Monotonic Alpha-divergence Minimisation

Kamélia Daudel

Postdoc at the University of Oxford
kamelia.daudel@gmail.com

CIRM
29/09/2021

Joint work with Randal Douc and François Roueff

Introduction

- 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 complex models, we can only evaluate $p(y|\mathscr{D})$ up to the constant $p(\mathscr{D})$.

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

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

$$q^\star = \mathrm{arginf}_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \ ,$$

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

Important aspects in VI (i) Choice of D (ii) Choice of $\mathcal{Q}$

Introduction

- 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 complex models, we can only evaluate $p(y|\mathscr{D})$ up to the constant $p(\mathscr{D})$.

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

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

$$q^\star = \mathrm{arginf}_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \ ,$$

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

Important aspects in VI (i) Choice of D (ii) Choice of $\mathcal{Q}$

Introduction

- 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 complex models, we can only evaluate $p(y|\mathscr{D})$ up to the constant $p(\mathscr{D})$.

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

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

$$q^\star = \operatorname{arginf}_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \,,$$

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

Introduction

- 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 complex models, we can only evaluate $p(y|\mathscr{D})$ up to the constant $p(\mathscr{D})$.

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

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

$$q^\star = \mathrm{arginf}_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) ,$$

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

Important aspects in VI (i) Choice of D (ii) Choice of $\mathcal{Q}$

Introduction

- 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 complex models, we can only evaluate $p(y|\mathscr{D})$ up to the constant $p(\mathscr{D})$.

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

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

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

$$q^\star = \mathrm{arginf}_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \ ,$$

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

Important aspects in VI (i) Choice of D (ii) Choice of $\mathcal{Q}$

Introduction

- 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 complex models, we can only evaluate $p(y|\mathscr{D})$ up to the constant $p(\mathscr{D})$.

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

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

$$q^\star = \operatorname{arginf}_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \ ,$$

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

Important aspects in VI (i) Choice of D (ii) Choice of $\mathcal{Q}$

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

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

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 $\mathbb{P}$

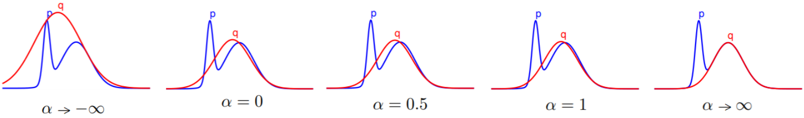
$$D_\alpha(\mathbb{Q}||\mathbb{P}) = \int_{\mathsf{Y}} f_\alpha\left(\frac{q(y)}{p(y)}\right) p(y)\nu(\mathrm{d}y) \ ,$$

where

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

1. A flexible family of divergences...

Figure: In red, the Gaussian which minimises the α-divergence to a mixture of two Gaussian for a varying α



Adapted from **Divergence Measures and Message Passing**. T. Minka (2005). Technical Report MSR-TR-2005-173

The α-divergence family (2)

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...

$$\begin{aligned} q^\star &= \operatorname{arginf}_{q\in\mathcal{Q}} D_\alpha(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \\ &= \operatorname{arginf}_{q\in\mathcal{Q}} \Psi_\alpha(q; p) \end{aligned}$$

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

The α-divergence family (2)

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...

$$\begin{aligned} q^\star &= \mathrm{arginf}_{q\in\mathcal{Q}} D_\alpha(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \\ &= \mathrm{arginf}_{q\in\mathcal{Q}} \Psi_\alpha(q;p) \end{aligned}$$

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

Our approach

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 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 / 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)

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.

1. (Weights) and (Components) permit 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)

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.

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

2. The dependency is simpler in (Weights)

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

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

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

Conditions for a monotonic decrease (2)

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.

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

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

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

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

Conditions for a monotonic decrease (2)

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.

1. (Weights) and (Components) permit 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 $(\alpha-1)\kappa \geqslant 0$

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). *To appear in the Annals of Statistics.*

Core insight :
The mixture weights update is gradient-based, η_n plays the role of a learning rate

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). *To appear in the Annals of Statistics.*

Core insight :

The mixture weights update is gradient-based, η_n plays the role of a learning rate

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). *To appear in the Annals of Statistics.*

Core insight :
The mixture weights update is gradient-based, η_n plays the role of a learning rate

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

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

- Gradient-based approach

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

where $\gamma_{j,n} \in (0, 1]$, $c_{j,n} > 0$,

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

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

$\rightarrow$ Question : How do this relate to / improve on the existing literature?

