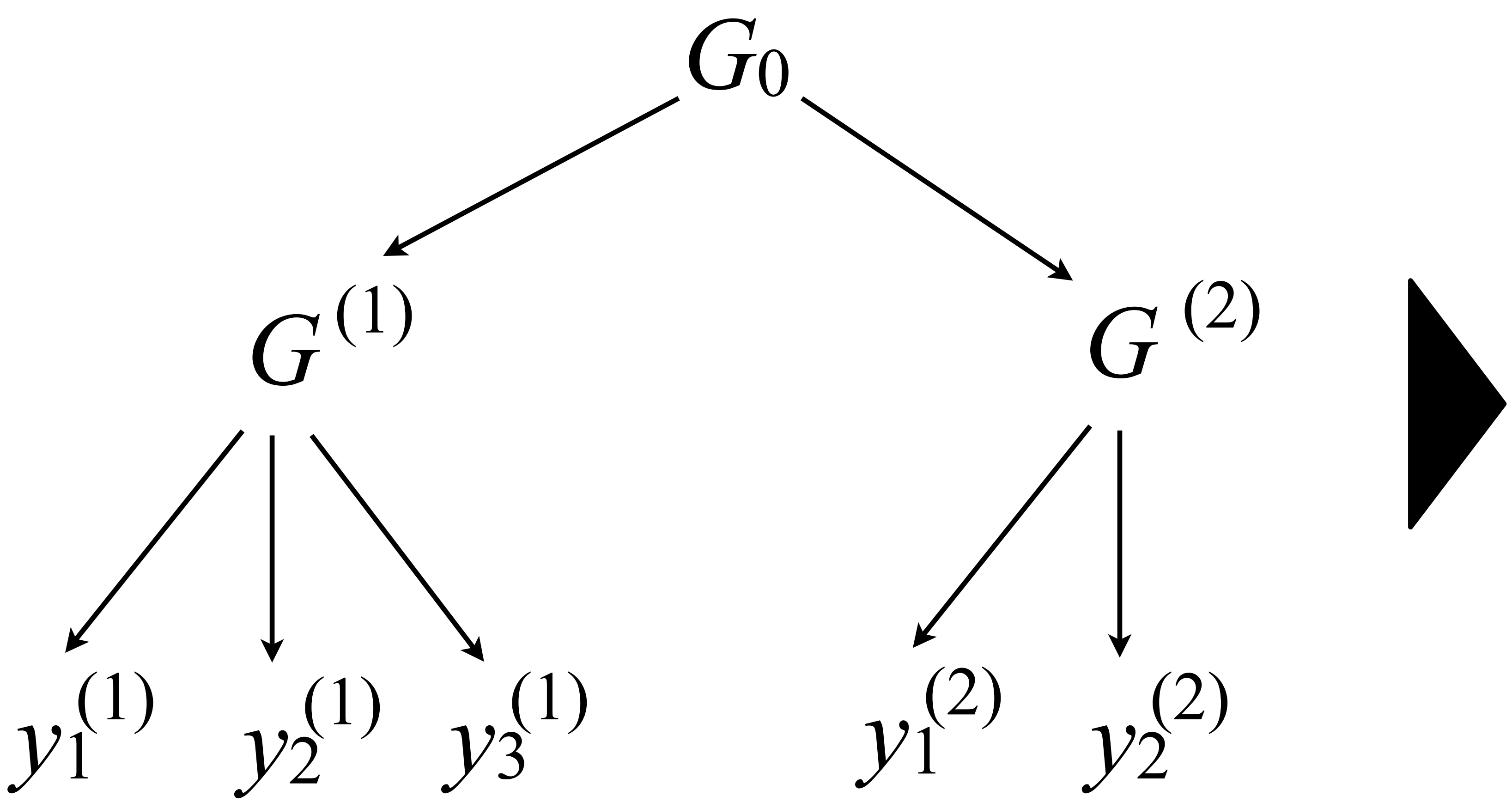


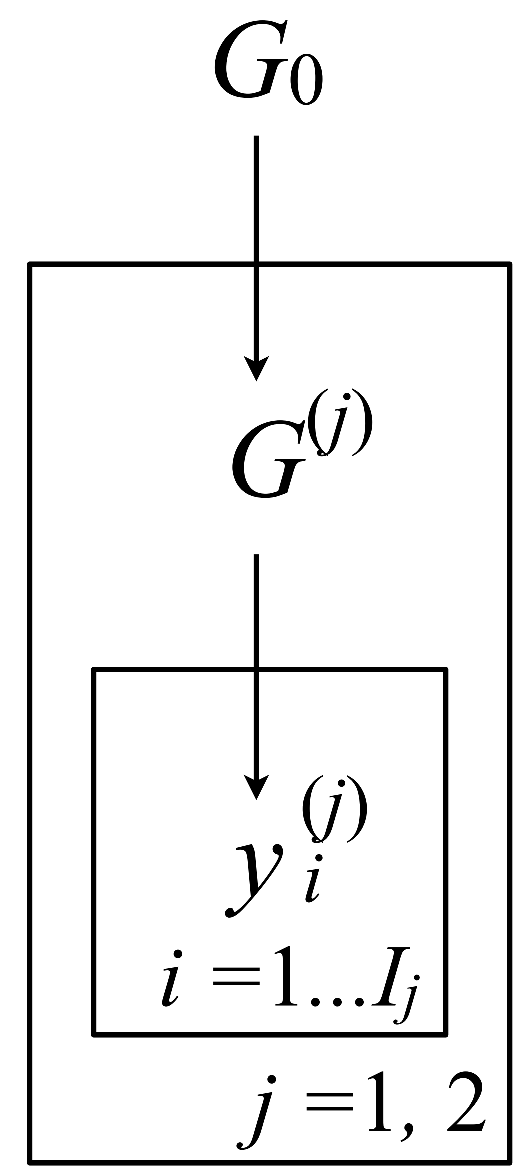
Hierarchies with DPs

Hyper-parameter: $\alpha_0,\ \boldsymbol{G_0}$



Hierarchies with DPs





What distribution to put G_0?

Distribution on G_0

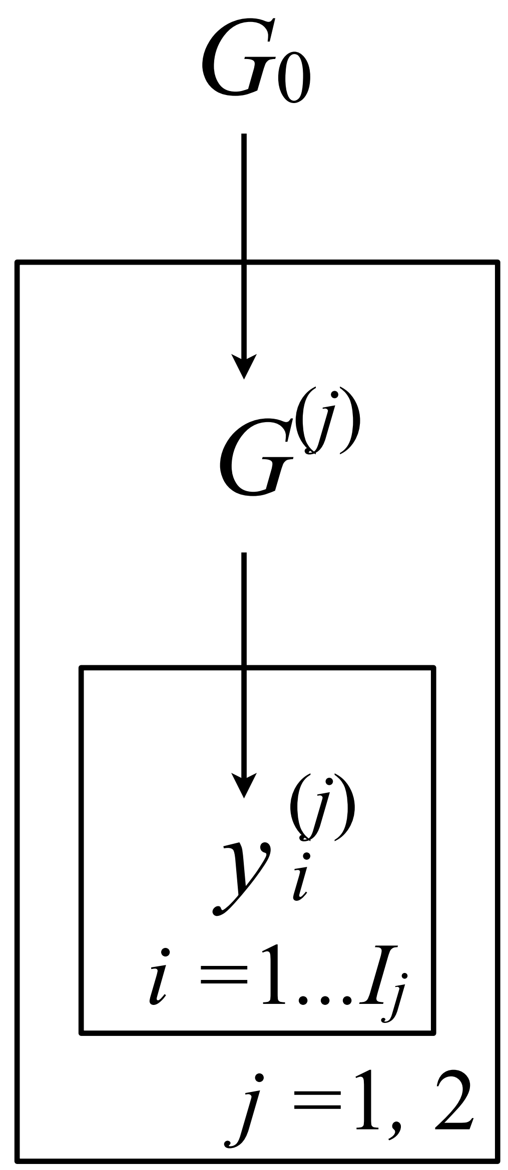
What distribution to put G_0?

