Fusion without diffusions

Ryan Chan

Murray Pollock [1], Gareth Roberts [1], Hongsheng Dai [2]

[1]University of Warwick

[2]University of Essex

24 March, 2020

Outline

Divide-and-conquer paradigm

We wish to carry out inferences based on subsets of data and then combine inferences. But why?

- Big Data
- Inference under privacy settings
- Combining views of multiple experts on a topic

Divide-and-conquer paradigm

We wish to carry out inferences based on subsets of data and then combine inferences. But why?

- Big Data
- Inference under privacy settings
- Combining views of multiple experts on a topic

Fusion Problem

Target of interest:

$$f(\boldsymbol{x}) \propto f_1(\boldsymbol{x}) \cdots f_C(\boldsymbol{x}) = \prod_{c=1}^{C} f_c(\boldsymbol{x})$$

where C is the number of cores / experts and f_c are *sub-posteriors*

- Many useful methods exist, but all involve approximation, e.g. Consensus Monte Carlo (Scott et al., 2016)

Fusion Problem

Target of interest:

$$f(\mathbf{x}) \propto f_1(\mathbf{x}) \cdots f_C(\mathbf{x}) = \prod_{c=1}^{C} f_c(\mathbf{x})$$

where C is the number of cores / experts and f_c are *sub-posteriors*

- Many useful methods exist, but all involve approximation, e.g. Consensus Monte Carlo (Scott et al., 2016)

Rejection sampler for $f(x) \propto \sqrt{f_1 f_2}$

Consider d-dimensional target density:

$$f(\boldsymbol{x}) \propto \sqrt{f_1(\boldsymbol{x}) \cdot f_2(\boldsymbol{x})}$$

Interested in expectations of the form:

$$\begin{aligned}\mathbb{E}_{\sqrt{f_1 f_2}}[h(X)] = \mathbb{E}_{\sqrt{f_1 f_2}}[h(X)] \\ := \int h(\boldsymbol{x}) \cdot \sqrt{f_1(\boldsymbol{x}) \cdot f_2(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x}\end{aligned}$$

Rejection sampler for $f(x) \propto \sqrt{f_1 f_2}$

Consider d-dimensional target density:

$$f(\boldsymbol{x}) \propto \sqrt{f_1(\boldsymbol{x}) \cdot f_2(\boldsymbol{x})}$$

Interested in expectations of the form:

$$\begin{aligned}\mathbb{E}_{\sqrt{f_1 f_2}}[h(X)] = \mathbb{E}_{\sqrt{f_1 f_2}}[h(X)] \\ := \int h(\boldsymbol{x}) \cdot \sqrt{f_1(\boldsymbol{x}) \cdot f_2(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x}\end{aligned}$$

Rejection sampler for $f(x) \propto \sqrt{f_1 f_2}$

We can construct a rejection sampler, since:

$$\mathbb{E}_{\sqrt{f_1 f_2}}[h(X)] := \int h(\boldsymbol{x}) \cdot \sqrt{f_1(\boldsymbol{x}) \cdot f_2(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x}$$
$$= \int h \cdot \sqrt{f_1 \cdot f_2} \cdot \left[\frac{f_1 \vee f_2}{f_1 \vee f_2}\right] \cdot \left[\frac{f_1 + f_2}{f_1 + f_2}\right] \, \mathrm{d}\boldsymbol{x}$$
$$= \int h \cdot \left[\frac{\sqrt{f_1 \cdot f_2}}{f_1 \vee f_2}\right] \cdot \left[\frac{f_1 \vee f_2}{f_1 + f_2}\right] \cdot \left[f_1 + f_2\right] \, \mathrm{d}\boldsymbol{x}$$

where $x \vee y = \max\{x, y\}$

Rejection sampler for $f(x) \propto \sqrt{f_1 f_2}$

We can construct a rejection sampler, since:

