Monotonic Alpha-divergence Minimisation for Variational Inference

Kamélia Daudel

University of Oxford
kamelia.daudel@stats.ox.ac.uk

StatML CDT
09/12/2021

Joint work with Randal Douc and François Roueff

Outline

Outline

Bayesian statistics

- Compute / sample from the posterior density of the latent variables y given the data $\mathscr{D}$

$$p(y|\mathscr{D}) = \frac{p(\mathscr{D}, y)}{p(\mathscr{D})} \,.$$

- Problem : for many important models, we can only evaluate $p(y|\mathscr{D})$ up to the constant $p(\mathscr{D})$.

Bayesian statistics

- Compute / sample from the posterior density of the latent variables y given the data $\mathscr{D}$

$$p(y|\mathscr{D}) = \frac{p(\mathscr{D}, y)}{p(\mathscr{D})} \ .$$

- Problem : for many important models, we can only evaluate $p(y|\mathscr{D})$ up to the constant $p(\mathscr{D})$.

Variational Inference in a nutshell

$\rightarrow$ Variational Inference : inference is seen as an optimisation problem.

❶ Posit a *simpler* variational family $\mathcal{Q}$, where $q \in \mathcal{Q}$.

❷ Fit q to obtain the best approximation to the posterior density

$$\inf_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \ ,$$

where D is a measure of dissimilarity between the variational distribution $\mathbb{Q}$ and the posterior distribution $\mathbb{P}_{|\mathscr{D}}$

Typically, D : exclusive Kullback-Leibler (KL) divergence and $\mathcal{Q}$: parametric family (e.g. Mean-field)

$$\begin{cases} D_{KL}(\mathbb{Q}||\mathbb{P}) = \int_{\mathsf{Y}} \log\left(\frac{q(y)}{p(y)}\right) q(y)\nu(\mathrm{d}y) \\ \mathcal{Q} = \{q : y \mapsto k(\theta, y) \ : \ \theta \in \mathsf{T}\} \end{cases}$$

Question : How to choose D and $\mathcal{Q}$?

- Can we select alternative/more general D?
- Can we design more expressive variational families $\mathcal{Q}$?

Variational Inference in a nutshell

$\rightarrow$ Variational Inference : inference is seen as an optimisation problem.

❶ Posit a *simpler* variational family $\mathcal{Q}$, where $q \in \mathcal{Q}$.

❷ Fit q to obtain the best approximation to the posterior density

$$\inf_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \ ,$$

where D is a measure of dissimilarity between the variational distribution $\mathbb{Q}$ and the posterior distribution $\mathbb{P}_{|\mathscr{D}}$

Typically, D : exclusive Kullback-Leibler (KL) divergence and $\mathcal{Q}$: parametric family (e.g. Mean-field)

$$\begin{cases} D_{KL}(\mathbb{Q}||\mathbb{P}) = \int_{\mathsf{Y}} \log\left(\frac{q(y)}{p(y)}\right) q(y)\nu(\mathrm{d}y) \\ \mathcal{Q} = \{q : y \mapsto k(\theta, y) \ : \ \theta \in \mathsf{T}\} \end{cases}$$

Question : How to choose D and $\mathcal{Q}$?

- Can we select alternative/more general D?
- Can we design more expressive variational families $\mathcal{Q}$?

Variational Inference in a nutshell

$\rightarrow$ Variational Inference : inference is seen as an optimisation problem.

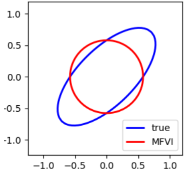
1. Posit a *simpler* variational family $\mathcal{Q}$, where $q \in \mathcal{Q}$.
2. Fit q to obtain the best approximation to the posterior density

$$\inf_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \ ,$$

where D is a measure of dissimilarity between the variational distribution $\mathbb{Q}$ and the posterior distribution $\mathbb{P}_{|\mathscr{D}}$

Typically, D : exclusive Kullback-Leibler (KL) divergence and $\mathcal{Q}$: parametric family (e.g. Mean-field)

$$\begin{cases} D_{KL}(\mathbb{Q}||\mathbb{P}) = \int_{\mathsf{Y}} \log\left(\dfrac{q(y)}{p(y)}\right) q(y)\nu(\mathrm{d}y) \\ \mathcal{Q} = \{q : y \mapsto k(\theta, y) \ : \ \theta \in \mathsf{T}\} \end{cases}$$



Question : How to choose D and $\mathcal{Q}$?

- Can we select alternative/more general D?
- Can we design more expressive variational families $\mathcal{Q}$?

Variational Inference in a nutshell

→ Variational Inference : inference is seen as an optimisation problem.

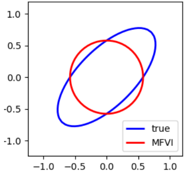
1. Posit a *simpler* variational family $\mathcal{Q}$, where $q \in \mathcal{Q}$.
2. Fit q to obtain the best approximation to the posterior density

$$\inf_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \ ,$$

where D is a measure of dissimilarity between the variational distribution $\mathbb{Q}$ and the posterior distribution $\mathbb{P}_{|\mathscr{D}}$

Typically, D : exclusive Kullback-Leibler (KL) divergence and $\mathcal{Q}$: parametric family (e.g. Mean-field)

$$\begin{cases} D_{KL}(\mathbb{Q}||\mathbb{P}) = \int_{\mathsf{Y}} \log\left(\frac{q(y)}{p(y)}\right) q(y)\nu(\mathrm{d}y) \\ \mathcal{Q} = \{q : y \mapsto k(\theta, y) \ : \ \theta \in \mathsf{T}\} \end{cases}$$



Question : How to choose D and $\mathcal{Q}$?

- Can we select alternative/more general D?
- Can we design more expressive variational families $\mathcal{Q}$?

Variational Inference in a nutshell

$\rightarrow$ Variational Inference : inference is seen as an optimisation problem.

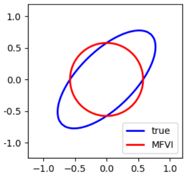
1. Posit a *simpler* variational family $\mathcal{Q}$, where $q \in \mathcal{Q}$.
2. Fit q to obtain the best approximation to the posterior density

$$\inf_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \ ,$$

where D is a measure of dissimilarity between the variational distribution $\mathbb{Q}$ and the posterior distribution $\mathbb{P}_{|\mathscr{D}}$

Typically, D : exclusive Kullback-Leibler (KL) divergence and $\mathcal{Q}$: parametric family (e.g. Mean-field)

$$\begin{cases} D_{KL}(\mathbb{Q}||\mathbb{P}) = \int_{\mathsf{Y}} \log\left(\dfrac{q(y)}{p(y)}\right) q(y)\nu(\mathrm{d}y) \\ \mathcal{Q} = \{q : y \mapsto k(\theta, y) \ : \ \theta \in \mathsf{T}\} \end{cases}$$



Question : How to choose D and $\mathcal{Q}$?

- Can we select alternative/more general D?
- Can we design more expressive variational families $\mathcal{Q}$?

Variational Inference in a nutshell

$\rightarrow$ Variational Inference : inference is seen as an optimisation problem.

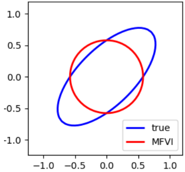
1. Posit a *simpler* variational family $\mathcal{Q}$, where $q \in \mathcal{Q}$.
2. Fit q to obtain the best approximation to the posterior density

$$\inf_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \ ,$$

where D is a measure of dissimilarity between the variational distribution $\mathbb{Q}$ and the posterior distribution $\mathbb{P}_{|\mathscr{D}}$

Typically, D : exclusive Kullback-Leibler (KL) divergence and $\mathcal{Q}$: parametric family (e.g. Mean-field)

$$\begin{cases} D_{KL}(\mathbb{Q}||\mathbb{P}) = \int_{\mathsf{Y}} \log\left(\dfrac{q(y)}{p(y)}\right) q(y)\nu(\mathrm{d}y) \\ \mathcal{Q} = \{q : y \mapsto k(\theta, y) \ : \ \theta \in \mathsf{T}\} \end{cases}$$



Question : How to choose D and $\mathcal{Q}$?