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

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

- Gradient-based approach

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

where $\gamma_{j,n} \in (0,1]$, $c_{j,n} > 0$,

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

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

$\rightarrow$ Question : How do this relate to / improve on the existing literature?

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

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

- Gradient-based approach

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

where $\gamma_{j,n} \in (0,1]$, $c_{j,n} > 0$,

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

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

→ Question : How do this relate to / improve on the existing literature?

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

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

- Gradient-based approach

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

where $\gamma_{j,n} \in (0, 1]$, $c_{j,n} > 0$,

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

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

$\rightarrow$ Question : How do this relate to / improve on the existing literature?

Maximisation approach

The M-PMC algorithm a.k.a 'Integrated EM'

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

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

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

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

$\rightarrow$ We recover the M-PMC algorithm when $\alpha = 0$, $\eta_n = 1$ and $\kappa = 0$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n and κ, where η_n acts as a learning rate
2. We extend the systematic decrease property to $\alpha \in [0, 1)$

The M-PMC algorithm a.k.a 'Integrated EM'

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

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

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

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

$\rightarrow$ We recover the M-PMC algorithm when $\alpha = 0$, $\eta_n = 1$ and $\kappa = 0$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n and κ, where η_n acts as a learning rate
2. We extend the systematic decrease property to $\alpha \in [0, 1)$

The M-PMC algorithm a.k.a 'Integrated EM'

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

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

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

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

$\rightarrow$ We recover the M-PMC algorithm when $\alpha = 0$, $\eta_n = 1$ and $\kappa = 0$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n and κ, where η_n acts as a learning rate
2. We extend the systematic decrease property to $\alpha \in [0, 1)$

The M-PMC algorithm a.k.a 'Integrated EM'

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

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

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

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

$\rightarrow$ We recover the M-PMC algorithm when $\alpha = 0$, $\eta_n = 1$ and $\kappa = 0$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n and κ, where η_n acts as a learning rate
2. We extend the systematic decrease property to $\alpha \in [0, 1)$

The M-PMC algorithm a.k.a ‘Integrated EM’

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

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

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

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

$\rightarrow$ We recover the M-PMC algorithm when $\alpha = 0$, $\eta_n = 1$ and $\kappa = 0$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n and κ, where η_n acts as a learning rate
2. We extend the systematic decrease property to $\alpha \in [0, 1)$

The M-PMC algorithm a.k.a 'Integrated EM'

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

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

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

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

$\rightarrow$ We recover the M-PMC algorithm when $\alpha = 0$, $\eta_n = 1$ and $\kappa = 0$

We have generalised an integrated EM algorithm for mixture models optimisation

1. We introduce η_n and κ, where η_n acts as a learning rate
2. We extend the systematic decrease property to $\alpha \in [0, 1)$

Application to GMMs

$\rightarrow$ Gaussian kernels : $k(\theta_j, y) = \mathcal{N}(y; m_j, \Sigma_j)$ with $\theta_j = (m_j, \Sigma_j) \in \mathsf{T}$

Algorithm 1: α-divergence minimisation for GMMs

At iteration n,

For all $j = 1 \ldots J$, set

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

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

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

$\rightarrow$ In practice : M i.i.d samples generated from q_n at iteration n

$$\hat{\gamma}_{j,\alpha}^n(y) = \frac{k(\theta_{j,n}, y)}{q_n(y)}\left(\frac{\mu_{\boldsymbol{\lambda}_n,\Theta_n} k(y)}{p(y)}\right)^{\alpha-1}$$

Application to GMMs

→ Gaussian kernels : $k(\theta_j, y) = \mathcal{N}(y; m_j, \Sigma_j)$ with $\theta_j = (m_j, \Sigma_j) \in \mathsf{T}$

Algorithm 1: α-divergence minimisation for GMMs

At iteration n,
For all $j = 1 \ldots J$, set

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

→ In practice : M i.i.d samples generated from q_n at iteration n

$$
\hat{\gamma}_{j,\alpha}^{n}(y) = \frac{k(\theta_{j,n}, y)}{q_n(y)} \left(\frac{\mu_{\boldsymbol{\lambda}_n, \Theta_n} k(y)}{p(y)}\right)^{\alpha - 1}
$$

Application to GMMs

$\rightarrow$ Gaussian kernels : $k(\theta_j, y) = \mathcal{N}(y; m_j, \Sigma_j)$ with $\theta_j = (m_j, \Sigma_j) \in \mathsf{T}$

