

# Dissertation Defense: Numerical Methods for Approximating High-Dimensional Posterior Distributions

Jameson Quinn

12/9/19

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{\int_{\theta' \in \Theta} P(x|\theta')P(\theta')d\theta'}$$

Always good to start a stats talk with Bayes' Thm.

You've all seen this before.

As you know, the trickiest part is the denominator.

If integral is over hi-D space, naive numerical methods for estimating will have unacceptably high variance.

So you need tricks.

# Structure of thesis

- ▶ Chapter 1: online data assimilation in spatiotemporal systems
- ▶ Chapter 2: new method for variational inference on latent variable models
  - ▶ Contributions: Laplace guide families; analytic amortization
- ▶ Chapter 3: application to ecological inference (EI)
  - ▶ Contributions: Extensible model for EI; full algorithm and implementation of Laplace VI for this model

SAY VI XXXXXXXXXXXXXXXX

Ch. 2: The VI framework is to assume the posterior is well-approximated... construct a new guide family that's able to...

Ch 3: More than a simple application. The model is more realistic and more extensible than the most common method, and applying VI here requires several tricks.

You will notice a few changes to what I sent you, particularly in the chapter 3 results; I will point them out as we go along

## Collaborator on Ch. 2-3: Mira Bernstein

- ▶ On most things, equal collaborator and coauthor
- ▶ All the major motivating ideas, and  $>95\%$  of the coding, is mine
- ▶ We checked that this is OK

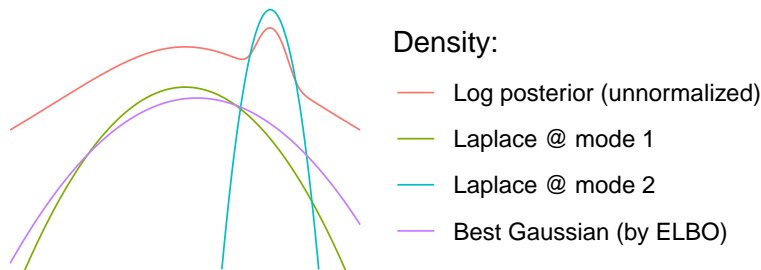
# Variational Inference

Approximate w/ guide distribution  $q_\phi(\boldsymbol{\theta})$ ; choose  $\phi$  to minimize KL:

$$\hat{\phi} = \operatorname{argmin}_{\phi} [D_{\text{KL}} (q_{\phi}(\boldsymbol{\theta}) \parallel p(\boldsymbol{\theta}|\mathbf{x}))].$$

Equivalent to maximizing ELBO:

$$\text{ELBO}(\phi) := E_{q_{\phi}} [\log p(\mathbf{x}, \boldsymbol{\theta}) - \log q_{\phi}(\boldsymbol{\theta})]$$



## \textcolor{red}{\scriptsize Re center, breathe. minimize}

# Computational tool: Pyro

Released in 2017 and still under very active development, pyro is a cutting-edge python package for black-box VI.

- ▶ Stochastic optimization (hill-climbing)
- ▶ Automatic differentiation via PyTorch ML

Explain automatic differentiation

All the significant software engineering I had to do

# Choosing a guide family

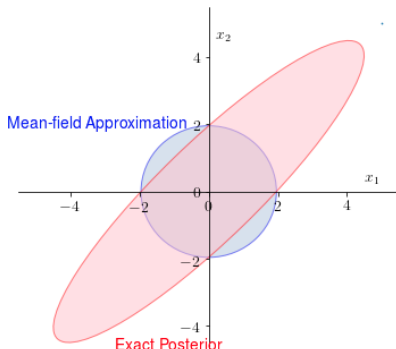
This talk will focus on Gaussian guide families.

The first obvious possibility for the guide family of a  $d$ -parameter model is just the unrestricted set of Gaussians.

- ▶ Mean:  $d$  guide parameters (1 per model parameter)
- ▶ Covariances:  $\mathcal{O}(d^2)$  guide parameters

can't capture posterior correlations

systematically underestimates posterior marginals



# Meanfield guide family

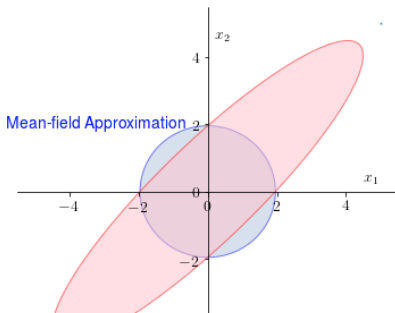
A common assumption is posterior independence of parameters, referred to as “meanfield” guides. Thus guide parameters:

- ▶ Mean:  $d$  guide parameters (1 per model parameter)
- ▶ Variances:  $d$  guide parameters (1 per model parameter; diagonal covariance matrix)