$$\mathbb{E}_{\sqrt{f_1 f_2}}[h(X)] := \int h(\mathbf{x}) \cdot \sqrt{f_1(\mathbf{x}) \cdot f_2(\mathbf{x})} \, \mathrm{d}\mathbf{x}$$

$$= \int h \cdot \sqrt{f_1 \cdot f_2} \cdot \left[\frac{f_1 \vee f_2}{f_1 \vee f_2}\right] \cdot \left[\frac{f_1 + f_2}{f_1 + f_2}\right] \, \mathrm{d}\mathbf{x}$$

$$= \int h \cdot \left[\frac{\sqrt{f_1 \cdot f_2}}{f_1 \vee f_2}\right] \cdot \left[\frac{f_1 \vee f_2}{f_1 + f_2}\right] \cdot \left[f_1 + f_2\right] \, \mathrm{d}\mathbf{x}$$

where $x \vee y = \max\{x, y\}$

Rejection sampler for $f(x) \propto \sqrt{f_1 f_2}$

We can construct a rejection sampler, since:

$$\begin{aligned}
\mathbb{E}_{\sqrt{f_1 f_2}}[h(X)] &:= \int h(\boldsymbol{x}) \cdot \sqrt{f_1(\boldsymbol{x}) \cdot f_2(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x} \\
&= \int h \cdot \sqrt{f_1 \cdot f_2} \cdot \left[\frac{f_1 \vee f_2}{f_1 \vee f_2}\right] \cdot \left[\frac{f_1 + f_2}{f_1 + f_2}\right] \, \mathrm{d}\boldsymbol{x} \\
&= \int h \cdot \left[\frac{\sqrt{f_1 \cdot f_2}}{f_1 \vee f_2}\right] \cdot \left[\frac{f_1 \vee f_2}{f_1 + f_2}\right] \cdot \left[f_1 + f_2\right] \, \mathrm{d}\boldsymbol{x}
\end{aligned}$$

where $x \vee y = \max\{x, y\}$

Rejection sampler for $f(x) \propto \sqrt{f_1 f_2}$

Therefore,

$$\mathbb{E}_f[h(X)] := \int h(\boldsymbol{x}) \cdot \sqrt{f_1(\boldsymbol{x}) \cdot f_2(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x}$$

$$= \mathbb{E}_{f_1+f_2}\left[h(X) \cdot \left[\frac{\sqrt{f_1(X) \cdot f_2(X)}}{f_1(X) \vee f_2(X)}\right] \cdot \left[\frac{f_1(X) \vee f_2(X)}{f_1(X) + f_2(X)}\right]\right]$$

$$= \mathbb{E}_{f_1+f_2}\left[h(X) \cdot \rho_1(X) \cdot \rho_2(X)\right] \quad (1)$$

where $x \vee y = \max\{x, y\}$, $\rho_1(X) = \frac{f_1(X) \vee f_2(X)}{f_1(X)+f_2(X)} \in [0, 1]$ and $\rho_2(X) = \frac{\sqrt{f_1(X) \cdot f_2(X)}}{f_1(X) \vee f_2(X)} \in [0, 1]$

Rejection sampler for $f(x) \propto \sqrt{f_1 f_2}$

Therefore,

$$\begin{aligned}
\mathbb{E}_f[h(X)] &:= \int h(\boldsymbol{x}) \cdot \sqrt{f_1(\boldsymbol{x}) \cdot f_2(\boldsymbol{x})}\ \mathrm{d}\boldsymbol{x} \\
&= \mathbb{E}_{f_1+f_2}\left[h(X) \cdot \left[\frac{\sqrt{f_1(X) \cdot f_2(X)}}{f_1(X) \vee f_2(X)}\right] \cdot \left[\frac{f_1(X) \vee f_2(X)}{f_1(X) + f_2(X)}\right]\right] \\
&= \mathbb{E}_{f_1+f_2}\left[h(X) \cdot \rho_1(X) \cdot \rho_2(X)\right] \qquad (1)
\end{aligned}$$

where $x \vee y = \max\{x, y\}$, $\rho_1(X) = \frac{f_1(X) \vee f_2(X)}{f_1(X) + f_2(X)} \in [0, 1]$ and $\rho_2(X) = \frac{\sqrt{f_1(X) \cdot f_2(X)}}{f_1(X) \vee f_2(X)} \in [0, 1]$