Algorithm 1: α-divergence minimisation for GMMs

At iteration n,
For all $j = 1 \dots J$, set

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

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

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

$\rightarrow$ In practice : M i.i.d samples generated from q_n at iteration n

$$\hat{\gamma}_{j,\alpha}^n(y) = \frac{k(\theta_{j,n}, y)}{q_n(y)} \left(\frac{\mu_{\boldsymbol{\lambda}_n,\Theta_n} k(y)}{p(y)}\right)^{\alpha-1}$$

Improving on the M-PMC algorithm

Target : $p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$, $d = 16$

Parameters

$\alpha = 0$, $\eta_n = \eta$

$M = 200$, $J = 100$

$q_n(y) = \sum_{j=1}^{J} \lambda_{j,n} k(\theta_{j,n}, y)$

$\rightarrow$ varying η and κ

Improving on the M-PMC algorithm

Target : $p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$, $d = 16$

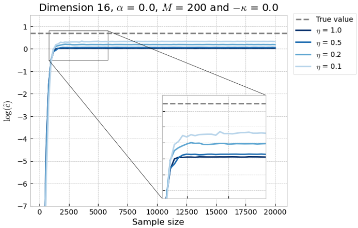
<u>Parameters</u>

$\alpha = 0$, $\eta_n = \eta$

$M = 200$, $J = 100$

$q_n(y) = \sum_{j=1}^{J} \lambda_{j,n} k(\theta_{j,n}, y)$

$\rightarrow$ varying η and κ



Improving on the M-PMC algorithm

Target : $p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$, $d = 16$

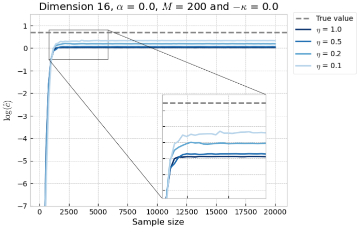
<u>Parameters</u>

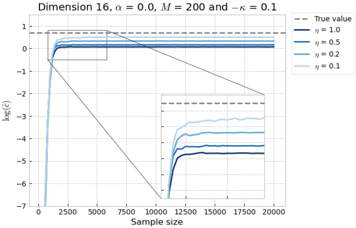
$\alpha = 0,\ \eta_n = \eta$

$M = 200,\ J = 100$

$q_n(y) = \sum_{j=1}^{J} \lambda_{j,n} k(\theta_{j,n}, y)$

$\rightarrow$ varying η and κ





Improving on the M-PMC algorithm

Target : $p(y) = 2 \times \left[0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})\right]$, $d = 16$

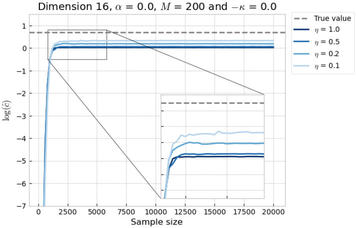
<u>Parameters</u>

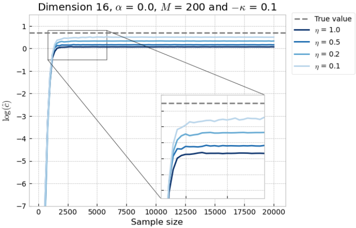
$\alpha = 0$, $\eta_n = \eta$

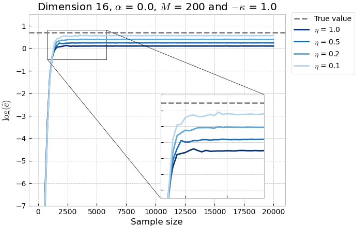
$M = 200$, $J = 100$

$q_n(y) = \sum_{j=1}^{J} \lambda_{j,n} k(\theta_{j,n}, y)$

$\rightarrow$ varying η and κ







Gradient-based approach

Gradient-based approach and Gradient Descent

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

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

where $\gamma_{j,n} \in (0,1]$, $c_{j,n} > 0$,

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

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

Set $p = p(\cdot, \mathscr{D})$, $\gamma_{j,n} := \gamma_n \in (0,1]$. Usual gradient descent steps on Θ for

- α-divergence minimisation : $c_{j,n} = \lambda_{j,n}$
- Rényi's α-divergence minimisation :
$c_{j,n} = \lambda_{j,n}(\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n,\Theta_n} k(y)^\alpha p(y)^{1-\alpha}\nu(\mathrm{d}y))^{-1}$

$\rightarrow$ Problem : $\lambda_{j,n}$ appears as a multiplicative factor, which could prevent learning!