Problem:

can't capture posterior correlations

systematically underestimates posterior marginals





# Who will guide us?

Among Gaussian guide families:

- ▶ Set of all normals, with unrestricted covariance, is too big
- ▶ Meanfield subfamily doesn't actually contain any good approximations
- ▶ We want subfamily that contains at least some good approximations without being too big

# Introducing: Laplace family

Let's guarantee that the family contains the Laplace approximation around any posterior mode. This allows us to parametrize only the mean, and then derive the precision matrix by taking the observed information of the posterior:

$$\mathcal{I}_p(\theta^*) := -H[\log p(\theta)] \Big|_{\theta^*}$$

Thus, the guide parameters for a model  $p(\theta)$  would be  $\theta^*$ , defining the point at which to take a Laplace approximation.

- ▶ Means ( $\theta^*$ ):  $d$  guide parameters (1 per model parameter)
- ▶ Covariance: 0 guide parameters! Just compute  $\mathcal{I}_p(\theta^*)$ .

Don't ignore correlation. Don't optimize over it. Just get it by calculus.

# Boosting function

$\mathcal{I}_p$  not guaranteed to be positive definite. So define “boosting” function  $f(\mathcal{I}_p)$  s.t.:

- ▶ Guaranteed p.d.
- ▶ Smooth almost everywhere.
- ▶  $f(\mathcal{I}_p) \approx \mathcal{I}_p$  if  $\mathcal{I}_p$  already p.d.

A similar problem arises in optimization (quasi-Newton methods); solved via modified Cholesky algorithms (Surveyed in Fang, 2008; we use GMW81 by Gill, Murray, & Wright)

# Boosting family

XXXXX

Furthermore, we can parametrize  $f$  to create a boosting family  $f_{\psi}$ , for  $\psi_i \in \mathbb{R}_+^D$ , s.t. as  $\psi \rightarrow \vec{0}$ ,  $f(\mathcal{I}_p) \rightarrow \mathcal{I}_p$  if  $\mathcal{I}_p$  already p.d.

Boosting family is better than just boosting function.

D-dimensional so we can boost dif params dif.

Version of thesis sent previously has quasi-boosting which we're no longer using

# Formal definition of Laplace family

Let  $p(\boldsymbol{\theta})$  be a (twice-differentiable) probability density over  $\mathbb{R}^d$ .

Let  $\Theta \subseteq \mathbb{R}^d$ ,  $\Psi \subseteq \mathbb{R}_+^d$ , and let  $f_\psi$  be a boosting family.

Laplace guide:  $q_{\boldsymbol{\theta}^* \in \Theta, \psi \in \Psi}(\boldsymbol{\theta})$ , a  $d$ -dimensional Gaussian with mean  $\boldsymbol{\theta}^*$  and precision matrix  $f_\psi(\mathcal{I}_p(\boldsymbol{\theta}^*))$ .

Laplace guide family  $\mathcal{L}_{\Theta \times \Psi}(p, f_\Psi)$ :  $\{q_{\boldsymbol{\theta}^*, \psi} : \boldsymbol{\theta}^* \in \Theta, \psi \in \Psi\}$

Thus,  $2d$  guide parameters.

Note that capital greek letters can be subspaces

$\boldsymbol{\theta}^*$  tells mean;  $\psi$  tells how aggressively to boost

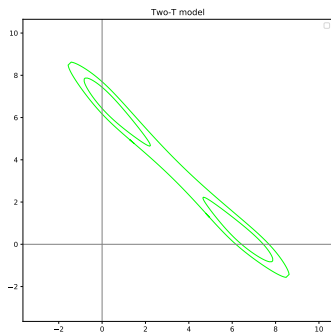
## Toy model

We want:  $p(T_1, T_2 | x = 7)$

$$x = T_1 + T_2 + \epsilon$$

$$T_i \sim \text{Student}T_\nu(0, 1); i \in \{1, 2\}$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

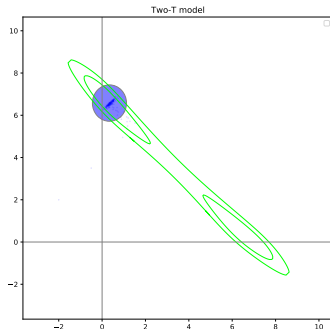


# Toy model

$$x = T_1 + T_2 + \epsilon$$

$$T_i \sim \text{Student}T_\nu(0, 1); i \in \{1, 2\}$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$



Simple model with bimodal posterior

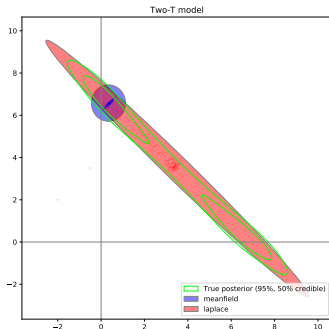
Shows several things: Importance of covariance; case where laplace of MAP isn't

