Automated Variational Inference

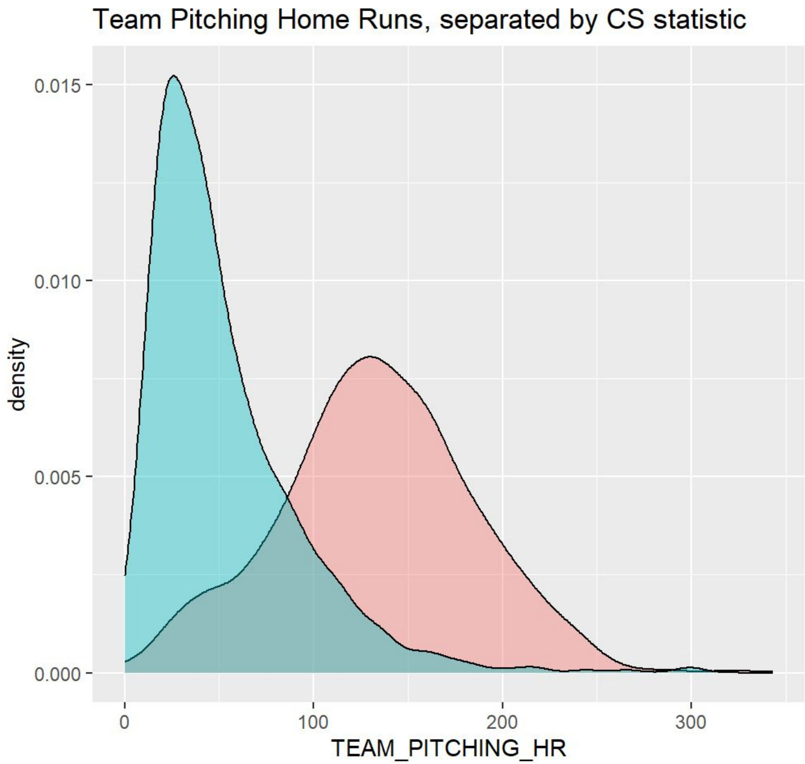
Daniela Toader & Radu Gaghi

TU Delft

Introduction

- Probabilistic **programs** define **p**
- Running the program gives the **prior p(x)**
- We are interested in the **posterior p(y | x)**
- Estimating **marginal likelihood p(x | y)** precisely is difficult

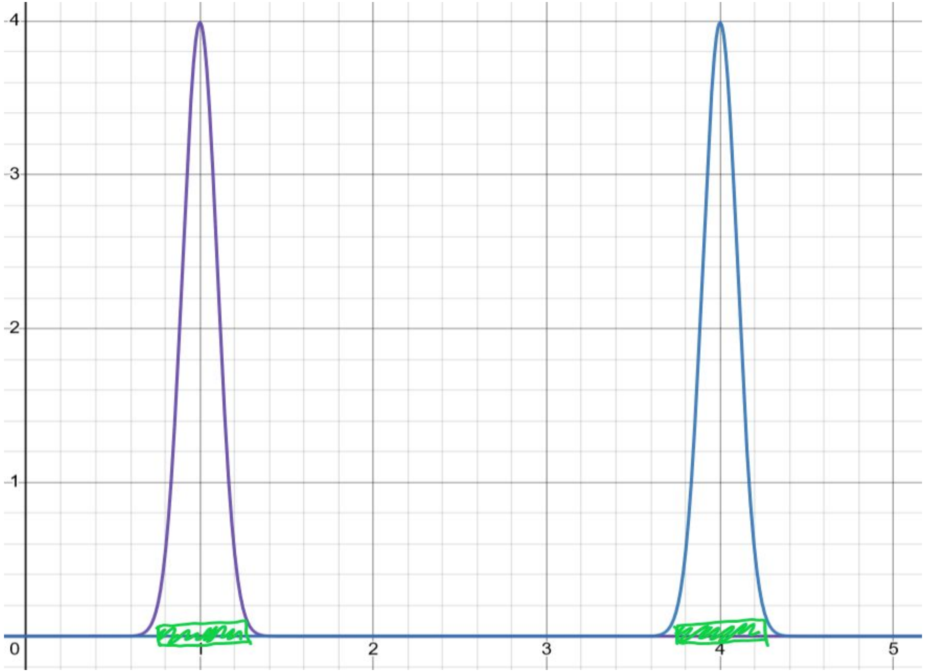
Why is sampling p(x | y) difficult?



a)

Source: https://rpubs.com/hillt5/blog2_621

Why is sampling p(x | y) difficult?



b)

Introduction

- What if we could "build" / "approximate" $p(x \mid y)$ in some other way?