$\rightarrow$ Solution enabled by our framework : $c_{j,n} = (\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y))^{-1}$

Gradient-based approach and Gradient Descent

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

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

where $\gamma_{j,n} \in (0,1]$, $c_{j,n} > 0$,

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

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

Set $p = p(\cdot, \mathscr{D})$, $\gamma_{j,n} := \gamma_n \in (0,1]$. Usual gradient descent steps on Θ for

- α-divergence minimisation : $c_{j,n} = \lambda_{j,n}$
- Rényi's α-divergence minimisation : $c_{j,n} = \lambda_{j,n}(\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n,\Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(\mathrm{d}y))^{-1}$

→ Problem : $\lambda_{j,n}$ appears as a multiplicative factor, which could prevent learning!

→ Solution enabled by our framework : $c_{j,n} = (\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y))^{-1}$

Gradient-based approach and Gradient Descent

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

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

where $\gamma_{j,n} \in (0,1]$, $c_{j,n} > 0$,

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

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

Set $p = p(\cdot,\mathscr{D})$, $\gamma_{j,n} := \gamma_n \in (0,1]$. Usual gradient descent steps on Θ for

- α-divergence minimisation : $c_{j,n} = \lambda_{j,n}$
- Rényi's α-divergence minimisation : $c_{j,n} = \lambda_{j,n}(\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n,\Theta_n} k(y)^\alpha p(y)^{1-\alpha}\nu(\mathrm{d}y))^{-1}$

→ Problem : $\lambda_{j,n}$ appears as a multiplicative factor, which could prevent learning!

→ Solution enabled by our framework : $c_{j,n} = (\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y))^{-1}$

Gradient-based approach and Gradient Descent

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

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

where $\gamma_{j,n} \in (0,1]$, $c_{j,n} > 0$,

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

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

Set $p = p(\cdot, \mathscr{D})$, $\gamma_{j,n} := \gamma_n \in (0,1]$. Usual gradient descent steps on Θ for

- α-divergence minimisation : $c_{j,n} = \lambda_{j,n}$
- Rényi's α-divergence minimisation :

$c_{j,n} = \lambda_{j,n}(\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n,\Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(\mathrm{d}y))^{-1}$

$\rightarrow$ Problem : $\lambda_{j,n}$ appears as a multiplicative factor, which could prevent learning!

$\rightarrow$ Solution enabled by our framework : $c_{j,n} = (\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y))^{-1}$

Gradient-based approach and Gradient Descent

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

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

where $\gamma_{j,n} \in (0,1]$, $c_{j,n} > 0$,

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

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

Set $p = p(\cdot, \mathscr{D})$, $\gamma_{j,n} := \gamma_n \in (0,1]$. Usual gradient descent steps on Θ for

- α-divergence minimisation : $c_{j,n} = \lambda_{j,n}$
- Rényi's α-divergence minimisation :
$c_{j,n} = \lambda_{j,n}(\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(\mathrm{d}y))^{-1}$

$\rightarrow$ Problem : $\lambda_{j,n}$ appears as a multiplicative factor, which could prevent learning!

$\rightarrow$ Solution enabled by our framework : $c_{j,n} = (\int_{\mathsf{Y}} \gamma^n_{j,\alpha}(y)\nu(\mathrm{d}y))^{-1}$

Gradient-based approach and Gradient Descent

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

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

where $\gamma_{j,n} \in (0,1]$, $c_{j,n} > 0$,

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

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

Set $p = p(\cdot, \mathscr{D})$, $\gamma_{j,n} := \gamma_n \in (0,1]$. Usual gradient descent steps on Θ for

- α-divergence minimisation : $c_{j,n} = \lambda_{j,n}$
- Rényi's α-divergence minimisation :
$c_{j,n} = \lambda_{j,n}(\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n,\Theta_n} k(y)^{\alpha} p(y)^{1-\alpha}\nu(\mathrm{d}y))^{-1}$

$\rightarrow$ Problem : $\lambda_{j,n}$ appears as a multiplicative factor, which could prevent learning!

