Approximate Sampling for Binary Discrete Exponential Families, with Fixed Execution Time and Quality Guarantees

Carter T. Butts

Department of Sociology and

Institute for Mathematical Behavioral Sciences

University of California, Irvine
buttsc@uci.edu

MURI AHM, 6/3/11

This work was supported in part by ONR award N00014-08-1-1015 and NSF award BCS-0827027.

Exponential Families on Binary Vectors

- Let $Y = Y_1, \ldots, Y_N$ be a finite vector of binary-valued random variables with joint support $\mathcal{Y}_N$

- Wlg, we may write the joint pmf of Y in discrete exponential family form as

$$\Pr\left(Y = y \,|\theta\right) = \frac{\exp\left[\theta^T t\left(y\right)\right]}{\sum_{y' \in \mathcal{Y}_N} \exp\left[\theta^T t\left(y'\right)\right]} I_{\mathcal{Y}_N}\left(y\right) \tag{1}$$

  - $t : \mathcal{Y}_N \mapsto \mathbb{R}^p$ is a vector of sufficient statistics, $\theta \in \mathbb{R}^p$ is a parameter vector, and $I_{\mathcal{Y}_N}$ is an indicator function for membership in the support
  - In general, cannot compute due to intractability of $\sum_{y' \in \mathcal{Y}_N} \exp\left[\theta^T t\left(y'\right)\right]$

- Useful feature of above – full conditionals easily obtained:

$$\Pr\left(Y_i = 1 \,|\theta, Y_i^c = y_i^c\right) = \left[1 + \exp\left[\theta^T \left[t\left(y_i^-\right) - t\left(y_i^+\right)\right]\right]\right]^{-1} \tag{2}$$

  - Y_i^c refers to all elements of Y other than the ith, Y_i^+ refers to the vector Y w/ith entry is set to 1, and Y_i^- refers to Y w/ith entry set to 0

A Sequential Take

- Note that we may also write Y's joint distribution in sequential form:

$$\Pr\left(Y = y \,|\theta\right) = \Pr\left(Y_1 = y_1 \,|\theta\right) \Pr\left(Y_2 = y_2 \,|\theta, Y_1 = y_1\right) \ldots \times \Pr\left(Y_N = y_N \,|\theta, Y_1 = y_1, \ldots, Y_{N-1} = y_{N-1}\right) \quad (3)$$

  - But what are these "partial conditionals" in the sequence?

- Let $Y^c_{<i} = Y_1, \ldots, Y_{i-1}$. By definition,

$$\Pr\left(Y_i = 1 \,|\theta, Y^c_{<i} = y^c_{<i}\right) = \sum_{y' \in \mathcal{Y}_n : {y'}^c_{<i} = y^c_{<i}} \Pr\left(Y_i = 1 \,\middle|\theta, Y^c_i = {y'}^c_i\right) \Pr\left(Y^c_i = {y'}^c_i \,|\theta, Y^c_{<i} = y^c_{<i}\right). \quad (4)$$

  - So the "partial conditionals" are convex combinations of the full conditionals for Y_i
  - And, as noted, those full conditionals are often easy to work with....

Bounding the Partial Conditionals

- An implication of Eq 4: full conditionals can be used to bound partial conditionals (a la Butts (2010))
  - $\min_{y' \in \mathcal{Y}_N : y'^c_{<i} = y^c_{<i}} \Pr(Y_i = y_i | Y_i^c = y_i'^c) \leq \Pr(Y_i = y_i | Y_{<i} = y_{<i})$
  - $\max_{y' \in \mathcal{Y}_N : y'^c_{<i} = y^c_{<i}} \Pr(Y_i = y_i | Y_i^c = y_i'^c) \geq \Pr(Y_i = y_i | Y_{<i} = y_{<i})$
  - Minima/maxima are over all sequences preserving outcomes prior to Y_i
  - Follows immediately from convexity condition
- Using Eq 2, we can easily express the bounds in logit form
  - Define the following:
    - $\alpha_i \leq \min_{y' \in \mathcal{Y}_N : y'^c_{<i} = y^c_{<i}} \left[1 + \exp\left[\theta^T \left[t\left(y_i'^-\right) - t\left(y_i'^+\right)\right]\right]\right]^{-1}$
    - $\beta_i \equiv \Pr(Y_i = 1 | Y_{<i} = y_{<i})$
    - $\gamma_i \geq \max_{y' \in \mathcal{Y}_N : y'^c_{<i} = y^c_{<i}} \left[1 + \exp\left[\theta^T \left[t\left(y_i'^-\right) - t\left(y_i'^+\right)\right]\right]\right]^{-1}$
  - Then $\alpha_i \leq \beta_i \leq \gamma_i$ (i.e., α and γ sandwich the partial conditionals)

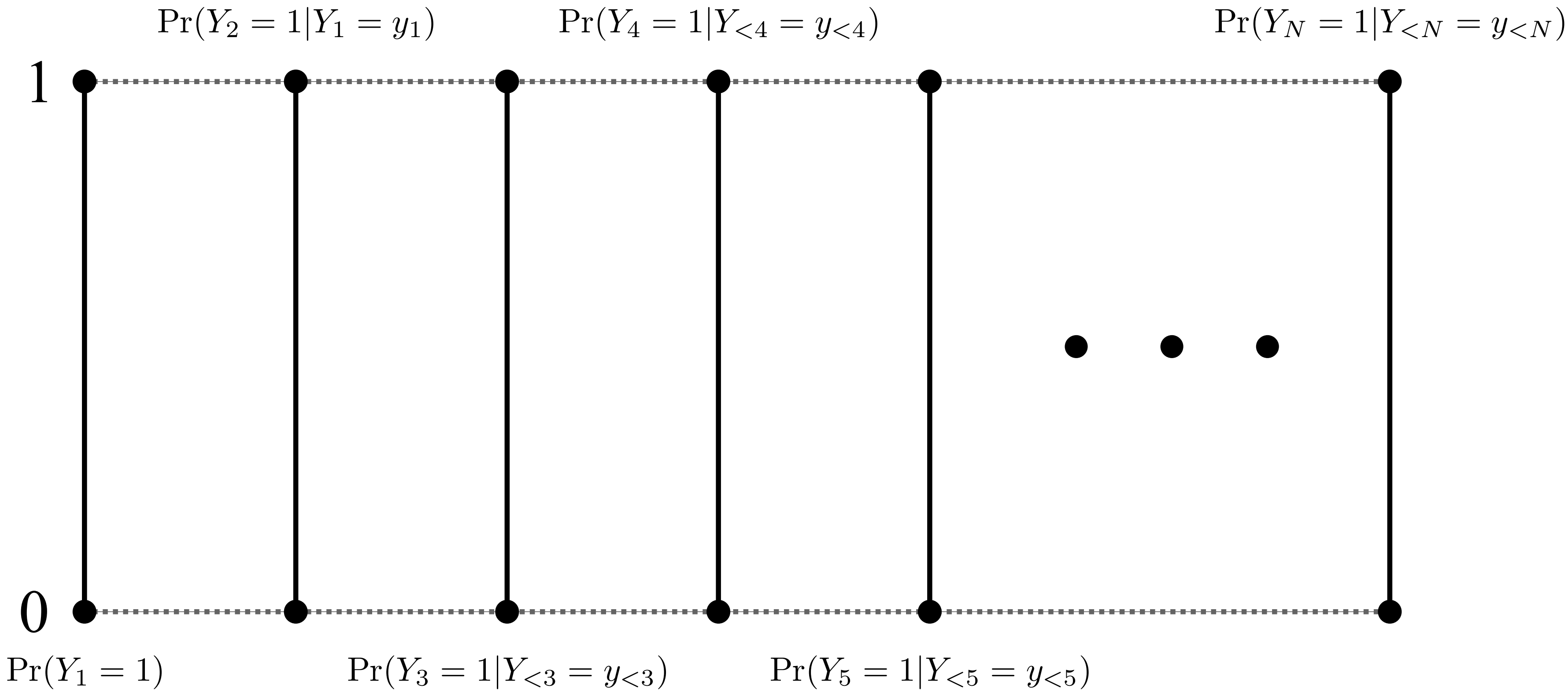
From Bounds to Simulation

- Simple, exact, and useless method of simulating a draw y from Y
  - Draw r from $R = R_1, \ldots, R_N$, w/$R_i \sim U(0,1)$
  - For $i \in 1, \ldots, N$, let $y_i = 1$ if $r_i < \Pr(Y_i = 1 | \theta, Y^c_{<i} = y^c_{<i}) = \beta_i$, else $y_i = 0$
- Problem: we don't know β_i! But sometimes, we don't have to...
  - If $r_i < \alpha_i$, then $r_i < \beta_i$, so we know that $y_i = 1$
  - If $r_i \geq \gamma_i$, then $r_i \geq \beta_i$, so we know that $y_i = 0$
  - In these cases, we say that α, γ "fix" y_i
- Further observation: when y_i is "fixed," this restricts later α, γ values
  - α and β can only "tighten" as more values are fixed
  - "Fixation cascade:" fixing y_i tightens bounds for y_{i+k}, fixing it (which may in turn fix others)
  - Sometimes, will fail; may have to get out and push
- This suggests an algorithm....