# Toy model

$$x = T_1 + T_2 + \epsilon$$

$$T_i \sim \text{Student}T_\nu(0, 1); i \in \{1, 2\}$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$



Simple model with bimodal posterior

Shows several things: Importance of covariance; case where laplace of MAP isn't



# Latent variable models (or: why hi-D?)

A latent variable model has 3 core elements:

- ▶ Global parameters:  $\gamma \in \Gamma \cong \mathbb{R}^g$ ,
- ▶ Latent parameter vectors:  $\lambda_1, \dots, \lambda_N \in \Lambda \cong \mathbb{R}^l$
- ▶ Observation vectors:  $\mathbf{x}_1, \dots, \mathbf{x}_N$

$$p(\gamma, \lambda_1, \dots, \lambda_N, \mathbf{x}_1, \dots, \mathbf{x}_N) = p(\gamma) \prod_{i=1}^N p(\lambda_i | \gamma) p(\mathbf{x}_i | \lambda_i, \gamma)$$

Laplace guide parameters:  $\gamma^*, \lambda_1^* \dots \lambda_N^*, \psi$

conditional independence

This is why we need a high-dimensional solution.

# Latent variable models: Block Arrowhead Hessians

$$\mathcal{I}_p(\boldsymbol{\theta}^*) = \begin{pmatrix} & \gamma & \lambda_1 & \lambda_2 & \dots & \lambda_N \\ \gamma & G & C_1 & C_2 & \dots & C_N \\ \lambda_1 & C_1^T & U_1 & 0 & \dots & 0 \\ \lambda_1 & C_2^T & 0 & U_2 & \dots & 0 \\ \vdots & \vdots & 0 & 0 & \ddots & 0 \\ \lambda_N & C_N^T & 0 & 0 & \dots & U_N \end{pmatrix}$$

- ▶ Easy to boost.
- ▶ Easy to sample from. Note that marginal covariance for  $\gamma$  is  $[\mathcal{I}_p(\boldsymbol{\theta}^*)^{-1}]_{\Gamma, \Gamma} = (G - \sum_i C_i U_i^{-1} C_i^T)^{-1}$

easy to sample from and easy to boost.

# SVI (Stochastic Variational Inference)

Two methods useful w/ LVM. Conceptually independent, but combine. 1st, standard:

At each optimization step, let  $\mathcal{S}$  be a random sample of units with  $p(i \in \mathcal{S}) = \pi_i$ .

Replace

$$\log p(\boldsymbol{\theta}, \mathbf{x}) := \log p(\boldsymbol{\gamma}) + \sum_{i=1}^N \left[ \log p(\boldsymbol{\lambda}_i | \boldsymbol{\gamma}) + \log p(\mathbf{x}_i | \boldsymbol{\lambda}_i, \boldsymbol{\gamma}) \right]$$

with the unbiased estimator

$$\log p_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}}, \mathbf{x}_{\mathcal{S}}) := \log p(\boldsymbol{\gamma}) + \frac{1}{\pi_i} \sum_{i \in \mathcal{S}} \left[ \log p(\boldsymbol{\lambda}_i | \boldsymbol{\gamma}) + \log p(\mathbf{x}_i | \boldsymbol{\lambda}_i, \boldsymbol{\gamma}) \right]$$

Then compute Laplace guide for the latter expression, and find the ELBO gradient.

If cheap way to predict bigger terms, weights; we haven't implemented!

# SVI (Stochastic Variational Inference): unbiased?

With Laplace guide, this makes:

- ▶ Log density: unbiased
- ▶ Guide covariance of globals for given  $\theta^*$ : Up to boosting, unbiased for both conditional precision and “marginal precision” (inverse of marginal covariance).
- ▶ ELBO and ELBO gradient: Not unbiased (unlike meanfield)

ELBO & gradient not unbiased as in meanfield, but seem to work well

# Amortization

Suppose we can find an analytic function for the (approximate) conditional MAP:  $f(\gamma, \mathbf{x}_i) \approx \operatorname{argmax}_{\lambda_i} p(\lambda_i | \gamma, \mathbf{x}_i)$

We can use it to restrict to a lower-dimensional subfamily of  $\mathcal{L}$ :

- ▶ Restrict  $\Theta$ : reduce the number of guide parameters by setting  $\lambda_i^*$  to  $f(\gamma^*, \mathbf{x}_i)$
- ▶ Also restrict  $\Psi$ : reuse the same boosting parameters for each unit

Laplace guide parameters:  $\gamma^*, \psi_\gamma, \psi_\lambda$

Note that it's MAP not MLE.

