University of California, Berkeley

Bayesian Bias Mitigation for Crowdsourcing

Fabian L. Wauthier, UC Berkeley

with Michael I. Jordan

9th of May, 2012

The Problem of Bias in Crowdsourcing

- Crowdsourcing: collect data from crowd and learn a model.

The Problem of Bias in Crowdsourcing

- Crowdsourcing: collect data from crowd and learn a model.
- E.g. Amazon Mechanical Turk

The Problem of Bias in Crowdsourcing

- Crowdsourcing: collect data from crowd and learn a model.
- E.g. Amazon Mechanical Turk
- Labelers may be malicious/unhelpful or tasks ambiguous/hard.

The Problem of Bias in Crowdsourcing

- Crowdsourcing: collect data from crowd and learn a model.
- E.g. Amazon Mechanical Turk
- Labelers may be malicious/unhelpful or tasks ambiguous/hard.
- $\Rightarrow$ Crowdsourced data is biased.
  - Problem is systemic. No easy fixes.
  - Effects on learned models can be significant.

The Problem of Bias in Crowdsourcing

- Crowdsourcing: collect data from crowd and learn a model.
- E.g. Amazon Mechanical Turk
- Labelers may be malicious/unhelpful or tasks ambiguous/hard.
- $\Rightarrow$ Crowdsourced data is biased.
  - Problem is systemic. No easy fixes.
  - Effects on learned models can be significant.

- **Problem: Can we still learn from partially biased data?**

Example: Scene Understanding

- "Robot: Get me the *brown guitar behind* the *couch*."

Example: Scene Understanding

- "Robot: Get me the *brown guitar behind* the *couch*."
- Human label data would be ambiguous:

Example: Scene Understanding

- "Robot: Get me the *brown guitar behind* the *couch*."
- Human label data would be ambiguous:
  - Is the guitar *brown* or *yellow*?

Example: Scene Understanding

- "Robot: Get me the *brown guitar behind* the *couch*."
- Human label data would be ambiguous:
  - Is the guitar *brown* or *yellow*?
  - Is it *behind* or *next* to the couch?

Example: Scene Understanding

- "Robot: Get me the *brown guitar behind* the *couch*."
- Human label data would be ambiguous:
  - Is the guitar *brown* or *yellow*?
  - Is it *behind* or *next* to the couch?
- There can be structural differences between labellers.

Example: Scene Understanding

- "Robot: Get me the *brown guitar behind* the *couch*."
- Human label data would be ambiguous:
  - Is the guitar *brown* or *yellow*?
  - Is it *behind* or *next* to the couch?
- There can be structural differences between labellers.
- How to learn from this data?

Current Methodologies

- Bias addressed in three stages of a pipeline:

Current Methodologies

- Bias addressed in three stages of a pipeline:
  1. Data collection: Active learning.
  2. Data curation: Screening/weighting of data.
  3. Learning: Noisy observation model.

Current Methodologies

- Bias addressed in three stages of a pipeline:
  1. Data collection: Active learning.
  2. Data curation: Screening/weighting of data.
  3. Learning: Noisy observation model.
- **Common Assumptions**:
  - There exists a *single* truth.
  - Can model bias *effects* as noise.

Current Methodologies

- Bias addressed in three stages of a pipeline:
  1. Data collection: Active learning.
  2. Data curation: Screening/weighting of data.
  3. Learning: Noisy observation model.
- **Common Assumptions**:
  - There exists a *single* truth.
  - Can model bias *effects* as noise.
- Inappropriate when tasks are subjective or particularly hard.

Overview

Contribution I: Bayesian Preference Model

BBMC Results

Contribution II: Approximate Active Learning

Active Learning Results

Conclusion

Overview

Contribution I: Bayesian Preference Model

BBMC Results

Contribution II: Approximate Active Learning

Active Learning Results

Conclusion

Contribution I: Bayesian Preference Model

- Unify pipeline steps in a Bayesian model.

Contribution I: Bayesian Preference Model

- Unify pipeline steps in a Bayesian model.
  - Model the *sources* of bias, not just *effects*.

Contribution I: Bayesian Preference Model

- Unify pipeline steps in a Bayesian model.
  - Model the *sources* of bias, not just *effects*.
  - Labelers express **accumulated**, **shared** preferences.

Contribution I: Bayesian Preference Model

- Unify pipeline steps in a Bayesian model.
  - Model the *sources* of bias, not just *effects*.
  - Labelers express **accumulated**, **shared** preferences.
- **Benefits:**
  - Allows multiple **inconsistent** labellings to coexist.

Contribution I: Bayesian Preference Model

- Unify pipeline steps in a Bayesian model.
  - Model the *sources* of bias, not just *effects*.
  - Labelers express **accumulated**, **shared** preferences.
- **Benefits:**
  - Allows multiple **inconsistent** labellings to coexist.
  - Active learning can be coherently integrated.

Contribution I: Bayesian Preference Model

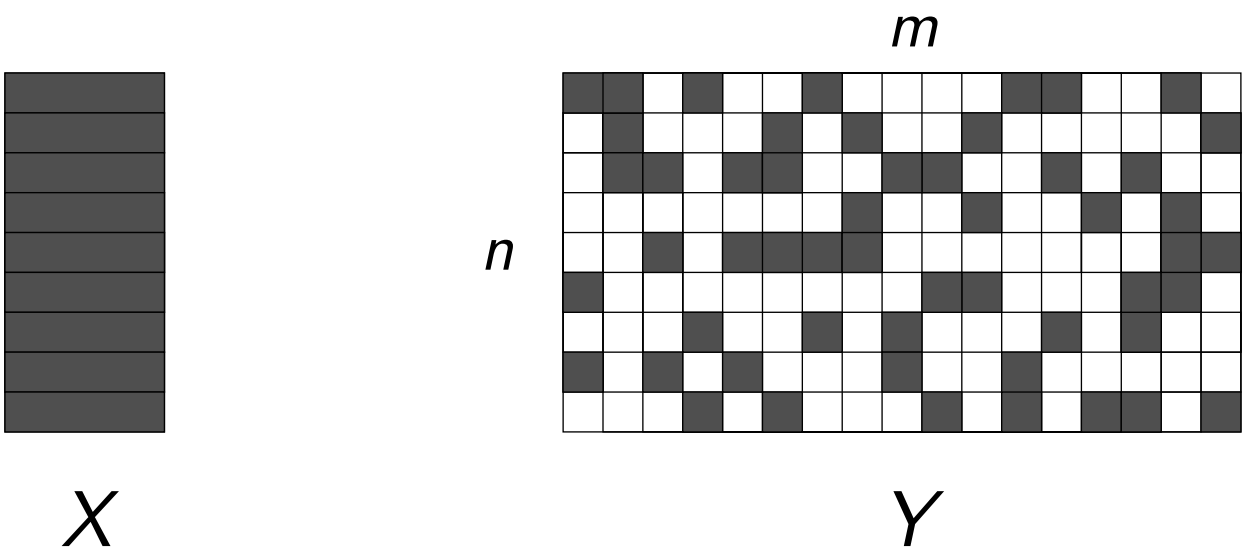
- Unify pipeline steps in a Bayesian model.
  - Model the *sources* of bias, not just *effects*.
  - Labelers express **accumulated**, **shared** preferences.
- **Benefits:**
  - Allows multiple **inconsistent** labellings to coexist.
  - Active learning can be coherently integrated.
  - Bayesian inference combines data curation and learning.

Input Data

- Tasks i, labelers l.
- Example task: "Is the guitar behind or next to the couch?"

Input Data

- Tasks i, labelers l.
- Example task: "Is the guitar behind or next to the couch?"
- Task covariates $x_i \in \mathbb{R}^d, i = 1, \ldots, n$ in X.
- Labels are $y_{i,l} \in \{-1, 0, +1\}, i = 1, \ldots, n; l = 1, \ldots, m$ in Y.



Bayesian Preference Model

Labelers express **accumulated, shared** preferences.

Bayesian Preference Model

Labelers express **accumulated, shared** preferences.

- Parameter γ_b models *effect* of preference $b = 1, \ldots, K$.

Bayesian Preference Model

Labelers express **accumulated, shared** preferences.

- Parameter γ_b models *effect* of preference $b = 1, \ldots, K$.
- $m \times K$ binary matrix Z models parameter sharing.

Bayesian Preference Model

Labelers express **accumulated, shared** preferences.

- Parameter γ_b models *effect* of preference $b = 1, \ldots, K$.
- $m \times K$ binary matrix Z models parameter sharing.
- If $z_{l,b} = 1$, labeler l expresses preference b.

Bayesian Preference Model

Labelers express **accumulated, shared** preferences.

- Parameter γ_b models *effect* of preference $b = 1, \ldots, K$.
- $m \times K$ binary matrix Z models parameter sharing.
- If $z_{l,b} = 1$, labeler l expresses preference b.
- Parameter β_l accumulates preferences:

$$\beta_l = \sum_b z_{l,b} \gamma_b$$

Bayesian Preference Model

Labelers express **accumulated, shared** preferences.

- Parameter γ_b models *effect* of preference $b = 1, \ldots, K$.
- $m \times K$ binary matrix Z models parameter sharing.
- If $z_{l,b} = 1$, labeler l expresses preference b.
- Parameter β_l accumulates preferences:

$$\beta_l = \sum_b z_{l,b} \gamma_b$$

- **Likelihood:**

$$p(Y|X, Z, \gamma) = \prod_l \prod_{i:y_{i,l} \neq 0} p(y_{i,l}|\beta_l^\top x_i)$$

Bayesian Preference Model

Labelers express **accumulated, shared** preferences.