TU Delft

Mean Field Approximation

- The probability of a trace is given by:

$$p(x) = \prod_{t=1}^{T} p_t(x_t \mid \psi_t(h_t))$$

where h_t is the history $(x_1, \ldots, x_{t-1})$ of the program up to ERP t

TUDelft

Mean Field Approximation

- We approximate p(x) with a simpler $p_\theta(x)$
- We use θ to parameterize p_θ, then we learn θ

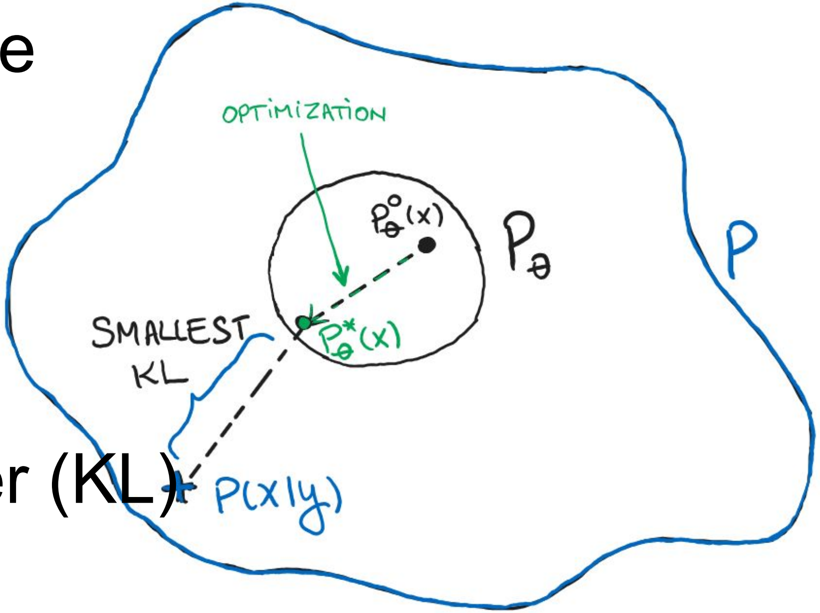
$$p(x) = \prod_{t=1}^{T} p_t(x_t \mid \psi_t(h_t))$$

$$p_\theta(x) = \prod_{t=1}^{T} p_\theta(x_t|\theta_t)$$

TU Delft

Mean Field Approximation

- Analytically intractable
- Stochastic Gradient Descent is our friend
- Enter Kullback-Leibler (KL) Divergence!



KL Divergence

- Measure of distance between distributions
- Essentially our reward function

$$
\begin{aligned}
KL(p_\theta, p(x|y)) &= \int_x p_\theta(x) \log\left(\frac{p_\theta(x)}{p(x|y)}\right) \\
&= \overset{\text{Bayes' rule}}{\boxed{\int_x p_\theta(x) \log\left(\frac{p_\theta(x)}{p(y|x)p(x)}\right)}} + \log p(y) = \\
&= \underbrace{\boxed{-L(\theta)}}_{\text{ELBO}} + \log p(y)
\end{aligned}
$$

TU Delft

KL Divergence and ELBO

$$KL(p_\theta, p(x|y)) \quad = \quad -L(\theta) + \log p(y)$$

where

$$L(\theta) \quad \triangleq \quad \int_x p_\theta(x) \log\left(\frac{p(y|x)p(x)}{p_\theta(x)}\right)$$

- KL ≥ 0 → - L(θ) + log(p(y)) ≥ 0
- log(p(y)) ≥ L(θ)

(constant)

TU Delft

Stochastic Gradient Optimization

$$-\nabla_\theta \, L(\theta) = \int_x \nabla_\theta \left(p_\theta(x) \log \left(\frac{p_\theta(x)}{p(y|x)p(x)} \right) \right) \quad (5)$$

product rule

$$= \int_x \nabla_\theta p_\theta(x) \left(\log \left(\frac{p_\theta(x)}{p(y|x)p(x)} \right) \right) + \int_x p_\theta(x) \left(\nabla_\theta \log(p_\theta(x)) \right) \quad (6)$$