- Can we select alternative/more general D?
- Can we design more expressive variational families $\mathcal{Q}$?

Variational Inference with the α-divergence family

$(\mathsf{Y}, \mathcal{Y}, \nu)$: measured space, ν is a σ-finite measure on $(\mathsf{Y}, \mathcal{Y})$.
$\mathbb{Q}$ and $\mathbb{P}$: $\mathbb{Q} \preceq \nu$, $\mathbb{P} \preceq \nu$ with $\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\nu} = q$, $\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\nu} = p$.

α-divergence between $\mathbb{Q}$ and $\mathbb{P}$

$$D_\alpha(\mathbb{Q}||\mathbb{P}) = \int_{\mathsf{Y}} f_\alpha\left(\frac{q(y)}{p(y)}\right) p(y)\nu(\mathrm{d}y) \ ,$$

where

$$f_\alpha = \begin{cases} \frac{1}{\alpha(\alpha-1)}\left[u^\alpha - 1 - \alpha(u-1)\right], & \text{if } \alpha \in \mathbb{R} \setminus \{0,1\}\,, \\ u\log(u) + 1 - u, & \text{if } \alpha = 1 \text{ (Exclusive KL)}, \\ -\log(u) + u - 1, & \text{if } \alpha = 0 \text{ (Inclusive KL)}. \end{cases}$$

Variational Inference with the α-divergence family

$(\mathsf{Y}, \mathcal{Y}, \nu)$: measured space, ν is a σ-finite measure on $(\mathsf{Y}, \mathcal{Y})$.
$\mathbb{Q}$ and $\mathbb{P}$: $\mathbb{Q} \preceq \nu$, $\mathbb{P} \preceq \nu$ with $\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\nu} = q$, $\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\nu} = p$.

α-divergence between $\mathbb{Q}$ and $\mathbb{P}$

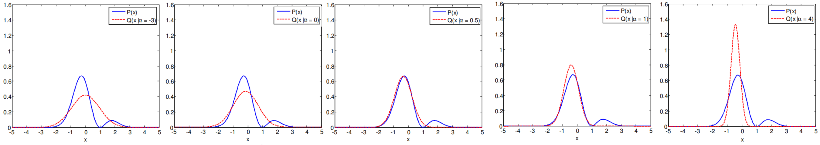
$$D_\alpha(\mathbb{Q}||\mathbb{P}) = \int_{\mathsf{Y}} f_\alpha\left(\frac{q(y)}{p(y)}\right) p(y)\nu(\mathrm{d}y) \ ,$$

where

$$f_\alpha = \begin{cases} \frac{1}{\alpha(\alpha-1)}\left[u^\alpha - 1 - \alpha(u-1)\right], & \text{if } \alpha \in \mathbb{R} \setminus \{0,1\}\,, \\ u\log(u) + 1 - u, & \text{if } \alpha = 1 \text{ (Exclusive KL)}, \\ -\log(u) + u - 1, & \text{if } \alpha = 0 \text{ (Inclusive KL)}. \end{cases}$$

❶ A flexible family of divergences...

Figure: In red, the Gaussian which minimises $D_\alpha(\mathbb{Q}||\mathbb{P})$ for a varying α



$\alpha = -3$ $\alpha = 0$ $\alpha = 0.5$ $\alpha = 1$ $\alpha = 4$

Adapted from V. Cevher's lecture notes (2008) https://www.ece.rice.edu/~vc3/elec633/AlphaDivergence.pdf

Variational Inference with the α-divergence family

α-divergence between $\mathbb{Q}$ and $\mathbb{P}$

$$D_\alpha(\mathbb{Q}||\mathbb{P}) = \int_{\mathsf{Y}} f_\alpha\left(\frac{q(y)}{p(y)}\right) p(y)\nu(\mathrm{d}y) \;,$$

where

$$f_\alpha = \begin{cases} \frac{1}{\alpha(\alpha-1)}\left[u^\alpha - 1 - \alpha(u-1)\right], & \text{if } \alpha \in \mathbb{R}\setminus\{0,1\}\,, \\ u\log(u) + 1 - u, & \text{if } \alpha = 1 \text{ (Forward KL)}, \\ -\log(u) + u - 1, & \text{if } \alpha = 0 \text{ (Reverse KL)}. \end{cases}$$

❶ A flexible family of divergences...

❷ ...suitable for Variational Inference purposes...

$$\inf_{q\in\mathcal{Q}} D_\alpha(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \Leftrightarrow \inf_{q\in\mathcal{Q}} \Psi_\alpha(q;p)$$

with $\Psi_\alpha(q;p) = \int_{\mathsf{Y}} f_\alpha\left(\frac{q(y)}{p(y)}\right) p(y)\nu(\mathrm{d}y)$ and $p = p(\cdot, \mathscr{D})$

Black-box alpha divergence minimization. J. Hernandez-Lobato et al. (2016). ICML

Rényi divergence variational inference. Y. Li and R. E Turner (2016). NeurIPS

Variational inference via χ-upper bound minimization A. Dieng et al. (2017). NeurIPS

Variational Inference with the α-divergence family

α-divergence between $\mathbb{Q}$ and $\mathbb{P}$

$$D_\alpha(\mathbb{Q}||\mathbb{P}) = \int_{\mathsf{Y}} f_\alpha\left(\frac{q(y)}{p(y)}\right) p(y)\nu(\mathrm{d}y) \;,$$

where

$$f_\alpha = \begin{cases} \frac{1}{\alpha(\alpha-1)}\left[u^\alpha - 1 - \alpha(u-1)\right], & \text{if } \alpha \in \mathbb{R}\setminus\{0,1\}\,, \\ u\log(u) + 1 - u, & \text{if } \alpha = 1 \text{ (Forward KL)}, \\ -\log(u) + u - 1, & \text{if } \alpha = 0 \text{ (Reverse KL)}. \end{cases}$$

1. A flexible family of divergences...
2. ...suitable for Variational Inference purposes...

$$\inf_{q\in\mathcal{Q}} D_\alpha(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \Leftrightarrow \inf_{q\in\mathcal{Q}} \Psi_\alpha(q;p)$$

with $\Psi_\alpha(q;p) = \int_{\mathsf{Y}} f_\alpha\left(\frac{q(y)}{p(y)}\right) p(y)\nu(\mathrm{d}y)$ and $p = p(\cdot,\mathscr{D})$

Black-box alpha divergence minimization. J. Hernandez-Lobato et al. (2016). ICML

Rényi divergence variational inference. Y. Li and R. E Turner (2016). NeurIPS

Variational inference via χ-upper bound minimization A. Dieng et al. (2017). NeurIPS

Outline

Monotonic Alpha-Divergence Minimisation

Monotonic Alpha-divergence Minimisation.
K. Daudel, R. Douc and F. Roueff (2021). https://arxiv.org/abs/2103.05684

Idea : Extend the typical variational parametric family

$$\mathcal{Q} = \{y \mapsto k(\theta, y) \ : \ \theta \in \mathsf{T}\}$$

by considering the mixture model variational family

$$\mathcal{Q} = \left\{ q : y \mapsto \mu_{\boldsymbol{\lambda},\Theta} k(y) := \sum_{j=1}^{J} \lambda_j k(\theta_j, y) \ : \ \boldsymbol{\lambda} \in \mathcal{S}_J, \Theta \in \mathsf{T}^J \right\}$$

and propose an update formula for $(\boldsymbol{\lambda}, \Theta)$ that ensures a systematic decrease in the α-divergence (i.e. Ψ_α) at each step.

