Coresets for automated, scalable Bayesian inference

Tamara Broderick
ITT Career Development Assistant Professor, MIT

With: Trevor Campbell, Jonathan H. Huggins
Bayesian inference

- Microcredit

Fuel consumption

Cybersecurity
Bayesian inference

- Microcredit

- Challenge: existing methods can be slow (and/or tedious, unreliable)

- Fuel consumption

- Cybersecurity
Bayesian inference

- Microcredit

- Challenge: existing methods can be slow (and/or tedious, unreliable)
- Our proposal: use efficient summarization of data
Bayesian inference

- Microcredit

- Challenge: existing methods can be slow (and/or tedious, unreliable)
- Our proposal: use efficient summarization of data
- Coresets for scalable, automated approximate Bayes algorithms with error bounds for finite data

- Fuel consumption

- Cybersecurity
Roadmap

- Approximate Bayes review
- The “core” of the data set
- Uniform data subsampling isn’t enough
- Importance sampling for “coresets”
- Optimization for “coresets”
Roadmap

• Approximate Bayes review
• The “core” of the data set
• Uniform data subsampling isn’t enough
• Importance sampling for “coresets”
• Optimization for “coresets”
Bayesian inference
Bayesian inference  \( p(\theta) \)
Bayesian inference

\[ p(y|\theta)p(\theta) \]
Bayesian inference

\[ p(\theta|y) \propto \theta \ p(y|\theta)p(\theta) \]
Bayesian inference

\[ p(\theta|y) \propto p(y|\theta)p(\theta) \]

\((x_n, y_n)\)
Bayesian inference

\[ p(\theta|y) \propto p(y|\theta)p(\theta) \]

Normal

\((x_n, y_n)\)

Phishing
Bayesian inference

\[ p(\theta \mid y) \propto_\theta p(y \mid \theta)p(\theta) \]

Normal

\((x_n, y_n)\)

\(\theta\)

Phishing
Bayesian inference

\[ p(\theta|y) \propto_\theta p(y|\theta)p(\theta) \]

Normal

\[(x_n, y_n)\]

\(\theta\)

Phishing
Bayesian inference

\[ p(\theta | y) \propto p(y | \theta) p(\theta) \]

Normal

\((x_n, y_n)\)

Phishing

Exact posterior

[Bishop 2006]
Bayesian inference

$$p(\theta|y) \propto p(y|\theta)p(\theta)$$

- MCMC: Eventually accurate but can be slow

[Bishop 2006]

[Bardenet, Doucet, Holmes 2015]
Bayesian inference

\[ p(\theta | y) \propto p(y | \theta) p(\theta) \]

- MCMC: Eventually accurate but can be slow
- (Mean-field) variational Bayes: (MF)VB

\[ (x_n, y_n) \]

Normal

Phishing

[\text{Bardenet, Doucet, Holmes 2015}]

[Bishop 2006]
Bayesian inference

\[ p(\theta|y) \propto p(y|\theta)p(\theta) \]

- MCMC: Eventually accurate but can be slow
- (Mean-field) variational Bayes: (MF)VB
  - Fast

\[ (x_n, y_n) \]

Normal

Phishing
Bayesian inference

\[ p(\theta|y) \propto p(y|\theta)p(\theta) \]

- MCMC: Eventually accurate but can be slow
- (Mean-field) variational Bayes: (MF)VB
  - Fast, streaming, distributed [Broderick, Boyd, Wibisono, Wilson, Jordan 2013]
  - (3.6M Wikipedia, 32 cores, ~hour) [Bardenet, Doucet, Holmes 2015]
Bayesian inference

\[ p(\theta | y) \propto p(y | \theta) p(\theta) \]

- MCMC: Eventually accurate but can be slow
- (Mean-field) variational Bayes: (MF)VB
  - Fast, streaming, distributed [Broderick, Boyd, Wibisono, Wilson, Jordan 2013]
    (3.6M Wikipedia, 32 cores, \sim\text{hour})
- Misestimation & lack of quality guarantees

[MacKay 2003; Bishop 2006; Wang, Titterington 2004; Turner, Sahani 2011; Fosdick 2013; Dunson 2014; Bardenet, Doucet, Holmes 2015; Opper, Winther 2003; Giordano, Broderick, Jordan 2015]
Bayesian inference

\[ p(\theta | y) \propto p(y | \theta) p(\theta) \]