Stochastic Gradient Optimization

- To derive the MC estimate, we use a few tricks

a) $\nabla \log p_\theta(x) = \frac{\nabla_\theta p_\theta(x)}{p_\theta(x)}$

b) $\int_x p_\theta(x) \nabla \log p_\theta(x) = 0$

Stochastic Gradient Optimization

$$
\begin{aligned}
-\nabla_\theta L(\theta) &= \int_x \nabla_\theta p_\theta(x) \left(\log \left(\frac{p_\theta(x)}{p(y|x)p(x)} \right) \right) + \overbrace{\int_x p_\theta(x) \left(\nabla_\theta \log(p_\theta(x)) \right)}^{= 0} && (6) \\
&= \int_x \nabla_\theta p_\theta(x) \left(\log \left(\frac{p_\theta(x)}{p(y|x)p(x)} \right) \right) && (7) \\
&\overset{\text{apply a)}}{=} \int_x p_\theta(x) \nabla_\theta \log(p_\theta(x)) \left(\log \left(\frac{p_\theta(x)}{p(y|x)p(x)} \right) \right) && (8) \\
&= \int_x p_\theta(x) \nabla_\theta \log(p_\theta(x)) \left(\log \left(\frac{p_\theta(x)}{p(y|x)p(x)} \right) \boxed{+ K} \right) && (9) \\
&\boxed{\approx \frac{1}{N} \sum_{x^j} \nabla_\theta \log p_\theta(x^j) \left(\log \left(\frac{p_\theta(x^j)}{p(y|x^j)p(x^j)} \right) + K \right)} && (10)
\end{aligned}
$$

Can now use MC estimate

a) $\nabla \log p_\theta(x) = \frac{\nabla_\theta p_\theta(x)}{p_\theta(x)}$

b) $\int_x p_\theta(x) \nabla \log p_\theta(x) = 0$

Mean-field approximation

Probabilistic program A

```
M = normal();
if M>1
  mu = complex_deterministic_func( M );
  X = normal( mu );
else
  X = rand();
end;
```

Mean-Field variational program A

```
M = normal( θ1 );
if M>1
  mu = complex_deterministic_func( M );
  X = normal( θ3 );
else
  X = rand(θ4,θ5);
end;
```

Compositional Variational Inference

- Initialize θ to arbitrary value
- Sample $x_t \sim p_\theta(x_t)$
- Compute:
  - $\log(p_\theta(x_t))$
  - $\log(p(x_t \mid h_t))$
  - $R_t = \log(p(x_t \mid h_t)) - \log(p_\theta(x_t))$
  - local gradient $\Psi_t = \nabla_{\theta t} \log(p_{\theta t}(x_t))$