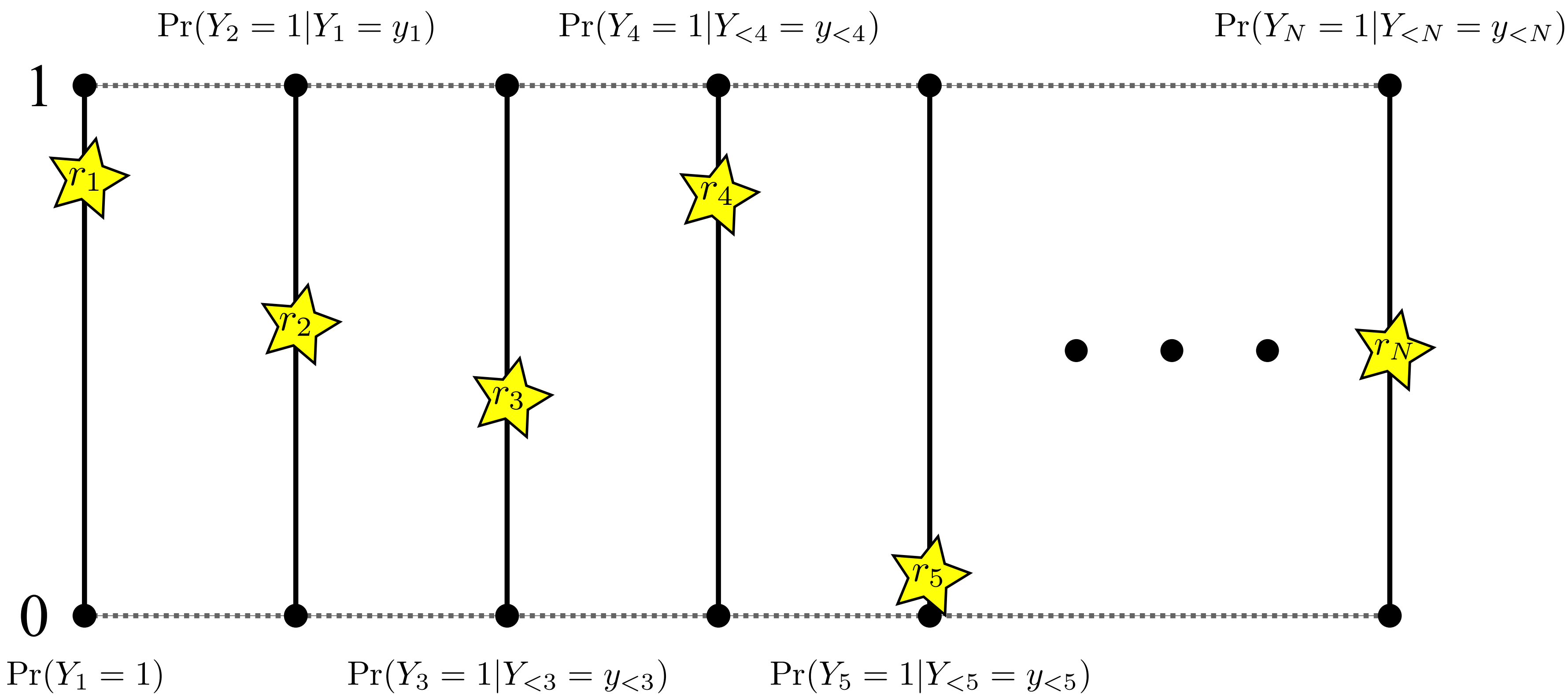
The “Bound Sampler”

- Basic algorithm for approximate simulation of Y †
  - For $i \in 1, \ldots, N$, perform the following steps:
    - *Initialization*: compute α_i, γ_i given $y_{<i}, \theta$
    - *Fixation*: if $r_i < \alpha_i$, set $y_i = 1$; if $r_i \geq \gamma_i$, set $y_i = 0$
    - *Perturbation*: if $\alpha_i \leq r_i < \gamma_i$, then “perturb” by drawing $\tau \sim U(\alpha_i, \gamma_i)$ and set $y_i = I(r_i < \tau_i)$ †
  - Resulting y vector is an approximate draw from Y
- Bound sampler has some interesting features
  - Draws are independent (unlike MCMC)
  - Worst-case time complexity fixed at $\mathcal{O}(Nf(N))$, where $f(N)$ is complexity of the initialization step (can be optimal, if $f(N)$ constant)
  - Can obtain quality guarantees: if y_i is fixed initially or by an unperturbed cascade, it *must* equal its “true” value!
    - “True” means wrt a coupled exact sampler with same R
    - Quality is a lower bound: unfixed y_i might still be correct, and fixed y_i certainly correct †

