

Monotonic Alpha-divergence Variational Inference

Kamélia Daudel

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

01/10/2021

Joint work with Randal Douc, François Portier and François Roueff

Outline

- 1 Introduction
- 2 Infinite-dimensional α -divergence minimisation
- 3 Monotonic α -divergence minimisation
- 4 Conclusion

Bayesian statistics

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

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

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

→ 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

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

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

Bayesian statistics

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

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

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

→ 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

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

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

Bayesian statistics

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

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

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

→ 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

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

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

Important aspects in Variational Inference

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

- Choice of the measure of dissimilarity D
- Choice of the approximating family \mathcal{Q}

Goal : construct a theoretically-sound **Variational Inference (VI)** framework

- that performs **monotonic α -divergence** minimisation
- and **enriches** the typical variational (parametric) family

Important aspects in Variational Inference

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

- Choice of the measure of dissimilarity D
- Choice of the approximating family \mathcal{Q}

Goal : construct a theoretically-sound **Variational Inference (VI)** framework

- that performs **monotonic α -divergence** minimisation
- and **enriches** the typical variational (parametric) family

Important aspects in Variational Inference

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

- Choice of the measure of dissimilarity D
- Choice of the approximating family \mathcal{Q}

Goal : construct a theoretically-sound **Variational Inference (VI)** framework

- that performs **monotonic α -divergence** minimisation
- and **enriches** the typical variational (parametric) family

Important aspects in Variational Inference

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

- Choice of the measure of dissimilarity D
→ Alternative/more general D beyond the Exclusive KL
- Choice of the approximating family \mathcal{Q}
→ Expressiveness of \mathcal{Q} beyond traditional parametric families

Goal : construct a theoretically-sound **Variational Inference (VI)** framework

- that performs **monotonic α -divergence** minimisation
- and **enriches** the typical variational (parametric) family

Important aspects in Variational Inference

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

- Choice of the measure of dissimilarity D
→ Alternative/more general D beyond the Exclusive KL
- Choice of the approximating family \mathcal{Q}
→ Expressiveness of \mathcal{Q} beyond traditional parametric families

Goal : construct a theoretically-sound **Variational Inference (VI)** framework

- that performs **monotonic α -divergence** minimisation
- and **enriches** the typical variational (parametric) family

Important aspects in Variational Inference

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

- Choice of the measure of dissimilarity D
→ Alternative/more general D beyond the Exclusive KL
- Choice of the approximating family \mathcal{Q}
→ Expressiveness of \mathcal{Q} beyond traditional parametric families

Goal : construct a theoretically-sound **Variational Inference (VI)** framework

- that performs **monotonic α -divergence** minimisation
- and **enriches** the typical variational (parametric) family

Important aspects in Variational Inference

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

- Choice of the measure of dissimilarity D
→ Alternative/more general D beyond the Exclusive KL
- Choice of the approximating family \mathcal{Q}
→ Expressiveness of \mathcal{Q} beyond traditional parametric families

Goal : construct a theoretically-sound **Variational Inference (VI)** framework

- that performs **monotonic α -divergence** minimisation
- and **enriches** the typical variational (parametric) family

Variational Inference with the α -divergence family

(Y, \mathcal{Y}, ν) : measured space, ν is a σ -finite measure on (Y, \mathcal{Y}) .

\mathbb{Q} and \mathbb{P} : $\mathbb{Q} \preceq \nu$, $\mathbb{P} \preceq \nu$ with $\frac{d\mathbb{Q}}{d\nu} = q$, $\frac{d\mathbb{P}}{d\nu} = p$.

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

$$D_{\alpha}(\mathbb{Q}||\mathbb{P}) = \int_Y f_{\alpha} \left(\frac{q(y)}{p(y)} \right) p(y) \nu(dy) ,$$

where

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

Variational Inference with the α -divergence family

(Y, \mathcal{Y}, ν) : measured space, ν is a σ -finite measure on (Y, \mathcal{Y}) .

\mathbb{Q} and \mathbb{P} : $\mathbb{Q} \preceq \nu$, $\mathbb{P} \preceq \nu$ with $\frac{d\mathbb{Q}}{d\nu} = q$, $\frac{d\mathbb{P}}{d\nu} = p$.