- MCMC: Eventually accurate but can be slow
- (Mean-field) variational Bayes: (MF)VB
  - Fast, streaming, distributed [Broderick, Boyd, Wibisono, Wilson, Jordan 2013]
    (3.6M Wikipedia, 32 cores, ~hour)
  - Misestimation & lack of quality guarantees
    [MacKay 2003; Bishop 2006; Wang, Titterington 2004; Turner, Sahani 2011; Fosdick 2013; Dunson 2014; Bardenet, Doucet, Holmes 2015; Opper, Winther 2003; Giordano, Broderick, Jordan 2015]
Bayesian inference

\[ p(\theta | y) \propto p(y | \theta) p(\theta) \]

- **MCMC**: Eventually accurate but can be slow
  - (Mean-field) variational Bayes: (MF)VB
    - Fast, streaming, distributed [Broderick, Boyd, Wibisono, Wilson, Jordan 2013]
      - (3.6M Wikipedia, 32 cores, ~hour)
    - Misestimation & lack of quality guarantees [MacKay 2003; Bishop 2006; Wang, Titterington 2004; Turner, Sahani 2011; Fosdick 2013; Dunson 2014; Bardenet, Doucet, Holmes 2015; Opper, Winther 2003; Giordano, Broderick, Jordan 2015]
- **Automation**: e.g. Stan, NUTS, ADVI
  - [http://mc-stan.org/ ; Hoffman, Gelman 2014; Kucukelbir, Tran, Ranganath, Gelman, Blei 2017]
Roadmap

- Approximate Bayes review
- The “core” of the data set
- Uniform data subsampling isn’t enough
- Importance sampling for “coresets”
- Optimization for “coresets”
Roadmap

- Approximate Bayes review
- The “core” of the data set
- Uniform data subsampling isn’t enough
- Importance sampling for “coresets”
- Optimization for “coresets”
Bayesian coresets
Bayesian coresets

• Observe: redundancies can exist even if data isn’t “tall”
Bayesian coresets

- Observe: redundancies can exist even if data isn’t “tall”
Bayesian coresets

- Observe: redundancies can exist even if data isn’t “tall”
Bayesian coresets

- Observe: redundancies can exist even if data isn’t “tall”
Bayesian coresets

- Observe: redundancies can exist even if data isn’t “tall”
- Coresets: pre-process data to get a smaller, weighted data set

[Agarwal et al 2005; Feldman & Langberg 2011]
Bayesian coresets

- Observe: redundancies can exist even if data isn’t “tall”
- Coresets: pre-process data to get a smaller, weighted data set

- Theoretical guarantees on quality

[Agarwal et al 2005; Feldman & Langberg 2011]
Bayesian coresets

- Observe: redundancies can exist even if data isn’t “tall”
- Coresets: pre-process data to get a smaller, weighted data set
- Theoretical guarantees on quality
- Previous heuristics: data squashing, big data GPs
Bayesian coresets

- Observe: redundancies can exist even if data isn’t “tall”
- Coresets: pre-process data to get a smaller, weighted data set

- Theoretical guarantees on quality
- Previous heuristics: data squashing, big data GPs
- Cf. subsampling

[Agarwal et al 2005; Feldman & Langberg 2011; DuMouchel et al 1999; Madigan et al 1999]
Bayesian coresets

- Observe: redundancies can exist even if data isn’t “tall”
- Coresets: pre-process data to get a smaller, weighted data set

- Theoretical guarantees on quality
- Previous heuristics: data squashing, big data GPs
- Cf. subsampling

[Agarwal et al 2005; Feldman & Langberg 2011; DuMouchel et al 1999; Madigan et al 1999]
Bayesian coresets

- Observe: redundancies can exist even if data isn’t “tall”
- Coresets: pre-process data to get a smaller, weighted data set

- Theoretical guarantees on quality
- Previous heuristics: data squashing, big data GPs
- Cf. subsampling

[Agarwal et al 2005; Feldman & Langberg 2011; DuMouchel et al 1999; Madigan et al 1999]
Bayesian coresets

• Observe: redundancies can exist even if data isn’t “tall”
• Coresets: pre-process data to get a smaller, weighted data set

• Theoretical guarantees on quality
• Previous heuristics: data squashing, big data GPs
• Cf. subsampling
• How to develop coresets for Bayes?

