Generative processes and graphical models

Jeremy R. Manning
Contextual Dynamics Lab

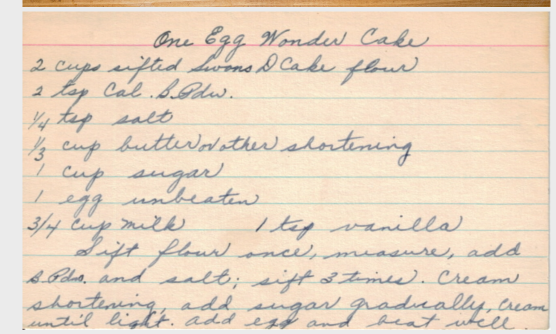
February 13, 2019

Generative process

- A generative process describes a recipe for generating data
- We will be designing “custom” probability distributions (building on top of other distributions) that we can draw the data from

Baking a cake

- Assumptions: the ingredients and amounts
- Recipe: combine the ingredients in a particular way. This might entail building “interim cake parts” (e.g. suspension of eggs + oil; mixture of dry ingredients; batter; batter + pan raw cake; etc.)
- Output: given the starting ingredients and particulars of the recipe, how likely is it that you’ll get any given cake instantiation? E.g. what are the chances that you’ll get the precise amount and distribution of frosting, the precise sprinkles configurations, precise air pocket placements, etc.