Conditions for a monotonic decrease

Optimisation problem

$$\inf_{\boldsymbol{\lambda} \in \mathcal{S}_J, \Theta \in \mathsf{T}^J} \Psi_\alpha(\mu_{\boldsymbol{\lambda},\Theta} k; p) \quad \text{with} \quad \Psi_\alpha(\mu_{\boldsymbol{\lambda},\Theta} k; p) = \int_{\mathsf{Y}} f_\alpha \left(\frac{\mu_{\boldsymbol{\lambda},\Theta} k(y)}{p(y)} \right) p(y) \nu(\mathrm{d}y)$$

(A1) For all $(\theta, y) \in \mathsf{T} \times \mathsf{Y}$, $k(\theta, y) > 0$, $p(y) \geqslant 0$ and $\int_{\mathsf{Y}} p(y)\nu(\mathrm{d}y) < \infty$.

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\boldsymbol{\lambda}_n, \Theta_n)_{n \geqslant 1}$ so that: $\Psi_\alpha(\mu_{\boldsymbol{\lambda}_1,\Theta_1} k; p) < \infty$ and $\forall n \geqslant 1$,

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}} \right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Weights)}$$

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Components)}$$

where $\gamma_{j,\alpha}^n(y) = k(\theta_{j,n}, y) \left(\frac{\mu_{\boldsymbol{\lambda}_n,\Theta_n} k(y)}{p(y)} \right)^{\alpha - 1}$, yields a systematic decrease in Ψ_α at each step.

Conditions for a monotonic decrease

Optimisation problem

$$\inf_{\boldsymbol{\lambda} \in \mathcal{S}_J, \Theta \in \mathsf{T}^J} \Psi_\alpha(\mu_{\boldsymbol{\lambda},\Theta} k; p) \quad \text{with} \quad \Psi_\alpha(\mu_{\boldsymbol{\lambda},\Theta} k; p) = \int_{\mathsf{Y}} f_\alpha\left(\frac{\mu_{\boldsymbol{\lambda},\Theta} k(y)}{p(y)}\right) p(y) \nu(\mathrm{d}y)$$

(A1) For all $(\theta, y) \in \mathsf{T} \times \mathsf{Y}$, $k(\theta, y) > 0$, $p(y) \geqslant 0$ and $\int_{\mathsf{Y}} p(y)\nu(\mathrm{d}y) < \infty$.

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\boldsymbol{\lambda}_n, \Theta_n)_{n \geqslant 1}$ so that: $\Psi_\alpha(\mu_{\boldsymbol{\lambda}_1,\Theta_1} k; p) < \infty$ and $\forall n \geqslant 1$,

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log\left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}}\right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Weights)}$$

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log\left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)}\right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Components)}$$

where $\gamma_{j,\alpha}^n(y) = k(\theta_{j,n}, y) \left(\frac{\mu_{\boldsymbol{\lambda}_n,\Theta_n} k(y)}{p(y)}\right)^{\alpha - 1}$, yields a systematic decrease in Ψ_α at each step.

Conditions for a monotonic decrease

Optimisation problem

$$\inf_{\boldsymbol{\lambda} \in \mathcal{S}_J, \Theta \in \mathsf{T}^J} \Psi_\alpha(\mu_{\boldsymbol{\lambda},\Theta} k; p) \quad \text{with} \quad \Psi_\alpha(\mu_{\boldsymbol{\lambda},\Theta} k; p) = \int_{\mathsf{Y}} f_\alpha \left(\frac{\mu_{\boldsymbol{\lambda},\Theta} k(y)}{p(y)} \right) p(y) \nu(\mathrm{d}y)$$

(A1) For all $(\theta, y) \in \mathsf{T} \times \mathsf{Y}$, $k(\theta, y) > 0$, $p(y) \geqslant 0$ and $\int_{\mathsf{Y}} p(y)\nu(\mathrm{d}y) < \infty$.

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\boldsymbol{\lambda}_n, \Theta_n)_{n \geqslant 1}$ so that:
$\Psi_\alpha(\mu_{\boldsymbol{\lambda}_1,\Theta_1} k; p) < \infty$ and $\forall n \geqslant 1$,

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}} \right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Weights)}$$

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Components)}$$

where $\gamma_{j,\alpha}^n(y) = k(\theta_{j,n}, y) \left(\frac{\mu_{\boldsymbol{\lambda}_n,\Theta_n} k(y)}{p(y)} \right)^{\alpha - 1}$, yields a systematic decrease in Ψ_α at each step.

Conditions for a monotonic decrease (2)

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^{n}(y) \log\left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}}\right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Weights)}$$

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^{n}(y) \log\left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)}\right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Components)}$$

1. (Weights) and (Components) permit separate/simultaneous updates
2. The dependency is simpler in (Weights)

→ (Weights) holds for $\boldsymbol{\lambda}_{n+1}$ such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^{J} \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^{n}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}, \quad j = 1 \ldots J$$

where $\eta_n \in (0,1]$ and κ is such that $(\alpha-1)\kappa \geqslant 0$

Conditions for a monotonic decrease (2)

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^{n}(y) \log \left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}} \right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Weights)}$$

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^{n}(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Components)}$$

1. (Weights) and (Components) permit separate/simultaneous updates

2. The dependency is simpler in (Weights)

   $\rightarrow$ (Weights) holds for $\boldsymbol{\lambda}_{n+1}$ such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y) \nu(\mathrm{d}y) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^{J} \lambda_{\ell,n} \left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^{n}(y) \nu(\mathrm{d}y) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \ldots J$$

   where $\eta_n \in (0, 1]$ and κ is such that $(\alpha - 1)\kappa \geqslant 0$

Conditions for a monotonic decrease (2)

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^{n}(y) \log \left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}} \right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Weights)}$$

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^{n}(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Components)}$$

1. (Weights) and (Components) permit separate/simultaneous updates
2. The dependency is simpler in (Weights)

→ (Weights) holds for $\boldsymbol{\lambda}_{n+1}$ such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y) \nu(\mathrm{d}y) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^{J} \lambda_{\ell,n} \left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^{n}(y) \nu(\mathrm{d}y) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \ldots J$$

where $\eta_n \in (0, 1]$ and κ is such that $(\alpha - 1)\kappa \geqslant 0$

Conditions for a monotonic decrease (2)

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^{n}(y) \log\left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}}\right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Weights)}$$

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^{n}(y) \log\left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)}\right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Components)}$$

1. (Weights) and (Components) permit separate/simultaneous updates
2. The dependency is simpler in (Weights)

→ (Weights) holds for $\boldsymbol{\lambda}_{n+1}$ such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^{J} \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^{n}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}, \quad j = 1 \ldots J$$

where $\eta_n \in (0,1]$ and κ is such that $(\alpha-1)\kappa \geqslant 0$

Understanding the mixture weights update

(Weights) and (Components) hold for $\boldsymbol{\lambda}_{n+1}$ and Θ_{n+1} such that:

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}\ , \quad j = 1 \ldots J$$

$$\Theta_{n+1} = \Theta_n$$

where $\eta_n \in (0,1]$ and κ is such that $(\alpha-1)\kappa \geqslant 0$

→ We recover the Power Descent algorithm from

Infinite-dimensional gradient-based descent for alpha-divergence minimisation.
K. Daudel, R. Douc and F. Portier (2021). *Ann. Statist. 49 (4) 2250 - 2270.*

Core insights :

1. The mixture weights update is gradient-based, η_n plays the role of a learning rate
2. We can improve on the Power Descent by proposing simultaneous updates for Θ with convergence guarantees!