- Parameter γ_b models *effect* of preference $b = 1, \ldots, K$.
- $m \times K$ binary matrix Z models parameter sharing.
- If $z_{l,b} = 1$, labeler l expresses preference b.
- Parameter β_l accumulates preferences:

$$\beta_l = \sum_b z_{l,b} \gamma_b$$

- **Likelihood:**

$$p(Y|X, Z, \gamma) = \prod_l \prod_{i:y_{i,l} \neq 0} p(y_{i,l}|\beta_l^\top x_i)$$

- Similar preferences $\Rightarrow$ similar labelling.

Priors

- **Prior on γ_b:** $p(\gamma_b) = \mathcal{N}(0, \sigma^2 I)$ for each b.

Priors

- **Prior on γ_b:** $p(\gamma_b) = \mathcal{N}(0, \sigma^2 I)$ for each b.
- **Prior on Z:** fix Z to be $m \times K$.

Priors

- **Prior on γ_b:** $p(\gamma_b) = \mathcal{N}(0, \sigma^2 I)$ for each b.
- **Prior on Z:** fix Z to be $m \times K$.

$$\pi_b | \alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right), b = 1, \ldots, K \tag{1}$$

$$\tag{2}$$

Priors

- **Prior on γ_b:** $p(\gamma_b) = \mathcal{N}(0, \sigma^2 I)$ for each b.
- **Prior on Z:** fix Z to be $m \times K$.

$$\pi_b | \alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right), b = 1, \ldots, K \tag{1}$$

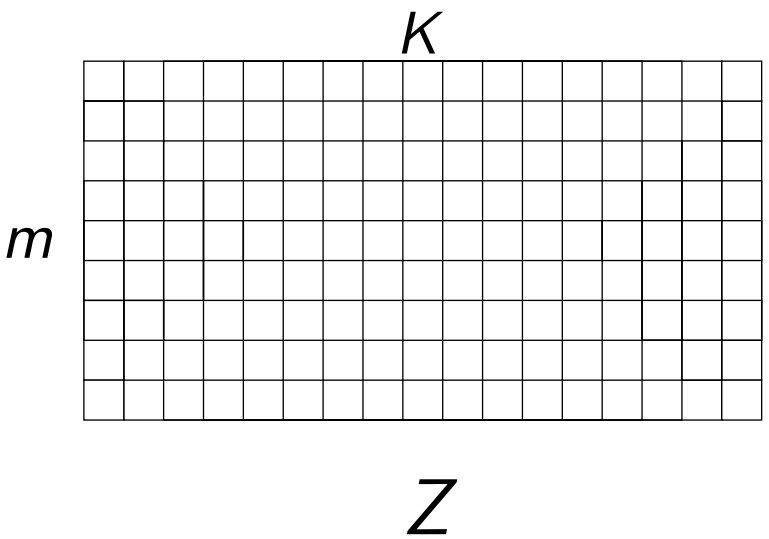
$$z_{l,b} | \pi_b \sim \text{Bern}\left(\pi_b\right), l = 1, \ldots, m \tag{2}$$

Priors

- **Prior on γ_b:** $p(\gamma_b) = \mathcal{N}(0, \sigma^2 I)$ for each b.
- **Prior on Z:** fix Z to be $m \times K$.

$$\pi_b | \alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right), b = 1, \ldots, K \tag{1}$$

$$z_{l,b} | \pi_b \sim \text{Bern}\left(\pi_b\right), l = 1, \ldots, m \tag{2}$$

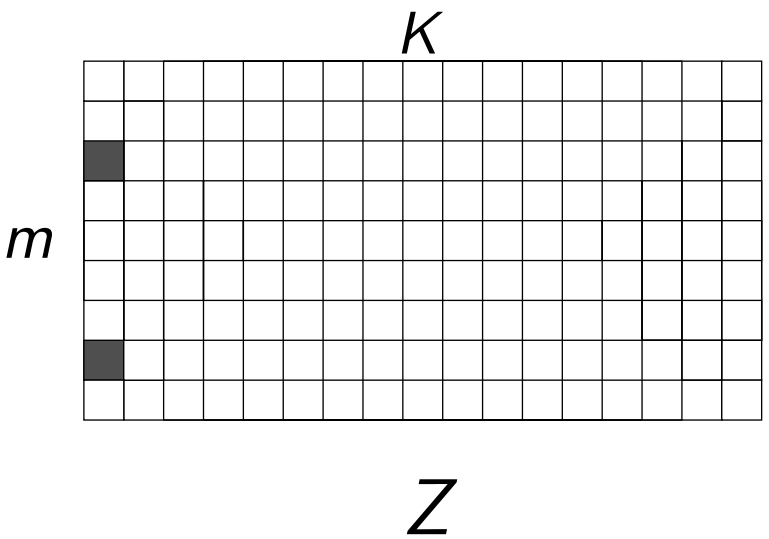


Priors

- **Prior on γ_b:** $p(\gamma_b) = \mathcal{N}(0, \sigma^2 I)$ for each b.
- **Prior on Z:** fix Z to be $m \times K$.

$$\pi_b | \alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right), b = 1, \ldots, K \tag{1}$$

$$z_{l,b} | \pi_b \sim \text{Bern}\left(\pi_b\right), l = 1, \ldots, m \tag{2}$$

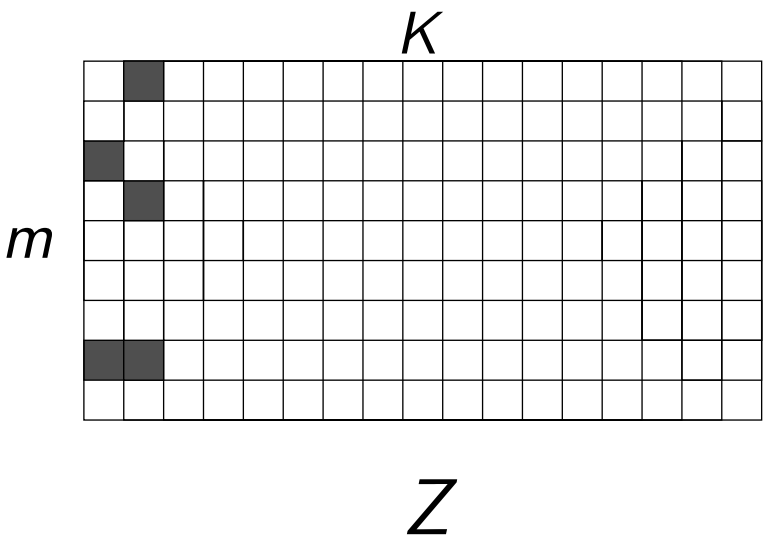


Priors

- **Prior on γ_b:** $p(\gamma_b) = \mathcal{N}(0, \sigma^2 I)$ for each b.
- **Prior on Z:** fix Z to be $m \times K$.

$$\pi_b | \alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right), b = 1, \ldots, K \tag{1}$$

$$z_{l,b} | \pi_b \sim \text{Bern}\left(\pi_b\right), l = 1, \ldots, m \tag{2}$$

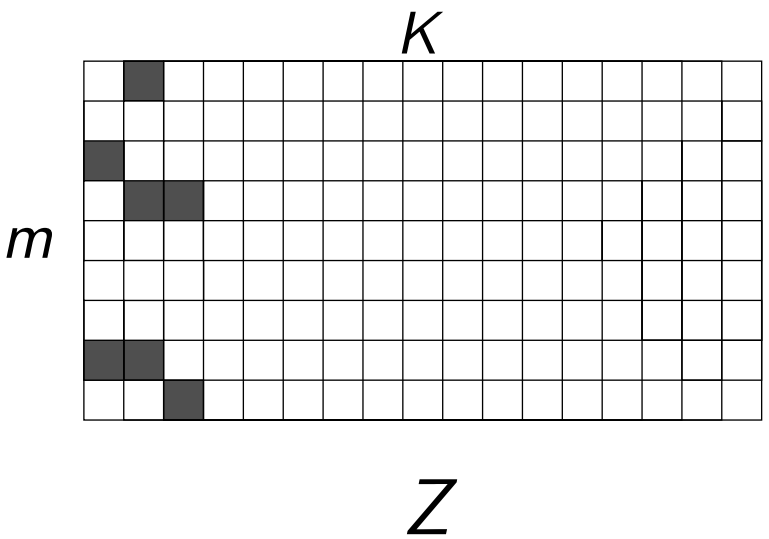


Priors

- **Prior on γ_b:** $p(\gamma_b) = \mathcal{N}(0, \sigma^2 I)$ for each b.
- **Prior on Z:** fix Z to be $m \times K$.

$$\pi_b | \alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right), b = 1, \ldots, K \tag{1}$$

$$z_{l,b} | \pi_b \sim \text{Bern}\left(\pi_b\right), l = 1, \ldots, m \tag{2}$$

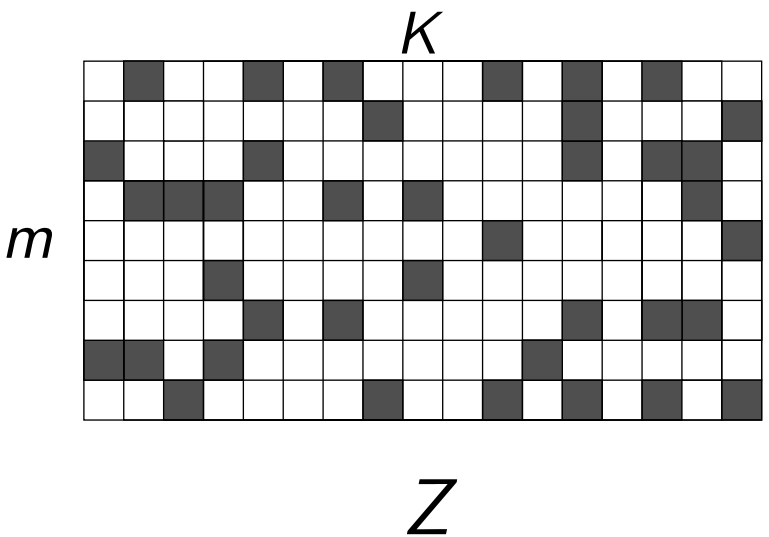


Priors

- **Prior on γ_b:** $p(\gamma_b) = \mathcal{N}(0, \sigma^2 I)$ for each b.
- **Prior on Z:** fix Z to be $m \times K$.