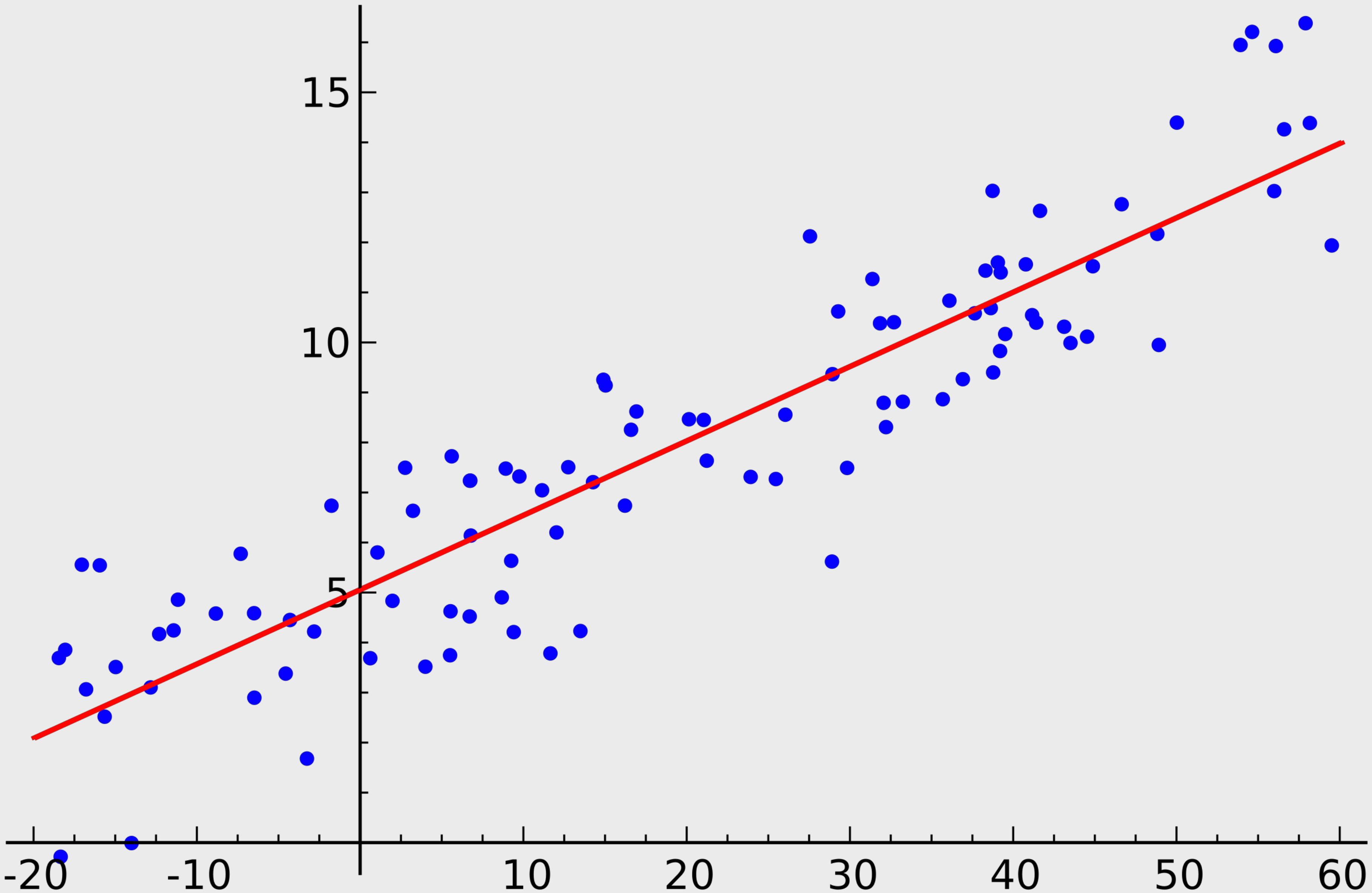
One Egg Wonder Cake

2 cups sifted Swans D Cake flour
2 tsp Cal. B.Pdw.
1/4 tsp salt
1/3 cup butter or other shortening
1 cup sugar
1 egg unbeaten
3/4 cup milk 1 tsp vanilla
Sift flour once, measure, add B.Pdw. and salt; sift 3 times. Cream shortening, add sugar gradually, cream until light. add egg and beat well

Example model: linear regression

- Suppose we have two variables, x and y
- Given **arbitrary** values of x, we want to predict what the corresponding y value is
- We'll use the equation for a line to generate our predictions: $y = mx + b$
  - What assumptions are we making?
  - What does "fitting the model" mean?
- Observations: $(x0, y0), (x1, y1), \ldots, (xN, yN)$ — i.e. a bunch of x and y values we know about

Example model: linear regression



Generative process for linear regression

- Draw the intercept b ~ N(mu_b, sigma_b)
- Draw the slope m ~ N(mu_m, sigma_m)
- For each x-value we want to make a prediction at (x1…N), draw yn ~ N(m(xn) + b, sigma_y)

- Key concepts:
  - x1…N are **observed values**— we don't need to fit them
  - mu_b, sigma_b, mu_m, sigma_m, and sigma_y are **hyperparameters**— we choose them a priori (we pretend they're observed)
  - The last step (drawing yn from a Normal distribution) tells us about the **likelihood**. What is p(yn | m, b, sigma_y)?

Graphical models

- Graphical models are a notation for describing generative processes
- They also tell us about how the joint probability distribution over the data and latent variables ("parameters") factorizes into the products of marginal and conditional distributions
- Each variable (observed or latent) is represented by a circle; hyperparameters are represented by dots
- Observed variables are shaded circles. Latent variables are open circles. Hyperparameters are also shaded.
- Lines connecting the circles tell us which parts of the model depend on which other things.

Graphical models

- Draw the intercept b ~ N(mu_b, sigma_b)
- Draw the slope m ~ N(mu_m, sigma_m)
- For each x-value we want to make a prediction at (x1…N), draw yn ~ N(m(xn) + b, sigma_y)