[Agarwal et al 2005; Feldman & Langberg 2011; DuMouchel et al 1999; Madigan et al 1999; Huggins, Campbell, Broderick 2016; Campbell, Broderick 2017; Campbell, Broderick 2018]
Bayesian coresets

- Posterior \( p(\theta|y) \propto \theta \cdot p(y|\theta)p(\theta) \)
Bayesian coresets

- Posterior \( p(\theta|y) \propto p(y|\theta)p(\theta) \)

- Log likelihood \( \mathcal{L}_n(\theta) := \log p(y_n|\theta) \), \( \mathcal{L}(\theta) := \sum_{n=1}^{N} \mathcal{L}_n(\theta) \)
Bayesian coresets

• Posterior $p(\theta|y) \propto_{\theta} p(y|\theta)p(\theta)$

• Log likelihood $\mathcal{L}_n(\theta) := \log p(y_n|\theta)$, $\mathcal{L}(\theta) := \sum_{n=1}^{N} \mathcal{L}_n(\theta)$

• Coreset log likelihood
Bayesian coresets

• Posterior \( p(\theta|y) \propto p(y|\theta)p(\theta) \)

• Log likelihood \( \mathcal{L}_n(\theta) := \log p(y_n|\theta) \), \( \mathcal{L}(\theta) := \sum_{n=1}^{N} \mathcal{L}_n(\theta) \)

• Coreset log likelihood

\[ \|w\|_0 \ll N \]
Bayesian coresets

- Posterior \( p(\theta|y) \propto \theta \cdot p(y|\theta)p(\theta) \)

- Log likelihood \( \mathcal{L}_n(\theta) := \log p(y_n|\theta), \quad \mathcal{L}(\theta) := \sum_{n=1}^{N} \mathcal{L}_n(\theta) \)

- Coreset log likelihood \( \mathcal{L}(w, \theta) := \sum_{n=1}^{N} w_n \mathcal{L}_n(\theta) \quad \text{s.t.} \quad \|w\|_0 \ll N \)
Bayesian coresets

- Posterior \( p(\theta|y) \propto_\theta p(y|\theta)p(\theta) \)
- Log likelihood \( \mathcal{L}_n(\theta) := \log p(y_n|\theta) \), \( \mathcal{L}(\theta) := \sum_{n=1}^{N} \mathcal{L}_n(\theta) \)
- Coreset log likelihood \( \mathcal{L}(w, \theta) := \sum_{n=1}^{N} w_n \mathcal{L}_n(\theta) \) s.t. \( \|w\|_0 \ll N \)
Bayesian coresets

• Posterior\[ p(\theta | y) \propto_{\theta} p(y | \theta)p(\theta) \]

• Log likelihood\[ \mathcal{L}_n(\theta) := \log p(y_n | \theta), \quad \mathcal{L}(\theta) := \sum_{n=1}^{N} \mathcal{L}_n(\theta) \]

• Coreset log likelihood\[ \mathcal{L}(w, \theta) := \sum_{n=1}^{N} w_n \mathcal{L}_n(\theta) \quad \text{s.t.} \quad \|w\|_0 \ll N \]

• \(\varepsilon\)-coreset: \[\|\mathcal{L}(w) - \mathcal{L}\| \leq \varepsilon\]
Bayesian coresets

- Posterior \( p(\theta | y) \propto p(y | \theta) p(\theta) \)
- Log likelihood \( \mathcal{L}_n(\theta) := \log p(y_n | \theta) \), \( \mathcal{L}(\theta) := \sum_{n=1}^{N} \mathcal{L}_n(\theta) \)
- Coreset log likelihood \( \mathcal{L}(w, \theta) := \sum_{n=1}^{N} w_n \mathcal{L}_n(\theta) \) s.t. \( \|w\|_0 \ll N \)
- \( \varepsilon \)-coreset: \( \|\mathcal{L}(w) - \mathcal{L}\| \leq \varepsilon \)
- Approximate posterior close in Wasserstein distance
Roadmap

• Approximate Bayes review
• The “core” of the data set
• Uniform data subsampling isn’t enough
• Importance sampling for “coresets”
• Optimization for “coresets”
Roadmap