$$\pi_b|\alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right), b = 1, \ldots, K \tag{1}$$

$$z_{l,b}|\pi_b \sim \text{Bern}\left(\pi_b\right), l = 1, \ldots, m \tag{2}$$

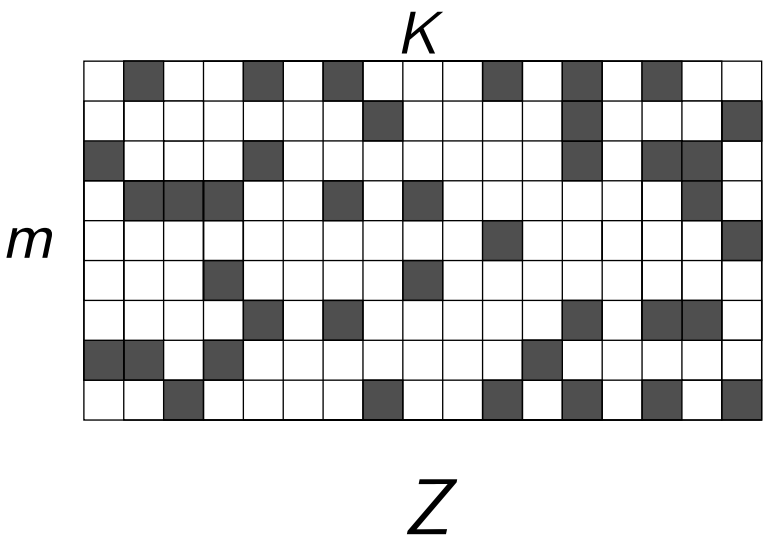


Priors

- **Prior on γ_b:** $p(\gamma_b) = \mathcal{N}(0, \sigma^2 I)$ for each b.
- **Prior on Z:** fix Z to be $m \times K$.

$$\pi_b | \alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right), b = 1, \ldots, K \tag{1}$$

$$z_{l,b} | \pi_b \sim \text{Bern}\left(\pi_b\right), l = 1, \ldots, m \tag{2}$$



- As $K \to \infty$, distribution over Z converges to the *Indian Buffet Process (IBP)*.

Complete model

$$p(Y, Z, \gamma|X) = p(Y|X, Z, \gamma)p(\gamma|Z)p(Z)$$

Complete model

$$p(Y, Z, \gamma|X) = p(Y|X, Z, \gamma)p(\gamma|Z)p(Z)$$

- Recall bias: different labellers can have different β's
  Example: Disagreement if guitar is behind/next to the couch.

Complete model

$$p(Y, Z, \gamma|X) = p(Y|X, Z, \gamma)p(\gamma|Z)p(Z)$$

- Recall bias: different labellers can have different β's
  Example: Disagreement if guitar is behind/next to the couch.
- Want to predict labeller l's labels.

Complete model

$$p(Y, Z, \gamma|X) = p(Y|X, Z, \gamma)p(\gamma|Z)p(Z)$$

- Recall bias: different labellers can have different β's
  Example: Disagreement if guitar is behind/next to the couch.
- Want to predict labeller l's labels.
- Labeller l could be in the crowd, or the *gold standard*.

Complete model

$$p(Y, Z, \gamma|X) = p(Y|X, Z, \gamma)p(\gamma|Z)p(Z)$$

- Recall bias: different labellers can have different β's
  Example: Disagreement if guitar is behind/next to the couch.
- Want to predict labeller l's labels.
- Labeller l could be in the crowd, or the *gold standard*.
- Required inference: $p(\beta_l|X, Y)$, or equivalently $p(z_{l,b}, \gamma_b, b = 1, \ldots, K|X, Y)$.

Complete model

$$p(Y, Z, \gamma|X) = p(Y|X, Z, \gamma)p(\gamma|Z)p(Z)$$

- Recall bias: different labellers can have different β's
  Example: Disagreement if guitar is behind/next to the couch.
- Want to predict labeller l's labels.
- Labeller l could be in the crowd, or the *gold standard*.
- Required inference: $p(\beta_l|X, Y)$, or equivalently $p(z_{l,b}, \gamma_b, b = 1, \ldots, K|X, Y)$.
- Model is complex. Exact inference intractable.

Complete model

$$p(Y, Z, \gamma|X) = p(Y|X, Z, \gamma)p(\gamma|Z)p(Z)$$

- Recall bias: different labellers can have different β's
  Example: Disagreement if guitar is behind/next to the couch.
- Want to predict labeller l's labels.
- Labeller l could be in the crowd, or the *gold standard*.
- Required inference: $p(\beta_l|X, Y)$, or equivalently $p(z_{l,b}, \gamma_b, b = 1, \ldots, K|X, Y)$.
- Model is complex. Exact inference intractable.
- Possible alternatives: *Gibbs sampling*, *variational inference*, *slice sampling*, etc.

Overview

Contribution I: Bayesian Preference Model

BBMC Results

Contribution II: Approximate Active Learning

Active Learning Results

Conclusion

Results: Synthetic Data

- X is 2000×4 Gaussian matrix

Results: Synthetic Data

- X is 2000×4 Gaussian matrix
- Z is 30×2 uniform binary matrix ($m = 30, K = 2$).

Results: Synthetic Data

- X is 2000×4 Gaussian matrix
- Z is 30×2 uniform binary matrix ($m = 30, K = 2$).
- γ_b Gaussian $b = 1, 2$. $\beta_l = \sum_b z_{l,b}\gamma_b$.

Results: Synthetic Data

- X is 2000×4 Gaussian matrix
- Z is 30×2 uniform binary matrix ($m = 30, K = 2$).
- γ_b Gaussian $b = 1, 2$. $\beta_l = \sum_b z_{l,b}\gamma_b$.
- Observation probability $\epsilon = 0.1$.

$$y_{i,l} = \begin{cases} 0 & \text{w.p. } (1-\epsilon) \\ +1 & \text{w.p. } \epsilon\Phi(x_i^\top \beta_l) \\ -1 & \text{o.w.} \end{cases}$$

Results: Synthetic Data

- X is 2000×4 Gaussian matrix
- Z is 30×2 uniform binary matrix ($m = 30, K = 2$).
- γ_b Gaussian $b = 1, 2$. $\beta_l = \sum_b z_{l,b}\gamma_b$.
- Observation probability $\epsilon = 0.1$.

$$y_{i,l} = \begin{cases} 0 & \text{w.p. } (1-\epsilon) \\ +1 & \text{w.p. } \epsilon\Phi(x_i^\top \beta_l) \\ -1 & \text{o.w.} \end{cases}$$

- Inference: want to recover β_1 (say).

Results: Synthetic Data

- X is 2000×4 Gaussian matrix
- Z is 30×2 uniform binary matrix ($m = 30, K = 2$).
- γ_b Gaussian $b = 1, 2$. $\beta_l = \sum_b z_{l,b}\gamma_b$.
- Observation probability $\epsilon = 0.1$.

$$y_{i,l} = \begin{cases} 0 & \text{w.p. } (1-\epsilon) \\ +1 & \text{w.p. } \epsilon\Phi(x_i^\top \beta_l) \\ -1 & \text{o.w.} \end{cases}$$

- Inference: want to recover β_1 (say).
- Requires $p(z_{1,b}, \gamma_b, b = 1 \ldots, K|, X, Y)$.

Results: Synthetic Data

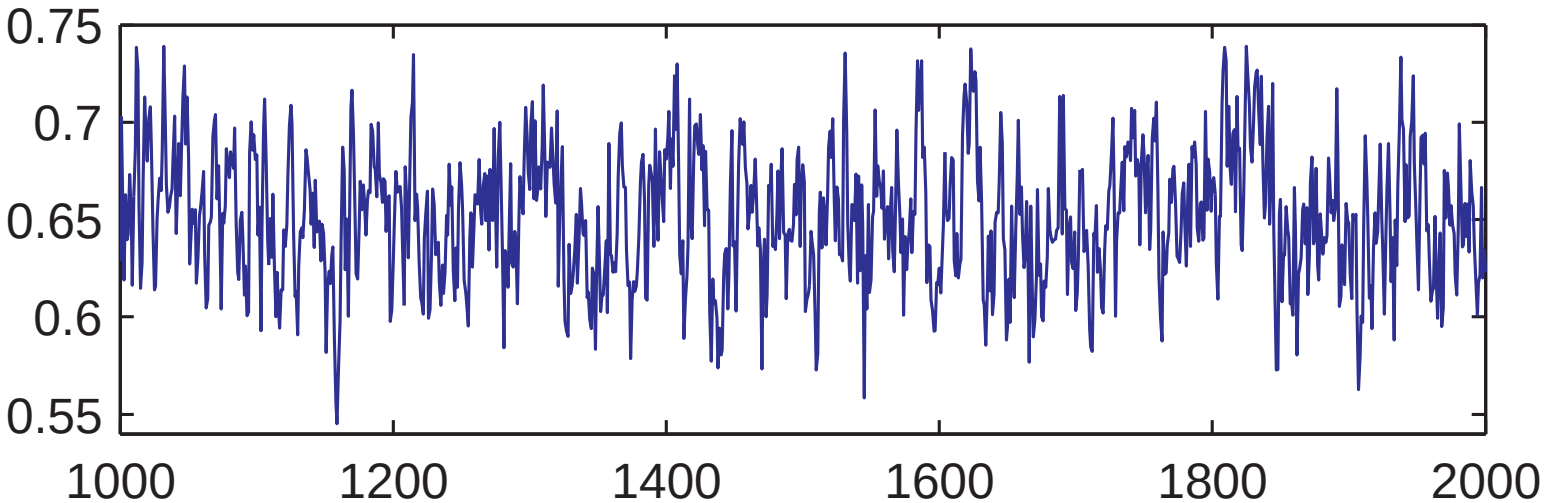
- X is 2000×4 Gaussian matrix
- Z is 30×2 uniform binary matrix ($m = 30, K = 2$).
- γ_b Gaussian $b = 1, 2$. $\beta_l = \sum_b z_{l,b}\gamma_b$.
- Observation probability $\epsilon = 0.1$.