MAP is not perfect but it's close.

computationally cheap refinement is available

## Multisite model

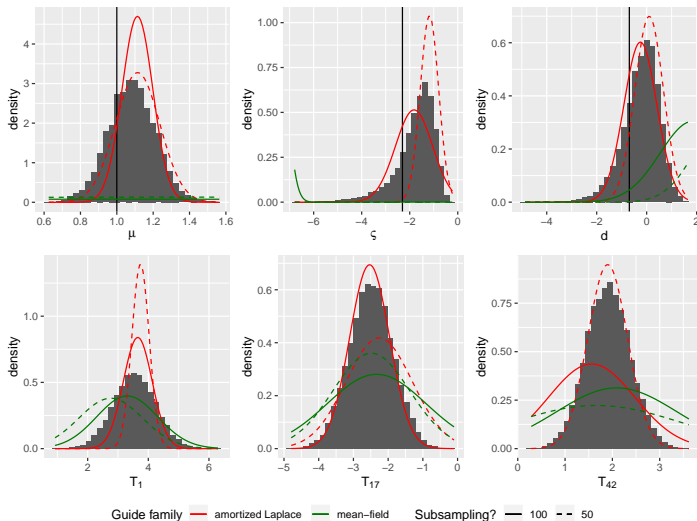
$$d := \log(\nu - \nu_{\min}) \sim \mathcal{N}(1, 1.5^2)$$

$$\varsigma := \log(\sigma - \sigma_{\min}) \sim \mathcal{N}(0, 2^2)$$

$$\mu \sim \mathcal{N}(0, 20)$$

$$\nu_{\min} = 2.5, \sigma_{\min} = \max(s_i) * 1.9$$

# Results



Changes: df; boosting (once for each diagonal block)

# Ecological inference (EI)

EI: inferring individual behavior from aggregated data.

Motivating example: voting behavior by racial or other groups

Race	Candidate			$n_u$
	X	Y	Z	
	White	?	?	400
	Black	?	?	200
	Hispanic	?	?	100
	Other	?	?	100
$v_u$				800

Mainly comes up in voting rights cases, so I'll talk about it in this setting.

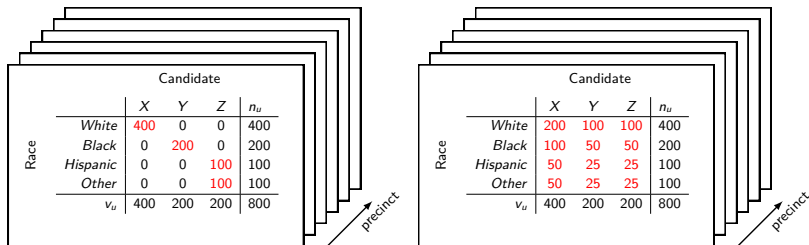
Z could represent not voting.



# Ecological inference (EI)

EI: inferring individual behavior from aggregated data.

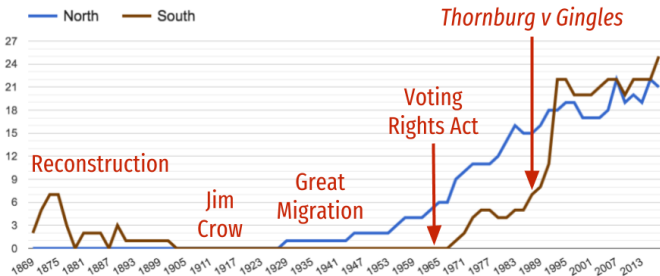
Motivating example: voting behavior by racial or other groups



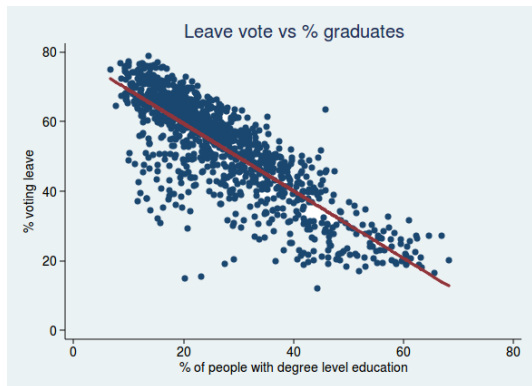
## Thornburg v. Gingles, 1986

When you can show racially polarized voting, a minority community is entitled to a majority-minority district. Result:

### Number of African-Americans in Congress



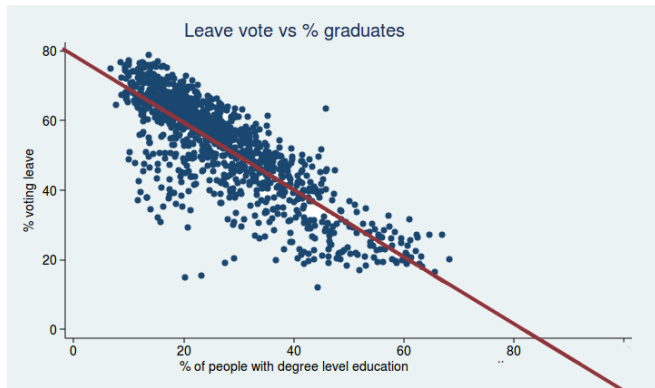
## First attempt: Ecological regression (ER)



Brexit voting data. (Example by Adam Jacobs.)