Rejection sampler for $f(x) \propto \sqrt{f_1 f_2}$

Result: a rejection sampler for $f \propto \sqrt{f_1 f_2}$ with proposal $f_1 + f_2$ and acceptance probability:

$$\rho_1(X) \cdot \rho_2(X) = \frac{\sqrt{f_1(X) \cdot f_2(X)}}{f_1(X) + f_2(X)} \tag{2}$$

The $\sqrt{f_1 f_2}$ Algorithm

The algorithm for simulating from $f \propto \sqrt{f_1 f_2}$ proceeds as follows:

1. Simulate $X \sim f_1 + f_2$
2. Accept X with probability $\rho_1(X) \cdot \rho_2(X) = \frac{\sqrt{f_1(X) \cdot f_2(X)}}{f_1(X)+f_2(X)}$, else return to Step 1

Note: can target $f(x) \propto f_1 f_2$ by simply simulating from f_1^2 and f_2^2

The $\sqrt{f_1 f_2}$ Algorithm

The algorithm for simulating from $f \propto \sqrt{f_1 f_2}$ proceeds as follows:

1. Simulate $X \sim f_1 + f_2$
2. Accept X with probability $\rho_1(X) \cdot \rho_2(X) = \frac{\sqrt{f_1(X) \cdot f_2(X)}}{f_1(X)+f_2(X)}$, else return to Step 1

Note: can target $f(x) \propto f_1 f_2$ by simply simulating from f_1^2 and f_2^2

The $\sqrt{f_1 f_2}$ Algorithm

The algorithm for simulating from $f \propto \sqrt{f_1 f_2}$ proceeds as follows:

1. Simulate $X \sim f_1 + f_2$
2. Accept X with probability $\rho_1(X) \cdot \rho_2(X) = \frac{\sqrt{f_1(X) \cdot f_2(X)}}{f_1(X) + f_2(X)}$, else return to Step 1

Note: can target $f(x) \propto f_1 f_2$ by simply simulating from f_1^2 and f_2^2

The $\sqrt{f_1 f_2}$ Algorithm

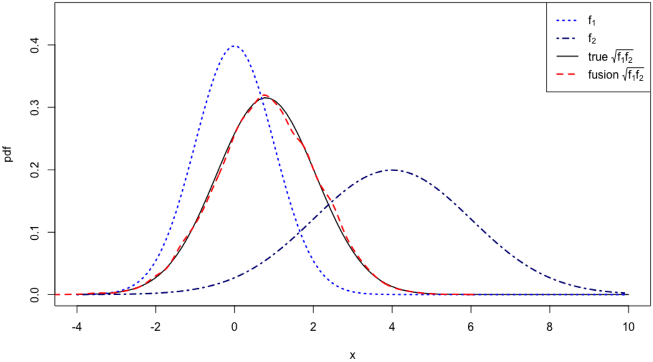
The algorithm for simulating from $f \propto \sqrt{f_1 f_2}$ proceeds as follows:

1. Simulate $X \sim f_1 + f_2$
2. Accept X with probability $\rho_1(X) \cdot \rho_2(X) = \frac{\sqrt{f_1(X) \cdot f_2(X)}}{f_1(X) + f_2(X)}$, else return to Step 1

Note: can target $f(x) \propto f_1 f_2$ by simply simulating from f_1^2 and f_2^2

A simple toy example

Consider $f \propto \sqrt{f_1 f_2}$, where $f_1 \sim \mathcal{N}(0,1)$ and $f_2 \sim \mathcal{N}(4,2)$.