First try: a continuous distribution, e.g. normal with random mean

$$\mu \sim \mathrm{N}(0, 1)$$

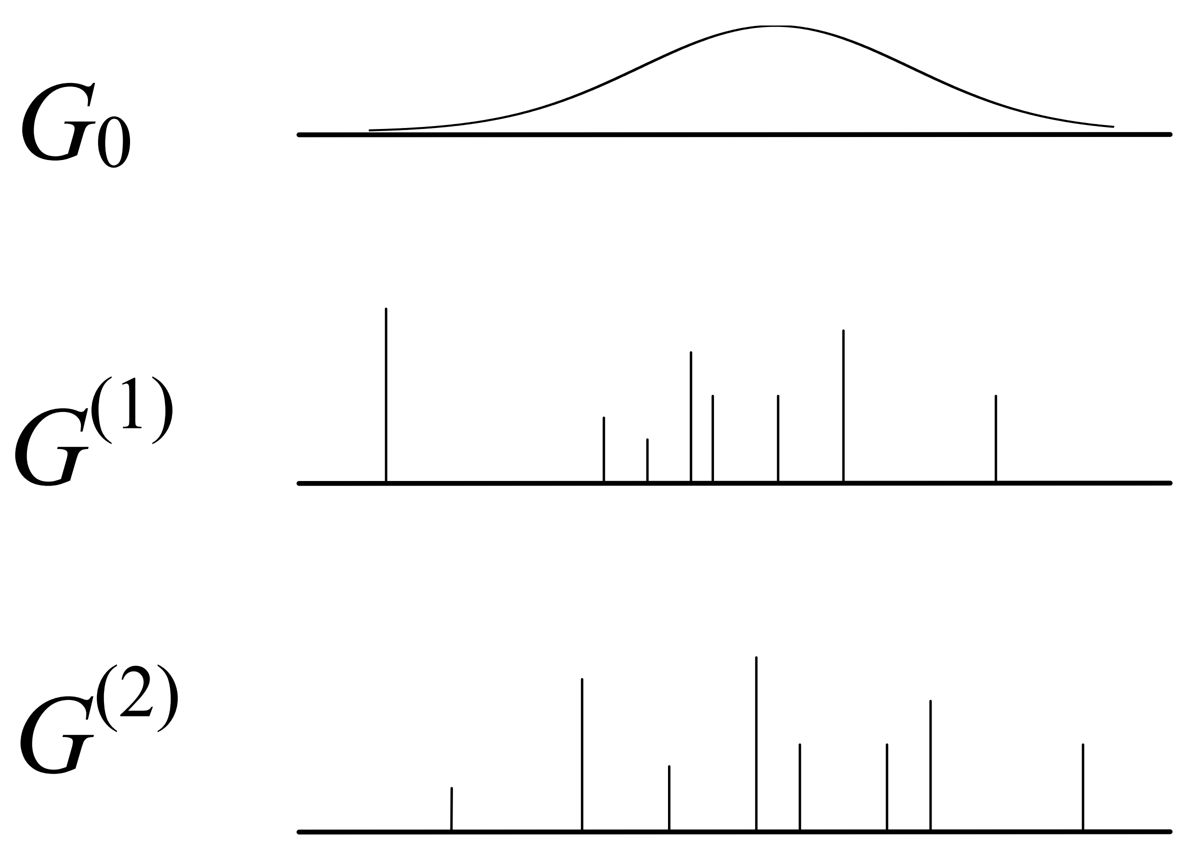
$$\mathrm{G}_0 = \mathrm{N}(\mu, 1)$$

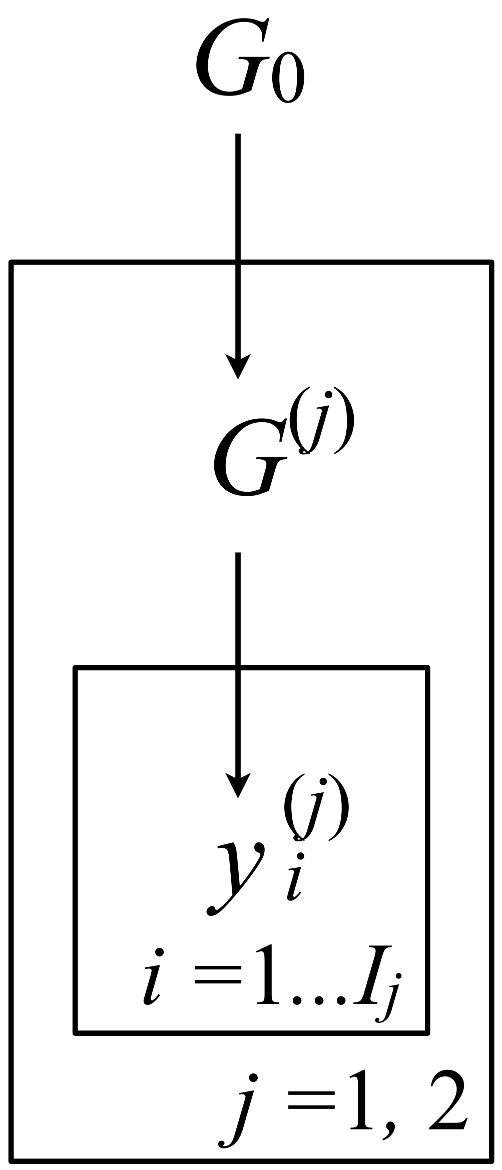
This does not work!



Distribution on G_0

Problem: with probability one, no atoms will be shared by $G^{(1)}$ and $G^{(2)}$: this means there will be no sharing of dishes across tasks/sub-problems





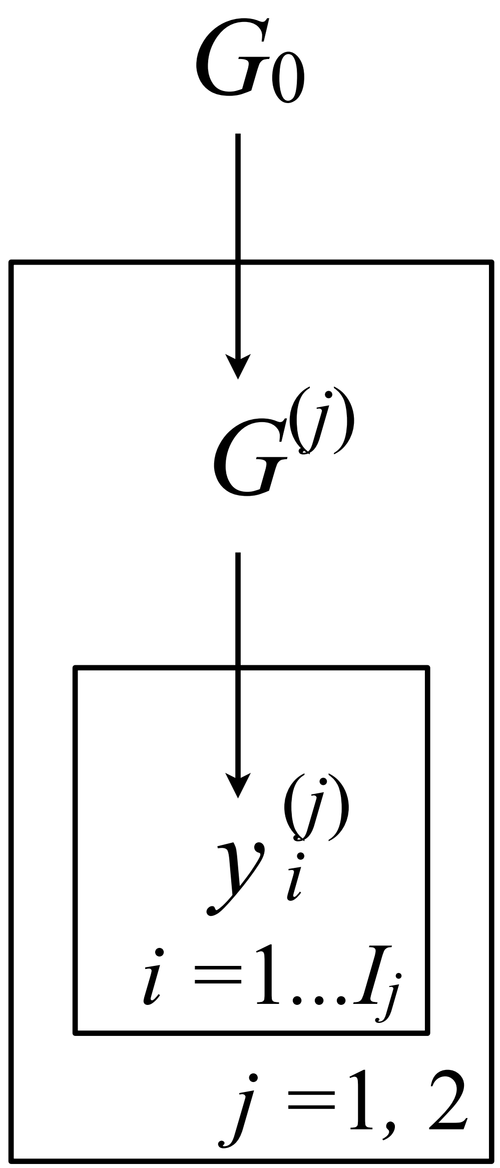
Distribution on G_0

What distribution to put G_0?