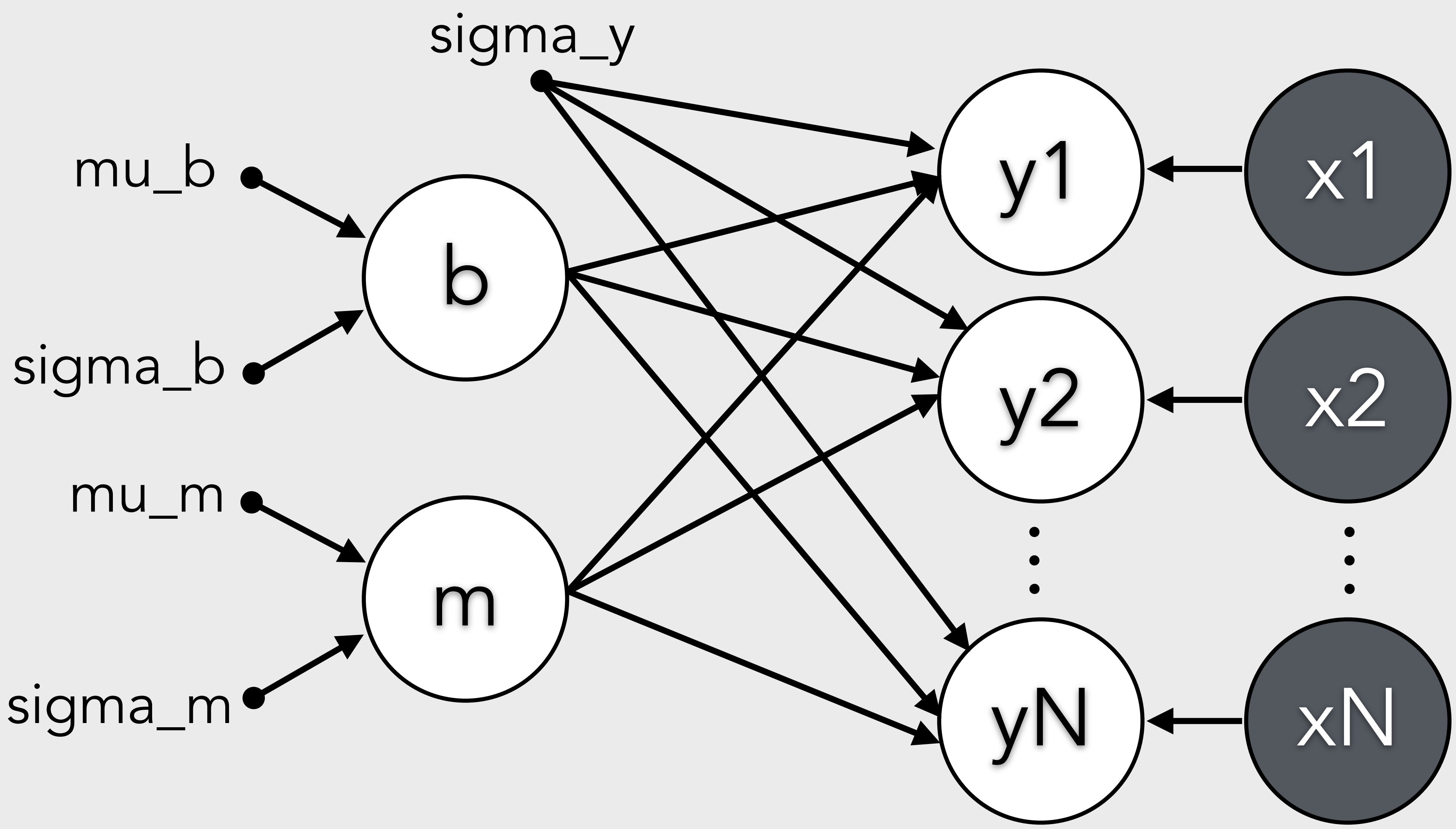