$$y_{i,l} = \begin{cases} 0 & \text{w.p. } (1-\epsilon) \\ +1 & \text{w.p. } \epsilon\Phi(x_i^\top \beta_l) \\ -1 & \text{o.w.} \end{cases}$$

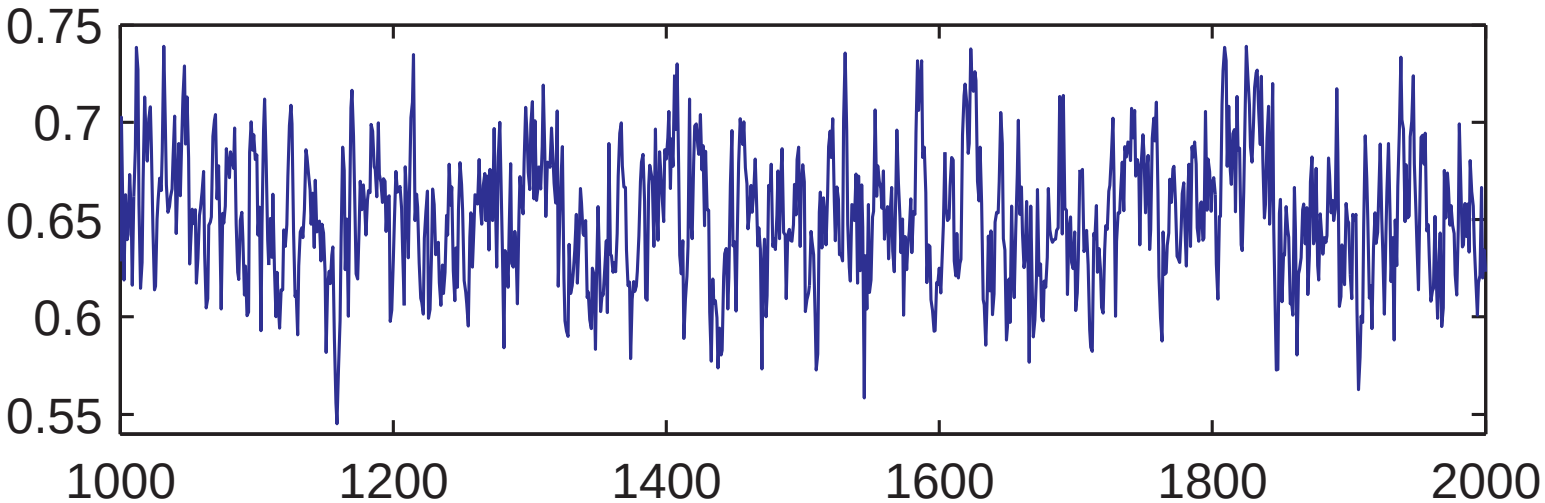
- Inference: want to recover β_1 (say).
- Requires $p(z_{1,b}, \gamma_b, b = 1 \ldots, K|, X, Y)$.
- For inference set $K = 10$.

- Latent Z mostly correct after 1000 Gibbs steps.

- Latent Z mostly correct after 1000 Gibbs steps.
- Gibbs sequence for $\gamma_{1,1}$.



- Latent Z mostly correct after 1000 Gibbs steps.
- Gibbs sequence for $\gamma_{1,1}$.

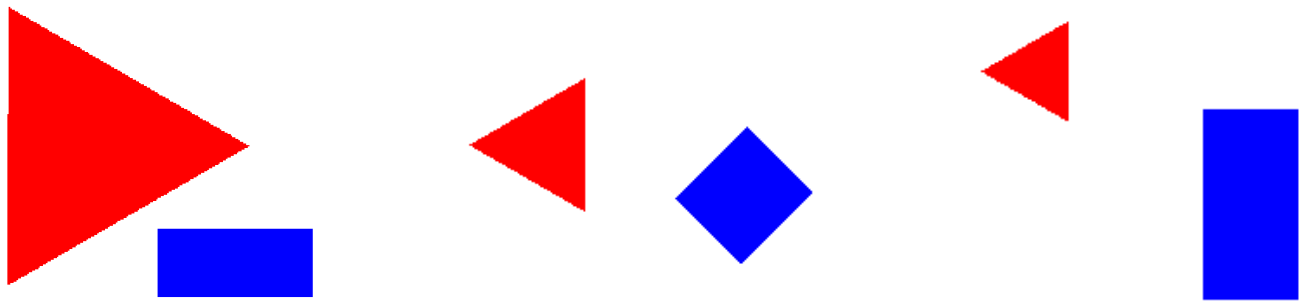


- True and posterior mean of β_1 after 1000 iterations burnin.

$$\beta_1 = \begin{bmatrix} 0.6915 \\ 0.0754 \\ -0.6815 \\ 0.6988 \end{bmatrix} \qquad \hat{\beta}_1 = \begin{bmatrix} 0.6514 \\ 0.0535 \\ -0.6473 \\ 0.6957 \end{bmatrix} \tag{3}$$

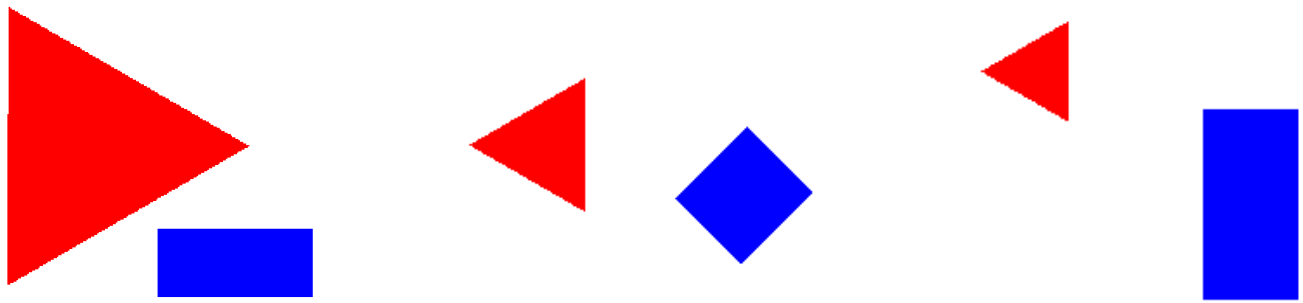
Results: Crowdsourced data

- **Task:** Is the triangle to the left or above the rectangle



Results: Crowdsourced data

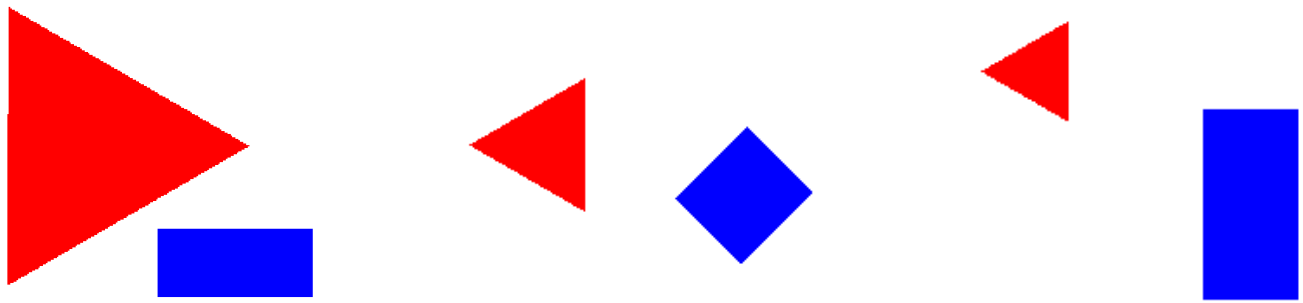
- **Task:** Is the triangle to the left or above the rectangle



- Labelled on Amazon Mechanical Turk: 523 tasks, 3 labels per task, 76 labellers.

Results: Crowdsourced data

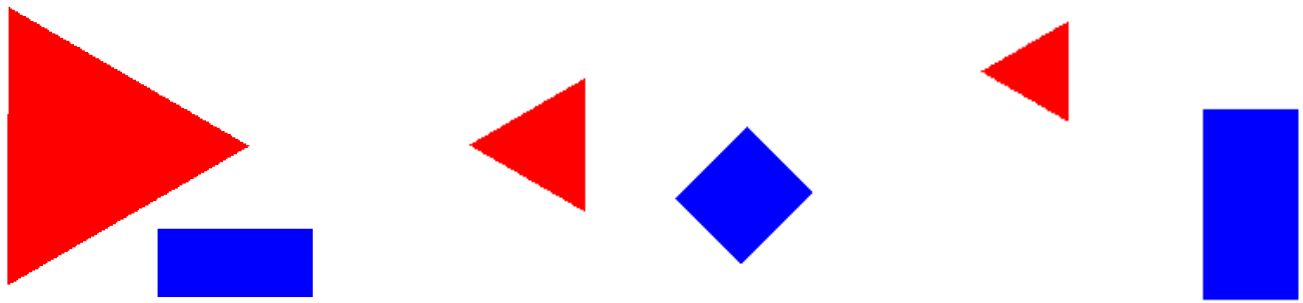
- **Task:** Is the triangle to the left or above the rectangle



- Labelled on Amazon Mechanical Turk: 523 tasks, 3 labels per task, 76 labellers.
- Want to predict gold standard: compare centroid positions.