A correct choice: a Dirichlet process !

$$\mathrm{G}_0 \sim \mathrm{DP}(\alpha_0, H)$$

$$\mathrm{G}^{(j)} | G_0 \sim \mathrm{DP}(\alpha'_0, G_0)$$



Pitman-Yor process

Another problem...

In some real-world datasets, Dirichlet processes do not have the right tail behavior!

Empirical observation: number of unique words (word types observed) in a natural language corpus containing n words tokens is $O(n^s)$ for $s \in [1/2, 1)$

Fact about DPs (proven last time): there are $O(\log n)$ tables in n draws from a DP

Note: DPs will still assign positive probability to $O(n^s)$ tables, might discourage it too much in practice

Solution: a generalized process

Pitman-Yor process: Start with the CRP, and boost the probability of table creation while preserving exchangeability

This has the same normalization as the DP: $\alpha_0 + n$

New customer

...

$\propto n_1 - d$

Join table #1, with already n_1 people sitting there

$\propto n_t - d$

Join table # t, with already n_t people sitting there

$\propto \alpha_0 + t\,d$

Create a new table

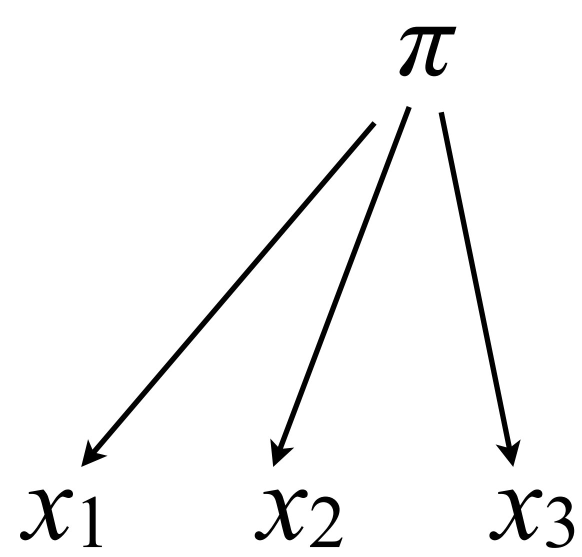
Discount: $d \in [0, 1]$

The Pitman-Yor (PY) process

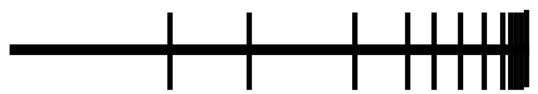
Exchangeability: we have shown last time an example where the seating plan is exchangeable, you will prove it in full generality in the assignment

Asymptotic number of tables: $O(n^s)$

De Finetti representation?



PY: stick breaking construction



Dirichlet process: defined $G = f(\beta, \theta)$ for an iid sequence of $\theta_i \sim G_0$ and:

$$\beta_i \sim \mathrm{Beta}(1, \alpha_0),$$

Pitman-Yor: Same but now beta's are not identically dist.:

$$\beta_i \sim \mathrm{Beta}(1 - d, \alpha_0 + i\, d)$$

Other stick breaking constructions?

Yes: For example as long as there is an epsilon > 0 s.t.,

$$\sum_{j=1}^{\infty} \mathbb{P}(\beta_j > \epsilon) = \infty$$

we get sticks with lengths that sum up to one

But: These are not all exchangeable! In fact the β_i's have to be of the form $\mathrm{Beta}(1 - d, \alpha_0 + i\, d)$ to have exchangeability!

Infinite HMM

Next topic: infinite HMMs

Motivation: *state splitting* in Markov chains

Setup: annotated sequence data, where we don't believe the annotation actually makes the chain Markovian

Example:

Noun	Adv	Verb	Noun
He	really	likes	swimming

Noun	Adv	Verb	Noun
I	really	like	swimming

Next topic: infinite HMMs

Solution: adding annotation on the hidden state

Example: an annotation -3PS when the sentence is 3th person singular

Noun-3PS	Adv-3PS	Verb-3PS	Noun
He	really	likes	swimming

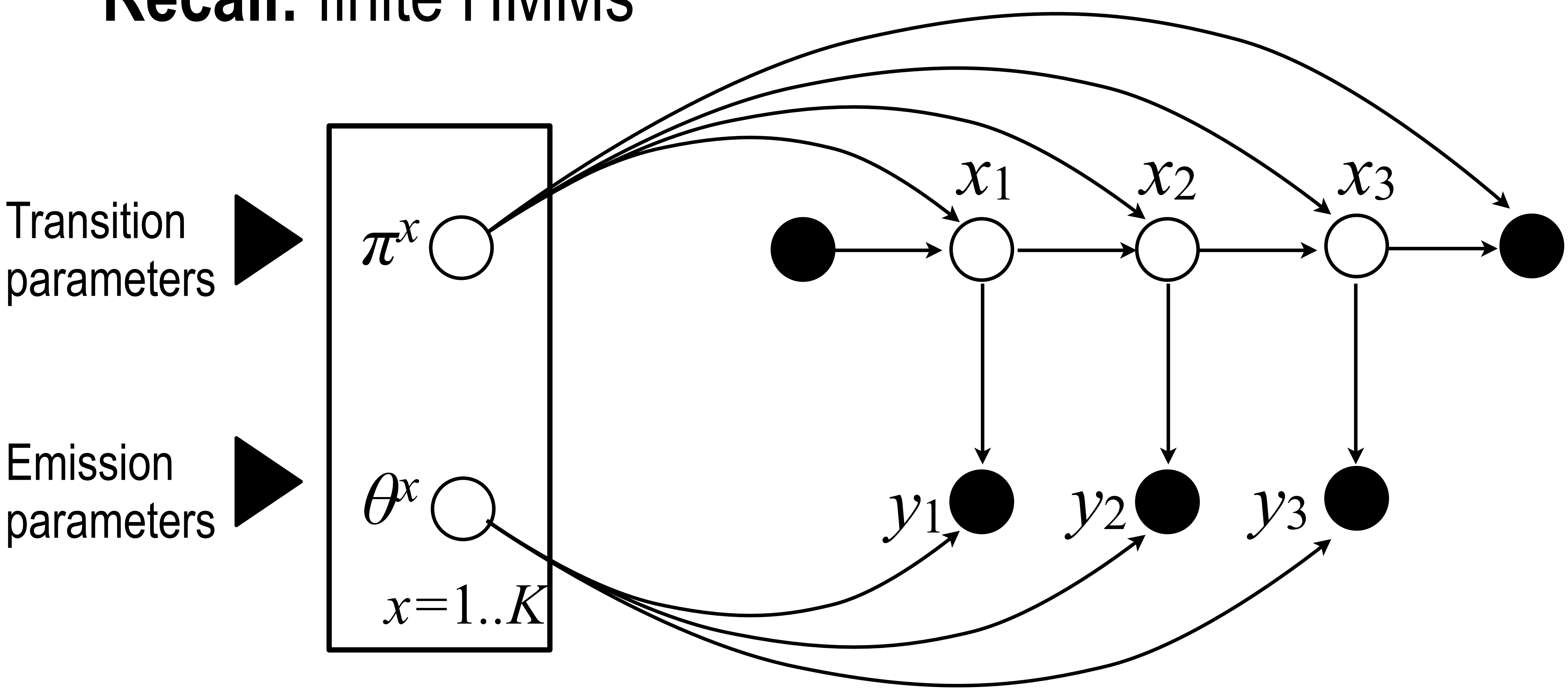
Noun	Adv	Verb	Noun
I	really	like	swimming