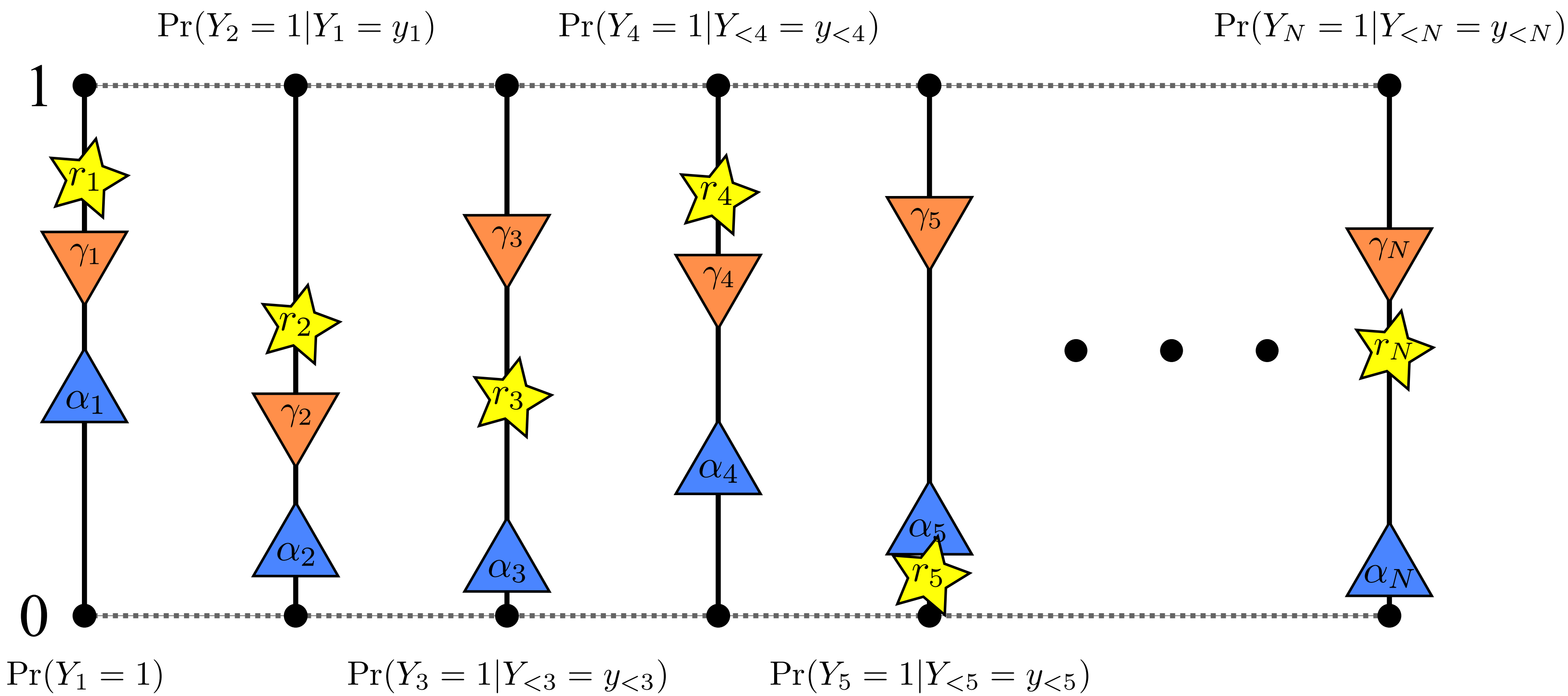
Sampling Process, Illustrated



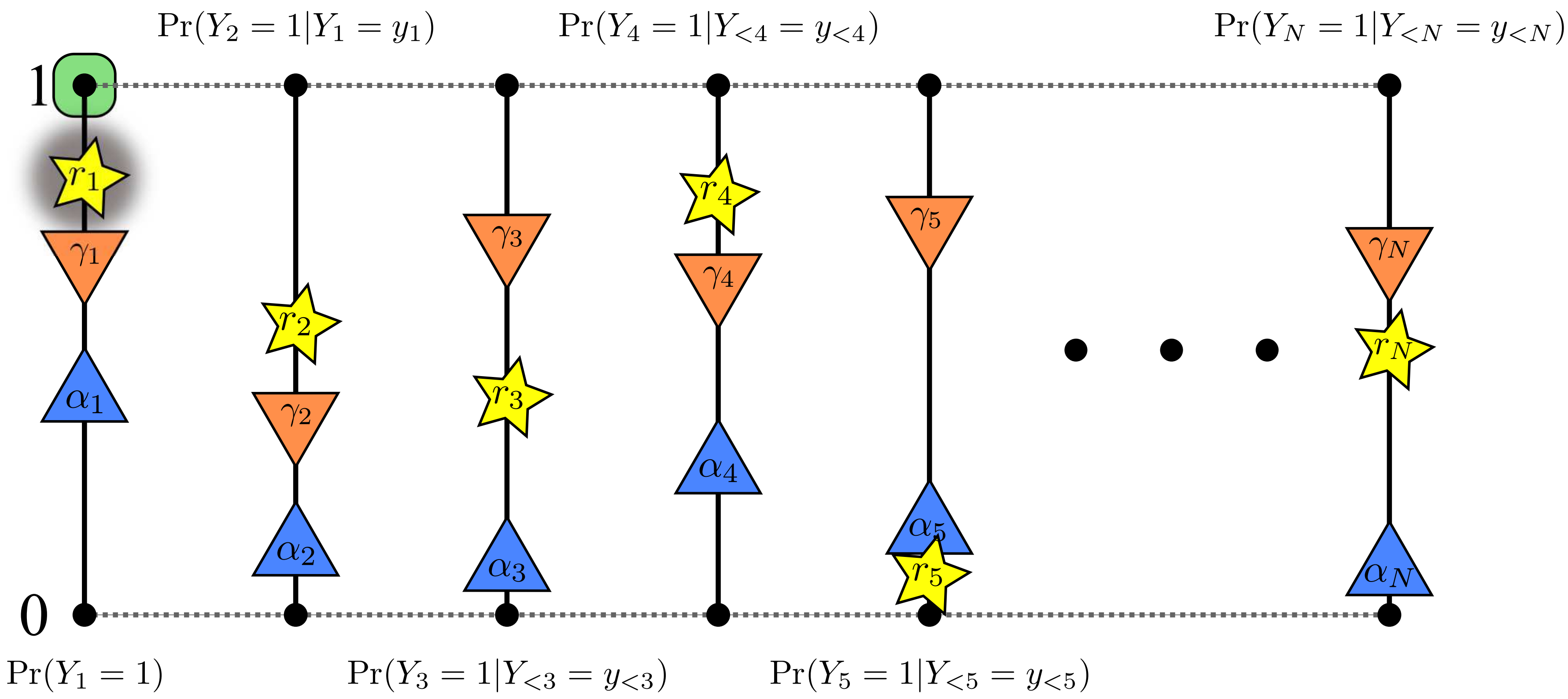
Sampling Process, Illustrated



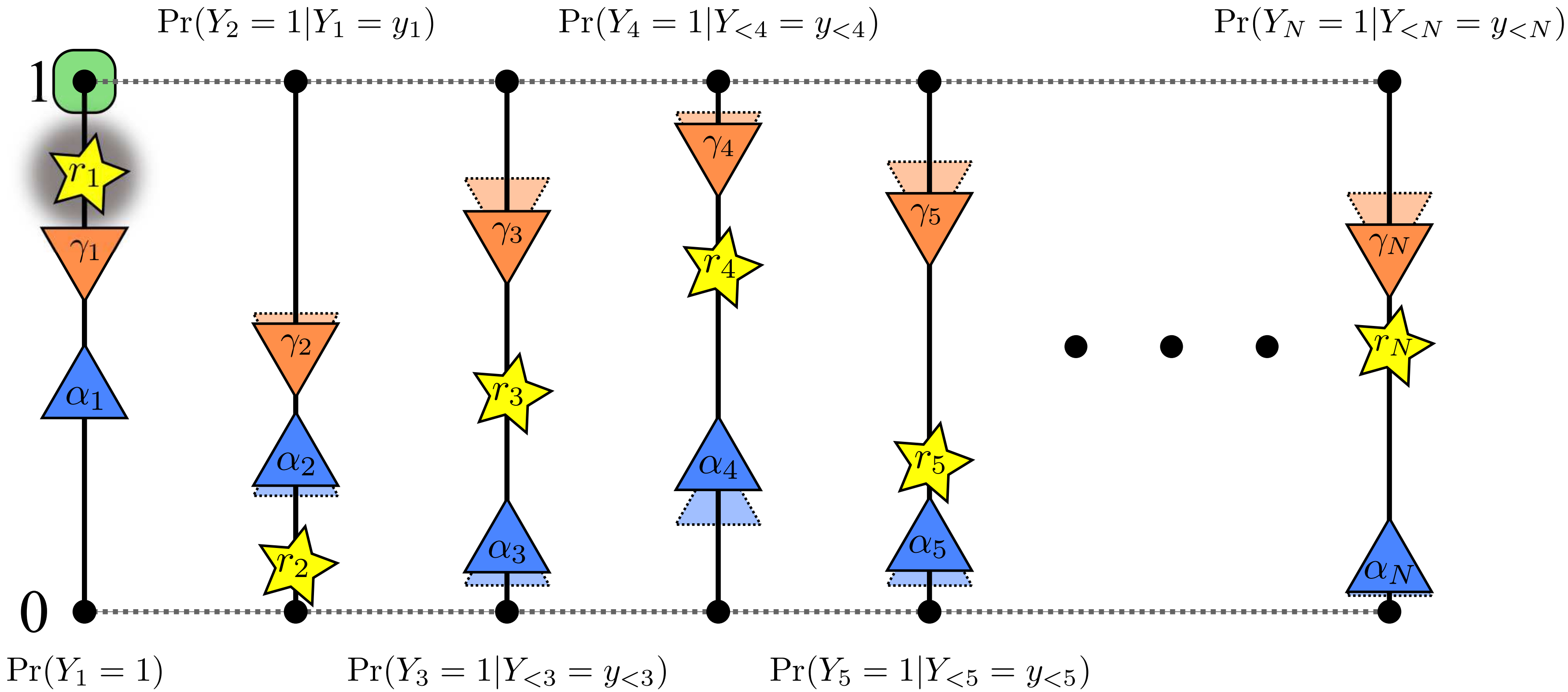
Sampling Process, Illustrated



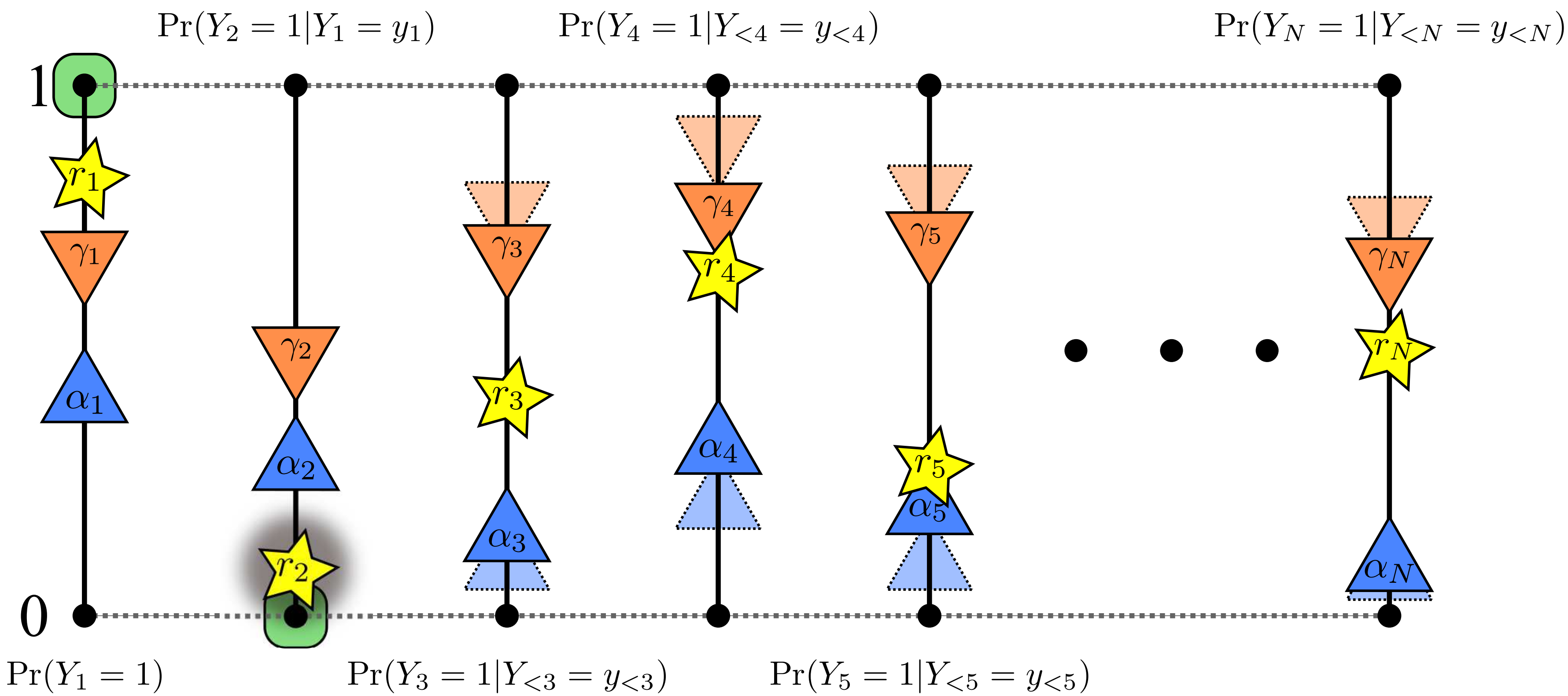
Sampling Process, Illustrated



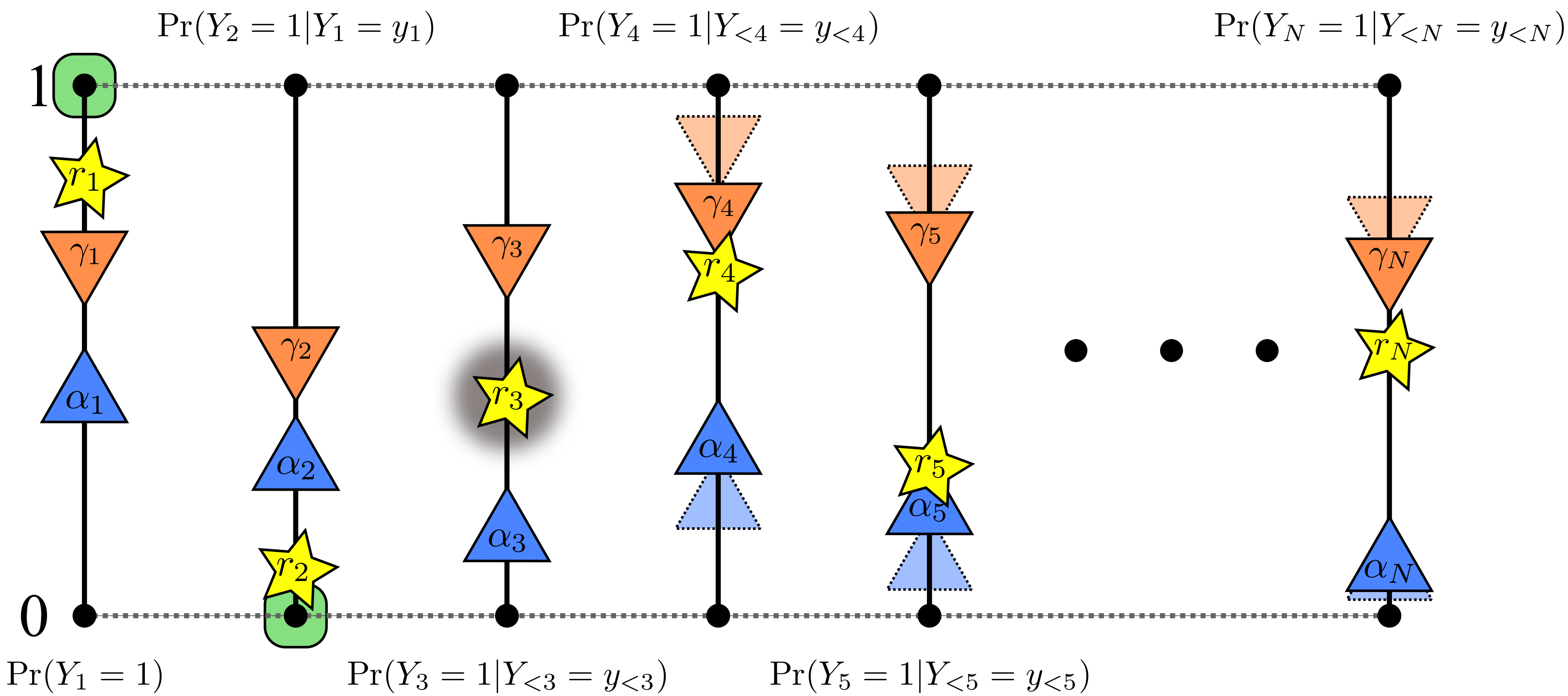
Sampling Process, Illustrated



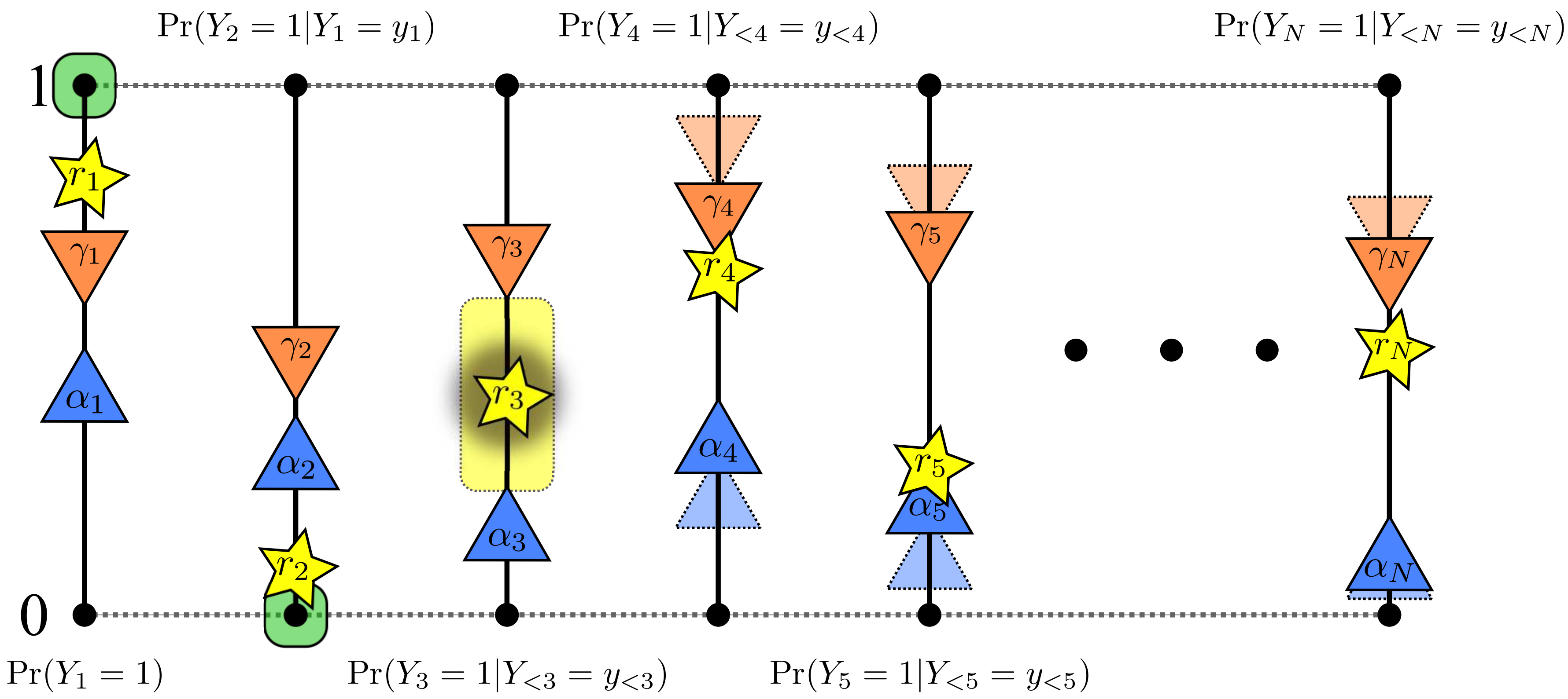