Want to know how Brexit support differed by education. So...

## First attempt: Ecological regression (ERrrrr...)



Brexit support: -16% of those with a degree???

Strong model assumption, which is incorrect: no precinct-level variation

# Comparison of models

ER  
(1953)

Global voting propensities  
(for each race)



Precinct vote totals  
by candidate (observed)

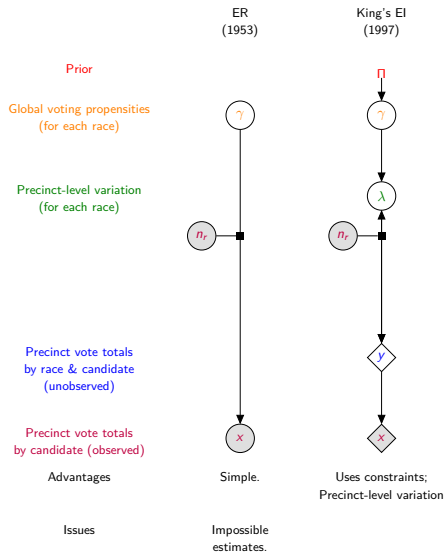
Advantages

Issues

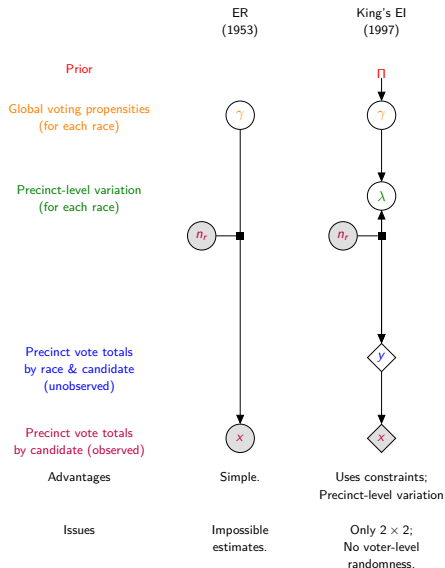
Simple.

Impossible  
estimates.

# Comparison of models

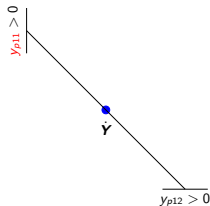


# Comparison of models



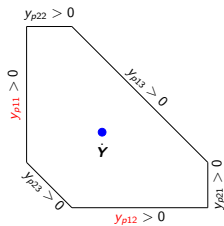
# Polytope

Race	Candidate			
	X	Y	$n_u$	
	White	50	50	100
	Black	70	70	140
	$v_u$	120	120	240



$y_u$

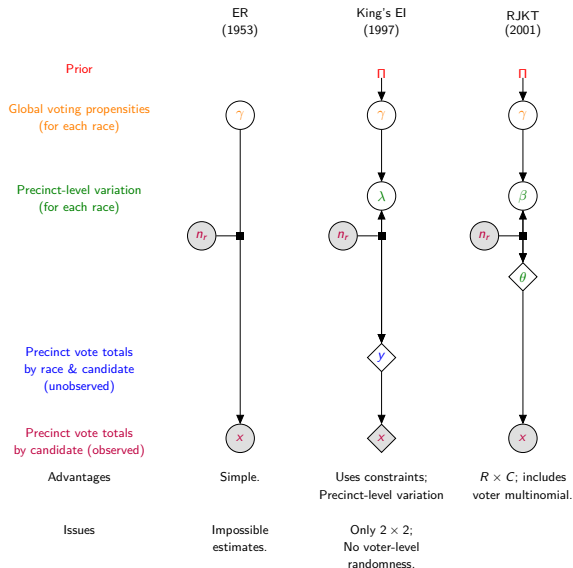
Race	Candidate				
	X	Y	Z	$n_u$	
	White	50	50	50	150
	Black	70	70	70	210
	$v_u$	120	120	120	360