State splitting: learn annotations (state splits) automatically from the training data. **How many splits?**

The infinite HMM

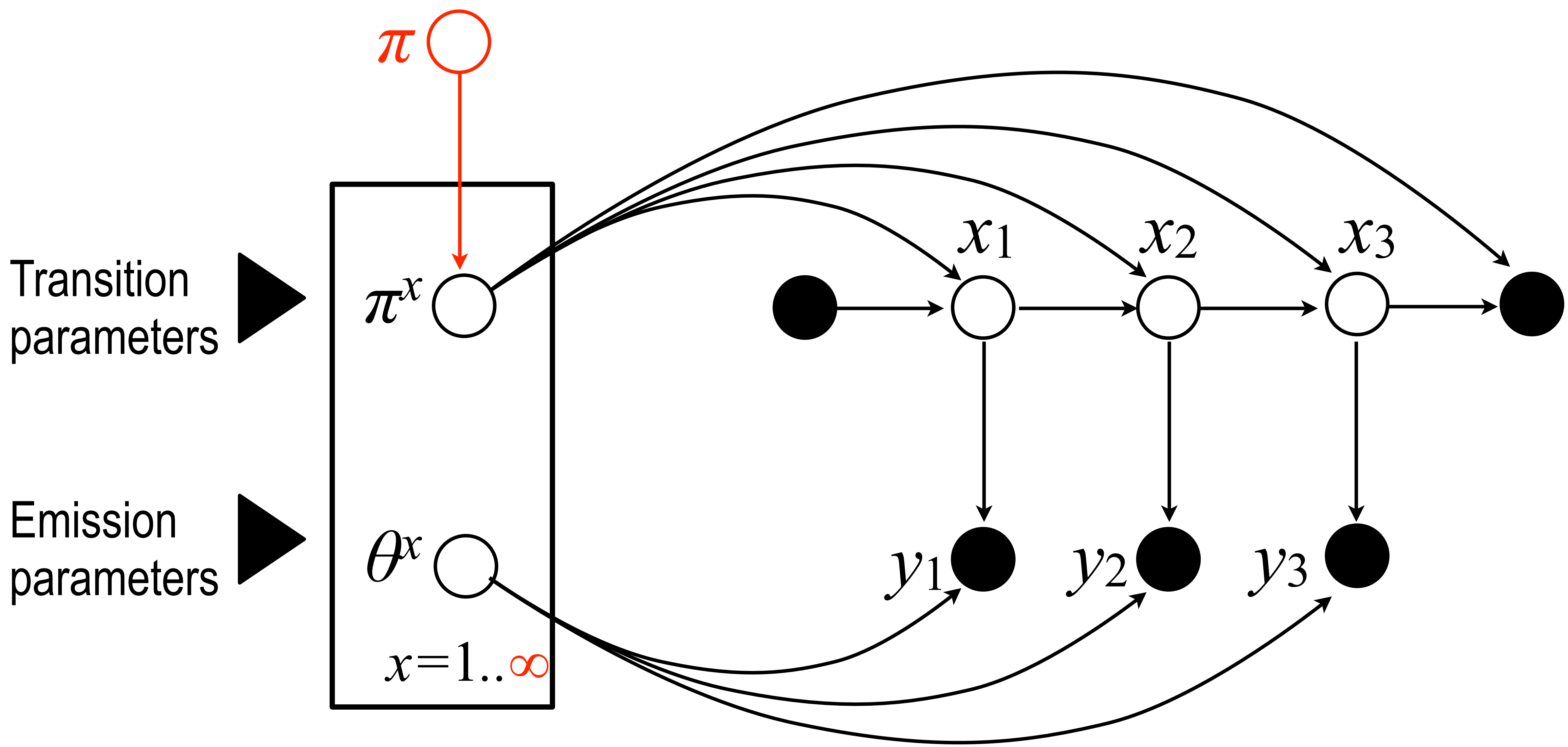
Motivation: an HMM without a bound on the number of hidden states

Recall: finite HMMs



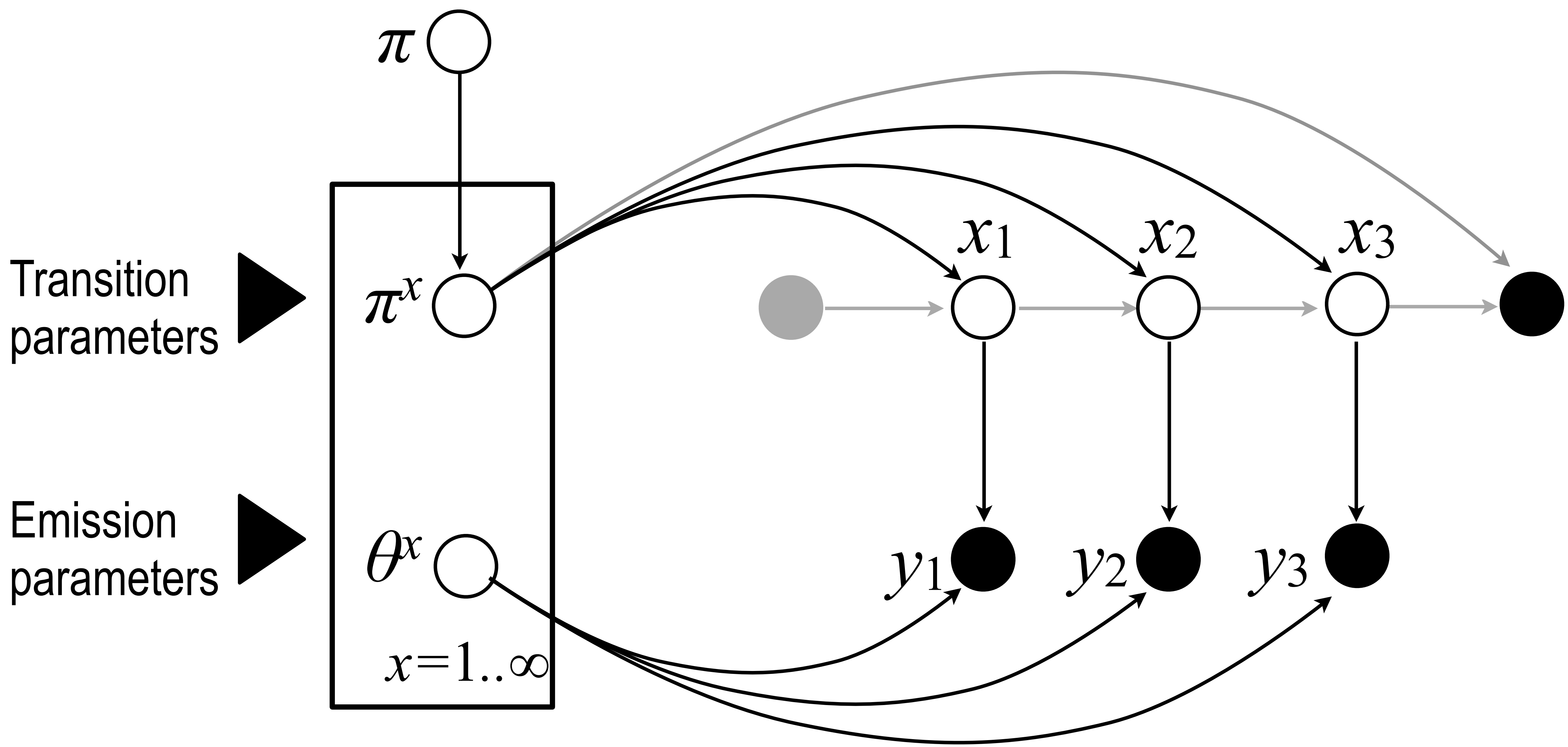
The infinite HMM

Infinite HMMs:



The infinite HMM

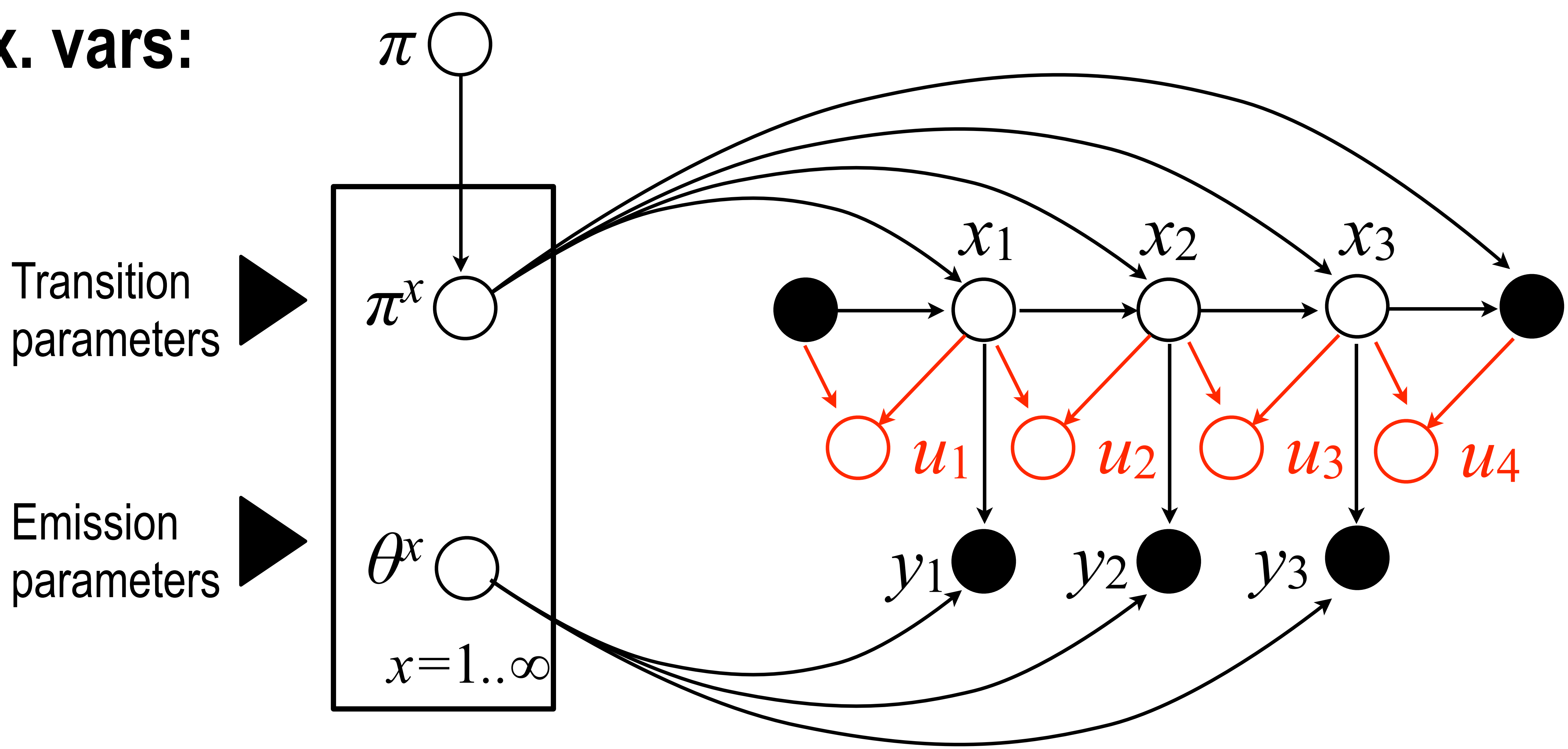
Infinite HMMs: connection with the Hierarchical Dirichlet process



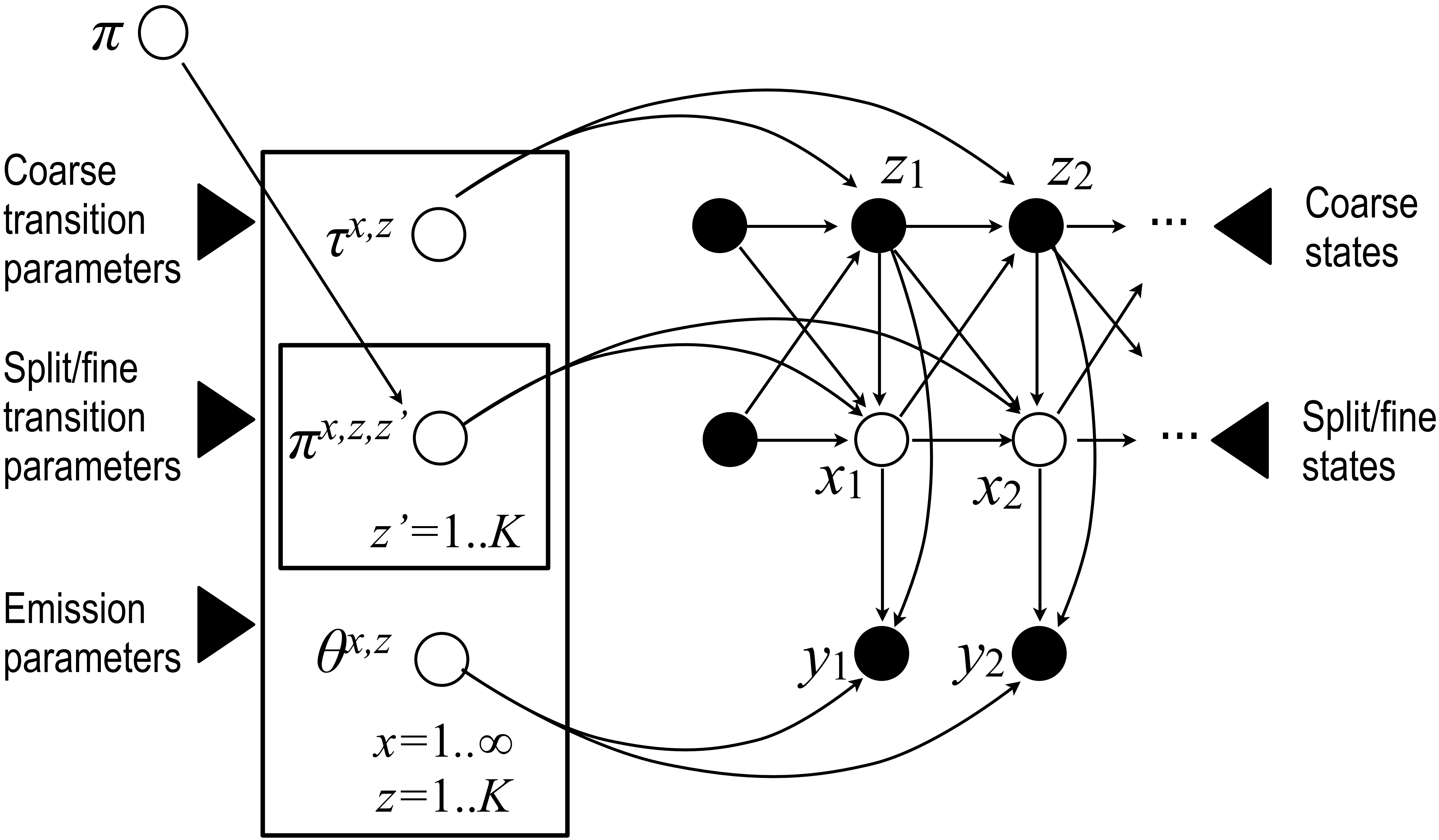
The infinite HMM

Computing the posterior: as usual, both a collapsed Gibbs sampler and a slice sampler are available