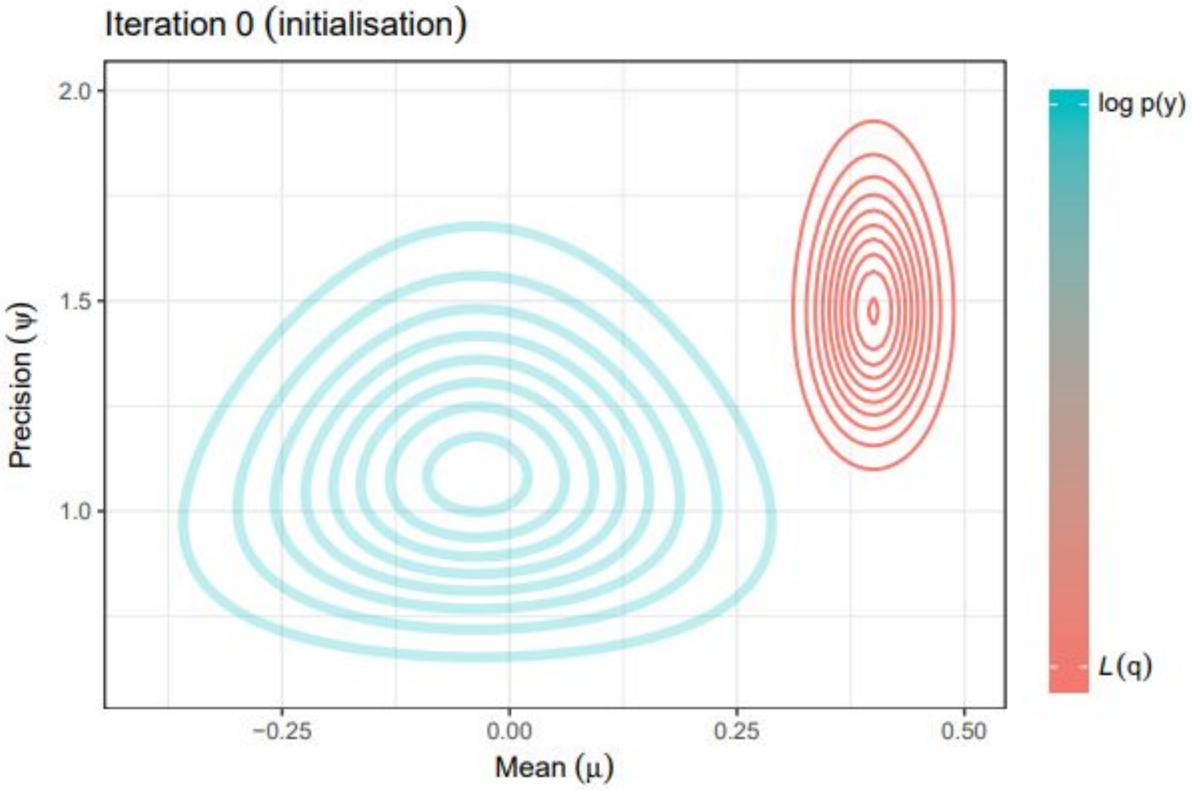
TU Delft

Computing the gain

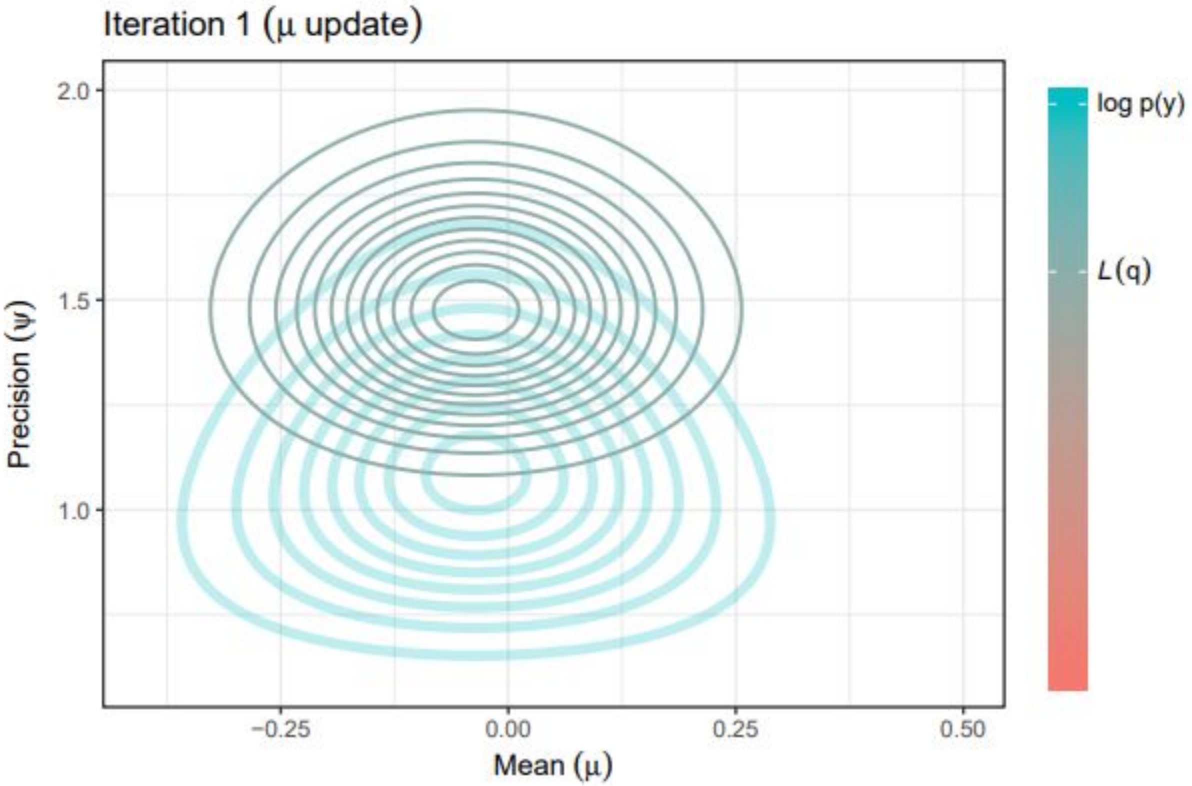
- Compute
  - log p(y | x)
  - the gain $R = \sum R_t + \log p(y \mid x) + K$
- Estimate at ERP t can be averaged over many sample traces for a more accurate estimate