$\rightarrow$ Solution enabled by our framework : $c_{j,n} = (\int_{\mathsf{Y}} \gamma_{j,\alpha}^{n}(y)\nu(\mathrm{d}y))^{-1}$

Application to GMMs (2)

$\rightarrow$ Gaussian kernels $k(\theta_j, y) = \mathcal{N}(y; \theta_j, \sigma^2 \boldsymbol{I_d})$ with $\Theta \in \mathsf{T}^J$, $\mathsf{T} = \mathbb{R}^d$ and $\sigma^2 > 0$

- Case 1 : $c_{j,n} = \lambda_{j,n}(\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(\mathrm{d}y))^{-1}$ with $\beta_{j,n} = \sigma^{-2}(1-\alpha)^{-1}$
- Case 2 : $c_{j,n} = (\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y) \nu(\mathrm{d}y))^{-1}$ with $\beta_{j,n} = \sigma^{-2}(1-\alpha)^{-1}$

$\rightarrow$ In practice : M i.i.d samples generated from q_n at iteration n

Application to GMMs (2)

$\rightarrow$ Gaussian kernels $k(\theta_j, y) = \mathcal{N}(y; \theta_j, \sigma^2 \boldsymbol{I_d})$ with $\Theta \in \mathsf{T}^J$, $\mathsf{T} = \mathbb{R}^d$ and $\sigma^2 > 0$

- Case 1 : $c_{j,n} = \lambda_{j,n}(\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(\mathrm{d}y))^{-1}$ with $\beta_{j,n} = \sigma^{-2}(1-\alpha)^{-1}$
- Case 2 : $c_{j,n} = (\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y) \nu(\mathrm{d}y))^{-1}$ with $\beta_{j,n} = \sigma^{-2}(1-\alpha)^{-1}$

$\rightarrow$ In practice : M i.i.d samples generated from q_n at iteration n

Application to GMMs (2)

$\rightarrow$ Gaussian kernels $k(\theta_j, y) = \mathcal{N}(y; \theta_j, \sigma^2 \boldsymbol{I_d})$ with $\Theta \in \mathsf{T}^J$, $\mathsf{T} = \mathbb{R}^d$ and $\sigma^2 > 0$

- Case 1 : $c_{j,n} = \lambda_{j,n}(\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(\mathrm{d}y))^{-1}$ with $\beta_{j,n} = \sigma^{-2}(1-\alpha)^{-1}$
- Case 2 : $c_{j,n} = (\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y) \nu(\mathrm{d}y))^{-1}$ with $\beta_{j,n} = \sigma^{-2}(1-\alpha)^{-1}$

$\rightarrow$ In practice : M i.i.d samples generated from q_n at iteration n

Application to GMMs (2)

$\rightarrow$ Gaussian kernels $k(\theta_j, y) = \mathcal{N}(y; \theta_j, \sigma^2 \boldsymbol{I_d})$ with $\Theta \in \mathsf{T}^J$, $\mathsf{T} = \mathbb{R}^d$ and $\sigma^2 > 0$

- Case 1 : $c_{j,n} = \lambda_{j,n} (\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(\mathrm{d}y))^{-1}$ with $\beta_{j,n} = \sigma^{-2}(1-\alpha)^{-1}$
- Case 2 : $c_{j,n} = (\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y) \nu(\mathrm{d}y))^{-1}$ with $\beta_{j,n} = \sigma^{-2}(1-\alpha)^{-1}$

Algorithm 2: α-divergence minimisation for GMMs (2)

At iteration n,

For all $j = 1 \dots J$, set

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

$$\theta_{j,n+1} = \begin{cases} \theta_{j,n} + \gamma_n \frac{\int_{\mathsf{Y}} \lambda_{j,n} \gamma_{j,\alpha}^n(y)(y - \theta_{j,n}) \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(\mathrm{d}y)} & \text{(Case 1)} \\ (1 - \gamma_n)\, \theta_{j,n} + \gamma_n \frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y) \; y \; \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y) \nu(\mathrm{d}y)} & \text{(Case 2)} \end{cases}$$

$\rightarrow$ In practice : M i.i.d samples generated from q_n at iteration n

Application to GMMs (2)

$\rightarrow$ Gaussian kernels $k(\theta_j, y) = \mathcal{N}(y; \theta_j, \sigma^2 \boldsymbol{I_d})$ with $\Theta \in \mathsf{T}^J$, $\mathsf{T} = \mathbb{R}^d$ and $\sigma^2 > 0$

- Case 1 : $c_{j,n} = \lambda_{j,n}(\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n,\Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(\mathrm{d}y))^{-1}$ with $\beta_{j,n} = \sigma^{-2}(1-\alpha)^{-1}$
- Case 2 : $c_{j,n} = (\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y))^{-1}$ with $\beta_{j,n} = \sigma^{-2}(1-\alpha)^{-1}$