• Approximate Bayes review
• The “core” of the data set
• Uniform data subsampling isn’t enough
• Importance sampling for “coresets”
• Optimization for “coresets”
Uniform subsampling revisited

Normal

Phishing
Uniform subsampling revisited

Normal

Phishing
Uniform subsampling revisited
Uniform subsampling revisited

- Normal
- Phishing
  - Might miss important data
Uniform subsampling revisited

- Might miss important data
Uniform subsampling revisited

- Normal
- Phishing
  - Might miss important data
Uniform subsampling revisited

- Normal
- Phishing

- Might miss important data
Uniform subsampling revisited

- Might miss important data
Uniform subsampling revisited

- Normal
- Phishing

- Might miss important data
Uniform subsampling revisited

- Might miss important data
Uniform subsampling revisited

• Might miss important data
Uniform subsampling revisited

- Normal
- Phishing

• Might miss important data
Uniform subsampling revisited

- Normal
- Phishing

- Might miss important data
Uniform subsampling revisited

- Normal
- Phishing

• Might miss important data
• Noisy estimates
Uniform subsampling revisited

- Might miss important data
- Noisy estimates

$M = 10$
Uniform subsampling revisited

- Might miss important data
- Noisy estimates

$M = 10$

$M = 100$

$M = 1000$
Roadmap

• Approximate Bayes review
• The “core” of the data set
• Uniform data subsampling isn’t enough
• Importance sampling for “coresets”
• Optimization for “coresets”
Roadmap

• Approximate Bayes review
• The “core” of the data set
• Uniform data subsampling isn’t enough
• Importance sampling for “coresets”
• Optimization for “coresets”
Importance sampling

Normal

Phishing
Importance sampling

Normal

Phishing
Importance sampling

Normal

Phishing
Importance sampling

- Normal
- Phishing
Importance sampling
Importance sampling
Importance sampling

\[ \sigma_n \propto \| \mathcal{L}_n \| \]
Importance sampling

\[ \sigma := \sum_{n=1}^{N} \| \mathcal{L}_n \| \]

\[ \sigma_n := \frac{\| \mathcal{L}_n \|}{\sigma} \]
**Importance sampling**

**Thm sketch (CB).** \( \delta \in (0,1) \). W.p. \( \geq 1 - \delta \), after \( M \) iterations,

\[
\| \mathcal{L}(w) - \mathcal{L} \| \leq \frac{\sigma \bar{\eta}}{\sqrt{M}} \left( 1 + \sqrt{2 \log \frac{1}{\delta}} \right)
\]
Importance sampling

**Thm sketch (CB).** $\delta \in (0,1)$. W.p. $\geq 1 - \delta$, after $M$ iterations,

$$\| L(w) - L \| \leq \frac{\sigma \bar{\eta}}{\sqrt{M}} \left( 1 + \sqrt{2 \log \frac{1}{\delta}} \right)$$

- Still noisy estimates

$M = 10$
Importance sampling

**Thm sketch (CB).** $\delta \in (0,1)$. W.p. $\geq 1 - \delta$, after $M$ iterations,

$$\| \mathcal{L}(w) - \mathcal{L} \| \leq \frac{\sigma \bar{\eta}}{\sqrt{M}} \left( 1 + \sqrt{2 \log \frac{1}{\delta}} \right)$$

- Still noisy estimates

$M = 10$

$M = 100$

$M = 1000$
Hilbert coresets

- Want a good coreset: \( \min_{w \in \mathbb{R}^N} \| \mathcal{L}(w) - \mathcal{L} \| \)

\[ \text{s.t. } w \geq 0, \|w\|_0 \leq M \]
Hilbert coresets

- Want a good coreset: \[
\min_{w \in \mathbb{R}^N} \| \mathcal{L}(w) - \mathcal{L} \|
\]

subject to \( w \geq 0, \|w\|_0 \leq M \)

\[
\exp(\mathcal{L}(\theta)) \quad \exp(\mathcal{L}_n(\theta))
\]
Hilbert coresets

- Want a good coreset: \( \min_{w \in \mathbb{R}^N} \| \mathcal{L}(w) - \mathcal{L} \| \)
  
  \[ \text{s.t. } w \geq 0, \|w\|_0 \leq M \]
Hilbert coresets

• Want a good coreset: \[
\min_{w \in \mathbb{R}^N} \| \mathcal{L}(w) - \mathcal{L} \|
\]
\[
\text{s.t. } w \geq 0, \|w\|_0 \leq M
\]