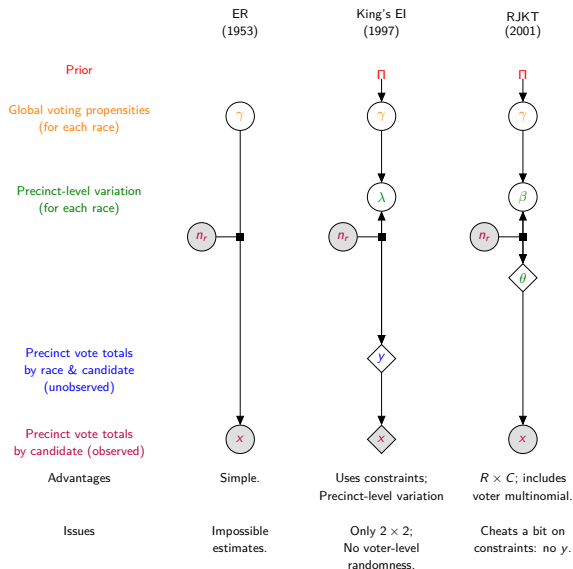
$y_u$



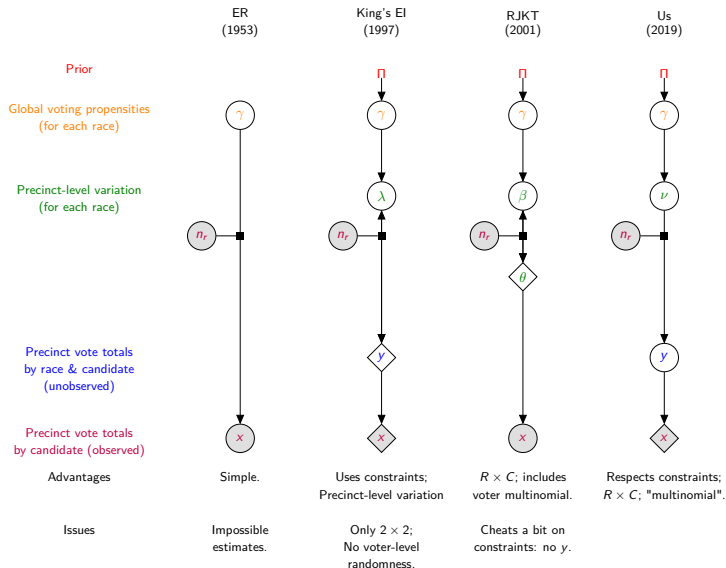
# Comparison of models



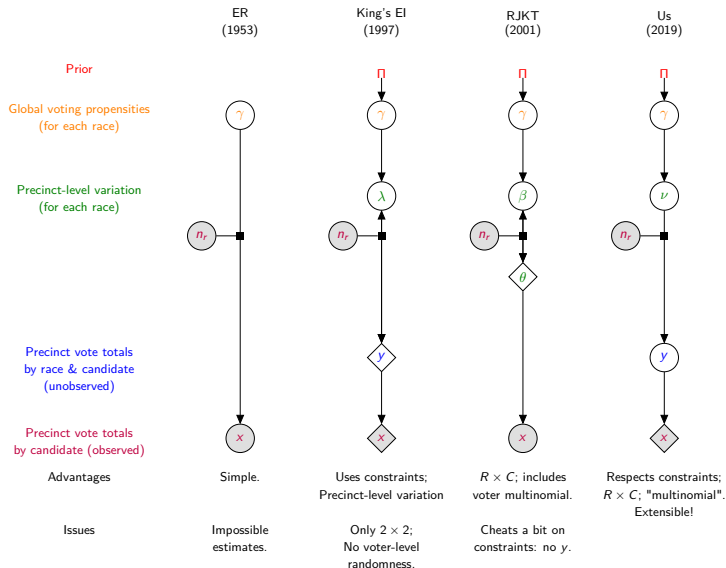
# Comparison of models



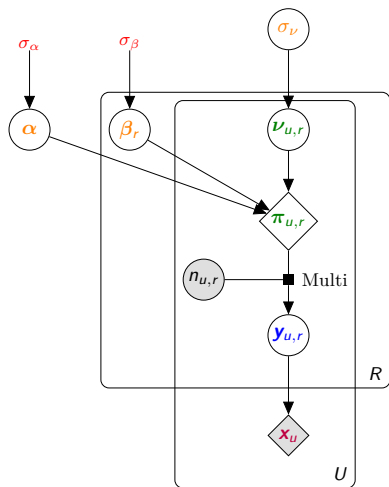
# Comparison of models



# Comparison of models



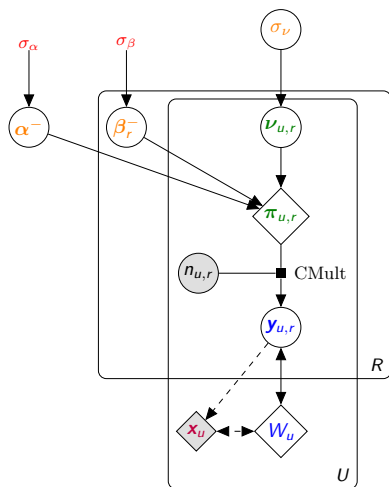
# Our model



Talk briefly about how you could add other Christmas tree ornaments

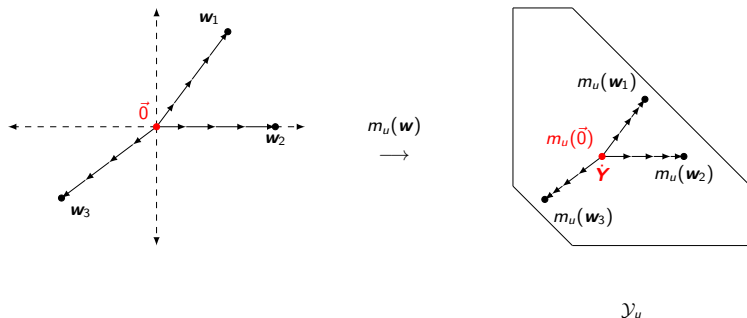
Talk about why this is hard to make a guide for

# Modified model



Cmult; Polytopize: ae smooth bijective map from  $R^n$  to polytope  
Pseudovoters because of boundary issues that arise from Cmult and polytopize  
All of these make the model itself slightly less-realistic, but make VI work.

# Polytopize



Let  $m_u(\mathbf{w}) := g(a(\mathbf{w}))$ , where  $a$  is affine projection,  $g$  is retraction

$b(M)$ : the intersection of ray  $\overrightarrow{\dot{Y}M}$  with boundary of  $\mathcal{Y}_u$

$$g(M) := \begin{cases} \dot{Y} & \text{if } M = \dot{Y}, \\ \dot{Y} + \exp\left(-\frac{|b(M) - \dot{Y}|}{|M - \dot{Y}|}\right) \cdot (b(M) - \dot{Y}) & \text{otherwise.} \end{cases}$$

► Note that  $m_u(\mathbf{0}) = \dot{Y}$ .

## Testing our EI on simulated data

- ▶ Data cleaning on racial breakdown by precinct in NC
- ▶ Used exit polls to calculate realistic  $\alpha$  and  $\beta$  for 2016 presidential election
- ▶  $3 \times 3$ :
  - ▶ Races: White/Black/Other
  - ▶ Candidates: Trump/Clinton/Other (where “other” includes both 3rd party and not voting)
- ▶ Simulated datasets from our model using three choices of  $\sigma_\nu$ : 0.02, 0.1, 0.3
- ▶ Ran LVI using pyro, RJKT using eiPack in R.