Algorithm 2: α-divergence minimisation for GMMs (2)

At iteration n,
For all $j = 1 \ldots J$, set

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

$$\theta_{j,n+1} = \begin{cases} \theta_{j,n} + \gamma_n \frac{\int_{\mathsf{Y}} \lambda_{j,n}\gamma_{j,\alpha}^n(y)(y-\theta_{j,n})\nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \mu_{\boldsymbol{\lambda}_n,\Theta_n} k(y)^\alpha p(y)^{1-\alpha}\nu(\mathrm{d}y)} & \text{(Case 1)} \\ (1-\gamma_n)\,\theta_{j,n} + \gamma_n \frac{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\ y\ \nu(\mathrm{d}y)}{\int_{\mathsf{Y}} \gamma_{j,\alpha}^n(y)\nu(\mathrm{d}y)} & \text{(Case 2)} \end{cases}$$

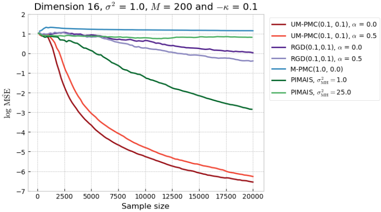
$\rightarrow$ In practice : M i.i.d samples generated from q_n at iteration n

Improving on Gradient Descent updates

Target : $p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]\ ,\ d = 16$

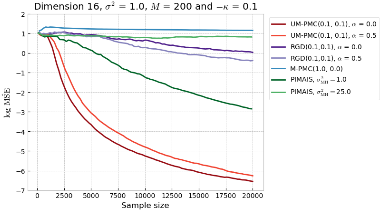
Improving on Gradient Descent updates

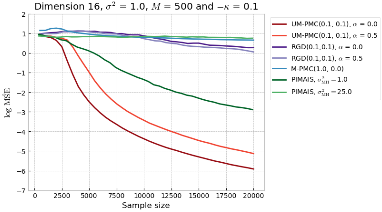
Target : $\quad p(y) = 2 \times \left[0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})\right]\ ,\ d = 16$



Improving on Gradient Descent updates

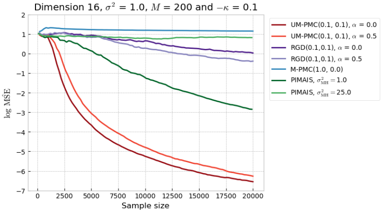
Target : $p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$, $d = 16$

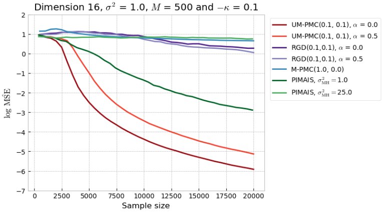


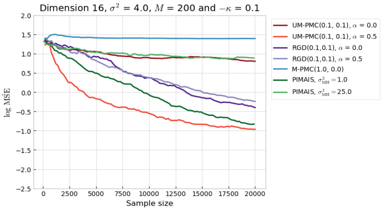


Improving on Gradient Descent updates

Target : $p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$, $d = 16$

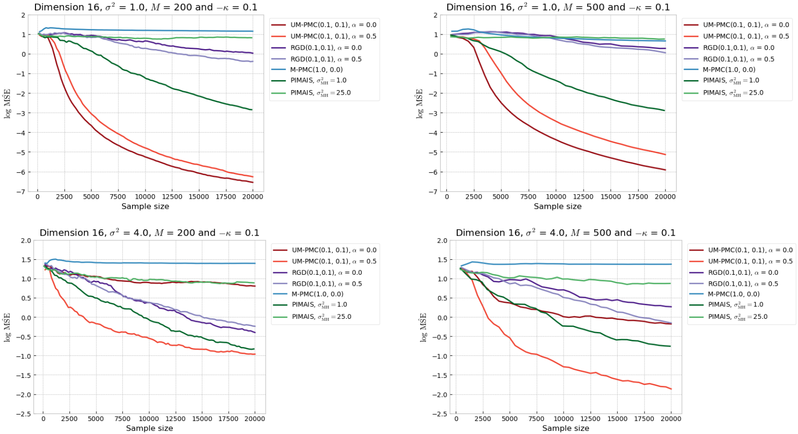






Improving on Gradient Descent updates

Target : $p(y) = 2 \times \left[0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})\right]\ ,\ d = 16$



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 compared to gradient-based approaches and to the M-PMC algorithm

Perspectives

- Additionnal convergence results
- ML applications (suggestions?)