Sampling Process, Illustrated



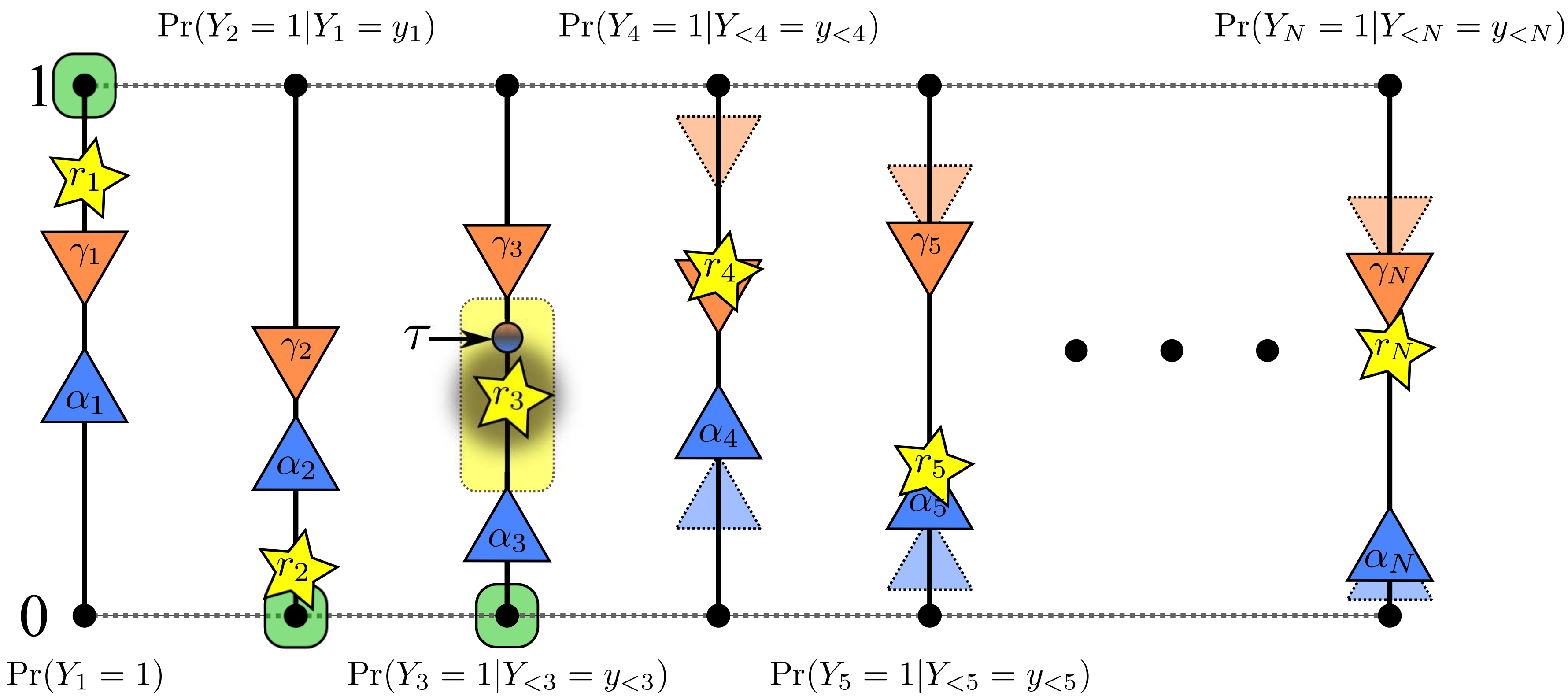
Sampling Process, Illustrated



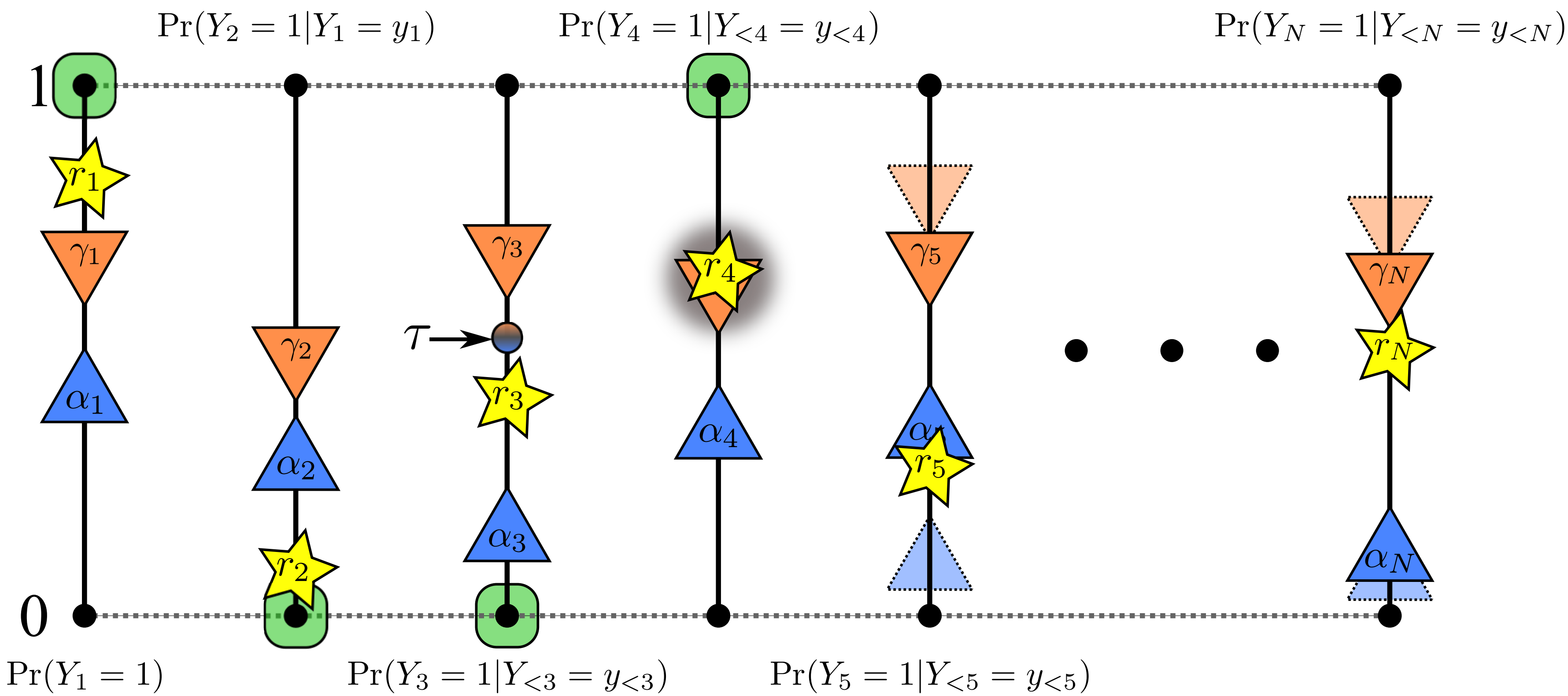
Sampling Process, Illustrated



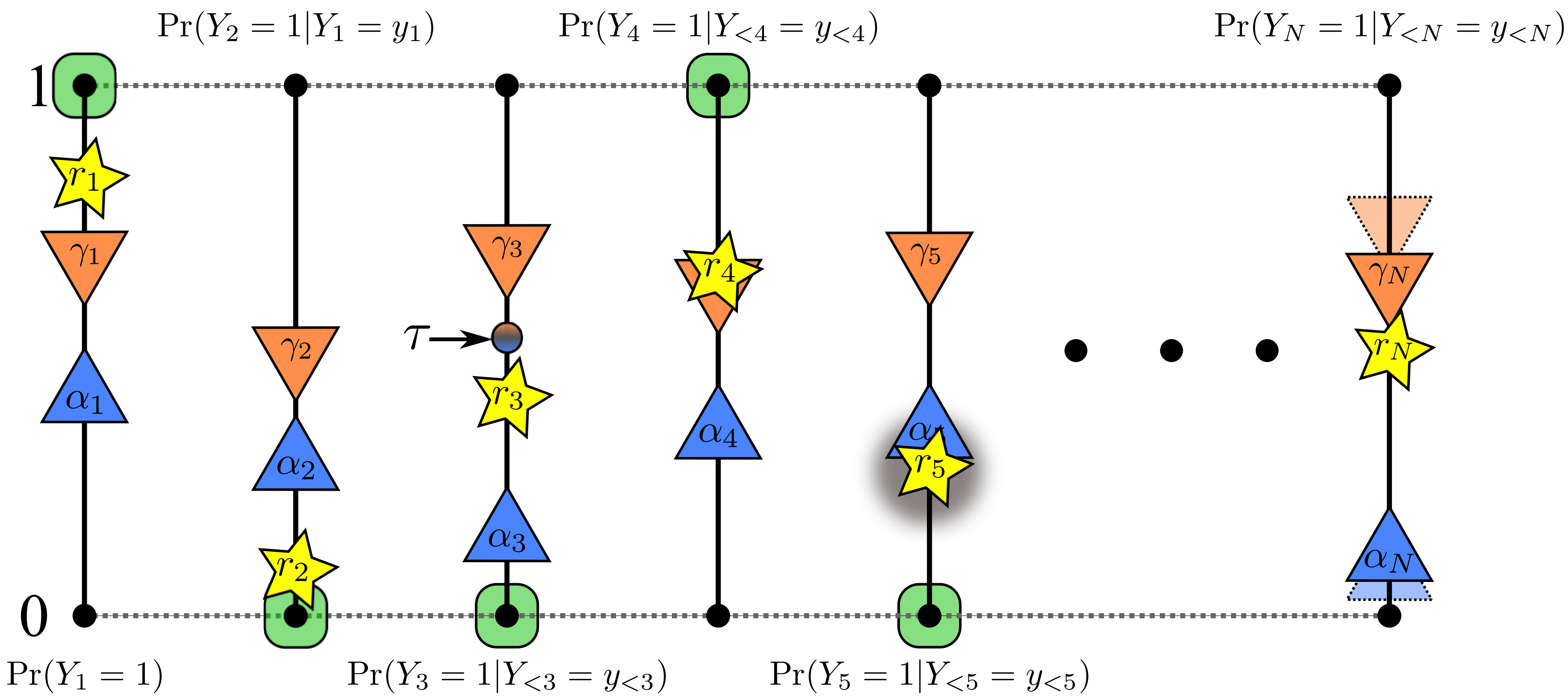
Sampling Process, Illustrated



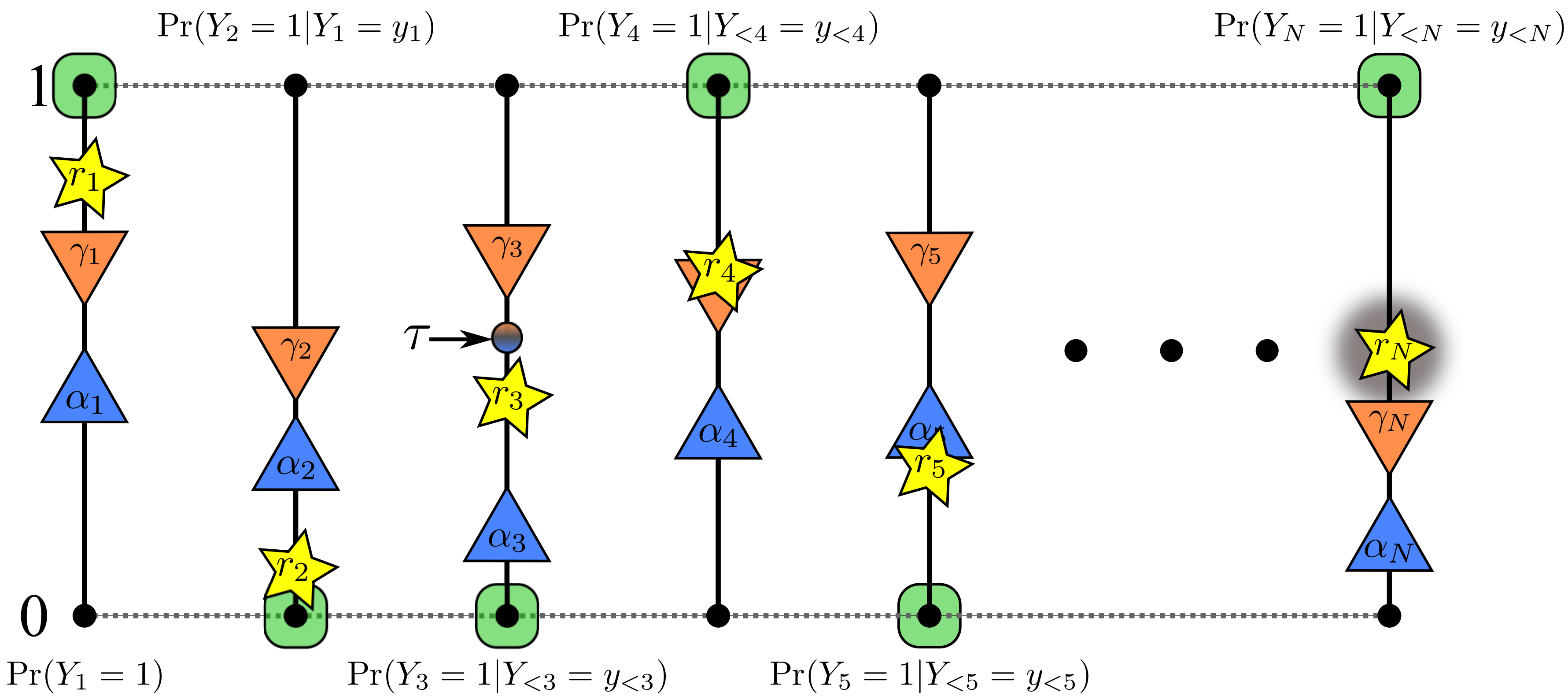
Sampling Process, Illustrated



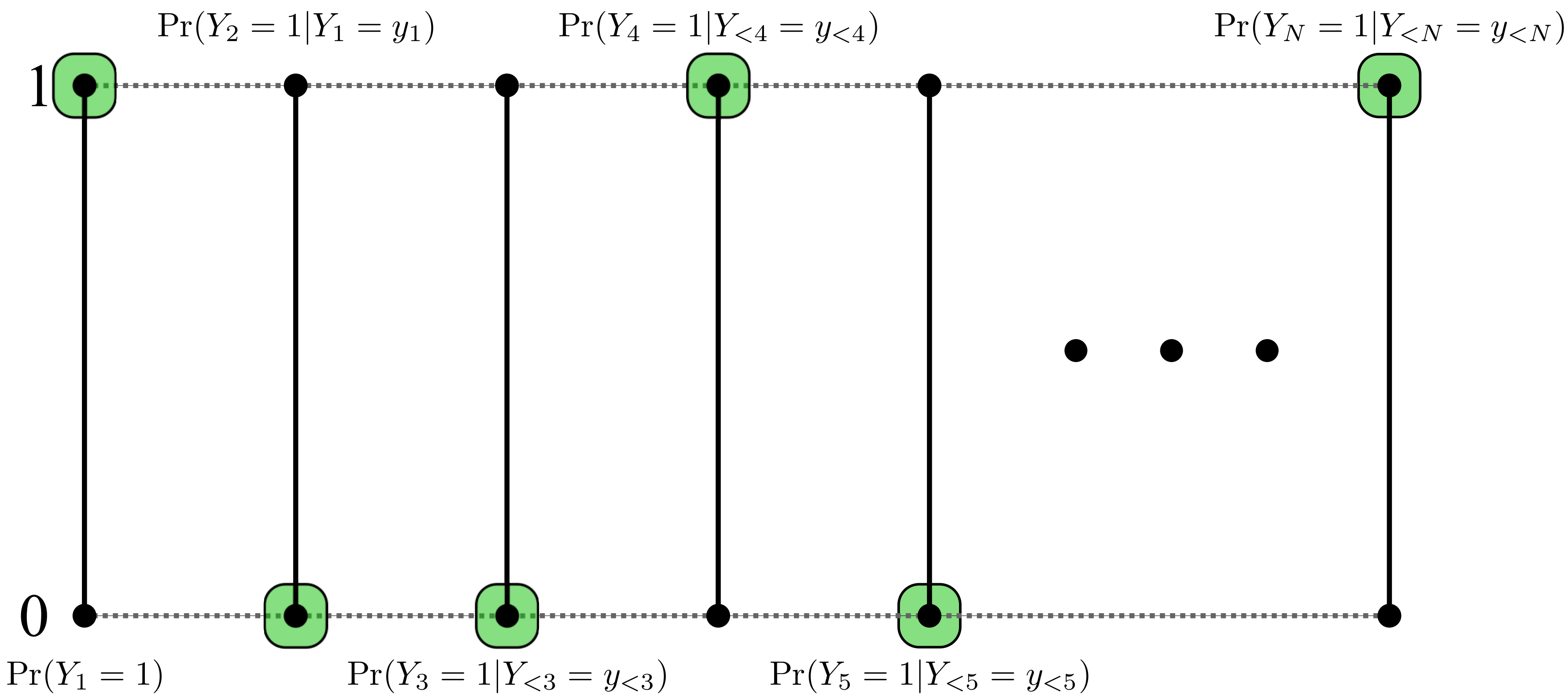