- ▶ Reporting  $\bar{Q}$  and  $s_Q$  for each race and candidate: the mean and s.d. of  $Q(z_i)$  where  $z_i$  is a sample from the fitted posterior and  $Q$  gives the percent of the given race who voted for the given candidate.



## El results

Results for  $\sigma_\nu = 0.02$

		Other/none		Clinton (D)		Trump (R)	
$\mathcal{A}$		$\bar{Q}$	$s_Q$	$\bar{Q}$	$s_Q$	$\bar{Q}$	$s_Q$
White	Truth	32.4%		22.6%		45.0%	
	RJKT	32.3%	0.059%	22.5%	0.069%	45.0%	0.057%
	LVI	32.3%	0.028%	22.7%	0.028%	45.0%	0.030%
Black	Truth	38.1%		56.6%		5.28%	
	RJKT	38.4%	0.32%	56.2%	0.19%	4.87%	0.19%
	LVI	37.9%	0.067%	56.3%	0.058%	5.79%	0.046%
Other	Truth	43.4%		32.7%		24.0%	
	RJKT	41.5%	0.65%	32.9%	0.41%	24.2%	0.72%
	LVI	44.4%	0.25%	32.4%	0.21%	23.3%	0.25%

Point out that this is different (better!) than what you originally sent, because:  
does not underestimate variance (fixed bug)  
corrected alphas and betas (so that overall percentages of people of each race voting for each candidate approximate the true 2016 data, as intended)  
improved amortization (optimize Y <U+2192> optimize W)

Conclusion: We are as good as RJKT, but we're just getting started

## El results

Results for  $\sigma_\nu = 0.3$

		Other/none		Clinton (D)		Trump (R)	
$\mathcal{A}$		$\bar{Q}$	$s_Q$	$\bar{Q}$	$s_Q$	$\bar{Q}$	$s_Q$
White	Truth	32.3%		22.7%		45.0%	
	RJKT	32.3%	0.080%	22.8%	0.11%	44.8%	0.14%
	LVI	33.2%	0.067%	23.5%	0.049%	43.0%	0.070%
Black	Truth	38.9%		55.6%		5.48%	
	RJKT	40.4%	0.39%	53.7%	0.38%	5.33%	0.20%
	LVI	36.3%	0.17%	54.6%	0.15%	9.21%	0.16%
Other	Truth	43.0%		32.7%		24.3%	
	RJKT	38.3%	1.1%	35.4%	0.52%	24.9%	1.1%
	LVI	45.0%	0.46%	33.6%	0.40%	21.7%	0.47%