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

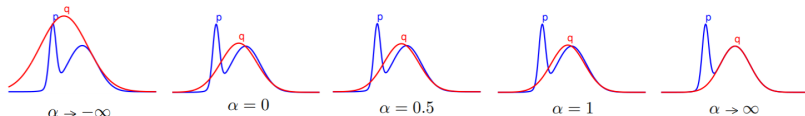
$$D_\alpha(\mathbb{Q}||\mathbb{P}) = \int_Y f_\alpha \left(\frac{q(y)}{p(y)} \right) p(y) \nu(dy) ,$$

where

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

❶ A **flexible** family of divergences...

Figure: In red, the Gaussian which minimises 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

Variational Inference with the α -divergence family

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

$$D_{\alpha}(\mathbb{Q}||\mathbb{P}) = \int_Y f_{\alpha} \left(\frac{q(y)}{p(y)} \right) p(y) \nu(dy) ,$$

where

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

- ❶ A flexible family of divergences...
- ❷ ...suitable for Variational Inference purposes...

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

$$\text{with } \Psi_{\alpha}(q; p) = \int_Y f_{\alpha} \left(\frac{q(y)}{p(y)} \right) p(y) \nu(dy) \text{ and } p = p(\cdot, \mathcal{D})$$

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}

$$D_{\alpha}(\mathbb{Q}||\mathbb{P}) = \int_Y f_{\alpha} \left(\frac{q(y)}{p(y)} \right) p(y) \nu(dy) ,$$

where

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

- ❶ A flexible family of divergences...
- ❷ ...suitable for Variational Inference purposes...

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

$$\text{with } \Psi_{\alpha}(q; p) = \int_Y f_{\alpha} \left(\frac{q(y)}{p(y)} \right) p(y) \nu(dy) \text{ and } p = p(\cdot, \mathcal{D})$$

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

- 1 Introduction
- 2 Infinite-dimensional α -divergence minimisation**
- 3 Monotonic α -divergence minimisation
- 4 Conclusion

Infinite-dimensional α -divergence minimisation

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

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

<https://doi.org/10.1214/20-AOS2035>.

Mixture weights optimisation for Alpha-Divergence Variational Inference.

K. Daudel and R. Douc (2021). To appear in NeurIPS2021

Idea : Extend the traditional variational parametric family

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

by putting a prior on the variational parameter θ

$$\mathcal{Q} = \left\{ q : y \mapsto \mu k(y) := \int_{\mathbb{T}} \mu(d\theta) k(\theta, y) : \mu \in \mathbb{M} \right\}$$

and propose an update formula for μ that ensures a systematic decrease in the α -divergence at each step

→ Mixture Models : $\mu = \sum_{j=1}^J \lambda_j \delta_{\theta_j}$

Infinite-dimensional α -divergence minimisation

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

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

<https://doi.org/10.1214/20-AOS2035>.

Mixture weights optimisation for Alpha-Divergence Variational Inference.

K. Daudel and R. Douc (2021). To appear in NeurIPS2021

Idea : Extend the traditional variational parametric family

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

by putting a prior on the variational parameter θ

$$\mathcal{Q} = \left\{ q : y \mapsto \mu k(y) := \int_{\mathbb{T}} \mu(d\theta) k(\theta, y) : \mu \in \mathbb{M} \right\}$$

and propose an update formula for μ that ensures a systematic decrease in the α -divergence at each step

$$\rightarrow \text{Mixture Models} : \mu = \sum_{j=1}^J \lambda_j \delta_{\theta_j}$$

The (α, Γ) -descent algorithm

Optimisation problem

$$\inf_{\mu \in \mathcal{M}} \Psi_{\alpha}(\mu k; p) \quad \text{with} \quad \Psi_{\alpha}(\mu k; p) := \int_{\mathcal{Y}} f_{\alpha} \left(\frac{\mu k(y)}{p(y)} \right) p(y) \nu(dy)$$

- p is a **nonnegative measurable function** defined on $(\mathcal{Y}, \mathcal{Y})$
- \mathcal{M} is a subset of $\mathcal{M}_1(\mathcal{T})$, the space of probability measures on \mathcal{T}
- $K : (\theta, A) \mapsto \int_A k(\theta, y) \nu(dy)$ is a Markov transition kernel defined on $\mathcal{T} \times \mathcal{Y}$ with density k