Case where f_1 and f_2 are unnormalised

If f_1 and f_2 are unnormalised, then simulation from $f_1 + f_2$ is non-trivial

Suppose $\boldsymbol{y}_1 \sim f_1(\boldsymbol{y})$ and $\boldsymbol{y}_2 \sim f_2(\boldsymbol{y})$. Let $U_1, U_2, U_3 \overset{\text{i.i.d}}{\sim} U[0,1]$ are independent $U[0,1]$ variables. Define the following events:

$$\mathcal{F}_1 = \{U_1 < f_2(\boldsymbol{y}_1)/f_1(\boldsymbol{y}_1)\}$$
$$\mathcal{F}_2 = \{U_2 < f_1(\boldsymbol{y}_2)/f_2(\boldsymbol{y}_2)\}$$

Case where f_1 and f_2 are unnormalised

If f_1 and f_2 are unnormalised, then simulation from $f_1 + f_2$ is non-trivial

Suppose $\boldsymbol{y}_1 \sim f_1(\boldsymbol{y})$ and $\boldsymbol{y}_2 \sim f_2(\boldsymbol{y})$. Let $U_1, U_2, U_3 \overset{\text{i.i.d}}{\sim} U[0,1]$ are independent $U[0,1]$ variables. Define the following events:

$$\mathcal{F}_1 = \{U_1 < f_2(\boldsymbol{y}_1)/f_1(\boldsymbol{y}_1)\}$$
$$\mathcal{F}_2 = \{U_2 < f_1(\boldsymbol{y}_2)/f_2(\boldsymbol{y}_2)\}$$

Case where f_1 and f_2 are unnormalised

Define $\boldsymbol{y}^*$ and a $0-1$ indicator I as

$$(\boldsymbol{y}^*, I) = \begin{cases} (\boldsymbol{y}^* = \boldsymbol{y}_1, I = 1) & \text{if } \bar{\mathcal{F}}_1 \cap \mathcal{F}_2 \cap \{U_3 \leq \frac{1}{2}\} \text{ is true;} \\ (\boldsymbol{y}^* = \boldsymbol{y}_2, I = 1) & \text{if } \mathcal{F}_1 \cap \bar{\mathcal{F}}_2 \cap \{U_3 \leq \frac{1}{2}\} \text{ is true;} \\ (\boldsymbol{y}^* = \boldsymbol{y}_1, I = 1) & \text{if } \mathcal{F}_1 \cap \mathcal{F}_2 \cap \{U_3 \leq \frac{1}{2}\} \text{ is true;} \\ (\boldsymbol{y}^* = \boldsymbol{y}_2, I = 1) & \text{if } \mathcal{F}_1 \cap \mathcal{F}_2 \cap \{U_3 > \frac{1}{2}\} \text{ is true;} \\ (\boldsymbol{y}^* = \cdot, I = 0) & \text{otherwise} \end{cases} \tag{3}$$

Lemma

Conditional on $I = 1$, $\boldsymbol{y}^*$ follows the distribution with density proportional to $f_1(\boldsymbol{y}) + f_2(\boldsymbol{y})$.

Case where f_1 and f_2 are unnormalised

Define $\boldsymbol{y}^*$ and a $0-1$ indicator I as

$$(\boldsymbol{y}^*, I) = \begin{cases} (\boldsymbol{y}^* = \boldsymbol{y}_1, I = 1) & \text{if } \bar{\mathcal{F}}_1 \cap \mathcal{F}_2 \cap \{U_3 \leq \frac{1}{2}\} \text{ is true;} \\ (\boldsymbol{y}^* = \boldsymbol{y}_2, I = 1) & \text{if } \mathcal{F}_1 \cap \bar{\mathcal{F}}_2 \cap \{U_3 \leq \frac{1}{2}\} \text{ is true;} \\ (\boldsymbol{y}^* = \boldsymbol{y}_1, I = 1) & \text{if } \mathcal{F}_1 \cap \mathcal{F}_2 \cap \{U_3 \leq \frac{1}{2}\} \text{ is true;} \\ (\boldsymbol{y}^* = \boldsymbol{y}_2, I = 1) & \text{if } \mathcal{F}_1 \cap \mathcal{F}_2 \cap \{U_3 > \frac{1}{2}\} \text{ is true;} \\ (\boldsymbol{y}^* = \cdot, I = 0) & \text{otherwise} \end{cases} \tag{3}$$