Results: Crowdsourced data

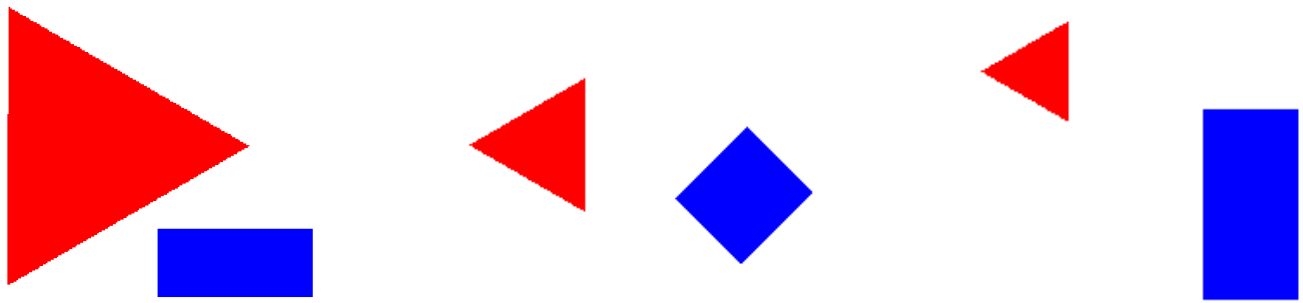
- **Task:** Is the triangle to the left or above the rectangle



- Labelled on Amazon Mechanical Turk: 523 tasks, 3 labels per task, 76 labellers.
- Want to predict gold standard: compare centroid positions.
- All 26 labellers with over 20 labels have error above 0.16.

Results: Crowdsourced data

- **Task:** Is the triangle to the left or above the rectangle



- Labelled on Amazon Mechanical Turk: 523 tasks, 3 labels per task, 76 labellers.
- Want to predict gold standard: compare centroid positions.
- All 26 labellers with over 20 labels have error above 0.16.
- Researcher also labels, and gives 60 gold standard labels.

- Averaged log likelihood and error rate on test set.
- Our model: BBMC.

	Algorithm	Final Loglik	Final Error
No Active Learning	GOLD	-3716 ± 1695	0.0547 ± 0.0102
	CONS	-421.1 ± 2.6	0.0935 ± 0.0031
	BBMC	$\mathbf{-219.1 \pm 3.1}$	$\mathbf{0.0309 \pm 0.0033}$

Overview

Contribution I: Bayesian Preference Model

BBMC Results

Contribution II: Approximate Active Learning

Active Learning Results

Conclusion

Active Learning

- Want to predict labeller l's labels. Need β_l.

Active Learning

- Want to predict labeller l's labels. Need β_l.
- Not all labellers are useful to infer β_l.

Active Learning

- Want to predict labeller l's labels. Need β_l.
- Not all labellers are useful to infer β_l.
- If l and l' share parameters $\Rightarrow$ can learn about β_l from l'.

$$\beta_l = \sum_b z_{l,b}\gamma_b \qquad \beta_{l'} = \sum_b z_{l',b}\gamma_b \tag{4}$$

Active Learning

- Want to predict labeller l's labels. Need β_l.
- Not all labellers are useful to infer β_l.
- If l and l' share parameters $\Rightarrow$ can learn about β_l from l'.

$$\beta_l = \sum_b z_{l,b}\gamma_b \qquad \beta_{l'} = \sum_b z_{l',b}\gamma_b \tag{4}$$

- Active learning: *repeatedly* select training data that helps learning β_l.

Active Learning

- Want to predict labeller l's labels. Need β_l.
- Not all labellers are useful to infer β_l.
- If l and l' share parameters $\Rightarrow$ can learn about β_l from l'.

$$\beta_l = \sum_b z_{l,b}\gamma_b \qquad \beta_{l'} = \sum_b z_{l',b}\gamma_b \tag{4}$$

- Active learning: *repeatedly* select training data that helps learning β_l.
- Goal: cheaper training data, faster learning.

Approximate inference and Active Learning

- Suppose we start with training data Y.

Approximate inference and Active Learning

- Suppose we start with training data Y.
- Query task-labeler pair (i, l) to maximize expected utility of adding it

$$(i, l) = \text{argmax}_{(i',l')} E_{y_{i',l'}} \left(U\left(p(\beta | y_{i',l'}, X, Y) \right) \right)$$

Approximate inference and Active Learning

- Suppose we start with training data Y.
- Query task-labeler pair (i, l) to maximize expected utility of adding it

$$(i, l) = \text{argmax}_{(i',l')} E_{y_{i',l'}} \left(U \left(p(\beta | y_{i',l'}, X, Y) \right) \right)$$

- Examples: $U(\cdot) = -\text{Entropy}(\cdot)$. $U_\mu(\cdot) = \|\text{Mean}(\cdot) - \mu\|_2^2$

Approximate inference and Active Learning

- Suppose we start with training data Y.
- Query task-labeler pair (i, l) to maximize expected utility of adding it

$$(i, l) = \text{argmax}_{(i',l')} E_{y_{i',l'}} \left(U \left(p(\beta | y_{i',l'}, X, Y) \right) \right)$$

- Examples: $U(\cdot) = -\text{Entropy}(\cdot)$. $U_\mu(\cdot) = \|\text{Mean}(\cdot) - \mu\|_2^2$
- For each (i', l') score, need posterior $p(\beta | y_{i',l'}, X, Y)$.

Approximate inference and Active Learning

- Suppose we start with training data Y.
- Query task-labeler pair (i, l) to maximize expected utility of adding it

$$(i, l) = \text{argmax}_{(i',l')} E_{y_{i',l'}} \left(U \left(p(\beta | y_{i',l'}, X, Y) \right) \right)$$

- Examples: $U(\cdot) = -\text{Entropy}(\cdot)$. $U_\mu(\cdot) = \|\text{Mean}(\cdot) - \mu\|_2^2$
- For each (i', l') score, need posterior $p(\beta | y_{i',l'}, X, Y)$.
- Gibbs sampling $\Rightarrow$ separate Gibbs samplers to score (i', l').

Approximate inference and Active Learning

- Suppose we start with training data Y.
- Query task-labeler pair (i, l) to maximize expected utility of adding it

$$(i, l) = \text{argmax}_{(i',l')} E_{y_{i',l'}} \left(U \left(p(\beta | y_{i',l'}, X, Y) \right) \right)$$

- Examples: $U(\cdot) = -\text{Entropy}(\cdot)$. $U_\mu(\cdot) = \|\text{Mean}(\cdot) - \mu\|_2^2$
- For each (i', l') score, need posterior $p(\beta | y_{i',l'}, X, Y)$.
- Gibbs sampling $\Rightarrow$ separate Gibbs samplers to score (i', l').

- We are already running one Gibbs sampler for basic inference.

Approximate inference and Active Learning

- Suppose we start with training data Y.
- Query task-labeler pair (i, l) to maximize expected utility of adding it

$$(i, l) = \text{argmax}_{(i',l')} E_{y_{i',l'}} \left(U \left(p(\beta | y_{i',l'}, X, Y) \right) \right)$$

- Examples: $U(\cdot) = -\text{Entropy}(\cdot)$. $U_\mu(\cdot) = \|\text{Mean}(\cdot) - \mu\|_2^2$
- For each (i', l') score, need posterior $p(\beta | y_{i',l'}, X, Y)$.
- Gibbs sampling $\Rightarrow$ separate Gibbs samplers to score (i', l').

- We are already running one Gibbs sampler for basic inference.
- **Problem: Can we avoid running the extra scoring chains?**

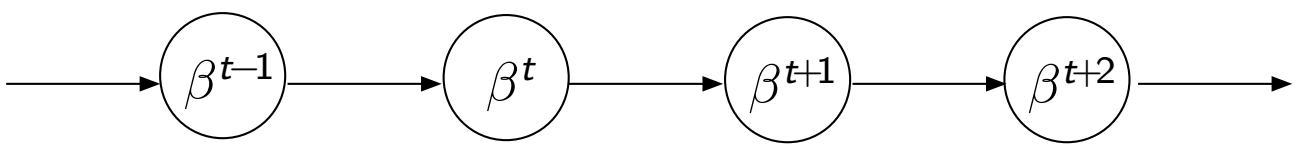
Contribution II: Approximate Active Learning

Contribution II: Approximate Active Learning

- Gibbs sampler for $p(\beta|X, Y)$ is a Markov chain for *inference*

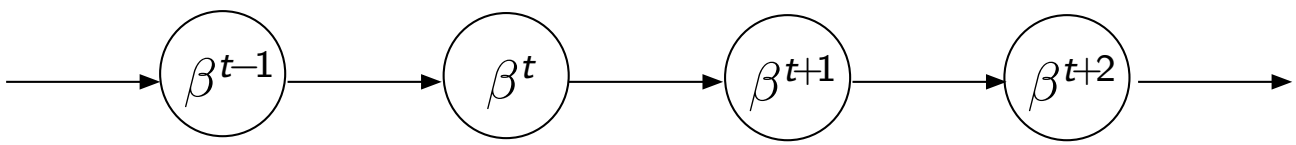
Contribution II: Approximate Active Learning