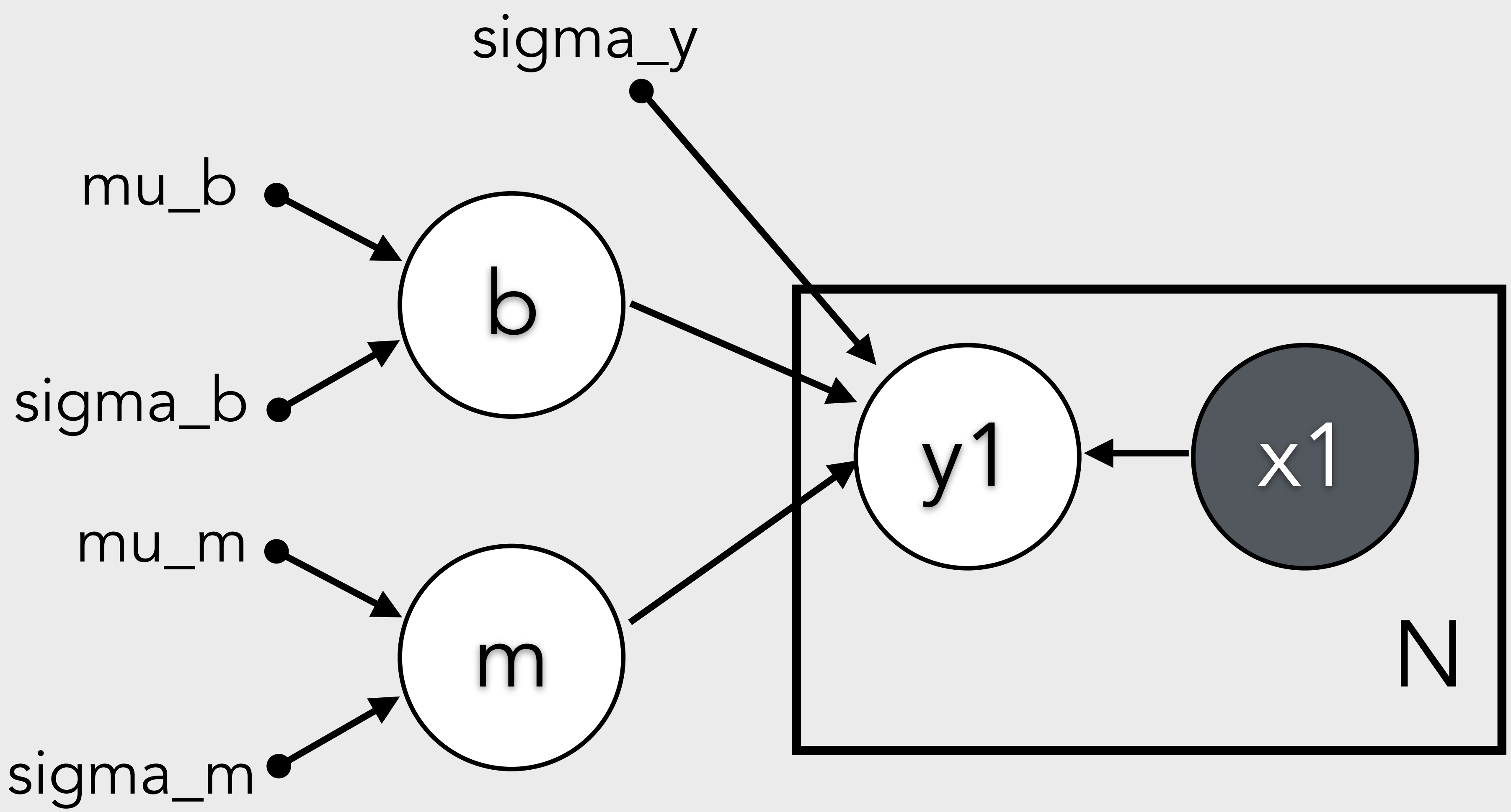