• need to consider (residual) error direction
Hilbert coresets

• Want a good coreset:
  \[
  \min_{w \in \mathbb{R}^N} \| \mathcal{L}(w) - \mathcal{L} \|
  \]
  s.t. \( w \geq 0, \|w\|_0 \leq M \)

• need to consider (residual) error direction
• sparse optimization
Roadmap

• Approximate Bayes review
• The “core” of the data set
• Uniform data subsampling isn’t enough
• Importance sampling for “coresets”
• Optimization for “coresets”
Roadmap

• Approximate Bayes review
• The “core” of the data set
• Uniform data subsampling isn’t enough
• Importance sampling for “coresets”
• Optimization for “coresets”
Frank-Wolfe

Convex optimization on a polytope $D$

[Jaggi 2013]
Frank-Wolfe

Convex optimization on a polytope $D$

- Repeat:
  1. Find gradient
  2. Find argmin point on plane in $D$
  3. Do line search between current point and argmin point

[Jaggi 2013]
Frank-Wolfe

Convex optimization on a polytope $D$

• Repeat:
  1. Find gradient
  2. Find argmin point on plane in $D$
  3. Do line search between current point and argmin point

• Convex combination of $M$ vertices after $M-1$ steps

[Jaggi 2013]
Frank-Wolfe

Convex optimization on a polytope $D$

- Repeat:
  1. Find gradient
  2. Find argmin point on plane in $D$
  3. Do line search between current point and argmin point

- Convex combination of $M$ vertices after $M-1$ steps

- Our problem: $\min_{w \in \mathbb{R}^N} \| \mathcal{L}(w) - \mathcal{L} \|$
Frank-Wolfe

Convex optimization on a polytope $D$

- Repeat:
  1. Find gradient
  2. Find argmin point on plane in $D$
  3. Do line search between current point and argmin point

- Convex combination of $M$ vertices after $M$-1 steps

- Our problem: $\min_{w \in \mathbb{R}^N} \| \mathcal{L}(w) - \mathcal{L} \|^2$

[Jaggi 2013]
Frank-Wolfe

Convex optimization on a polytope $D$

- Repeat:
  1. Find gradient
  2. Find argmin point on plane in $D$
  3. Do line search between current point and argmin point

- Convex combination of $M$ vertices after $M-1$ steps

- Our problem: \[ \min_{w \in \mathbb{R}^N} \| \mathcal{L}(w) - \mathcal{L} \|^2 \]

  \[ \text{s.t. } w \geq 0, \| w \|_0 \leq M \]
Frank-Wolfe

Convex optimization on a polytope \( D \)

- Repeat:
  1. Find gradient
  2. Find argmin point on plane in \( D \)
  3. Do line search between current point and argmin point

- Convex combination of \( M \) vertices after \( M - 1 \) steps

- Our problem:

\[
\min_{w \in \mathbb{R}^N} \| \mathcal{L}(w) - \mathcal{L} \|^2
\]

\[
\Delta^{N-1} := \left\{ w \in \mathbb{R}^N : \sum_{n=1}^{N} \sigma_n w_n = \sigma, w \geq 0 \right\}
\]
Frank-Wolfe

Convex optimization on a polytope $D$

• Repeat:
  1. Find gradient
  2. Find argmin point on plane in $D$
  3. Do line search between current point and argmin point

• Convex combination of $M$ vertices after $M - 1$ steps

• Our problem:

$$\min_{w \in \mathbb{R}^N} \| \mathcal{L}(w) - \mathcal{L} \|^2$$

$$\Delta^{N-1} := \left\{ w \in \mathbb{R}^N : \sum_{n=1}^{N} \sigma_n w_n = \sigma, w \geq 0 \right\}$$

**Thm sketch (CB).** After $M$ iterations,

$$\| \mathcal{L}(w) - \mathcal{L} \| \leq \frac{\sigma \bar{\eta}}{\sqrt{\alpha^2 M} + M}$$
Gaussian model (simulated)

- 10K pts; norms, inference: closed-form

Uniform subsampling

\[ M = 5 \]
Gaussian model (simulated)