Sampling Process, Illustrated



Sampling Process, Illustrated



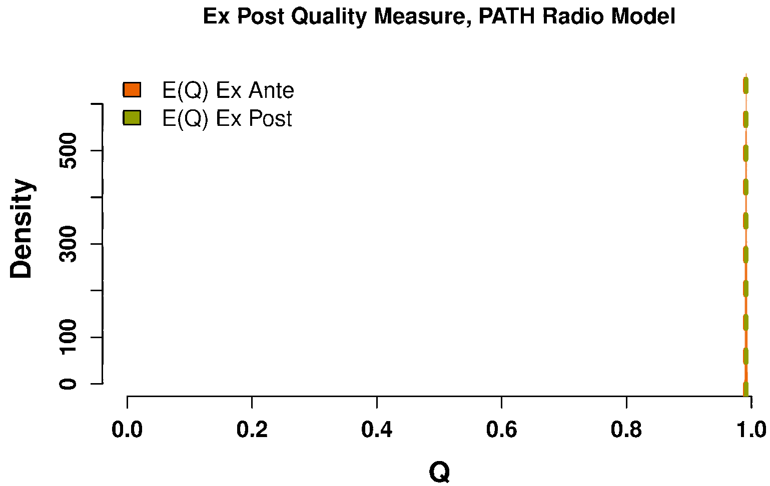
Sampling Process, Illustrated

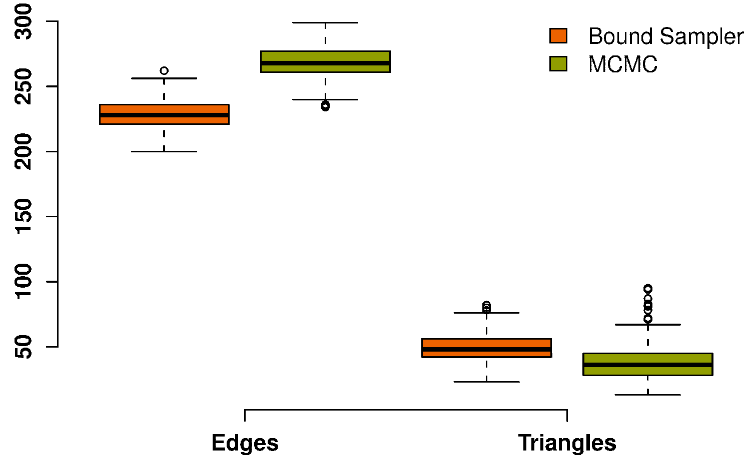


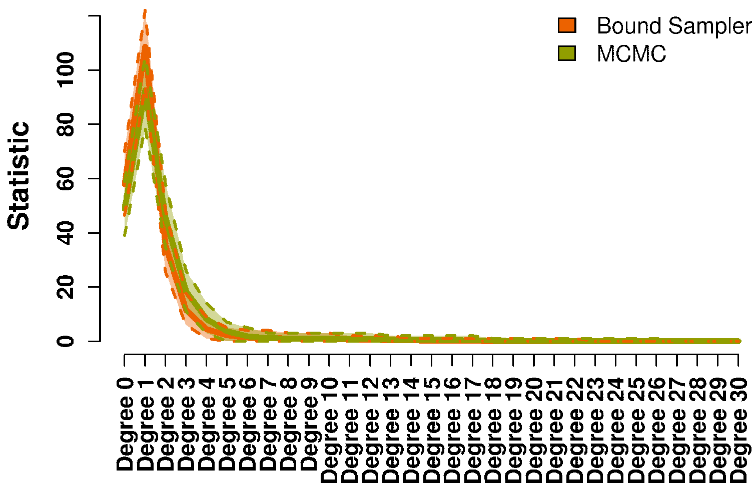
Application to Network Simulation

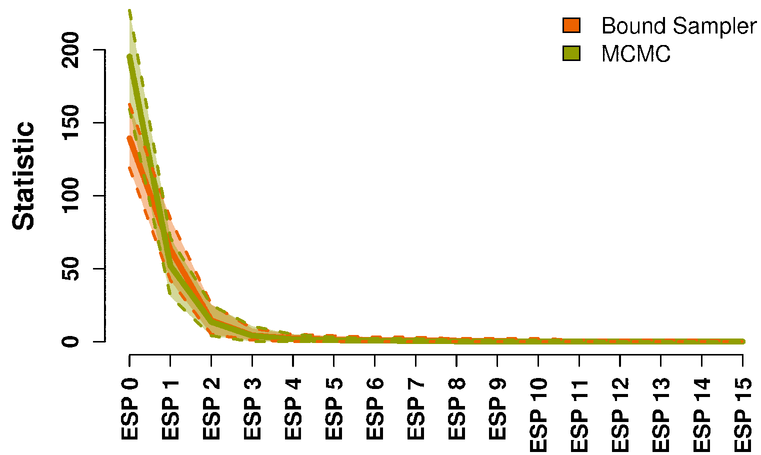
- Bound sampler can easily be used to simulate network processes
  - Write network model in ERG form (in terms of adjacency matrix)