$$-\nabla_\theta L(\theta) \approx \frac{1}{N} \sum_{x^j} \nabla_\theta \log p_\theta(x^j) \left(\log \left(\frac{p_\theta(x^j)}{p(y|x^j)p(x^j)} \right) + K \right) \tag{10}$$

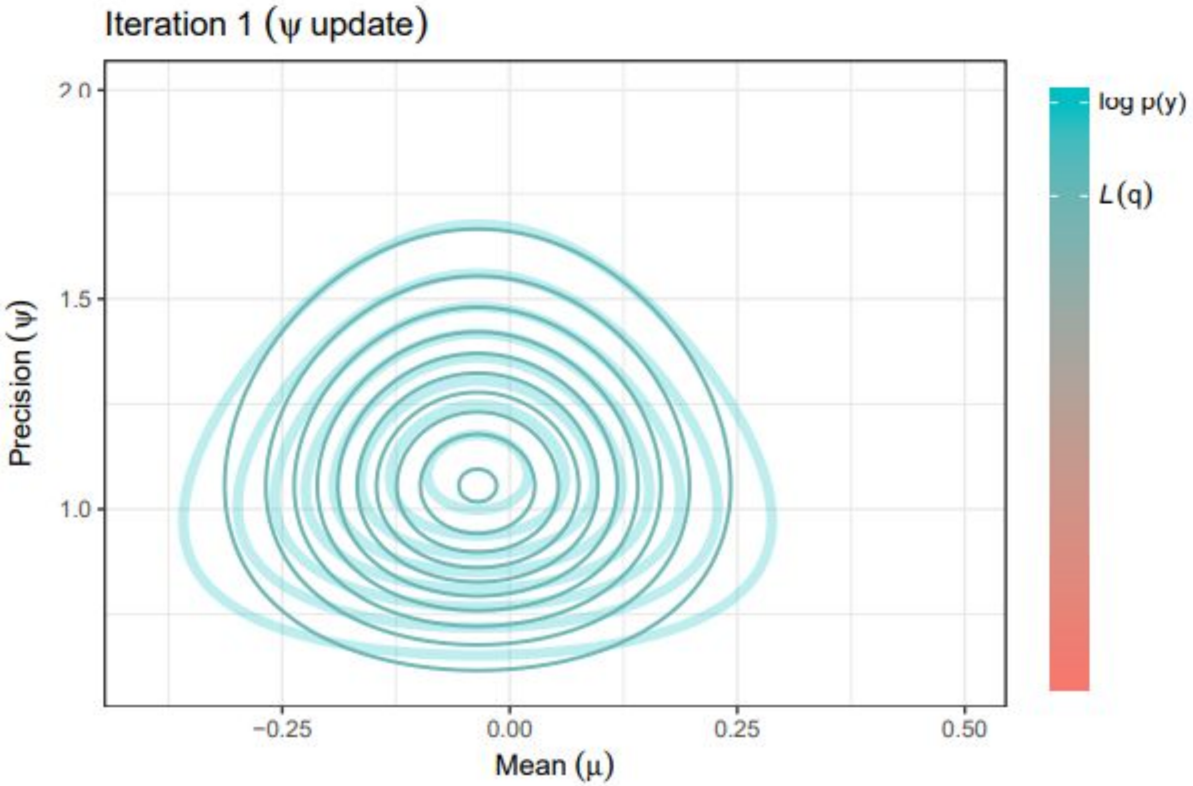
TU Delft

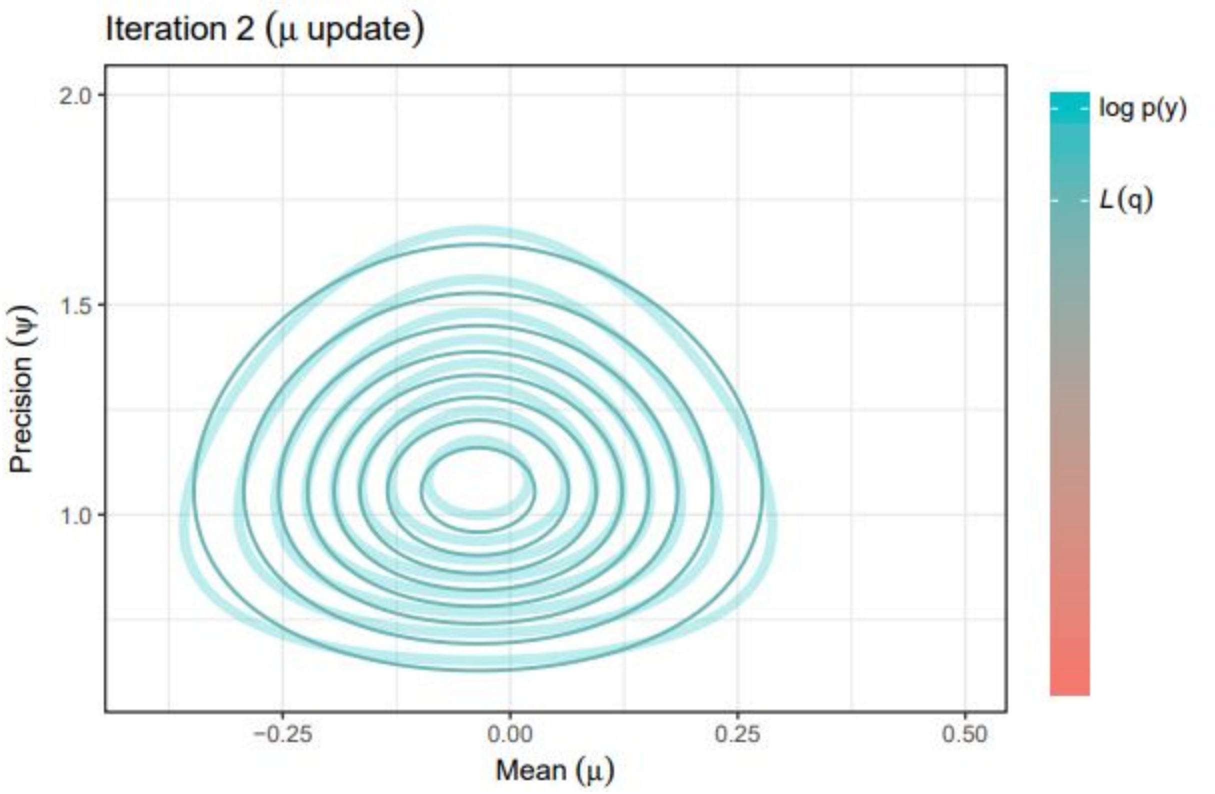


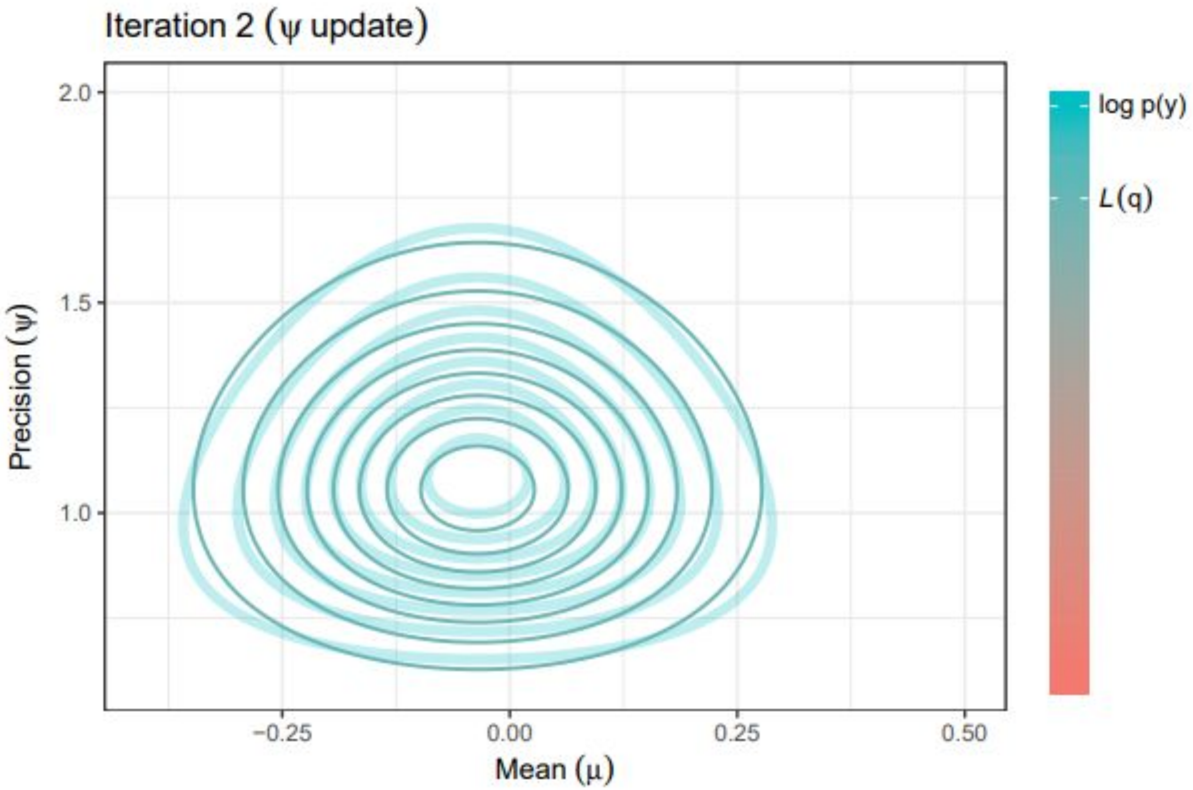