- Gibbs sampler for $p(\beta|X, Y)$ is a Markov chain for *inference*



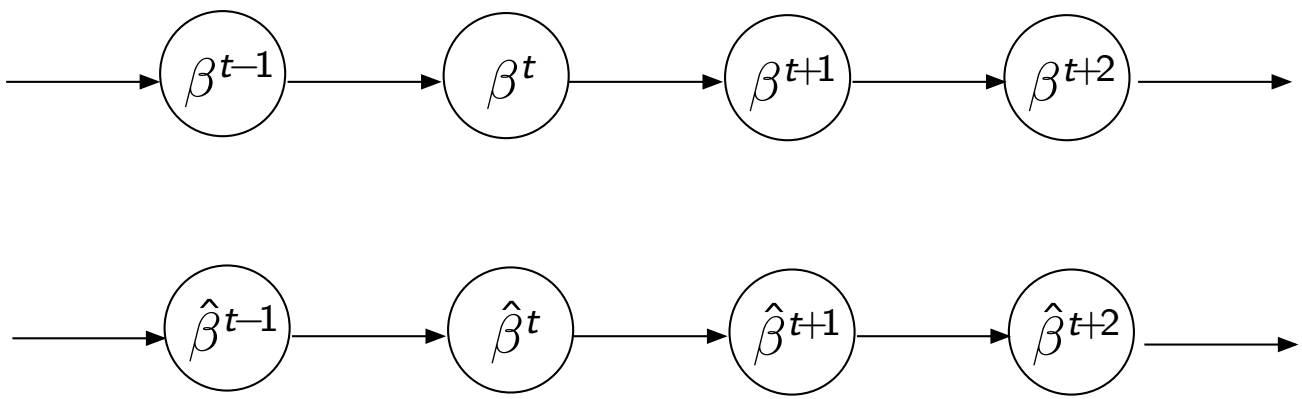
Contribution II: Approximate Active Learning

- Gibbs sampler for $p(\beta|X, Y)$ is a Markov chain for *inference*
- Sampler for $p(\beta|y_{i',l'}, X, Y)$ is a *perturbed* chain for *scoring*



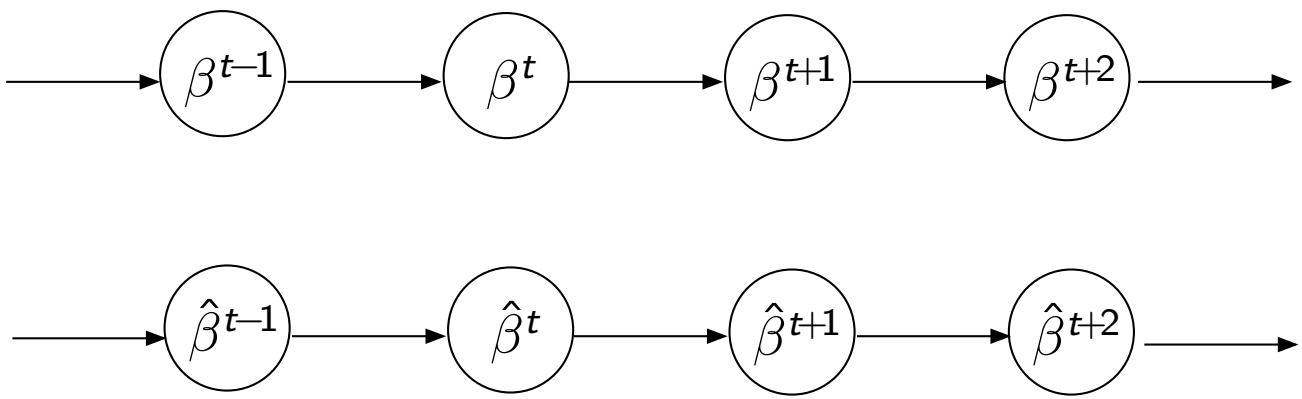
Contribution II: Approximate Active Learning

- Gibbs sampler for $p(\beta|X, Y)$ is a Markov chain for *inference*
- Sampler for $p(\beta|y_{i',l'}, X, Y)$ is a *perturbed* chain for *scoring*



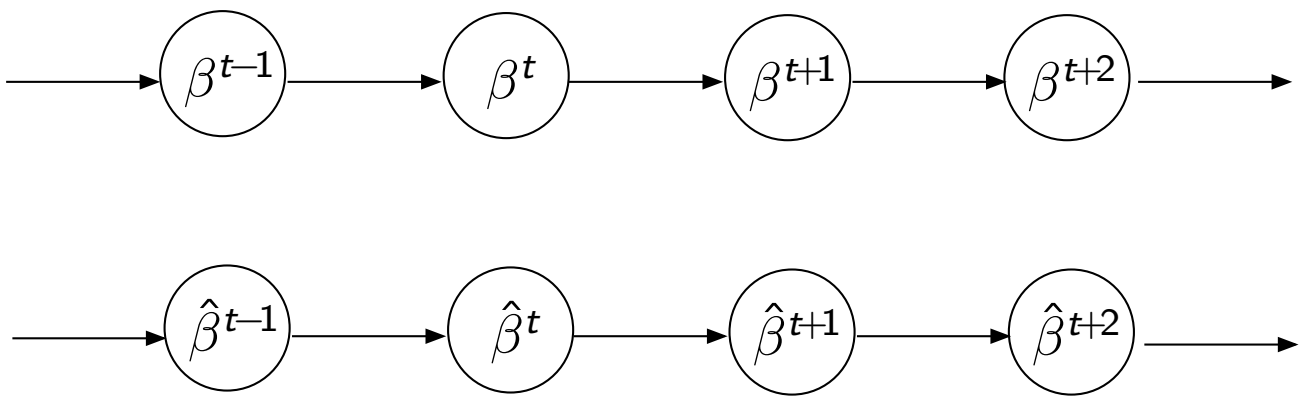
Contribution II: Approximate Active Learning

- Gibbs sampler for $p(\beta|X, Y)$ is a Markov chain for *inference*
- Sampler for $p(\beta|y_{i',l'}, X, Y)$ is a *perturbed* chain for *scoring*
- **Naïve** scoring:



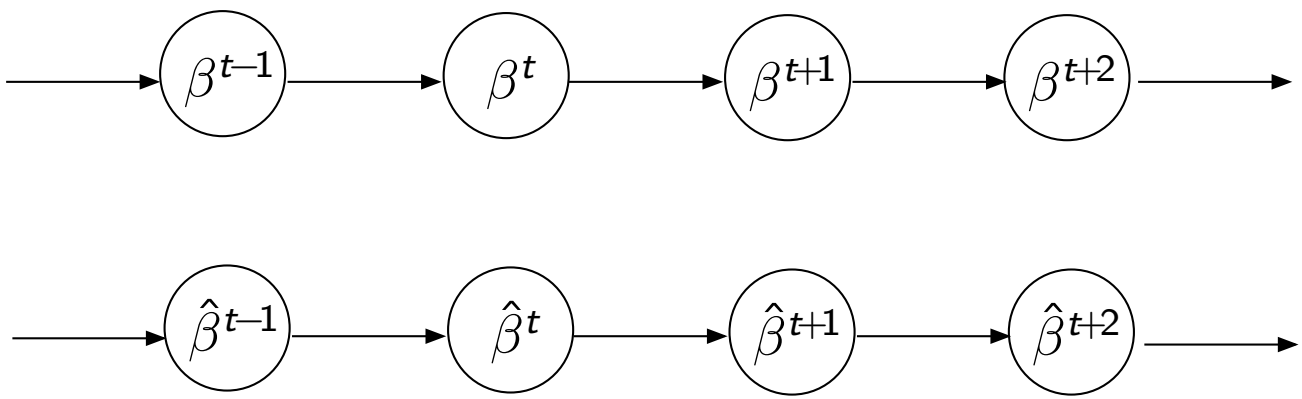
Contribution II: Approximate Active Learning

- Gibbs sampler for $p(\beta|X, Y)$ is a Markov chain for *inference*
- Sampler for $p(\beta|y_{i',l'}, X, Y)$ is a *perturbed* chain for *scoring*
- **Naïve** scoring:
  - Run perturbed chain; sample from stationary distribution.
  - Compute $U(p(\beta|y_{i',l'}, X, Y))$.



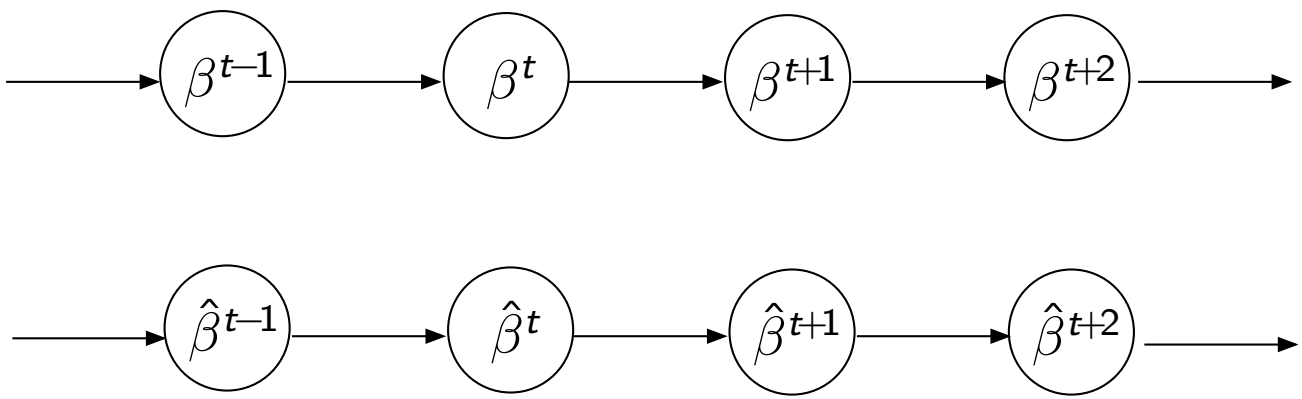
Contribution II: Approximate Active Learning

- Gibbs sampler for $p(\beta|X, Y)$ is a Markov chain for *inference*
- Sampler for $p(\beta|y_{i',l'}, X, Y)$ is a *perturbed* chain for *scoring*
- **Naïve** scoring:
  - Run perturbed chain; sample from stationary distribution.
  - Compute $U(p(\beta|y_{i',l'}, X, Y))$.
- **Our method**:



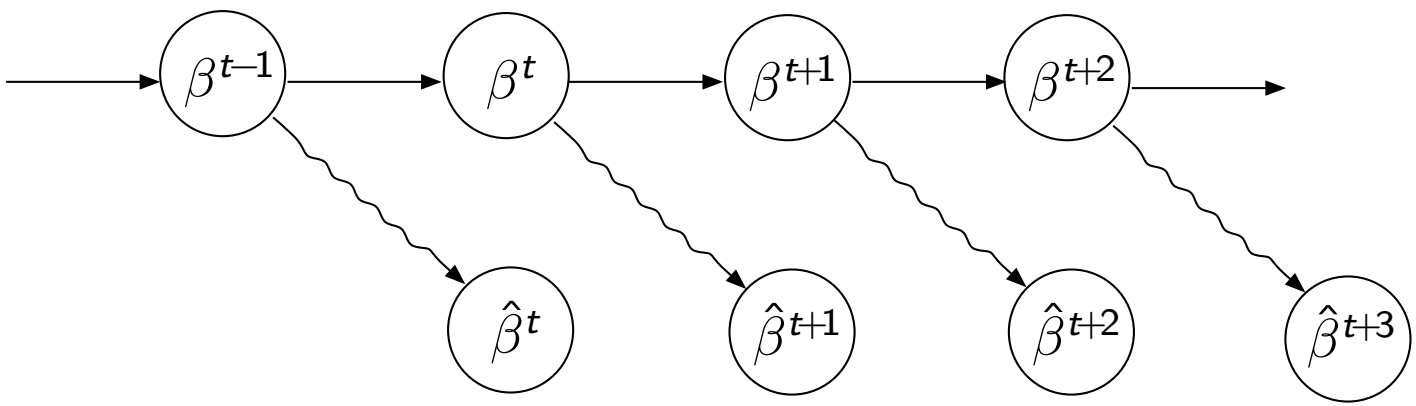
Contribution II: Approximate Active Learning

- Gibbs sampler for $p(\beta|X, Y)$ is a Markov chain for *inference*
- Sampler for $p(\beta|y_{i',l'}, X, Y)$ is a *perturbed* chain for *scoring*
- **Naïve** scoring:
  - Run perturbed chain; sample from stationary distribution.
  - Compute $U(p(\beta|y_{i',l'}, X, Y))$.
- **Our method**:
  - Get approximate samples of $p(\beta|y_{i',l'}, X, Y)$ by *transforming* samples of $p(\beta|X, Y)$.



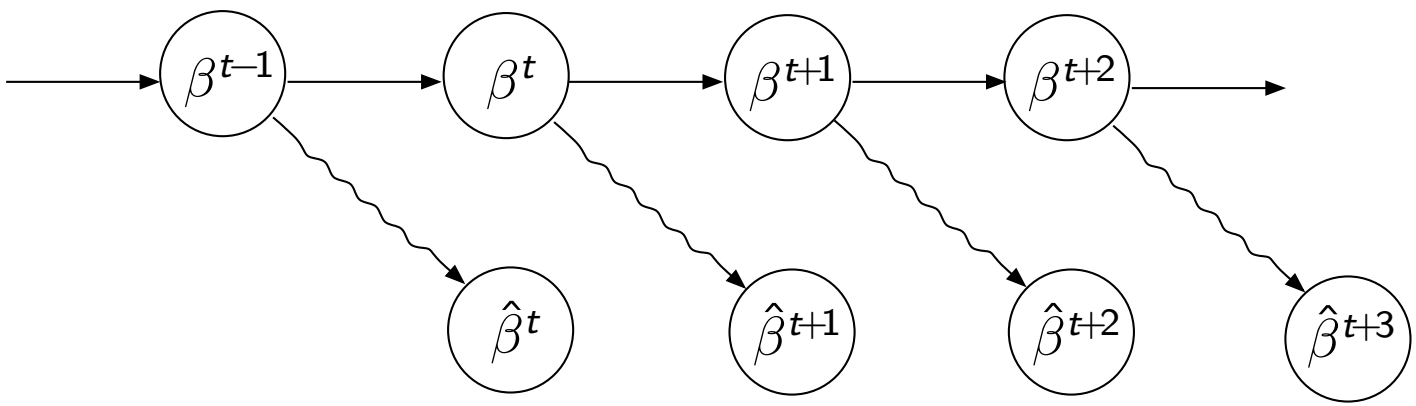
Contribution II: Approximate Active Learning

- Gibbs sampler for $p(\beta|X, Y)$ is a Markov chain for *inference*
- Sampler for $p(\beta|y_{i',l'}, X, Y)$ is a *perturbed* chain for *scoring*
- **Naïve** scoring:
  - Run perturbed chain; sample from stationary distribution.
  - Compute $U(p(\beta|y_{i',l'}, X, Y))$.
- **Our method**:
  - Get approximate samples of $p(\beta|y_{i',l'}, X, Y)$ by *transforming* samples of $p(\beta|X, Y)$.



Contribution II: Approximate Active Learning

- Gibbs sampler for $p(\beta|X, Y)$ is a Markov chain for *inference*
- Sampler for $p(\beta|y_{i',l'}, X, Y)$ is a *perturbed* chain for *scoring*
- **Naïve** scoring:
  - Run perturbed chain; sample from stationary distribution.
  - Compute $U(p(\beta|y_{i',l'}, X, Y))$.
- **Our method**:
  - Get approximate samples of $p(\beta|y_{i',l'}, X, Y)$ by *transforming* samples of $p(\beta|X, Y)$.
  - Approximate $U(p(\beta|y_{i',l'}, X, Y))$ from these.



Approximate Scoring for Active Learning

- Suppose chain $p(\beta^t|\beta^{t-1})$ and a perturbed chain $\hat{p}(\hat{\beta}^t|\hat{\beta}^{t-1})$.
- Stationary distributions are $p_\infty(\beta)$ and $\hat{p}_\infty(\hat{\beta})$.

Approximate Scoring for Active Learning

- Suppose chain $p(\beta^t|\beta^{t-1})$ and a perturbed chain $\hat{p}(\hat{\beta}^t|\hat{\beta}^{t-1})$.
- Stationary distributions are $p_\infty(\beta)$ and $\hat{p}_\infty(\hat{\beta})$.
- Let $\beta^s \sim p_\infty(\beta)$ $s = 1, \ldots, S$, and approximate

$$\hat{p}_\infty(\hat{\beta}) \approx \int \hat{p}(\hat{\beta}|\beta) p_\infty(\beta) d\beta \approx \frac{1}{S} \sum_{s=1}^{S} \hat{p}(\hat{\beta}|\beta^s).$$

Approximate Scoring for Active Learning

- Suppose chain $p(\beta^t|\beta^{t-1})$ and a perturbed chain $\hat{p}(\hat{\beta}^t|\hat{\beta}^{t-1})$.
- Stationary distributions are $p_\infty(\beta)$ and $\hat{p}_\infty(\hat{\beta})$.
- Let $\beta^s \sim p_\infty(\beta)$ $s = 1, \ldots, S$, and approximate

$$\hat{p}_\infty(\hat{\beta}) \approx \int \hat{p}(\hat{\beta}|\beta) p_\infty(\beta) d\beta \approx \frac{1}{S} \sum_{s=1}^{S} \hat{p}(\hat{\beta}|\beta^s).$$

- If $p_\infty(\beta) = \hat{p}_\infty(\beta)$, the first approximation is exact.

Approximate Scoring for Active Learning

- Suppose chain $p(\beta^t|\beta^{t-1})$ and a perturbed chain $\hat{p}(\hat{\beta}^t|\hat{\beta}^{t-1})$.
- Stationary distributions are $p_\infty(\beta)$ and $\hat{p}_\infty(\hat{\beta})$.
- Let $\beta^s \sim p_\infty(\beta)$ $s = 1, \ldots, S$, and approximate

$$\hat{p}_\infty(\hat{\beta}) \approx \int \hat{p}(\hat{\beta}|\beta) p_\infty(\beta) d\beta \approx \frac{1}{S} \sum_{s=1}^{S} \hat{p}(\hat{\beta}|\beta^s).$$

- If $p_\infty(\beta) = \hat{p}_\infty(\beta)$, the first approximation is exact.
- Specialize to active learning:
  - Unperturbed chain = Gibbs sampler for $p(\beta|X, Y)$.
  - Perturbed chain = Gibbs sampler for $p(\beta|y_{i',l'}, X, Y)$.

Special Case: Discrete Random Walks

- Suppose W is $n \times n$, positive, symmetric. $P = D^{-1}W$.

Special Case: Discrete Random Walks

- Suppose W is $n \times n$, positive, symmetric. $P = D^{-1}W$.
- Stationary distribution is left eigenvector of P. Decompose

$$A = D^{-1/2}WD^{-1/2} \tag{5}$$

$$= V\Lambda V^{\top}, \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n = 1 \tag{6}$$

$$p_{\infty} \propto D^{1/2}v_n \tag{7}$$

Special Case: Discrete Random Walks

- Suppose W is $n \times n$, positive, symmetric. $P = D^{-1}W$.
- Stationary distribution is left eigenvector of P. Decompose

$$A = D^{-1/2}WD^{-1/2} \tag{5}$$

$$= V\Lambda V^{\top}, \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n = 1 \tag{6}$$

$$p_{\infty} \propto D^{1/2}v_n \tag{7}$$

- Perturb the matrix: $\hat{W} = W + dW \geq 0$, with $dW\mathbf{1} = 0$.

Special Case: Discrete Random Walks

- Suppose W is $n \times n$, positive, symmetric. $P = D^{-1}W$.
- Stationary distribution is left eigenvector of P. Decompose

$$A = D^{-1/2} W D^{-1/2} \tag{5}$$

$$= V \Lambda V^{\top}, \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n = 1 \tag{6}$$

$$p_{\infty} \propto D^{1/2} v_n \tag{7}$$

- Perturb the matrix: $\hat{W} = W + dW \geq 0$, with $dW\mathbf{1} = 0$.
- Then $\hat{P} = D^{-1}\hat{W} = P + D^{-1}dW = P + dP$.