Plate notation:

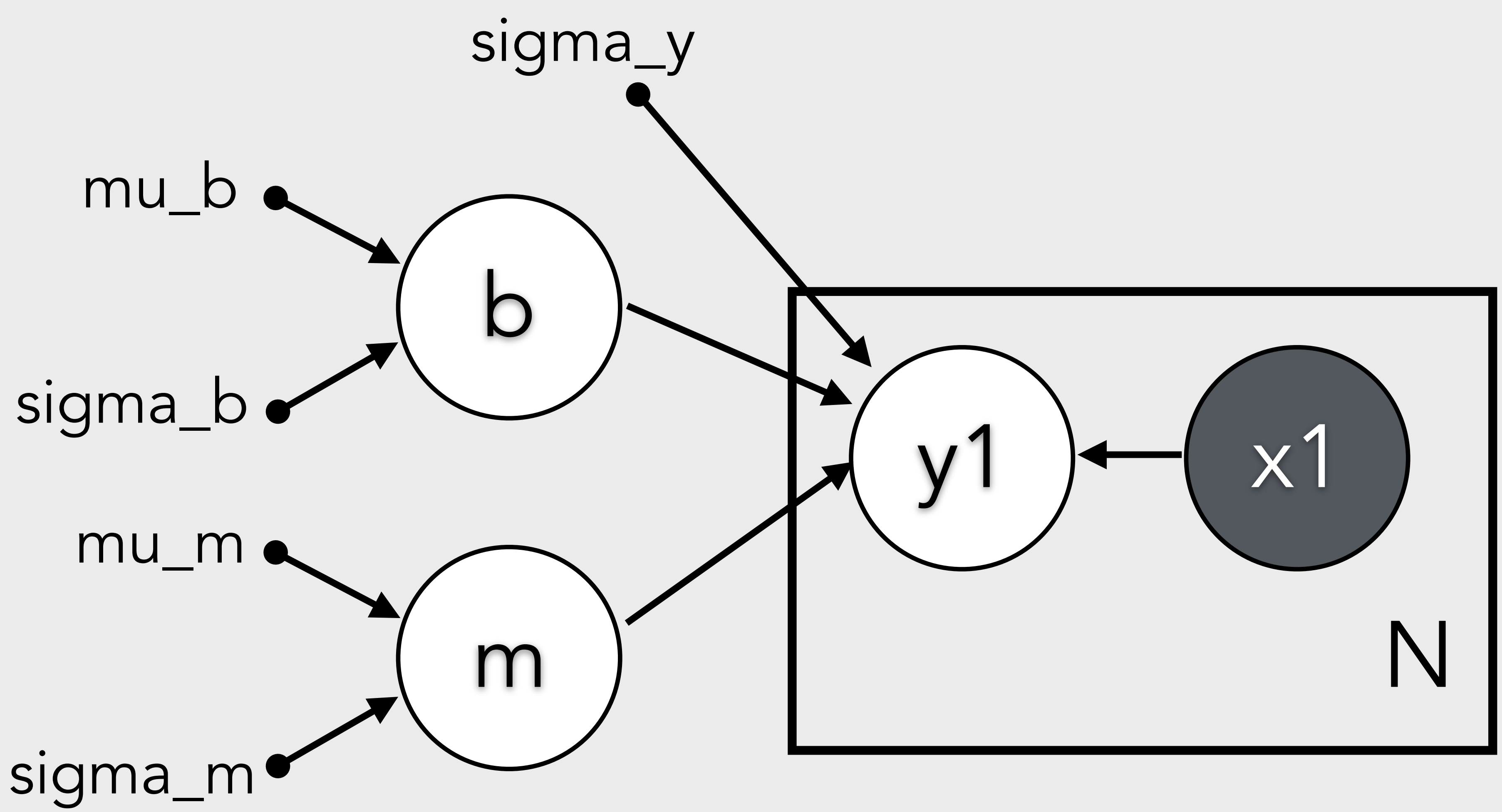
Plates show replicated structure

- Draw the intercept b ~ N(mu_b, sigma_b)
- Draw the slope m ~ N(mu_m, sigma_m)
- For each x-value we want to make a prediction at (x1…N), draw yn ~ N(m(xn) + b, sigma_y)