- 10K pts; norms, inference: closed-form

Uniform subsampling

\[ M = 5 \]  \hspace{2cm}  \[ M = 50 \]  \hspace{2cm}  \[ M = 500 \]
Gaussian model (simulated)

- 10K pts; norms, inference: closed-form

Uniform subsampling

Importance sampling

\[ M = 5 \quad M = 50 \quad M = 500 \]
Gaussian model (simulated)

- 10K pts; norms, inference: closed-form

Uniform subsampling

Importance sampling

Frank-Wolfe

\[ M = 5 \quad M = 50 \quad M = 500 \]
Gaussian model (simulated)

- 10K pts; norms, inference: closed-form
Logistic regression (simulated)

- 10K data points

Uniform subsampling

Importance sampling

Frank-Wolfe

\[ M = 10 \quad M = 100 \quad M = 1000 \]
Logistic regression (simulated)

- 10K data points
- similar for Poisson regression, spherical clustering

Uniform subsampling

Importance sampling

Frank-Wolfe

\[ M = 10 \quad M = 100 \quad M = 1000 \]
Real data experiments

Logistic regression

Poisson regression

lower error

Relative CPU Time

Relative 1-Wasserstein

uniform subsampling

Frank-Wolfe
Conclusions

- *Coresets* for **scalable, automated** approx. Bayes algorithms with **error bounds on quality for finite data**

- Get more accurate with more computation investment
Conclusions

• *Coresets* for **scalable, automated** approx. Bayes algorithms with **error bounds on quality for finite data**

• Get more accurate with more computation investment

• A start

• Lots of potential improvements/directions
Conclusions

- Coresets for **scalable, automated** approx. Bayes algorithms with **error bounds on quality for finite data**

- Get more accurate with more computation investment

- A start
  - Lots of potential improvements/ directions

[Campbell, Broderick 2018]
References


  * Code: https://github.com/trevorcampbell/bayesian-coresets


Practicalities
Practicalities

- Choice of norm
Practicalities

- Choice of norm
  - E.g. (weighted) Fisher information distance

\[ \| \mathcal{L}(w) - \mathcal{L} \|_{\tilde{\pi}, F}^2 := \mathbb{E}_{\tilde{\pi}} \left[ \| \nabla \mathcal{L}(\theta) - \nabla \mathcal{L}(w, \theta) \|_2^2 \right] \]
Practicalities

• Choice of norm
  
  • E.g. (weighted) Fisher information distance

\[
\| \mathcal{L}(w) - \mathcal{L} \|^2_{\hat{\pi}, F} := \mathbb{E}_{\hat{\pi}} \left[ \| \nabla \mathcal{L}(\theta) - \nabla \mathcal{L}(w, \theta) \|^2_2 \right]
\]

• Associated inner product:

\[
\langle \mathcal{L}_n, \mathcal{L}_m \rangle_{\hat{\pi}, F} := \mathbb{E}_{\hat{\pi}} \left[ \nabla \mathcal{L}_n(\theta)^T \nabla \mathcal{L}_m(\theta) \right]
\]
Practicalities

- Choice of norm
- E.g. (weighted) Fisher information distance
  \[ \| \mathcal{L}(w) - \mathcal{L} \|^2_{\hat{\pi}, F} := \mathbb{E}_{\hat{\pi}} \left[ \| \nabla \mathcal{L}(\theta) - \nabla \mathcal{L}(w, \theta) \|^2 \right] \]
- Associated inner product:
  \[ \langle \mathcal{L}_n, \mathcal{L}_m \rangle_{\hat{\pi}, F} := \mathbb{E}_{\hat{\pi}} \left[ \nabla \mathcal{L}_n(\theta)^T \nabla \mathcal{L}_m(\theta) \right] \]
Practicalities

• Choice of norm
  • E.g. (weighted) Fisher information distance
    \[ \| \mathcal{L}(w) - \mathcal{L} \|_{\hat{\pi}, F}^2 := \mathbb{E}_{\hat{\pi}} \left[ \| \nabla \mathcal{L}(\theta) - \nabla \mathcal{L}(w, \theta) \|_2^2 \right] \]
    • Associated inner product:
      \[ \langle \mathcal{L}_n, \mathcal{L}_m \rangle_{\hat{\pi}, F} := \mathbb{E}_{\hat{\pi}} \left[ \nabla \mathcal{L}_n(\theta)^T \nabla \mathcal{L}_m(\theta) \right] \]
• Random feature projection
Practicalities