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 compared to gradient-based approaches and to the M-PMC algorithm

Perspectives

- Additionnal convergence results
- ML applications (suggestions?)

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 compared to gradient-based approaches and to the M-PMC algorithm

Perspectives

- Additionnal convergence results
- ML applications (suggestions?)

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 compared to gradient-based approaches and to the M-PMC algorithm

Perspectives

- Additionnal convergence results
- ML applications (suggestions?)

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 compared to gradient-based approaches and to the M-PMC algorithm

Perspectives

- Additionnal convergence results
- ML applications (suggestions?)

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 compared to gradient-based approaches and to the M-PMC algorithm

Perspectives

- Additionnal convergence results
- ML applications (suggestions?)

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 compared to gradient-based approaches and to the M-PMC algorithm

Perspectives

- Additionnal convergence results
- ML applications (suggestions?)

Thank you for your attention!

kamelia.daudel@gmail.com

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). To appear in the Annals of Statistics.

Practical algorithm for GMMs optimisation

→ Gaussian kernels : $k(\theta_j, y) = \mathcal{N}(y; \theta_j, \sigma^2 \boldsymbol{I}_d)$ with $\theta_j \in \mathsf{T} = \mathbb{R}^d$ and $\sigma^2 > 0$

Algorithm 3: α-divergence minimisation for GMMs (constant σ^2)

At iteration n,

1. Draw independently M samples $(Y_{m,n})_{1 \leqslant m \leqslant M}$ from the proposal q_n.
2. For all $j = 1 \ldots J$, set

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

$$\theta_{j,n+1} = \begin{cases} \frac{\sum_{m=1}^{M} \hat{\gamma}_{j,\alpha}^{n}(Y_{m,n}) \cdot Y_{m,n}}{\sum_{m=1}^{M} \hat{\gamma}_{j,\alpha}^{n}(Y_{m,n})} & \text{(Maximisation)} \\ \theta_{j,n} + \frac{\lambda_{j,n} \sum_{m=1}^{M} \hat{\gamma}_{j,\alpha}^{n}(Y_{m,n}) \cdot (Y_{m,n} - \theta_{j,n})}{\sum_{j=1}^{J} \sum_{m=1}^{M} \lambda_{j,n} \hat{\gamma}_{j,\alpha}^{n}(Y_{m,n})} & \text{(GD-based)} \end{cases}$$

Here,

$$\hat{\gamma}_{j,\alpha}^{n}(y) = \frac{k(\theta_{j,n}, y)}{q_n(y)} \left(\frac{\mu_{\boldsymbol{\lambda}_n, \Theta_n} k(y)}{p(y)}\right)^{\alpha - 1}$$

Practical algorithm for GMMs optimisation

→ Gaussian kernels : $k(\theta_j, y) = \mathcal{N}(y; \theta_j, \sigma^2 \boldsymbol{I}_d)$ with $\theta_j \in \mathsf{T} = \mathbb{R}^d$ and $\sigma^2 > 0$

Algorithm 3: α-divergence minimisation for GMMs (constant σ^2)

At iteration n,

1. Draw independently M samples $(Y_{m,n})_{1 \leqslant m \leqslant M}$ from the proposal q_n.
2. For all $j = 1 \ldots J$, set

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

$$\theta_{j,n+1} = \begin{cases} \frac{\sum_{m=1}^{M} \hat{\gamma}_{j,\alpha}^{n}(Y_{m,n}) \cdot Y_{m,n}}{\sum_{m=1}^{M} \hat{\gamma}_{j,\alpha}^{n}(Y_{m,n})} & \text{(Maximisation)} \\ \theta_{j,n} + \frac{\lambda_{j,n} \sum_{m=1}^{M} \hat{\gamma}_{j,\alpha}^{n}(Y_{m,n}) \cdot (Y_{m,n} - \theta_{j,n})}{\sum_{j=1}^{J} \sum_{m=1}^{M} \lambda_{j,n} \hat{\gamma}_{j,\alpha}^{n}(Y_{m,n})} & \text{(GD-based)} \end{cases}$$

Here,

$$\hat{\gamma}_{j,\alpha}^{n}(y) = \frac{k(\theta_{j,n}, y)}{q_n(y)} \left(\frac{\mu_{\boldsymbol{\lambda}_n, \Theta_n} k(y)}{p(y)}\right)^{\alpha - 1}$$