Aux. vars:



State splitting and iHMM



Limitation of iHMMs/DPs

There are many useful splits. Examples:

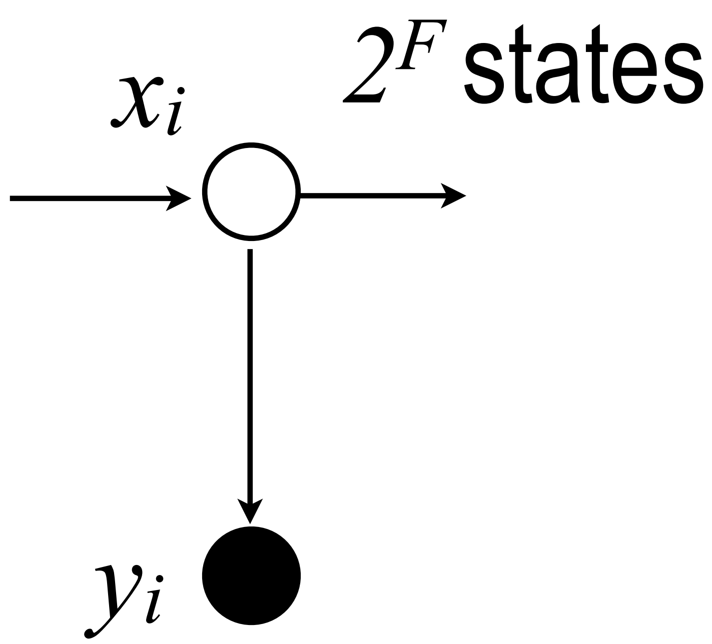
-3PS : when the sentence is 3th person singular
-INT : when the sentence is interrogative
-PAS : when the sentence is in the passive voice
...

Problem: representing the parameters of N splits takes $O(2^N)$ memory

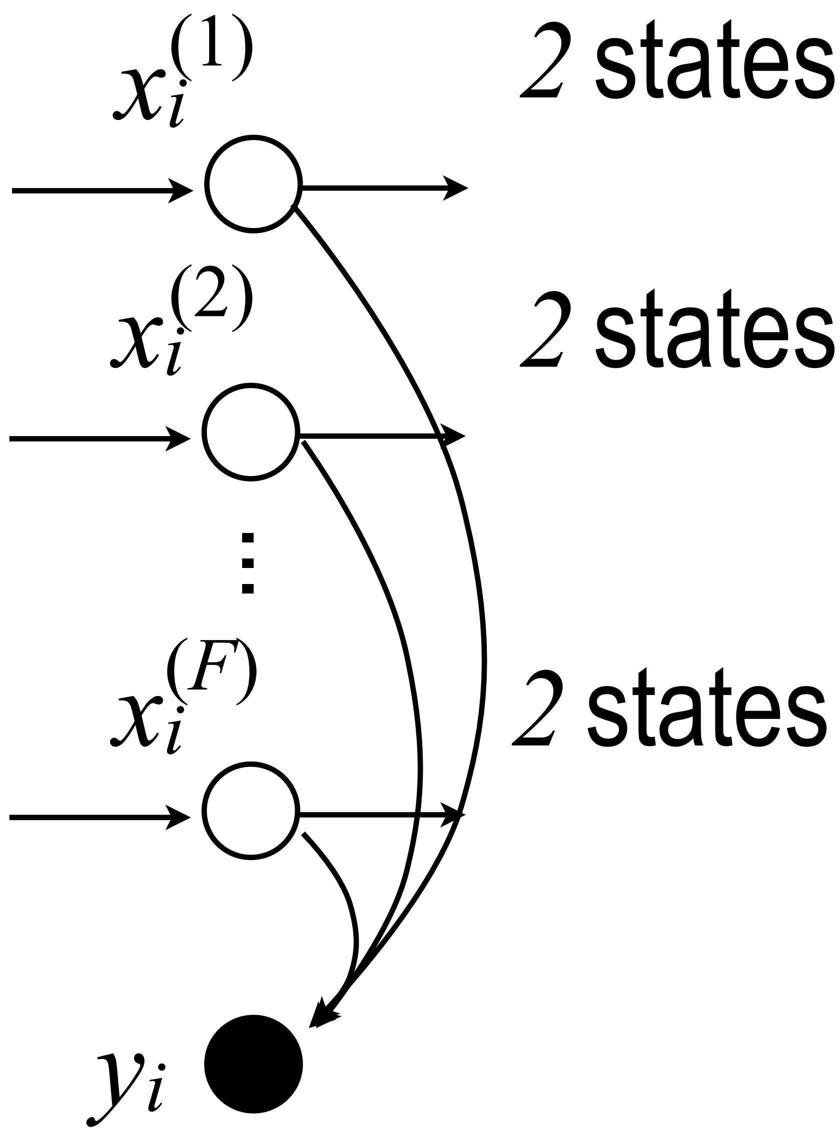
Solution: feature-based representations

Feature based representations

State-split



Feature



How many features? Will see soon a solution: Beta process

Another motivation

Input: Number of times people chose the row object over the column object.

	Phone 1	Phone 2	Phone 3
Phone 1	-	2	7
Phone 2	6	-	7
Phone 3	1	1	-

7 people chose Phone 1 over Phone 3

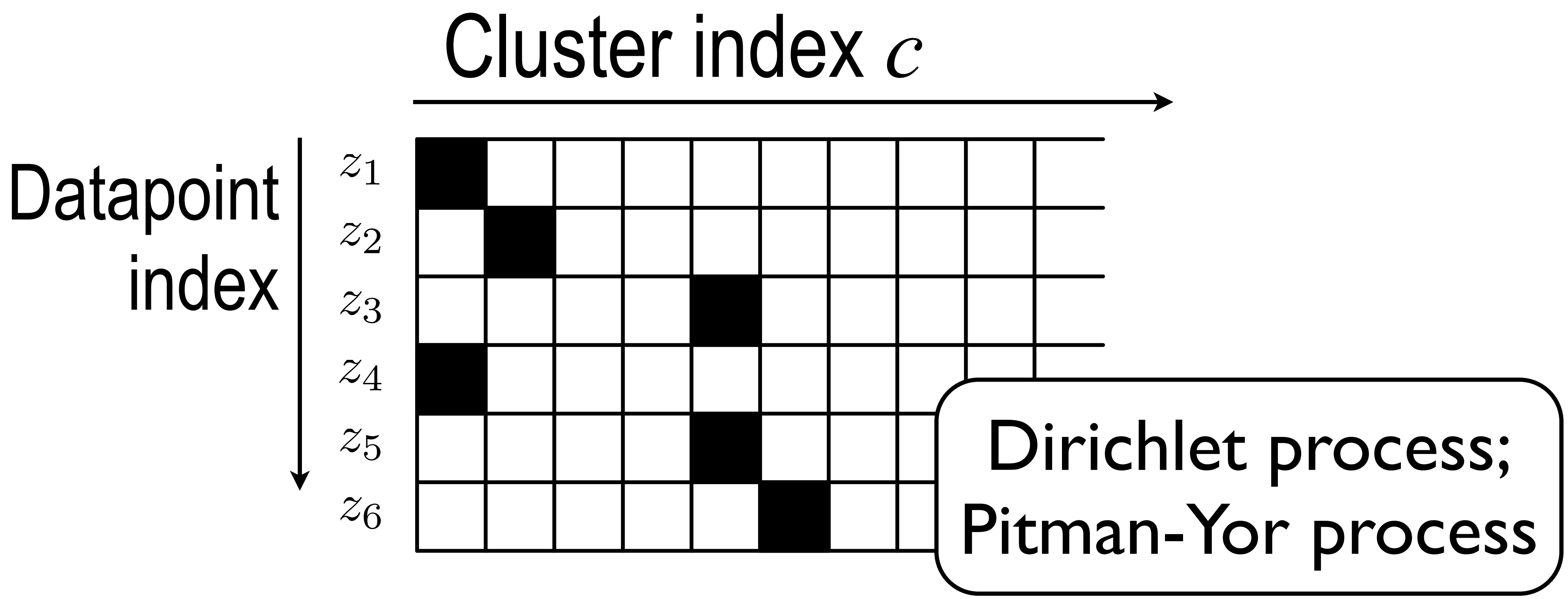
Desired output: latent features governing these choices