Lemma

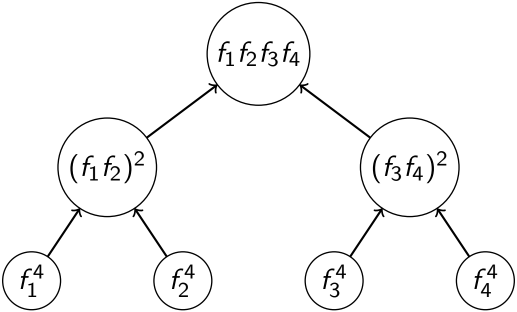
Conditional on $I = 1$, $\boldsymbol{y}^*$ follows the distribution with density proportional to $f_1(\boldsymbol{y}) + f_2(\boldsymbol{y})$.

Example

(fill in example for Bayesian Logistic Regression with 7 coefficients)

Extending to more than 2 sub-posteriors

We can adopt a *hierarchical* approach to fusion, e.g. for $f \propto f_1 f_2 f_3 f_4$:



Problem of conflicting sub-posteriors

We can rewrite the algorithm for simulating from $f \propto f_1 f_2$ as follows:

1. Simulate $X \sim f_1 + f_2$
2. With probability $\rho_1(X) = \frac{\max\{f_1(X), f_2(X)\}}{(f_1(X)+f_2(X))}$, continue, else return to Step 1
3. With probability $\rho_2(X) = \frac{\sqrt{f_1(X) f_2(X)}}{\max\{f_1(X), f_2(X)\}}$, accept X as a sample from f, else return to Step 1

Problem of conflicting sub-posteriors

We can rewrite the algorithm for simulating from $f \propto f_1 f_2$ as follows:

1. Simulate $X \sim f_1 + f_2$
2. With probability $\rho_1(X) = \frac{\max\{f_1(X), f_2(X)\}}{(f_1(X)+f_2(X))}$, continue, else return to Step 1
3. With probability $\rho_2(X) = \frac{\sqrt{f_1(X) f_2(X)}}{\max\{f_1(X), f_2(X)\}}$, accept X as a sample from f, else return to Step 1

Problem of conflicting sub-posteriors

We can rewrite the algorithm for simulating from $f \propto f_1 f_2$ as follows:

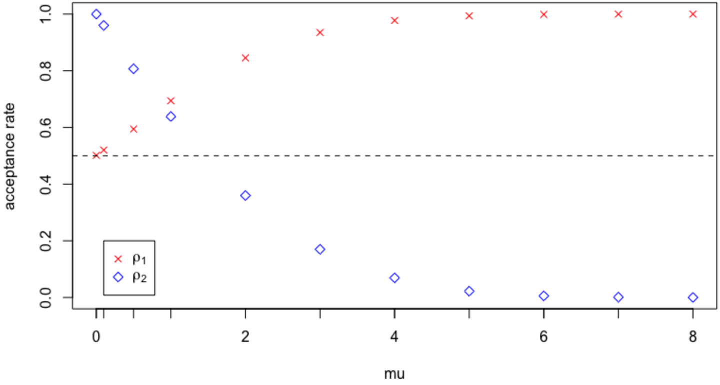
1. Simulate $X \sim f_1 + f_2$
2. With probability $\rho_1(X) = \frac{\max\{f_1(X), f_2(X)\}}{(f_1(X)+f_2(X))}$, continue, else return to Step 1
3. With probability $\rho_2(X) = \frac{\sqrt{f_1(X) f_2(X)}}{\max\{f_1(X), f_2(X)\}}$, accept X as a sample from f, else return to Step 1