Understanding the mixture weights update

(Weights) and (Components) hold for $\boldsymbol{\lambda}_{n+1}$ and Θ_{n+1} such that:

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}\,, \quad j = 1\dots J$$

$$\Theta_{n+1} = \Theta_n$$

where $\eta_n \in (0,1]$ and κ is such that $(\alpha-1)\kappa \geqslant 0$

$\rightarrow$ We recover the Power Descent algorithm from

Infinite-dimensional gradient-based descent for alpha-divergence minimisation.

K. Daudel, R. Douc and F. Portier (2021). *Ann. Statist. 49 (4) 2250 - 2270.*

Core insights :

1. The mixture weights update is gradient-based, η_n plays the role of a learning rate
2. We can improve on the Power Descent by proposing simultaneous updates for Θ with convergence guarantees!

Understanding the mixture weights update

(Weights) and (Components) hold for $\boldsymbol{\lambda}_{n+1}$ and Θ_{n+1} such that:

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma^n_{\ell,\alpha}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}\;, \quad j = 1\ldots J$$

$$\Theta_{n+1} = \Theta_n$$

where $\eta_n \in (0,1]$ and κ is such that $(\alpha-1)\kappa \geqslant 0$

$\rightarrow$ We recover the Power Descent algorithm from

Infinite-dimensional gradient-based descent for alpha-divergence minimisation.

K. Daudel, R. Douc and F. Portier (2021). *Ann. Statist. 49 (4) 2250 - 2270.*

Core insights :

1. The mixture weights update is gradient-based, η_n plays the role of a learning rate
2. We can improve on the Power Descent by proposing simultaneous updates for Θ with convergence guarantees!

Understanding the mixture weights update

(Weights) and (Components) hold for $\boldsymbol{\lambda}_{n+1}$ and Θ_{n+1} such that:

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma^n_{\ell,\alpha}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}\,, \quad j = 1 \ldots J$$

$$\Theta_{n+1} = \Theta_n$$

where $\eta_n \in (0,1]$ and κ is such that $(\alpha-1)\kappa \geqslant 0$

$\rightarrow$ We recover the Power Descent algorithm from

Infinite-dimensional gradient-based descent for alpha-divergence minimisation.

K. Daudel, R. Douc and F. Portier (2021). *Ann. Statist. 49 (4) 2250 - 2270.*

Core insights :

1. The mixture weights update is gradient-based, η_n plays the role of a learning rate
2. We can improve on the Power Descent by proposing simultaneous updates for Θ with convergence guarantees!

Understanding the mixture weights update

(Weights) and (Components) hold for $\boldsymbol{\lambda}_{n+1}$ and Θ_{n+1} such that:

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y) \nu(\mathrm{d}y) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^n(y) \nu(\mathrm{d}y) + (\alpha - 1)\kappa \right]^{\eta_n}} , \quad j = 1 \ldots J$$

$$\Theta_{n+1} = \Theta_n$$

where $\eta_n \in (0, 1]$ and κ is such that $(\alpha - 1)\kappa \geqslant 0$

$\rightarrow$ We recover the Power Descent algorithm from

Infinite-dimensional gradient-based descent for alpha-divergence minimisation.

K. Daudel, R. Douc and F. Portier (2021). *Ann. Statist. 49 (4) 2250 - 2270.*

Core insights :

1. The mixture weights update is gradient-based, η_n plays the role of a learning rate
2. We can improve on the Power Descent by proposing simultaneous updates for Θ with convergence guarantees!

Towards simultaneous updates

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^{n}(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Components)}$$

- Maximisation approach : for all $j = 1 \ldots J$,

$$\theta_{j,n+1} = \mathrm{argmax}_{\theta \in \mathsf{T}} \int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y) \log(k(\theta, y)) \nu(\mathrm{d}y)$$

- Gradient-based approach : for all $j = 1 \ldots J$, $\gamma_{j,n} \in (0, 1]$, $c_{j,n} > 0$

$$\theta_{j,n+1} = \theta_{j,n} - \frac{\gamma_{j,n}}{\beta_{j,n}} \nabla g_{j,n}(\theta)|_{\theta = \theta_{j,n}}$$

where $g_{j,n}$ is assumed to be $\beta_{j,n}$-smooth on $\mathsf{T} = \mathbb{R}^d$ with

$$g_{j,n}(\theta) = c_{j,n} \int_{\mathsf{Y}} \frac{\gamma_{j,\alpha}^{n}(y)}{\alpha - 1} \log \left(\frac{k(\theta, y)}{k(\theta_{j,n}, y)} \right) \nu(\mathrm{d}y) \,.$$

Towards simultaneous updates

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^{n}(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Components)}$$

- Maximisation approach : for all $j = 1 \dots J$,

$$\theta_{j,n+1} = \operatorname{argmax}_{\theta \in \mathsf{T}} \int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y) \log(k(\theta, y)) \nu(\mathrm{d}y)$$

- Gradient-based approach : for all $j = 1 \dots J$, $\gamma_{j,n} \in (0, 1]$, $c_{j,n} > 0$

$$\theta_{j,n+1} = \theta_{j,n} - \frac{\gamma_{j,n}}{\beta_{j,n}} \nabla g_{j,n}(\theta)|_{\theta = \theta_{j,n}}$$

where $g_{j,n}$ is assumed to be $\beta_{j,n}$-smooth on $\mathsf{T} = \mathbb{R}^d$ with

$$g_{j,n}(\theta) = c_{j,n} \int_{\mathsf{Y}} \frac{\gamma_{j,\alpha}^{n}(y)}{\alpha - 1} \log \left(\frac{k(\theta, y)}{k(\theta_{j,n}, y)} \right) \nu(\mathrm{d}y) \, .$$

Towards simultaneous updates

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^{n}(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Components)}$$

- Maximisation approach : for all $j = 1 \dots J$, $a_{j,n} > 0$, $b_{j,n} \geqslant 0$ and

$$\theta_{j,n+1} = \mathrm{argmax}_{\theta \in \mathsf{T}} \int_{\mathsf{Y}} \left[a_{j,n} \gamma_{j,\alpha}^{n}(y) + b_{j,n} k(\theta_{j,n}, y) \right] \log(k(\theta, y)) \nu(\mathrm{d}y)$$

- Gradient-based approach : for all $j = 1 \dots J$, $\gamma_{j,n} \in (0, 1]$, $c_{j,n} > 0$

$$\theta_{j,n+1} = \theta_{j,n} - \frac{\gamma_{j,n}}{\beta_{j,n}} \nabla g_{j,n}(\theta)|_{\theta=\theta_{j,n}}$$

where $g_{j,n}$ is assumed to be $\beta_{j,n}$-smooth on $\mathsf{T} = \mathbb{R}^d$ with

$$g_{j,n}(\theta) = c_{j,n} \int_{\mathsf{Y}} \frac{\gamma_{j,\alpha}^{n}(y)}{\alpha - 1} \log \left(\frac{k(\theta, y)}{k(\theta_{j,n}, y)} \right) \nu(\mathrm{d}y) \; .$$

Towards simultaneous updates

$$\int_{\mathsf{Y}} \sum_{j=1}^{J} \lambda_{j,n} \gamma_{j,\alpha}^{n}(y) \log\left(\frac{k(\theta_{j,n+1},y)}{k(\theta_{j,n},y)}\right) \nu(\mathrm{d}y) \geqslant 0 \qquad \text{(Components)}$$

- Maximisation approach : for all $j = 1 \dots J$, $a_{j,n} > 0$, $b_{j,n} \geqslant 0$ and

$$\theta_{j,n+1} = \mathrm{argmax}_{\theta \in \mathsf{T}} \int_{\mathsf{Y}} \left[a_{j,n}\gamma_{j,\alpha}^{n}(y) + b_{j,n}k(\theta_{j,n},y)\right] \log(k(\theta,y))\nu(\mathrm{d}y)$$