Practical algorithm for GMMs optimisation

$\rightarrow$ Gaussian kernels : $k(\theta_j, y) = \mathcal{N}(y; \theta_j, \sigma^2 \boldsymbol{I}_d)$ with $\theta_j \in \mathsf{T} = \mathbb{R}^d$ and $\sigma^2 > 0$

Algorithm 3: α-divergence minimisation for GMMs (constant σ^2)

At iteration n,

1. Draw independently M samples $(Y_{m,n})_{1 \leqslant m \leqslant M}$ from the proposal q_n.
2. For all $j = 1 \dots J$, set

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

$$\theta_{j,n+1} = \begin{cases} \frac{\sum_{m=1}^M \hat{\gamma}_{j,\alpha}^n(Y_{m,n}) \cdot Y_{m,n}}{\sum_{m=1}^M \hat{\gamma}_{j,\alpha}^n(Y_{m,n})} & \text{(Maximisation)} \\ \theta_{j,n} + \frac{\lambda_{j,n} \sum_{m=1}^M \hat{\gamma}_{j,\alpha}^n(Y_{m,n}) \cdot (Y_{m,n} - \theta_{j,n})}{\sum_{j=1}^J \sum_{m=1}^M \lambda_{j,n} \hat{\gamma}_{j,\alpha}^n(Y_{m,n})} & \text{(GD-based)} \end{cases}$$

Here,

$$\hat{\gamma}_{j,\alpha}^n(y) = \frac{k(\theta_{j,n}, y)}{q_n(y)} \left(\frac{\mu_{\boldsymbol{\lambda}_n, \Theta_n} k(y)}{p(y)} \right)^{\alpha - 1}$$

Additionnal Numerical Experiments

Target : $p(y) = 2 \times \left[0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})\right]\ ,\ d = 16$

Parameters

$\alpha = 0$, $\eta_n = \eta$, $M = 200$

$q_n(y) = \sum_{j=1}^{J} \lambda_{j,n} k(\theta_{j,n}, y)$

vs

$q_n(y) = J^{-1} \sum_{j=1}^{J} k(\theta_{j,n}, y)$

$\rightarrow$ varying η and κ

Additionnal Numerical Experiments

Target : $p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$, $d = 16$

Parameters

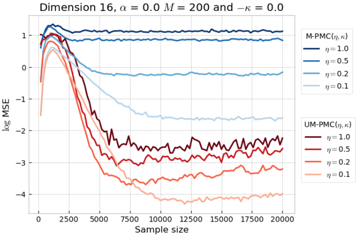
$\alpha = 0$, $\eta_n = \eta$, $M = 200$

$q_n(y) = \sum_{j=1}^{J} \lambda_{j,n} k(\theta_{j,n}, y)$

vs

$q_n(y) = J^{-1} \sum_{j=1}^{J} k(\theta_{j,n}, y)$

$\rightarrow$ varying η and κ



Additionnal Numerical Experiments

Target : $p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$, $d = 16$

<u>Parameters</u>

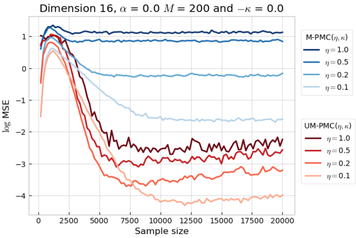
$\alpha = 0,\ \eta_n = \eta,\ M = 200$

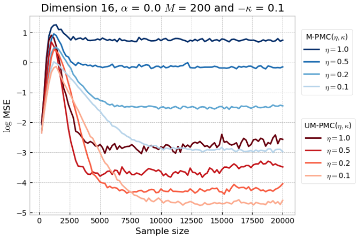
$q_n(y) = \sum_{j=1}^{J} \lambda_{j,n} k(\theta_{j,n}, y)$

vs

$q_n(y) = J^{-1} \sum_{j=1}^{J} k(\theta_{j,n}, y)$

$\rightarrow$ varying η and κ





Additionnal Numerical Experiments

Target : $p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$, $d = 16$

<u>Parameters</u>

$\alpha = 0,\ \eta_n = \eta,\ M = 200$

$q_n(y) = \sum_{j=1}^{J} \lambda_{j,n} k(\theta_{j,n}, y)$

vs

$q_n(y) = J^{-1} \sum_{j=1}^{J} k(\theta_{j,n}, y)$

$\rightarrow$ varying η and κ