Problem of conflicting sub-posteriors

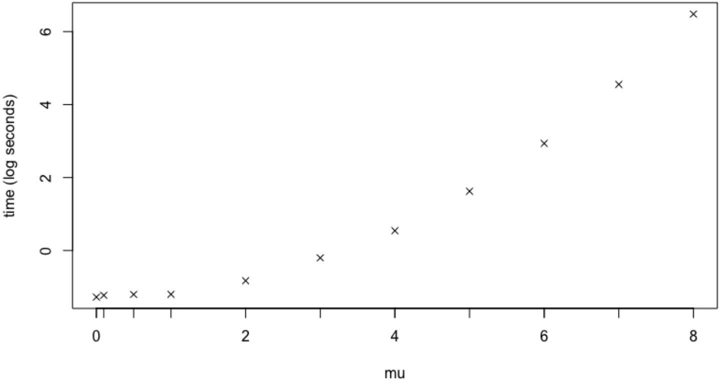
We can rewrite the algorithm for simulating from $f \propto f_1 f_2$ as follows:

1. Simulate $X \sim f_1 + f_2$
2. With probability $\rho_1(X) = \frac{\max\{f_1(X), f_2(X)\}}{(f_1(X)+f_2(X))}$, continue, else return to Step 1
3. With probability $\rho_2(X) = \frac{\sqrt{f_1(X) f_2(X)}}{\max\{f_1(X), f_2(X)\}}$, accept X as a sample from f, else return to Step 1

Problem of conflicting sub-posteriors



Problem of conflicting sub-posteriors



Problem of large data-sizes

Algorithm has $\mathcal{O}(n)$ per iteration cost

Looking to use unbiased estimators for the acceptance probability, since:

$$\rho_1(X) \cdot \rho_2(X) = \frac{\sqrt{f_1(X) \cdot f_2(X)}}{f_1(X) + f_2(X)} = \sqrt{\frac{f_1(X)}{f_1(X) + f_2(X)}} \cdot \sqrt{\frac{f_2(X)}{f_1(X) + f_2(X)}}$$

Problem of large data-sizes

Algorithm has $\mathcal{O}(n)$ per iteration cost
Looking to use unbiased estimators for the acceptance probability, since:

$$\rho_1(X) \cdot \rho_2(X) = \frac{\sqrt{f_1(X) \cdot f_2(X)}}{f_1(X) + f_2(X)}$$
$$= \sqrt{\frac{f_1(X)}{f_1(X) + f_2(X)}} \cdot \sqrt{\frac{f_2(X)}{f_1(X) + f_2(X)}}$$

Conclusion and ongoing work

- We've developed a simple rejection sampling algorithm that allows for perfect simulation from $f \propto \sqrt{f_1 f_2}$ by means of simulating from f_1 and f_2
  - However, currently is only useful for small examples
- There is ongoing work on *Monte Carlo Fusion* (which will be spoken about in the next talk by Gareth), which is more suitable for harder cases

Conclusion and ongoing work

- We've developed a simple rejection sampling algorithm that allows for perfect simulation from $f \propto \sqrt{f_1 f_2}$ by means of simulating from f_1 and f_2
  - However, currently is only useful for small examples
- There is ongoing work on *Monte Carlo Fusion* (which will be spoken about in the next talk by Gareth), which is more suitable for harder cases

Conclusion and ongoing work

- We've developed a simple rejection sampling algorithm that allows for perfect simulation from $f \propto \sqrt{f_1 f_2}$ by means of simulating from f_1 and f_2
  - However, currently is only useful for small examples
- There is ongoing work on *Monte Carlo Fusion* (which will be spoken about in the next talk by Gareth), which is more suitable for harder cases

References

- Dai, H., Pollock, M., and Roberts, G. (2019). Monte Carlo Fusion. *Journal of Applied Probability*, 56(1):174-191.
- Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and Mc-Culloch, R. E. (2016). Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management, 11(2):78–88.*