  - “Vectorize” the adjacency matrix via row or column concatenation
  - Apply the bound sampler to the resulting vector
- Potential advantages
  - Fixed execution time (and can be very fast)
  - Unsupervised (e.g., no convergence diagnostics)
  - Quality guarantees
- Potential drawbacks
  - Fast implementation can require smart data structures (need cheap initialization)
  - If it doesn’t work well, you’re screwed (but could use smarter perturbation heuristics); can’t trade time for quality

Example: 9/11 PATH Radio Communications

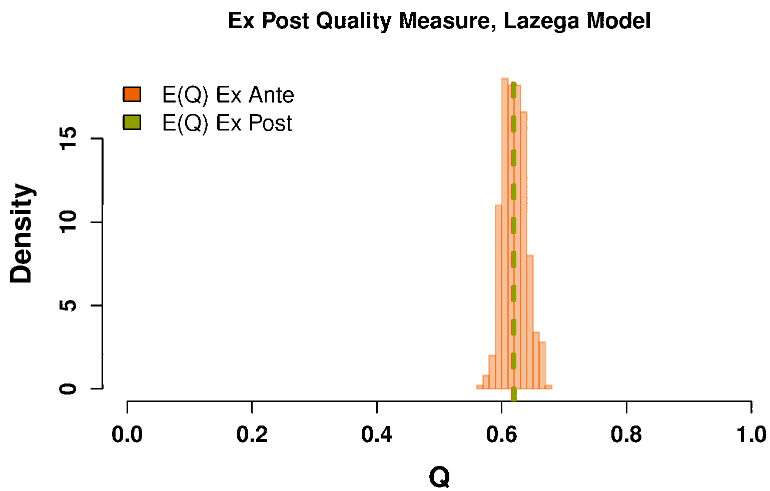


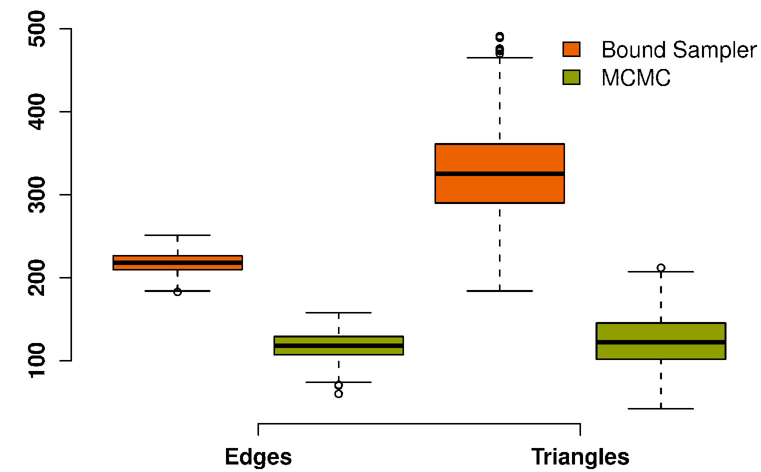


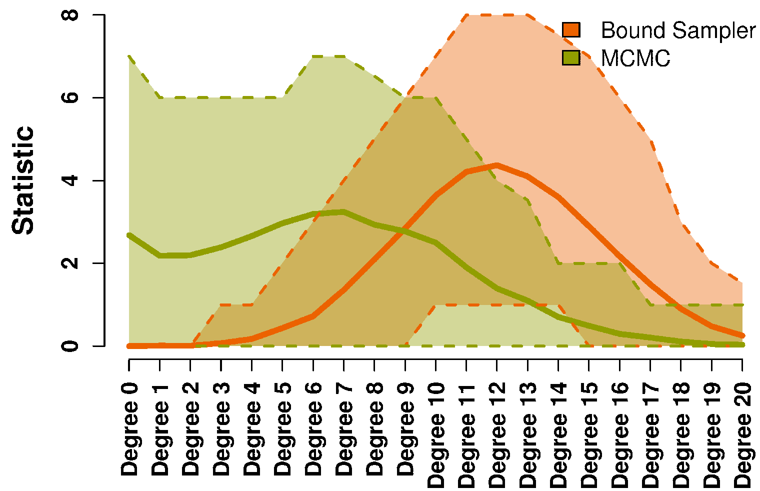


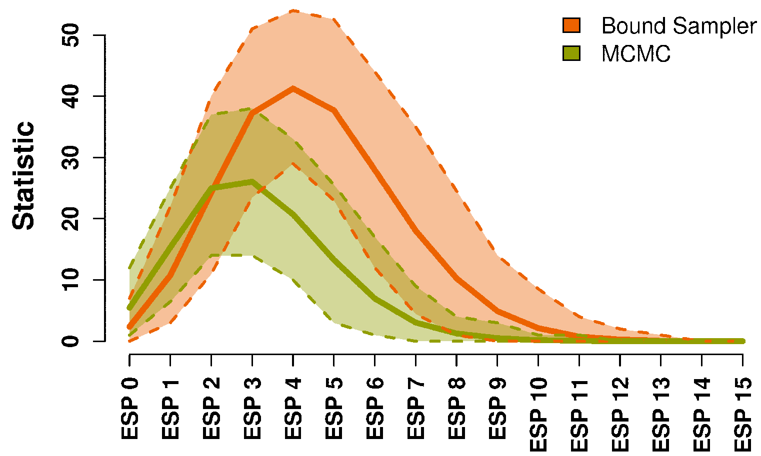


Example: Hunter and Handcock Lazega Model (Uniform Perturbations)