- Gradient-based approach : for all $j = 1 \dots J$, $\gamma_{j,n} \in (0,1]$, $c_{j,n} > 0$

$$\theta_{j,n+1} = \theta_{j,n} - \frac{\gamma_{j,n}}{\beta_{j,n}} \nabla g_{j,n}(\theta)|_{\theta=\theta_{j,n}}$$

where $g_{j,n}$ is assumed to be $\beta_{j,n}$-smooth on $\mathsf{T} = \mathbb{R}^d$ with

$$g_{j,n}(\theta) = c_{j,n} \int_{\mathsf{Y}} \frac{\gamma_{j,\alpha}^{n}(y)}{\alpha - 1} \log\left(\frac{k(\theta,y)}{k(\theta_{j,n},y)}\right) \nu(\mathrm{d}y) \ .$$

Related work

Maximisation approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma^n_{\ell,\alpha}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}$$

$$m_{j,n+1} = (1-\gamma_{j,n})m_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)y\ \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y)}$$

$$\Sigma_{j,n+1} = (1-\gamma_{j,n})\tilde{\Sigma}_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)(y-m_{j,n+1})(y-m_{j,n+1})^T\nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y)}$$

NB : $\gamma_{j,n} = (a_{j,n}\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y) + b_{j,n})^{-1}$ with $b_{j,n} \geqslant 0$

Adaptive importance sampling in general mixture classes. O. Cappé, R. Douc, A. Guillin, J-M Marin and C. P Robert (2008). Statistics and Computing, 18(4):447–459

→ The M-PMC algorithm a.k.a 'Integrated EM' corresponds to

$$\alpha = 0,\ \eta_n = 1,\ \kappa = 0 \ \text{ and } \ \gamma_{j,n} = 1$$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n, κ and $\gamma_{j,n}$, where η_n and $\gamma_{j,n}$ act as learning rates
2. We extend the systematic decrease property to $\alpha \in [0, 1)$

Maximisation approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}$$

$$m_{j,n+1} = (1-\gamma_{j,n})m_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)y\ \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)}$$

$$\Sigma_{j,n+1} = (1-\gamma_{j,n})\tilde{\Sigma}_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)(y-m_{j,n+1})(y-m_{j,n+1})^T\nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)}$$

NB : $\gamma_{j,n} = (a_{j,n}\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + b_{j,n})^{-1}$ with $b_{j,n} \geqslant 0$

Adaptive importance sampling in general mixture classes. O. Cappé, R. Douc, A. Guillin, J-M Marin and C. P Robert (2008). Statistics and Computing, 18(4):447–459

$\rightarrow$ The M-PMC algorithm a.k.a 'Integrated EM' corresponds to

$$\alpha = 0,\ \eta_n = 1,\ \kappa = 0 \ \text{ and } \ \gamma_{j,n} = 1$$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n, κ and $\gamma_{j,n}$, where η_n and $\gamma_{j,n}$ act as learning rates
2. We extend the systematic decrease property to $\alpha \in [0,1)$

Maximisation approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}$$

$$m_{j,n+1} = (1-\gamma_{j,n})m_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y) y \ \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)}$$

$$\Sigma_{j,n+1} = (1-\gamma_{j,n})\tilde{\Sigma}_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)(y-m_{j,n+1})(y-m_{j,n+1})^T \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)}$$

NB : $\gamma_{j,n} = (a_{j,n}\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + b_{j,n})^{-1}$ with $b_{j,n} \geqslant 0$

Adaptive importance sampling in general mixture classes. O. Cappé, R. Douc, A. Guillin, J-M Marin and C. P Robert (2008). Statistics and Computing, 18(4):447–459

→ The M-PMC algorithm a.k.a 'Integrated EM' corresponds to

$$\alpha = 0,\ \eta_n = 1,\ \kappa = 0 \ \text{ and } \ \gamma_{j,n} = 1$$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n, κ and $\gamma_{j,n}$, where η_n and $\gamma_{j,n}$ act as learning rates
2. We extend the systematic decrease property to $\alpha \in [0,1)$

Maximisation approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma^n_{\ell,\alpha}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}$$

$$m_{j,n+1} = (1-\gamma_{j,n})m_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)y\ \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y)}$$

$$\Sigma_{j,n+1} = (1-\gamma_{j,n})\tilde{\Sigma}_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)(y-m_{j,n+1})(y-m_{j,n+1})^T\nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y)}$$

NB : $\gamma_{j,n} = (a_{j,n}\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y) + b_{j,n})^{-1}$ with $b_{j,n} \geqslant 0$

Adaptive importance sampling in general mixture classes. O. Cappé, R. Douc, A. Guillin, J-M Marin and C. P Robert (2008). Statistics and Computing, 18(4):447–459

$\rightarrow$ The M-PMC algorithm a.k.a 'Integrated EM' corresponds to

$$\alpha = 0,\ \eta_n = 1,\ \kappa = 0 \ \text{ and } \ \gamma_{j,n} = 1$$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n, κ and $\gamma_{j,n}$, where η_n and $\gamma_{j,n}$ act as learning rates
2. We extend the systematic decrease property to $\alpha \in [0,1)$

Maximisation approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}$$

$$m_{j,n+1} = (1-\gamma_{j,n})m_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)y\ \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)}$$

$$\Sigma_{j,n+1} = (1-\gamma_{j,n})\tilde{\Sigma}_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)(y-m_{j,n+1})(y-m_{j,n+1})^T\nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)}$$

NB : $\gamma_{j,n} = (a_{j,n}\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + b_{j,n})^{-1}$ with $b_{j,n} \geqslant 0$

Adaptive importance sampling in general mixture classes. O. Cappé, R. Douc, A. Guillin, J-M Marin and C. P Robert (2008). Statistics and Computing, 18(4):447–459

$\rightarrow$ The M-PMC algorithm a.k.a 'Integrated EM' corresponds to

$$\alpha = 0,\ \eta_n = 1,\ \kappa = 0 \ \text{ and } \ \gamma_{j,n} = 1$$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n, κ and $\gamma_{j,n}$, where η_n and $\gamma_{j,n}$ act as learning rates
2. We extend the systematic decrease property to $\alpha \in [0,1)$

Maximisation approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}$$

$$m_{j,n+1} = (1-\gamma_{j,n})m_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)y\ \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)}$$

$$\Sigma_{j,n+1} = (1-\gamma_{j,n})\tilde{\Sigma}_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)(y-m_{j,n+1})(y-m_{j,n+1})^T\nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)}$$

NB : $\gamma_{j,n} = (a_{j,n}\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + b_{j,n})^{-1}$ with $b_{j,n} \geqslant 0$

Adaptive importance sampling in general mixture classes. O. Cappé, R. Douc, A. Guillin, J-M Marin and C. P Robert (2008). Statistics and Computing, 18(4):447–459

$\rightarrow$ The M-PMC algorithm a.k.a 'Integrated EM' corresponds to

$$\alpha = 0,\ \eta_n = 1,\ \kappa = 0 \ \text{ and } \ \gamma_{j,n} = 1$$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n, κ and $\gamma_{j,n}$, where η_n and $\gamma_{j,n}$ act as learning rates
2. We extend the systematic decrease property to $\alpha \in [0,1)$

Maximisation approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}$$

$$m_{j,n+1} = (1-\gamma_{j,n})m_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)y\ \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)}$$

$$\Sigma_{j,n+1} = (1-\gamma_{j,n})\tilde{\Sigma}_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)(y-m_{j,n+1})(y-m_{j,n+1})^T\nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)}$$

NB : $\gamma_{j,n} = (a_{j,n}\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + b_{j,n})^{-1}$ with $b_{j,n} \geqslant 0$

Adaptive importance sampling in general mixture classes. O. Cappé, R. Douc, A. Guillin, J-M Marin and C. P Robert (2008). Statistics and Computing, 18(4):447–459