	Phone	Camera	Internet	Flip-phone	Cheap
Phone 1	✓	✓	✓		
Phone 2	✓	✓			✓
Phone 3	✓		✓	✓	

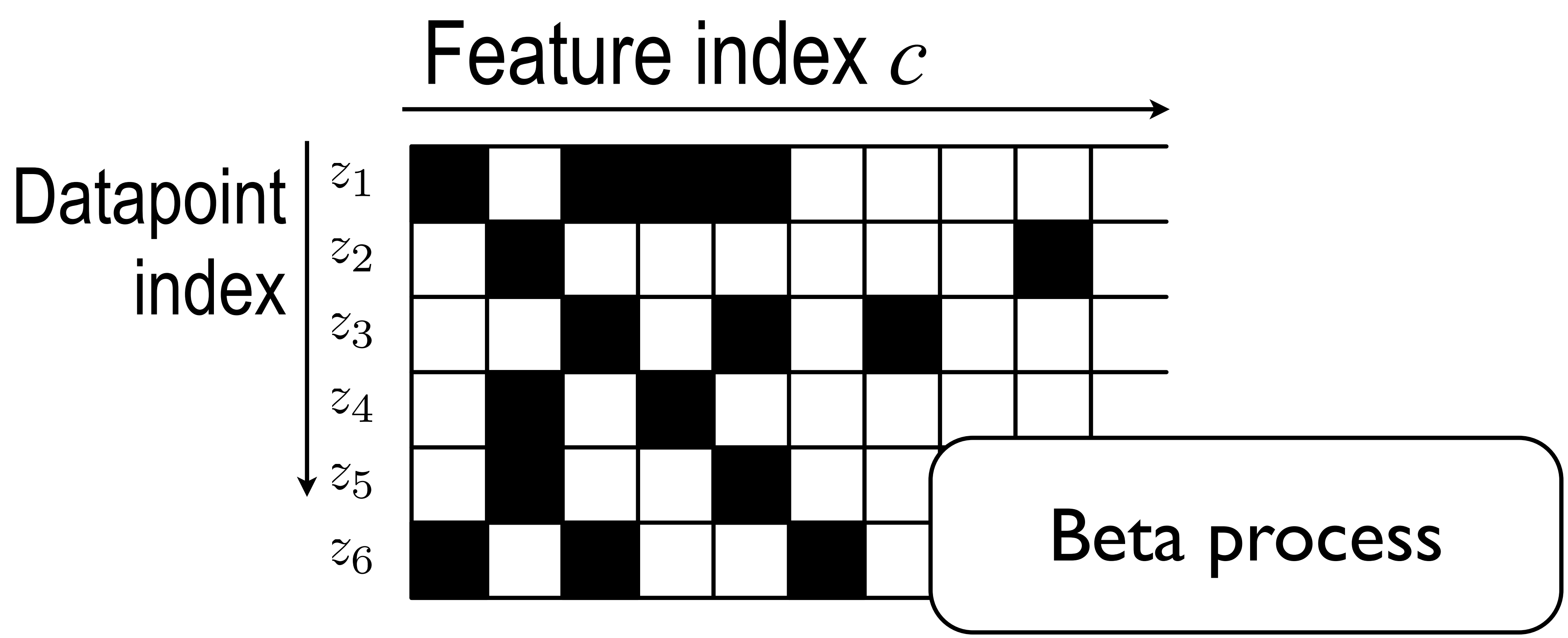
Slide from Kurt Miller

Beta process

Mixture indicator priors:

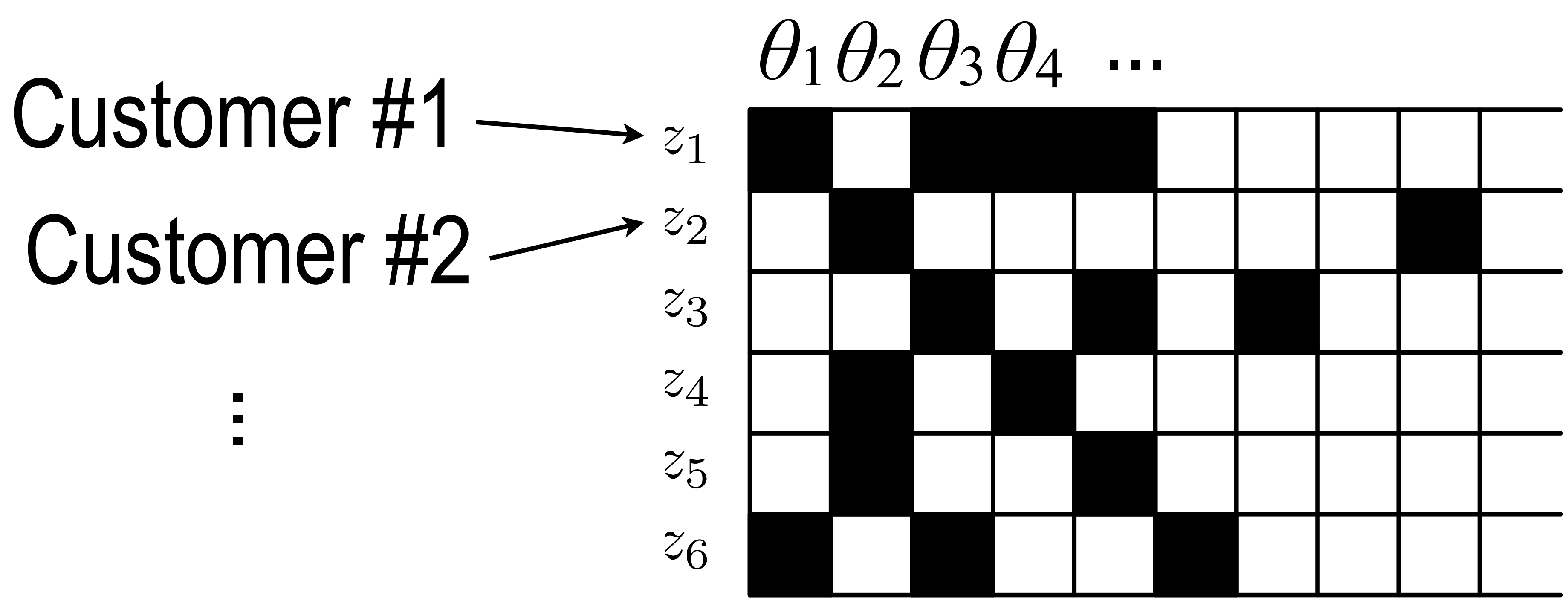


Feature indicator priors:



Predictive distribution: restaurant metaphor

Instead of a sit-down restaurant, think of a buffet with an infinite sequence of dishes θ_i sampled by customers

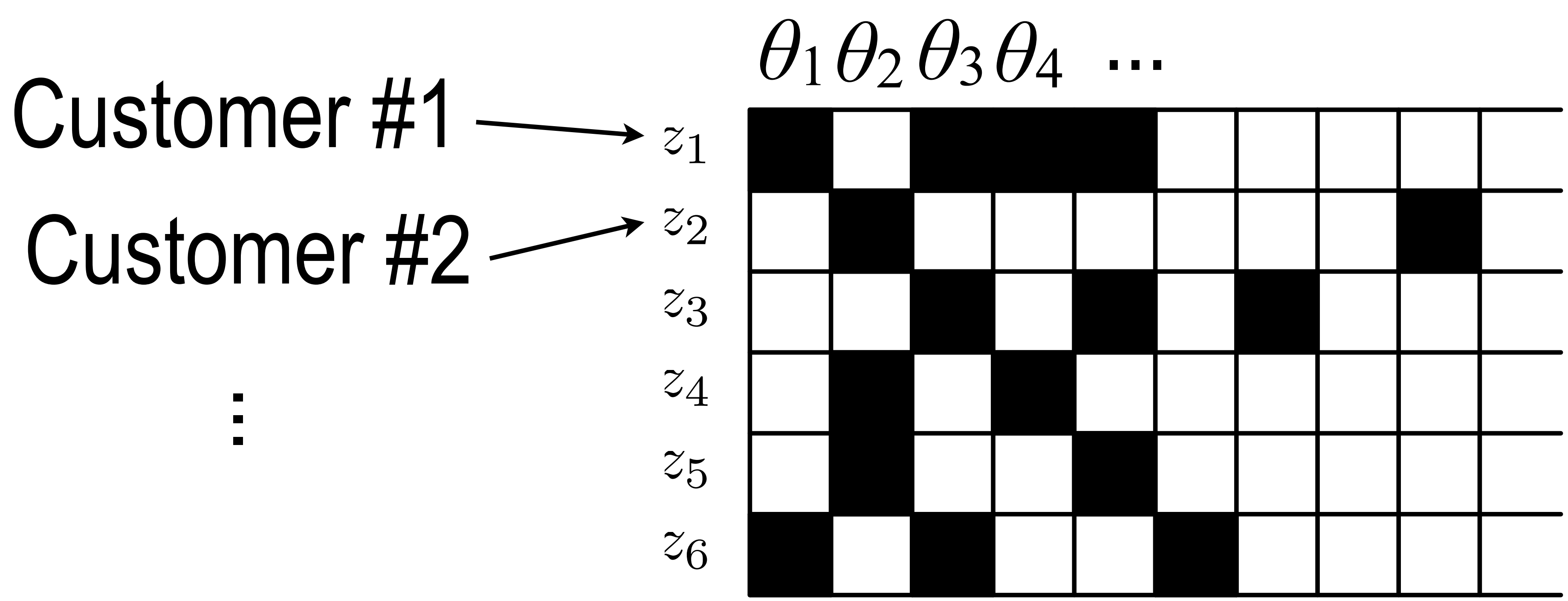


Obvious: order of the columns not important/exchangeable (because the θ_i's will be generated iid)

Less obvious: how to make the order of the rows exchangeable

Predictive distribution: restaurant metaphor

Instead of a sit-down restaurant, think of a buffet with an infinite sequence of dishes θ_i sampled by customers

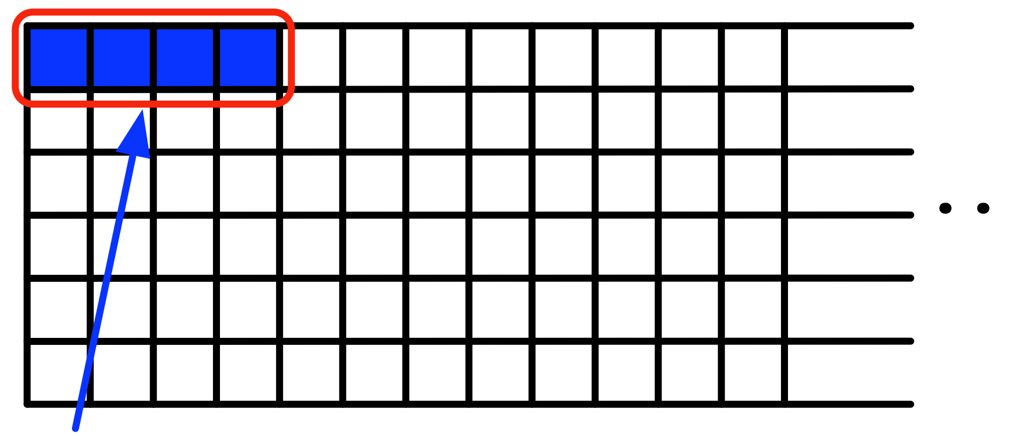


Obvious: order of the columns not important/exchangeable (because the θ_i's will be generated iid)

Less obvious: how to make the order of the rows exchangeable

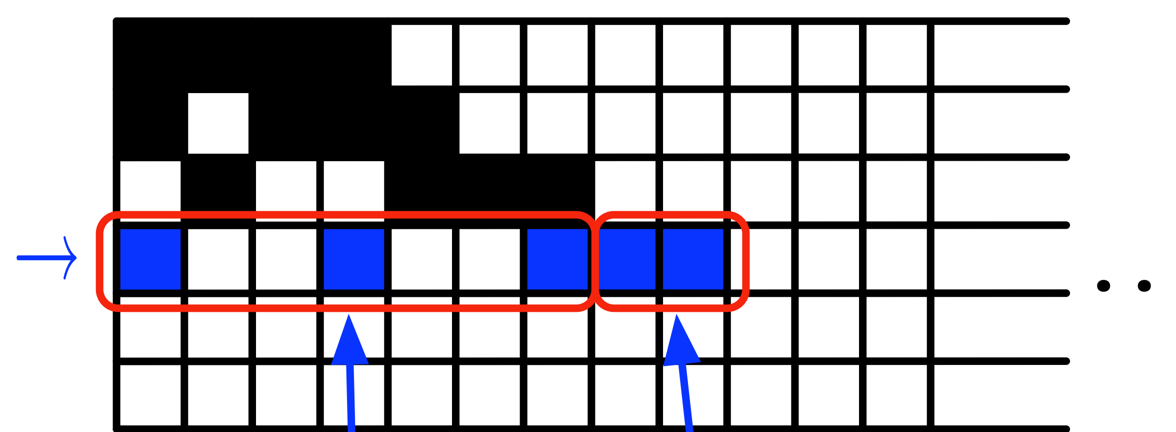
Predictive distribution: restaurant metaphor

First customer:



Sample a Poisson(α) number of dishes.

Fourth customer:



Sample a Poisson(α / i) number of new dishes.

Sample previously tried dishes in proportion to the number of people who have previously tried them.

(Example on the board)

Slide from Kurt Miller