Special Case: Discrete Random Walks

- Matrix perturbation theory:

$$\tilde{p}_\infty \approx p_\infty + D^{1/2} \left[\sum_{k \neq n} \frac{v_k v_k^\top}{1 - \lambda_k} \right] dP^\top D^{-1/2} p_\infty \qquad (8)$$

Special Case: Discrete Random Walks

- Matrix perturbation theory:

$$\tilde{p}_\infty \approx p_\infty + D^{1/2} \left[\sum_{k \neq n} \frac{v_k v_k^\top}{1 - \lambda_k} \right] dP^\top D^{-1/2} p_\infty \qquad (8)$$

- Works for discrete random walks, but not in general.

Special Case: Discrete Random Walks

- Matrix perturbation theory:

$$\tilde{p}_\infty \approx p_\infty + D^{1/2} \left[\sum_{k \neq n} \frac{v_k v_k^\top}{1 - \lambda_k} \right] dP^\top D^{-1/2} p_\infty \tag{8}$$

- Works for discrete random walks, but not in general.
- Our method is general and approximates:

$$\hat{p}_\infty \approx \hat{P}^\top p_\infty = p_\infty + dP^\top p_\infty. \tag{9}$$

Special Case: Discrete Random Walks

- Matrix perturbation theory:

$$\tilde{p}_\infty \approx p_\infty + D^{1/2} \left[\sum_{k \neq n} \frac{v_k v_k^\top}{1 - \lambda_k} \right] dP^\top D^{-1/2} p_\infty \tag{8}$$

- Works for discrete random walks, but not in general.
- Our method is general and approximates:

$$\hat{p}_\infty \approx \hat{P}^\top p_\infty = p_\infty + dP^\top p_\infty. \tag{9}$$

- If $D = I$ then accuracy depends on spectral gap.

$$\|\tilde{p}_\infty - \hat{p}_\infty\| \leq \max\left(1, \frac{1}{1 - \lambda_{n-1}}\right) \left\| dP^\top p_\infty \right\|. \tag{10}$$

Overview

Contribution I: Bayesian Preference Model

BBMC Results

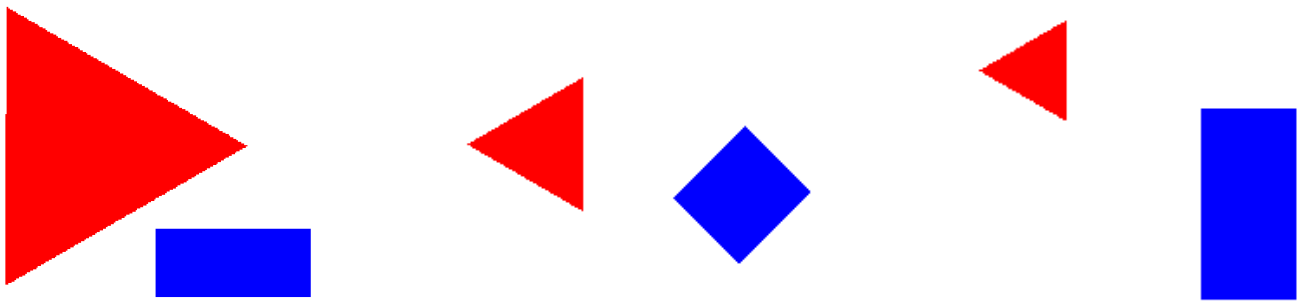
Contribution II: Approximate Active Learning

Active Learning Results

Conclusion

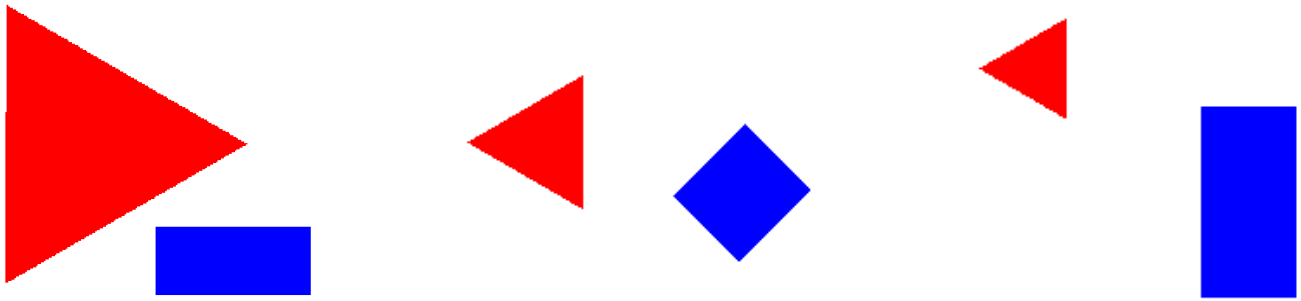
Results: Crowdsourced data

- **Task:** Is the triangle to the left or above the rectangle



Results: Crowdsourced data

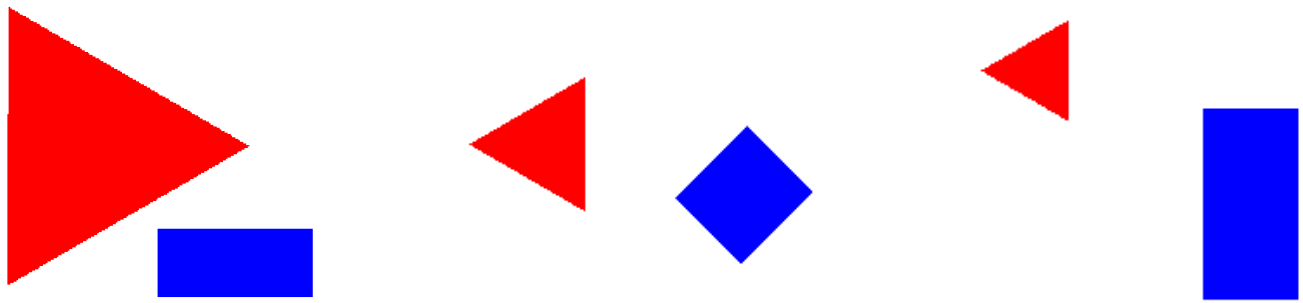
- **Task:** Is the triangle to the left or above the rectangle



- Active learning methods can query 100 labels.

Results: Crowdsourced data

- **Task:** Is the triangle to the left or above the rectangle



- Active learning methods can query 100 labels.
- Here: only query gold standard (could be other labeller).

- Averaged log likelihood and error rate on test set.
- BBMC and BBMC-ACT: us with/without active learning.

	Algorithm	Final Loglik	Final Error
No Active Learning	GOLD	-3716 ± 1695	0.0547 ± 0.0102
	CONS	-421.1 ± 2.6	0.0935 ± 0.0031
	BBMC	$\mathbf{-219.1 \pm 3.1}$	$\mathbf{0.0309 \pm 0.0033}$

- Averaged log likelihood and error rate on test set.
- BBMC and BBMC-ACT: us with/without active learning.

	Algorithm	Final Loglik	Final Error
No Active Learning	GOLD	-3716 ± 1695	0.0547 ± 0.0102
	CONS	-421.1 ± 2.6	0.0935 ± 0.0031
	BBMC	$\mathbf{-219.1 \pm 3.1}$	$\mathbf{0.0309 \pm 0.0033}$
Active Learning	GOLD-ACT	-1957 ± 696	0.0290 ± 0.0037
	CONS-ACT	-396.1 ± 3.6	0.0906 ± 0.0024
	RAND-ACT	-186.0 ± 2.2	0.0292 ± 0.0029
	DIS-ACT	-198.3 ± 5.8	0.0392 ± 0.0052
	MCMC-ACT	-196.1 ± 6.7	0.0492 ± 0.0050
	BBMC-ACT	$\mathbf{-160.8 \pm 3.9}$	$\mathbf{0.0188 \pm 0.0018}$

Conclusion

- Bayesian model to mitigate label bias.

Conclusion

- Bayesian model to mitigate label bias.
  - Unifies crowdsourcing pipeline into one model.

Conclusion

- Bayesian model to mitigate label bias.
  - Unifies crowdsourcing pipeline into one model.
  - Labelers express **accumulated**, **shared** preferences.

Conclusion

- Bayesian model to mitigate label bias.
  - Unifies crowdsourcing pipeline into one model.
  - Labelers express **accumulated**, **shared** preferences.
  - Scales well in the number of tasks.

Conclusion

- Bayesian model to mitigate label bias.
  - Unifies crowdsourcing pipeline into one model.
  - Labelers express **accumulated**, **shared** preferences.
  - Scales well in the number of tasks.
  - Performs well when consensus labels cannot be estimated.

Conclusion

- Bayesian model to mitigate label bias.
  - Unifies crowdsourcing pipeline into one model.
  - Labelers express **accumulated**, **shared** preferences.
  - Scales well in the number of tasks.
  - Performs well when consensus labels cannot be estimated.
- Approximate active learning for Gibbs sampling inference.

Conclusion

- Bayesian model to mitigate label bias.
  - Unifies crowdsourcing pipeline into one model.
  - Labelers express **accumulated**, **shared** preferences.
  - Scales well in the number of tasks.
  - Performs well when consensus labels cannot be estimated.
- Approximate active learning for Gibbs sampling inference.
  - Fast scoring by reusing Gibbs samples.

Conclusion

- Bayesian model to mitigate label bias.
  - Unifies crowdsourcing pipeline into one model.
  - Labelers express **accumulated**, **shared** preferences.
  - Scales well in the number of tasks.
  - Performs well when consensus labels cannot be estimated.
- Approximate active learning for Gibbs sampling inference.
  - Fast scoring by reusing Gibbs samples.
  - Outperforms naïve MCMC scoring.

Questions?