$\rightarrow$ The M-PMC algorithm a.k.a 'Integrated EM' corresponds to

$$\alpha = 0,\ \eta_n = 1,\ \kappa = 0 \ \text{ and } \ \gamma_{j,n} = 1$$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n, κ and $\gamma_{j,n}$, where η_n and $\gamma_{j,n}$ act as learning rates
2. We extend the systematic decrease property to $\alpha \in [0,1)$

Gradient-based approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma^n_{\ell,\alpha}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}$$

$$\text{(RGD)} \quad m_{j,n+1} = m_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \lambda_{j,n}\gamma^n_{j,\alpha}(y)(y-m_{j,n})\nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \mu_n k(y)^\alpha p(y)^{1-\alpha}\nu(\mathrm{d}y)}$$

$$\text{(MG)} \quad m_{j,n+1} = (1-\gamma_{j,n})\, m_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)y\ \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y)}$$

- **(RGD).** Set $p = p(\cdot, \mathscr{D})$, $\gamma_{j,n} := \gamma_n \in (0,1]$. Usual gradient descent steps on Θ for Rényi's α-divergence minimisation

Rényi divergence variational inference. Y. Li and R. E Turner (2016). NeurIPS

NB : We provide simultaneous updates for $\boldsymbol{\lambda}$ that preserve the convergence!

- **(MG).** 'EM-like' : coincides with the mixture means update from the maximisation approach.

NB : Our maximisation approach gives updates for the covariance matrices too!

Gradient-based approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}$$

$$\text{(RGD)} \quad m_{j,n+1} = m_{j,n} + \gamma_{j,n} \frac{\int_{\mathsf{Y}} \lambda_{j,n}\gamma_{j,\alpha}^n(y)(y - m_{j,n})\nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \mu_n k(y)^\alpha p(y)^{1-\alpha}\nu(\mathrm{d}y)}$$

$$\text{(MG)} \quad m_{j,n+1} = (1-\gamma_{j,n})\, m_{j,n} + \gamma_{j,n} \frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y) y \; \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)}$$

- **(RGD).** Set $p = p(\cdot, \mathscr{D})$, $\gamma_{j,n} := \gamma_n \in (0,1]$. Usual gradient descent steps on Θ for Rényi's α-divergence minimisation

Rényi divergence variational inference. Y. Li and R. E Turner (2016). NeurIPS

NB : We provide simultaneous updates for $\boldsymbol{\lambda}$ that preserve the convergence!

- **(MG).** 'EM-like' : coincides with the mixture means update from the maximisation approach.

NB : Our maximisation approach gives updates for the covariance matrices too!

Gradient-based approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^{J} \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^{n}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}$$

$$\text{(RGD)} \quad m_{j,n+1} = m_{j,n} + \gamma_{j,n} \frac{\int_{\mathsf{Y}} \lambda_{j,n}\gamma_{j,\alpha}^{n}(y)(y-m_{j,n})\nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \mu_n k(y)^{\alpha} p(y)^{1-\alpha}\nu(\mathrm{d}y)}$$

$$\text{(MG)} \quad m_{j,n+1} = (1-\gamma_{j,n})\, m_{j,n} + \gamma_{j,n} \frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y) y \; \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y)\nu(\mathrm{d}y)}$$

- **(RGD).** Set $p = p(\cdot, \mathscr{D})$, $\gamma_{j,n} := \gamma_n \in (0,1]$. Usual gradient descent steps on Θ for Rényi's α-divergence minimisation

Rényi divergence variational inference. Y. Li and R. E Turner (2016). NeurIPS

NB : We provide simultaneous updates for $\boldsymbol{\lambda}$ that preserve the convergence!

- **(MG).** 'EM-like' : coincides with the mixture means update from the maximisation approach.

NB : Our maximisation approach gives updates for the covariance matrices too!

Gradient-based approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^n(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}$$

$$\text{(RGD)} \quad m_{j,n+1} = m_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \lambda_{j,n}\gamma_{j,\alpha}^n(y)(y-m_{j,n})\nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \mu_n k(y)^\alpha p(y)^{1-\alpha}\nu(\mathrm{d}y)}$$

$$\text{(MG)} \quad m_{j,n+1} = (1-\gamma_{j,n})\, m_{j,n} + \gamma_{j,n}\frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)y\ \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)}$$

- **(RGD).** Set $p = p(\cdot, \mathscr{D})$, $\gamma_{j,n} := \gamma_n \in (0,1]$. Usual gradient descent steps on Θ for Rényi's α-divergence minimisation

Rényi divergence variational inference. Y. Li and R. E Turner (2016). NeurIPS

NB : We provide simultaneous updates for $\boldsymbol{\lambda}$ that preserve the convergence!

- **(MG).** 'EM-like' : coincides with the mixture means update from the maximisation approach.

NB : Our maximisation approach gives updates for the covariance matrices too!

Gradient-based approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y) \nu(\mathrm{d}y) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^n(y) \nu(\mathrm{d}y) + (\alpha - 1)\kappa \right]^{\eta_n}}$$

$$\text{(RGD)} \quad m_{j,n+1} = m_{j,n} + \gamma_{j,n} \frac{\int_{\mathsf{Y}} \lambda_{j,n} \gamma_{j,\alpha}^n(y)(y - m_{j,n}) \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \mu_n k(y)^\alpha p(y)^{1-\alpha} \nu(\mathrm{d}y)}$$

$$\text{(MG)} \quad m_{j,n+1} = (1 - \gamma_{j,n})\, m_{j,n} + \gamma_{j,n} \frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y) y \ \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y) \nu(\mathrm{d}y)}$$

- **(RGD).** Set $p = p(\cdot, \mathscr{D})$, $\gamma_{j,n} := \gamma_n \in (0, 1]$. Usual gradient descent steps on Θ for Rényi's α-divergence minimisation

Rényi divergence variational inference. Y. Li and R. E Turner (2016). NeurIPS

NB : We provide simultaneous updates for $\boldsymbol{\lambda}$ that preserve the convergence!

- **(MG).** 'EM-like' : coincides with the mixture means update from the maximisation approach.

NB : Our maximisation approach gives updates for the covariance matrices too!

Gradient-based approach (Gaussian case)

$$\lambda_{j,n+1} = \frac{\lambda_{j,n}\left[\int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}{\sum_{\ell=1}^{J} \lambda_{\ell,n}\left[\int_{\mathsf{Y}} \gamma_{\ell,\alpha}^{n}(y)\nu(\mathrm{d}y) + (\alpha-1)\kappa\right]^{\eta_n}}$$

$$\text{(RGD)} \quad m_{j,n+1} = m_{j,n} + \gamma_{j,n} \frac{\int_{\mathsf{Y}} \lambda_{j,n}\gamma_{j,\alpha}^{n}(y)(y - m_{j,n})\nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \mu_n k(y)^{\alpha} p(y)^{1-\alpha}\nu(\mathrm{d}y)}$$

$$\text{(MG)} \quad m_{j,n+1} = (1-\gamma_{j,n})\, m_{j,n} + \gamma_{j,n} \frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y) y \; \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y)\nu(\mathrm{d}y)}$$

• **(RGD).** Set $p = p(\cdot, \mathscr{D})$, $\gamma_{j,n} := \gamma_n \in (0,1]$. Usual gradient descent steps on Θ for Rényi's α-divergence minimisation

Rényi divergence variational inference. Y. Li and R. E Turner (2016). NeurIPS

NB : We provide simultaneous updates for $\boldsymbol{\lambda}$ that preserve the convergence!

• **(MG).** 'EM-like' : coincides with the mixture means update from the maximisation approach.

NB : Our maximisation approach gives updates for the covariance matrices too!