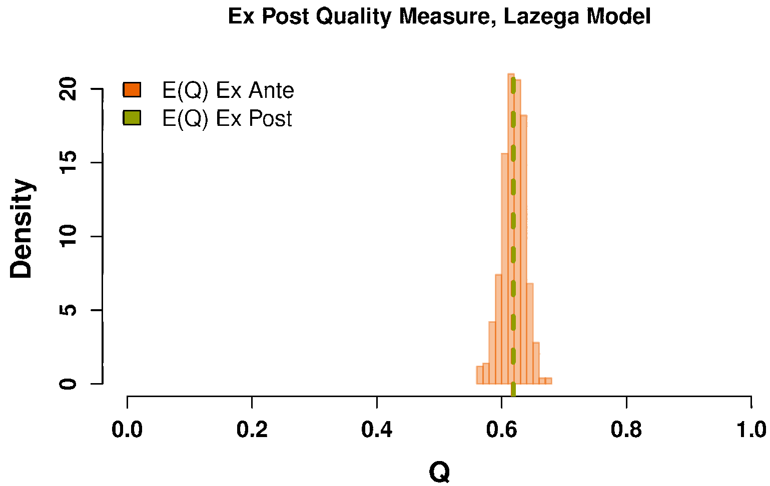


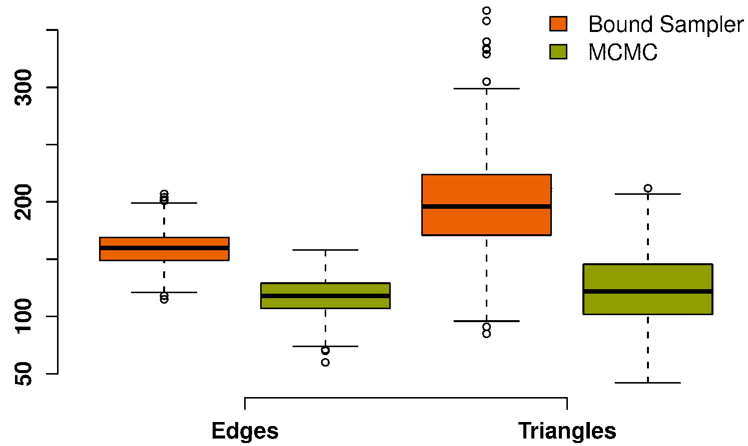


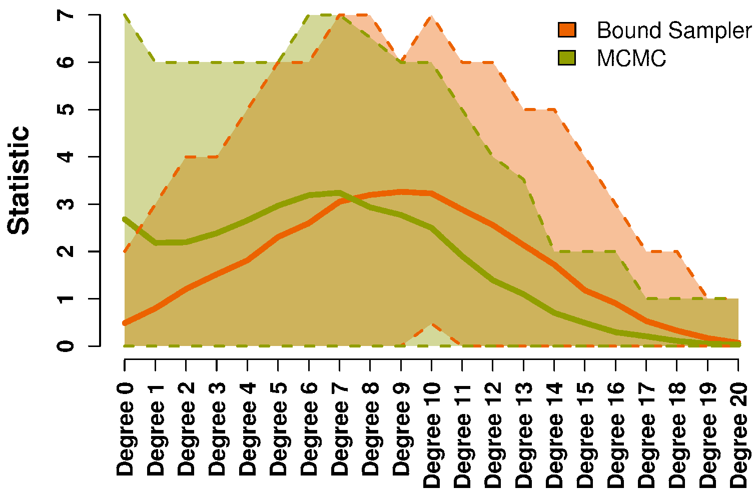


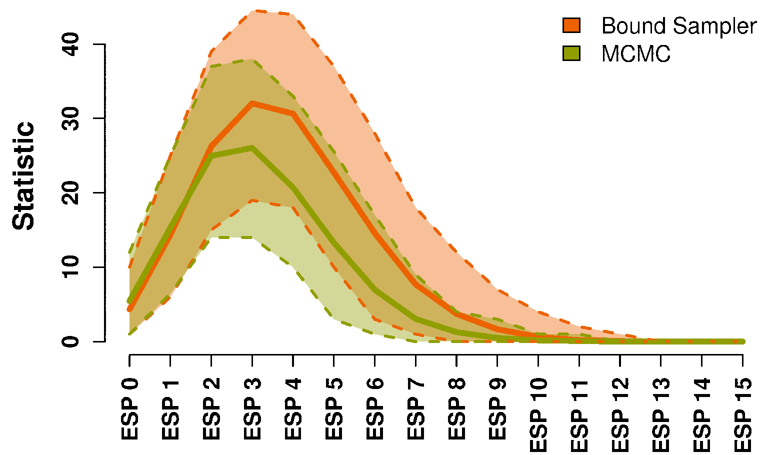


Example: Hunter and Handcock Lazega Model (Pseudo-marginal Perturbations)









Conclusion

- Bound sampler: an interesting direction for ERG simulation
  - Fixed execution time (can be worst-case optimal)
  - Ex ante and ex post quality guarantees
  - Unsupervised execution (no convergence checking)
- Ongoing issues
  - When is it good enough for particular applications?
  - Better perturbation tricks (the real key to success!)
  - Smart implementation for typical graph statistics (min/max tracking)

The End

Thanks for your attention!

Further Issues: Quality Assessment

- Using β to threshold R yields an exact draw from Y; the sampler approximates this using the (α, γ) bounds on β, plus perturbations (when bounds are insufficient)
- Let $y^t|r$ be the unique draw from Y corresponding to realization r of R; the "quality' of draw $y|r$ from the bound sampler is the similarity of y and y^t
- Simple measure: $Q(y) = (N - D_H(y, y^t))/N$, where D_H is the Hamming distance
  - Don't know $D_H(y, y^t)$, but can create an upper bound by treating all y_i not fixed by a perturbation-free cascade as incorrect; this give an ex post lower bound on Q
  - Computing unconditional values for (α_i, γ_i) leads to the ex ante lower bound on $\mathbf{E}Q(Y)$, $1/N \sum_{i=1}^{N} (\gamma_i - \alpha_i)$
- Can also answer more nuanced questions about y^t using y
  - Considering all y' consistent with the fixed values of y allows exact bounds on all properties of y^t (but perhaps loose ones!)
  - Given multiple draws, can bound expectations for properties of Y by expectations on upper/lower bounds from bound sampler draws
- Note: $Q(y)$ is *not* independent of y^t! Resampling until one gets a high-quality draw will introduce bias! (No free lunch, alas....) [Return]

Further Issues: Perturbation

- When we "perturb" the algorithm, we are really *estimating* the unknown β_i
  - The threshold, τ, is our "estimator"
- By default, select uniformly between bounds, but can do better
  - Clearly, $\mathbf{E}\beta = \sum_{i=1}^{N} Y_i/N$, so want to favor low/high τ when Y is sparse/dense
  - Taking an initial approximation to the pdf of β as a prior, can update based on (α, γ) and draw τ from the posterior
    - Diffuse beta distribution centered on expected density seems to be an easy improvement on the uniform
  - In theory, could use better approximations that take $y_{<i}$ into account; as $\tau \to \beta_i$, $y|r$ approaches its exact value
    - Have had good luck w/"pseudo-marginalization" method to approximate β_i by averaging full conditionals over a baseline Bernoulli model
- This is the principal route for improving simulation quality – clever ideas are welcome!

[Return]

Sampling Algorithm

for i in $1, \ldots, N$ **do**
Set $\alpha'_i := \min_{y' \in \mathcal{Y}_N : y'_{<i} = y_{<i}} \Pr\left(Y_i = 1 \,\middle|\, \theta, Y_i^c = {y'}_i^c\right)$
Set $\gamma'_i := \max_{y' \in \mathcal{Y}_N : y'_{<i} = y_{<i}} \Pr\left(Y_i = 1 \,\middle|\, \theta, Y_i^c = {y'}_i^c\right)$
Draw r_i from $U(0, 1)$
if $r_i < \alpha'_i$ **then**
Set $y_i = 1$
else if $r_i \geq \gamma'_i$ **then**
Set $y_i = 0$
else
Draw τ from $U(\alpha'_i, \beta'_i)$
if $r_i < \tau$ **then**
Set $y_i := 1$
else
Set $y_i := 0$
end if
end if
end for
return y

[Return]

1 References

Butts, C. T. (2010). Bernoulli graph bounds for general random graphs. Technical Report MBS 10-07, Irvine, CA.