Point out that this is different (better!) than what you originally sent, because:  
does not underestimate variance (fixed bug)  
corrected alphas and betas (so that overall percentages of people of each race voting for each candidate approximate the true 2016 data, as intended)  
did NOT improve amortization (optimize Y  $\rightarrow$  optimize W)

Conclusion: We are as good as RJKT, but we're just getting started

## Discussion/future work (Ch. 3)

- ▶ Including racial makeup of precinct as a covariate
- ▶ Hierarchical model (counties)
- ▶ Multiple elections
- ▶ Actual NC data
- ▶ Compare hierarchical model without EI, Standard RJKT, and our model
- ▶ Cross-validation

Say that this is the stuff we plan to include in final paper

## Discussion/future work (Ch. 2)

- ▶ More on subsampling:
  - ▶ General theory of how to assign weights to minimize variance of estimator in subsampling
  - ▶ Some theory to help choose sample size for SVI
- ▶ Investigate replacing normal with T in guide

(use Ch 3 as example)

Say that this will not be in current paper, which is basically done

# Thanks

Thank you!

# Directory of extra slides

- ▶ Prior work
- ▶ El amortization
- ▶ *Thornburg v. Gingles*
- ▶ Model (for discussing extensions)

## Non-meanfield prior work

- ▶ Copula VI (Han et al 2015): create arbitrary transformations, to allow “quasi-correlation matrices” for non-Gaussian families. The values for such matrices are unrestricted, though.
- ▶ Time-series (Zhang et al 2017): model-specific tricks.
- ▶ Hierarchical VI (Ranganath et al 2015): put a prior on the guide then marginalize it out. Relies on conjugacy.
- ▶ Variational Boosting (Miller et al 2016): Correlation structure: low-rank plus diagonal.
- ▶ Normalizing flows (Rezende et al 2015): Kinda like adding a step of MCMC after sampling from guide.
- ▶ “Laplace Variational Inference” (Wang & Blei, 2012): Use Laplace approximation to approve update step in a conjugate meanfield VI.
- ▶ Imaging (Zhang et al 2017): Use Laplace approximation around posterior mode for certain fixed-size subsets of parameters. No boosting.

# How our EI amortization works

Which variables are we amortizing:  $\mathbf{Y}$ ,  $\nu$ ,  $\sigma_{\nu}$

Steps:

- ▶ Find approximate mode of  $p(\mathbf{Y}|\alpha, \beta; \nu = 0)$  constrained to lie on polytope (this is linear algebra plus Stirling's approximation)
- ▶ Not yet done: One-dimensional Newton's method to find approximate mode of  $W$ . (Not the same thing, because there's Jacobian, mode of  $W$  is further away from boundary)
- ▶ Find approximate mode of  $p(\nu_u, \sigma_{\nu}|\gamma, \mathbf{W}_u)$  (ad hoc algorithm, but see next step)
- ▶ Newton's method (for free!!)

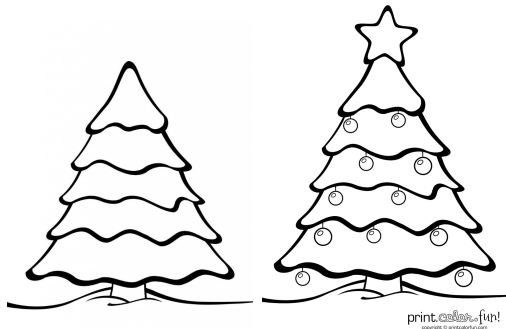


## *Thornburg v Gingles, 1986*

A majority-minority district must be created if:

1. A minority group is “sufficiently numerous and compact to form a majority in a single-member district”; and
2. The minority group is **"politically cohesive"**; and
3. The “majority **votes sufficiently as a bloc** to enable it . . . usually to defeat the minority’s preferred candidate.”

## Flexible model



$$\vec{y}_{p,r} = y_{p,r,c}|_{c=1}^C \sim \text{CMult} \left( n_{p,r}, \frac{\exp(\alpha_c + \beta_{r,c} + \nu_{r,c,p})|_{c=1}^C}{\sum_{c=1}^C \exp(\alpha_c + \beta_{r,c} + \nu_{r,c,p})} \right)$$

$$\alpha_c \sim \mathcal{N}(0, 2)$$

$$\beta_{r,c} \sim \mathcal{N}(0, 2)$$