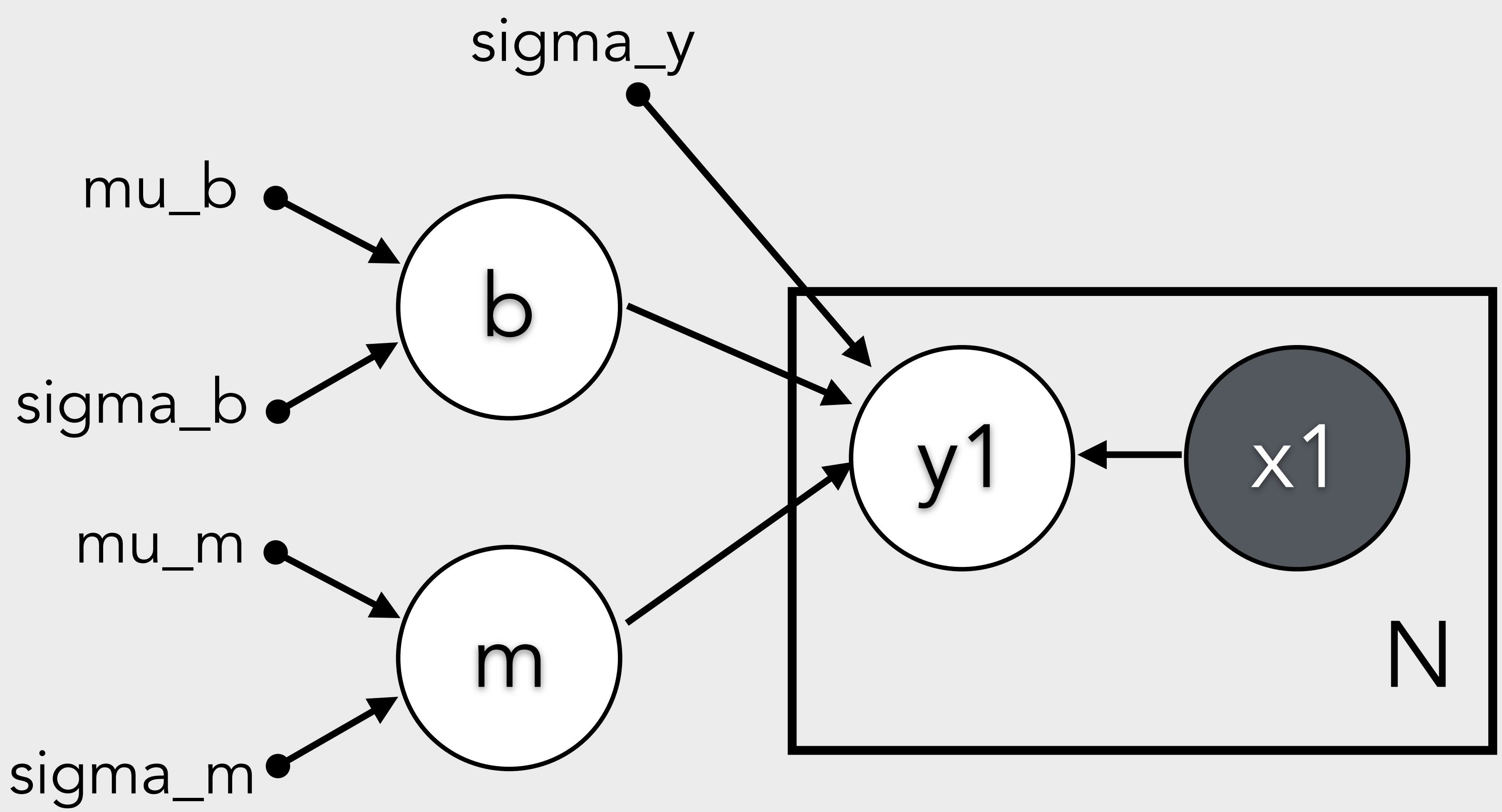
Graphical models and factorizing

- Every part of the model goes into the **joint distribution**
- Components where arrows **originate** go on the left of conditionals
- Arrow destinations go on the **right** of conditionals
- Components with not incoming arrows are **marginals**