Iteration 0 (initialisation)
2.0
1.5
1.0
Precision (ψ)
−0.25
0.00
0.25
0.50
Mean (μ)
log p(y)
L(q)



Iteration 1 (μ update)
2.0
1.5
1.0
Precision (ψ)
−0.25
0.00
0.25
0.50
Mean (μ)
log p(y)
L(q)


Iteration 1 (ψ update)
Precision (ψ)
2.0
1.5
1.0
−0.25
0.00
0.25
0.50
Mean (μ)
log p(y)
L(q)


Iteration 2 (μ update)
2.0
1.5
1.0
Precision (ψ)
−0.25
0.00
0.25
0.50
Mean (μ)
log p(y)
L(q)


Iteration 2 (ψ update)
2.0
1.5
1.0
Precision (ψ)
−0.25
0.00
0.25
0.50
Mean (μ)
log p(y)
L(q)

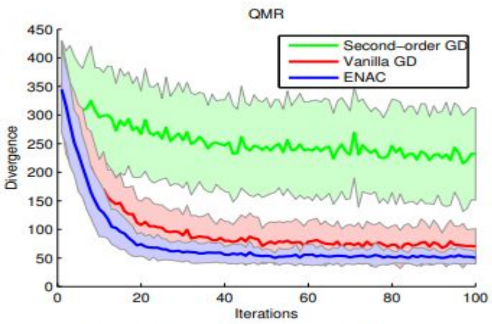
Experiments

- Which algorithm estimates the direction of the gradient best? How does SGD compare with the others?