- Choice of norm
  - E.g. (weighted) Fisher information distance
    \[ \| \mathcal{L}(w) - \mathcal{L} \|^2_{\hat{\pi}, F} := \mathbb{E}_{\hat{\pi}} \left[ \| \nabla \mathcal{L}(\theta) - \nabla \mathcal{L}(w, \theta) \|^2 \right] \]

  - Associated inner product:
    \[ \langle \mathcal{L}_n, \mathcal{L}_m \rangle_{\hat{\pi}, F} := \mathbb{E}_{\hat{\pi}} \left[ \nabla \mathcal{L}_n(\theta)^T \nabla \mathcal{L}_m(\theta) \right] \]

- Random feature projection
  \[ \langle \mathcal{L}_n, \mathcal{L}_m \rangle_{\hat{\pi}, F} \approx \frac{D}{J} \sum_{j=1}^{J} (\nabla \mathcal{L}_n(\theta_j))_{d_j} (\nabla \mathcal{L}_m(\theta_j))_{d_j}, \]
  \[ d_j \overset{iid}{\sim} \text{Unif}\{1, \ldots, D\}, \theta_j \overset{iid}{\sim} \hat{\pi} \]
Practicalities

- Choice of norm
  - E.g. (weighted) Fisher information distance
    \[ \| \mathcal{L}(w) - \mathcal{L} \|_{\hat{\pi},F}^2 := \mathbb{E}_{\hat{\pi}} \left[ \| \nabla \mathcal{L}(\theta) - \nabla \mathcal{L}(w, \theta) \|_2^2 \right] \]

- Associated inner product:
  \[ \langle \mathcal{L}_n, \mathcal{L}_m \rangle_{\hat{\pi},F} := \mathbb{E}_{\hat{\pi}} \left[ \nabla \mathcal{L}_n(\theta)^T \nabla \mathcal{L}_m(\theta) \right] \]

- Random feature projection
  \[ \langle \mathcal{L}_n, \mathcal{L}_m \rangle_{\hat{\pi},F} \approx \frac{D}{J} \sum_{j=1}^{J} \left( \nabla \mathcal{L}_n(\theta_j) \right)_{d_j} \left( \nabla \mathcal{L}_m(\theta_j) \right)_{d_j}, \]
  \[ d_j \overset{iid}{\sim} \text{Unif}\{1, \ldots, D\}, \theta_j \overset{iid}{\sim} \hat{\pi} \]

**Thm sketch (CB).** With high probability and large enough \( J \), a good coreset after random feat. proj. is a good coreset for \( (\mathcal{L}_n)_{n=1}^N \)
Full pipeline

\[ N \]  
dataset size
Full pipeline

\[ \text{cost } \hat{\pi} \]

\[ N \]

dataset size
Full pipeline

random feature projection

cost $\hat{\pi}$

$N$
dataset size

$J$
projection dim
Full pipeline

random feature projection

\[ O(NJ) \]

+ cost \( \hat{\pi} \)

\( N \) dataset size

\( J \) projection dim
Full pipeline

random feature projection $O(NJ)$

Frank-Wolfe $O(NJM)$

+ cost $\hat{\pi}$

$N$ dataset size

$M$ coreset size

$J$ projection dim
Full pipeline

random feature projection

\[ O(NJ) \]

Frank-Wolfe

\[ O(NJM) \]

MCMC

\[ O(MT) \]

+ cost \( \hat{\pi} \)

\[ N \]
dataset size

\[ M \]
coreset size

\[ J \]
projection dim

\[ T \]
MCMC steps
Full pipeline

- Random feature projection: $O(NJ)$
- Frank-Wolfe: $O(NJM)$
- MCMC: $O(MT)$

+ Cost $\hat{\pi}$

- $N$: dataset size
- $M$: coreset size
- $J$: projection dim
- $T$: MCMC steps

- vs. $O(NT)$
Full pipeline

- vs. $O(NT)$
- Can make streaming, distributed

$N$: dataset size
$M$: coreset size
$J$: projection dim
$T$: MCMC steps

+ cost $\hat{\pi}$

random feature projection
Frank-Wolfe
MCMC