Graphical models and factorizing

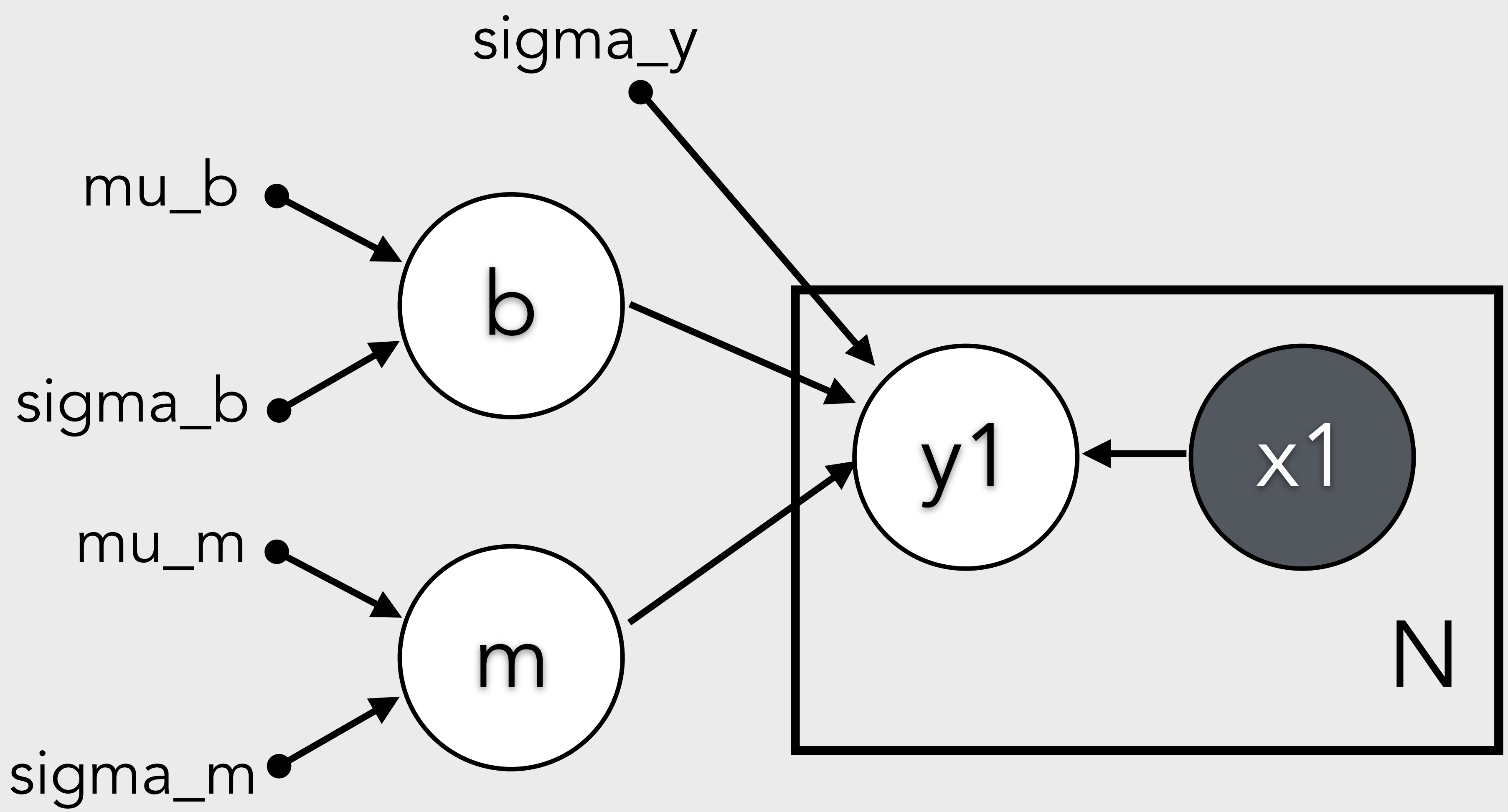
Joint distribution: $p(x_1...x_N, y_1...y_N, m, b, \sigma_y, \mu_b, \sigma_b, \mu_m, \sigma_m)$



Graphical models and factorizing

Joint distribution: $p(x_1 ... x_N, y_1 ... y_N, m, b, \sigma_y, \mu_b, \sigma_b, \mu_m, \sigma_m)$

Factorized joint distribution: $p(b|\mu_b, \sigma_b) p(m|\mu_m, \sigma_m) \prod_N p(x_n) p(y_n|x_n, b, m)$



Example: clustering

- For each of K clusters:
  - Select cluster center c_k
  - Select cluster covariance Sigma_k
- For each of N datapoints:
  - Select cluster assignment a(n)
  - Draw the data x_n ~ N(c_a(n), Sigma_a(n))

What's next?

- I propose that we proceed by reading the STAN reference manual: https://github.com/stan-dev/stan/releases/download/v2.17.1/stan-reference-2.17.1.pdf
- We'll carve out specific sections and/or find tutorials online, and then each coming week we'll tackle some aspect of STAN models