Outline

Monte Carlo approximations

Algorithm 1: Gaussian Mixture Models optimisation

At iteration n,

1. Draw independently M samples $(Y_{m,n})_{1 \leqslant m \leqslant M}$ from the proposal q_n. Define for all $j = 1 \dots J$, all $y \in \mathsf{Y}$ and all $n \geqslant 1$, $\hat{\gamma}^n_{j,\alpha}(y) = k(\theta_{j,n}, y)/q_n(y) \left(\mu_n k(y)/p(y)\right)^{\alpha-1}$.
2. For all $j = 1 \dots J$, set:

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\sum_{m=1}^M \hat{\gamma}^n_{j,\alpha}(Y_{m,n}) + (\alpha - 1)\kappa_n\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\sum_{m=1}^M \hat{\gamma}^n_{\ell,\alpha}(Y_{m,n}) + (\alpha - 1)\kappa_n\right]^{\eta_n}}$$

$$(RGD) \quad m_{j,n+1} = m_{j,n} + \gamma_n \frac{\lambda_{j,n} \sum_{m=1}^M \hat{\gamma}^n_{j,\alpha}(Y_{m,n}) \cdot (Y_{m,n} - \theta_{j,n})}{\sum_{j=1}^J \sum_{m=1}^M \lambda_{j,n} \hat{\gamma}^n_{j,\alpha}(Y_{m,n})}$$

$$(MG) \quad m_{j,n+1} = (1 - \gamma_n)\, m_{j,n} + \gamma_n \frac{\sum_{m=1}^M \hat{\gamma}^n_{j,\alpha}(Y_{m,n}) \cdot Y_{m,n}}{\sum_{m=1}^M \hat{\gamma}^n_{j,\alpha}(Y_{m,n})}$$

$\rightarrow$ Here $\hat{\gamma}^n_{j,\alpha}(y) = \frac{\gamma^n_{j,\alpha}(y)}{q_n(y)}$.

$\rightarrow$ We consider two samplers : $q_n = \mu_{\boldsymbol{\lambda}_n, \Theta_n}$ (IS-n) and $q_n = J^{-1} \sum_{j=1}^J k(\theta_{j,n}, \cdot)$ (IS-unif).

Monte Carlo approximations

Algorithm 1: Gaussian Mixture Models optimisation

At iteration n,

1. Draw independently M samples $(Y_{m,n})_{1 \leqslant m \leqslant M}$ from the proposal q_n. Define for all $j = 1 \dots J$, all $y \in \mathsf{Y}$ and all $n \geqslant 1$, $\hat{\gamma}^n_{j,\alpha}(y) = k(\theta_{j,n}, y)/q_n(y) \left(\mu_n k(y)/p(y)\right)^{\alpha-1}$.
2. For all $j = 1 \dots J$, set:

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\sum_{m=1}^M \hat{\gamma}^n_{j,\alpha}(Y_{m,n}) + (\alpha-1)\kappa_n\right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\sum_{m=1}^M \hat{\gamma}^n_{\ell,\alpha}(Y_{m,n}) + (\alpha-1)\kappa_n\right]^{\eta_n}}$$

$$(RGD) \quad m_{j,n+1} = m_{j,n} + \gamma_n \frac{\lambda_{j,n} \sum_{m=1}^M \hat{\gamma}^n_{j,\alpha}(Y_{m,n}) \cdot (Y_{m,n} - \theta_{j,n})}{\sum_{j=1}^J \sum_{m=1}^M \lambda_{j,n} \hat{\gamma}^n_{j,\alpha}(Y_{m,n})}$$

$$(MG) \quad m_{j,n+1} = (1-\gamma_n)\, m_{j,n} + \gamma_n \frac{\sum_{m=1}^M \hat{\gamma}^n_{j,\alpha}(Y_{m,n}) \cdot Y_{m,n}}{\sum_{m=1}^M \hat{\gamma}^n_{j,\alpha}(Y_{m,n})}$$

$\rightarrow$ Here $\hat{\gamma}^n_{j,\alpha}(y) = \frac{\gamma^n_{j,\alpha}(y)}{q_n(y)}$.

$\rightarrow$ We consider two samplers : $q_n = \mu_{\boldsymbol{\lambda}_n, \Theta_n}$ (IS-n) and $q_n = J^{-1} \sum_{j=1}^J k(\theta_{j,n}, \cdot)$ (IS-unif).

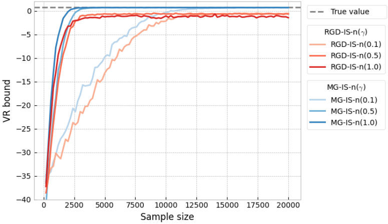
Comparing RGD to MG (fixed $\boldsymbol{\lambda}$)

$$\text{Target}: \quad p(y) = 2 \times \left[0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})\right]$$

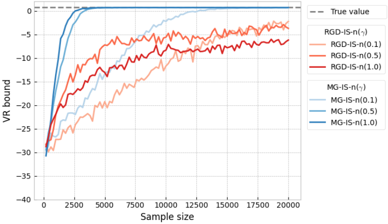
- MC estimate of the VR Bound averaged over 30 trials for RGD and MG.

[Here, $\alpha = 0.2$, $d = 16$, $M = 200$, $\kappa_n = 0$, $\eta_n = 0$. and $q_n = \mu_n k$.]

$J = 10$



$J = 50$



- LogMSE averaged over 30 trials for RGD and MG.

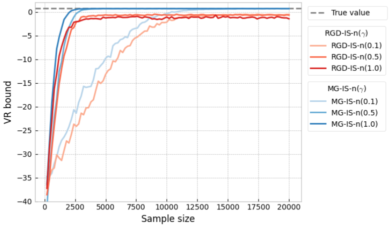
	$J = 10$			$J = 50$		
	$\gamma = 0.1$	$\gamma = 0.5$	$\gamma = 1.0$	$\gamma = 0.1$	$\gamma = 0.5$	$\gamma = 1.0$
RGD-IS-n(γ)	-0.081	-0.076	-0.218	-1.640	-1.673	-1.560
MG-IS-n(γ)	**-3.702**	**-1.875**	**-2.711**	**-2.760**	**-2.771**	**-2.788**

Comparing RGD to MG (fixed $\boldsymbol{\lambda}$)

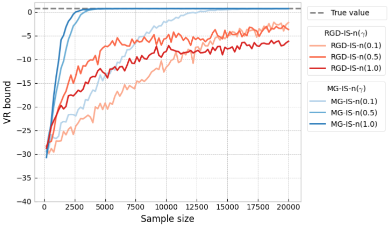
Target : $p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$

- MC estimate of the VR Bound averaged over 30 trials for RGD and MG. [Here, $\alpha = 0.2$, $d = 16$, $M = 200$, $\kappa_n = 0$, $\eta_n = 0$. and $q_n = \mu_n k$.]

$J = 10$



$J = 50$



- LogMSE averaged over 30 trials for RGD and MG.

	$J = 10$			$J = 50$		
	$\gamma = 0.1$	$\gamma = 0.5$	$\gamma = 1.0$	$\gamma = 0.1$	$\gamma = 0.5$	$\gamma = 1.0$
RGD-IS-n(γ)	-0.081	-0.076	-0.218	-1.640	-1.673	-1.560
MG-IS-n(γ)	**-3.702**	**-1.875**	**-2.711**	**-2.760**	**-2.771**	**-2.788**

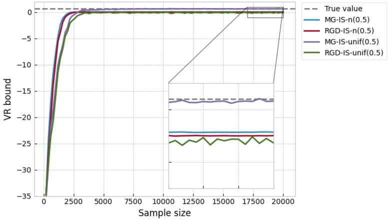
Comparing RGD to MG (varying $\boldsymbol{\lambda}$)

$$\text{Target :} \quad p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$$

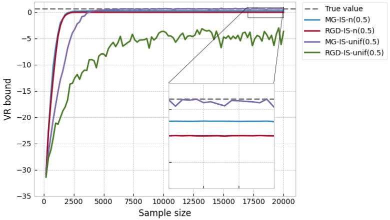
- MC estimate of the VR Bound averaged over 30 trials for RGD and MG.