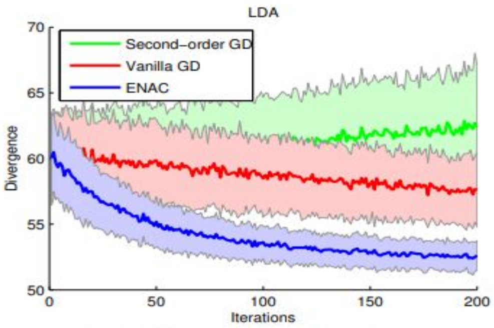
TUDelft

Experiments

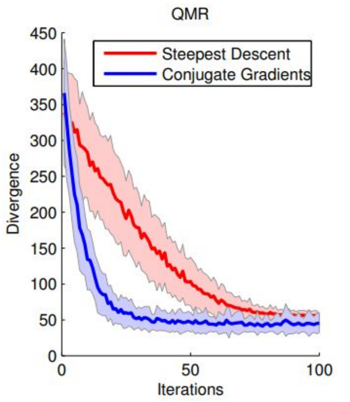
- ENAC > Vanilla GD (ours) > SOGD



(a) Results on QMR



(b) Results on LDA



(c) Steepest descent vs. conjugate gradients

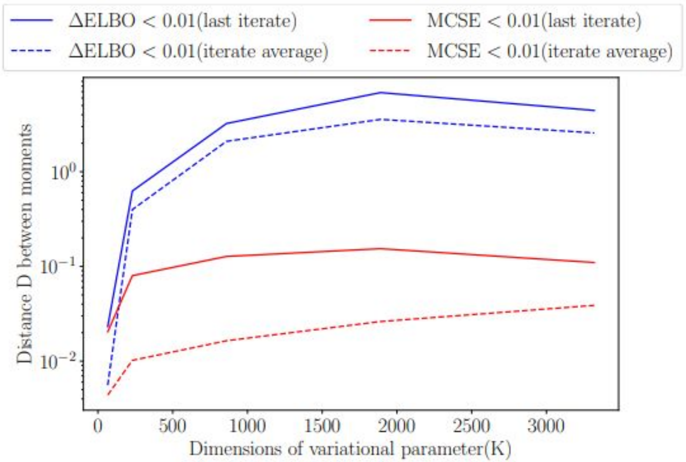
Takeaways

- Very fast approximate sampling from the posterior
- Much cheaper than using MCMC sampling

TU Delft

Related Work

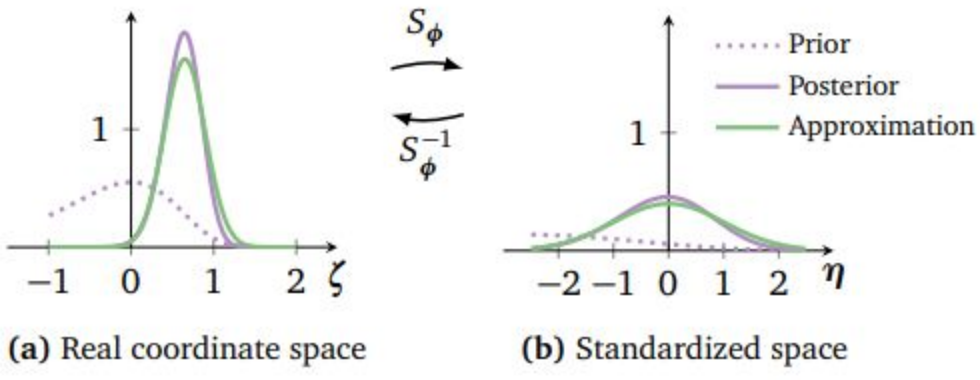
- High-dimensional posteriors are sometimes poorly approximated by SGD, a parallelizable approach is proposed [1]



$$\bar{\lambda} \equiv \frac{1}{T} \sum_{i=1}^{T} \lambda_{t+i},$$

Related Work

- Automatic Differentiation Variational Inference [2]



TU Delft

Related Work

- Rethinking Variational Inference for Probabilistic Programs with Stochastic Support [3]

Probabilistic program A

```
M = normal();
if M>1
  mu = complex_deterministic_func( M );
  X = normal( mu );
else
  X = rand();
end;
```

Questions?

References

[1] - Dhaka, A. K., Catalina, A., Andersen, M. R., Magnusson, M., Huggins, J., & Vehtari, A. (2020). Robust, accurate stochastic optimization for variational inference. Advances in Neural Information Processing Systems, 33, 10961-10973.

[2] - Ambrogioni, L., Lin, K., Fertig, E., Vikram, S., Hinne, M., Moore, D., & van Gerven, M. (2021, March). Automatic structured variational inference. In International Conference on Artificial Intelligence and Statistics (pp. 676-684). PMLR.

[3] - Reichelt, T., Ong, L., & Rainforth, T. (2022). Rethinking Variational Inference for Probabilistic Programs with Stochastic Support. Advances in Neural Information Processing Systems, 35, 15160-15175.

TU Delft