[Here, $\alpha = 0.2$, $d = 16$, $M = 200$, $\eta = 0.1$, $\kappa_n = 0$.]

$J = 10$



$J = 50$



- LogMSE averaged over 30 trials for RGD and MG.

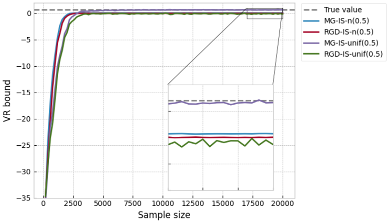
	$J = 10$			$J = 50$		
	$\gamma = 0.1$	$\gamma = 0.5$	$\gamma = 1.0$	$\gamma = 0.1$	$\gamma = 0.5$	$\gamma = 1.0$
RGD-IS-n(γ)	0.372	0.510	0.384	-0.616	-0.713	-0.778
MG-IS-n(γ)	1.104	1.074	0.387	1.135	-0.077	-0.060
RGD-IS-unif(γ)	0.359	0.469	0.458	-0.688	-0.670	-0.583
MG-IS-unif(γ)	**-0.200**	**-0.229**	**-0.515**	**-1.500**	**-1.462**	**-1.246**

Comparing RGD to MG (varying $\boldsymbol{\lambda}$)

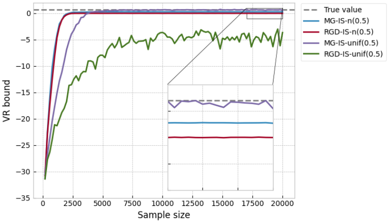
Target : $\quad p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$

- MC estimate of the VR Bound averaged over 30 trials for RGD and MG. [Here, $\alpha = 0.2$, $d = 16$, $M = 200$, $\eta = 0.1$, $\kappa_n = 0$.]

$J = 10$



$J = 50$



- LogMSE averaged over 30 trials for RGD and MG.

	$J = 10$			$J = 50$		
	$\gamma = 0.1$	$\gamma = 0.5$	$\gamma = 1.0$	$\gamma = 0.1$	$\gamma = 0.5$	$\gamma = 1.0$
RGD-IS-n(γ)	0.372	0.510	0.384	-0.616	-0.713	-0.778
MG-IS-n(γ)	1.104	1.074	0.387	1.135	-0.077	-0.060
RGD-IS-unif(γ)	0.359	0.469	0.458	-0.688	-0.670	-0.583
MG-IS-unif(γ)	**-0.200**	**-0.229**	**-0.515**	**-1.500**	**-1.462**	**-1.246**

Comparing RGD to MG (varying $\boldsymbol{\lambda}$, cont'd)

Target : $p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$

- LogMSE averaged over 30 trials for RGD and MG.

[Here, $\alpha = 0.2$, $d = 16$, $M = 200$, $\gamma = 0.5$, $\kappa_n = 0$.]

		$J = 10$			$J = 50$	
	$\eta = 0.05$	$\eta = 0.1$	$\eta = 0.5$	$\eta = 0.05$	$\eta = 0.1$	$\eta = 0.5$
RGD-IS-n(γ)	0.045	0.510	1.299	-1.355	-0.713	0.924
MG-IS-n(γ)	0.087	1.074	1.343	-1.205	-0.077	1.329
RGD-IS-unif(γ)	-0.018	0.469	1.328	-1.385	-0.670	0.928
MG-IS-unif(γ)	**-1.244**	**-0.229**	**1.100**	**-2.524**	**-1.462**	**0.309**

Comparing RGD to MG (varying $\boldsymbol{\lambda}$, cont'd)

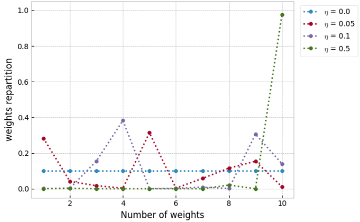
$$\text{Target}: \quad p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$$

- LogMSE averaged over 30 trials for RGD and MG.

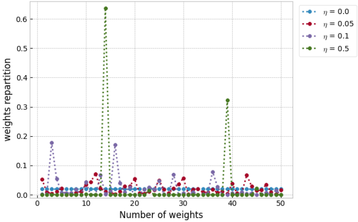
[Here, $\alpha = 0.2$, $d = 16$, $M = 200$, $\gamma = 0.5$, $\kappa_n = 0$.]

	$J = 10$			$J = 50$		
	$\eta = 0.05$	$\eta = 0.1$	$\eta = 0.5$	$\eta = 0.05$	$\eta = 0.1$	$\eta = 0.5$
RGD-IS-n(γ)	0.045	0.510	1.299	-1.355	-0.713	0.924
MG-IS-n(γ)	0.087	1.074	1.343	-1.205	-0.077	1.329
RGD-IS-unif(γ)	-0.018	0.469	1.328	-1.385	-0.670	0.928
MG-IS-unif(γ)	**-1.244**	**-0.229**	**1.100**	**-2.524**	**-1.462**	**0.309**

$J = 10$



$J = 50$



Outline

Conclusion

Novel framework for monotonic α-divergence minimisation

- applicable to mixture models optimisation
- mixture weights and mixture components parameters can be updated simultaneously
- empirical benefits of our general framework

Perspectives

- Additionnal convergence results
- Hyperparameters tuning
- ML applications

Conclusion

Novel framework for monotonic α-divergence minimisation

- applicable to mixture models optimisation
- mixture weights and mixture components parameters can be updated simultaneously
- empirical benefits of our general framework

Perspectives

- Additionnal convergence results
- Hyperparameters tuning
- ML applications

Conclusion

Novel framework for monotonic α-divergence minimisation

- applicable to mixture models optimisation
- mixture weights and mixture components parameters can be updated simultaneously
- empirical benefits of our general framework

Perspectives

- Additionnal convergence results
- Hyperparameters tuning
- ML applications

Conclusion

Novel framework for monotonic α-divergence minimisation

- applicable to mixture models optimisation
- mixture weights and mixture components parameters can be updated simultaneously
- empirical benefits of our general framework

Perspectives

- Additionnal convergence results
- Hyperparameters tuning
- ML applications

Conclusion

Novel framework for monotonic α-divergence minimisation

- applicable to mixture models optimisation
- mixture weights and mixture components parameters can be updated simultaneously
- empirical benefits of our general framework

Perspectives

- Additionnal convergence results
- Hyperparameters tuning
- ML applications

Conclusion

Novel framework for monotonic α-divergence minimisation

- applicable to mixture models optimisation
- mixture weights and mixture components parameters can be updated simultaneously
- empirical benefits of our general framework

Perspectives

- Additionnal convergence results
- Hyperparameters tuning
- ML applications

Conclusion

Novel framework for monotonic α-divergence minimisation

- applicable to mixture models optimisation
- mixture weights and mixture components parameters can be updated simultaneously
- empirical benefits of our general framework

Perspectives

- Additionnal convergence results
- Hyperparameters tuning
- ML applications

Conclusion

Novel framework for monotonic α-divergence minimisation

- applicable to mixture models optimisation
- mixture weights and mixture components parameters can be updated simultaneously
- empirical benefits of our general framework

Perspectives

- Additionnal convergence results
- Hyperparameters tuning
- ML applications

Thank you for your attention!

kamelia.daudel@stats.ox.ac.uk

Monotonic Alpha-divergence Minimisation
K. Daudel, R. Douc and F. Roueff (2021). https://arxiv.org/abs/2103.05684

Infinite-dimensional gradient-based descent for alpha-divergence minimisation.
K. Daudel, R. Douc and F. Portier (2020). Ann. Statist. 49 (4) 2250 - 2270.