The (α, Γ) -descent algorithm

Let $\mu_1 \in \mathcal{M}_1(\mathcal{T})$ be such that $\Psi_{\alpha}(\mu_1 k) < \infty$. The sequence of probability measures $(\mu_n)_{n \geq 1}$ is defined iteratively by

$$\mu_{n+1} = \mathcal{I}_{\alpha}(\mu_n), \quad n \geq 1$$

where for all $\mu \in \mathcal{M}_1(\mathcal{T})$ and all $\theta \in \mathcal{T}$,

$$\mathcal{I}_{\alpha}(\mu)(d\theta) = \frac{\mu(d\theta) \cdot \Gamma(b_{\mu, \alpha}(\theta) + \kappa)}{\mu(\Gamma(b_{\mu, \alpha} + \kappa))} \quad \text{with} \quad b_{\mu, \alpha}(\theta) = \int_{\mathcal{Y}} k(\theta, y) f'_{\alpha} \left(\frac{\mu k(y)}{p(y)} \right) \nu(dy)$$

The (α, Γ) -descent algorithm

Optimisation problem

$$\inf_{\mu \in M} \Psi_{\alpha}(\mu k; p) \quad \text{with} \quad \Psi_{\alpha}(\mu k; p) := \int_Y f_{\alpha} \left(\frac{\mu k(y)}{p(y)} \right) p(y) \nu(dy)$$

- p is a **nonnegative measurable function** defined on (Y, \mathcal{Y})
- M is a subset of $M_1(T)$, the space of probability measures on T
- $K : (\theta, A) \mapsto \int_A k(\theta, y) \nu(dy)$ is a Markov transition kernel defined on $T \times \mathcal{Y}$ with density k

The (α, Γ) -descent algorithm

Let $\mu_1 \in M_1(T)$ be such that $\Psi_{\alpha}(\mu_1 k) < \infty$. The sequence of probability measures $(\mu_n)_{n \geq 1}$ is defined iteratively by

$$\mu_{n+1} = \mathcal{I}_{\alpha}(\mu_n), \quad n \geq 1$$

where for all $\mu \in M_1(T)$ and all $\theta \in T$,

$$\mathcal{I}_{\alpha}(\mu)(d\theta) = \frac{\mu(d\theta) \cdot \Gamma(b_{\mu, \alpha}(\theta) + \kappa)}{\mu(\Gamma(b_{\mu, \alpha} + \kappa))} \quad \text{with} \quad b_{\mu, \alpha}(\theta) = \int_Y k(\theta, y) f'_{\alpha} \left(\frac{\mu k(y)}{p(y)} \right) \nu(dy)$$

The (α, Γ) -descent algorithm

Optimisation problem

$$\inf_{\mu \in \mathbf{M}} \Psi_{\alpha}(\mu k; p) \quad \text{with} \quad \Psi_{\alpha}(\mu k; p) := \int_{\mathbf{Y}} f_{\alpha} \left(\frac{\mu k(y)}{p(y)} \right) p(y) \nu(dy)$$

- p is a **nonnegative measurable function** defined on $(\mathbf{Y}, \mathcal{Y})$
- \mathbf{M} is a subset of $\mathbf{M}_1(\mathbf{T})$, the space of probability measures on \mathbf{T}
- $K : (\theta, A) \mapsto \int_A k(\theta, y) \nu(dy)$ is a Markov transition kernel defined on $\mathbf{T} \times \mathcal{Y}$ with density k

The (α, Γ) -descent algorithm

Let $\mu_1 \in \mathbf{M}_1(\mathbf{T})$ be such that $\Psi_{\alpha}(\mu_1 k) < \infty$. The sequence of probability measures $(\mu_n)_{n \geq 1}$ is defined iteratively by

$$\mu_{n+1} = \mathcal{I}_{\alpha}(\mu_n), \quad n \geq 1$$

where for all $\mu \in \mathbf{M}_1(\mathbf{T})$ and all $\theta \in \mathbf{T}$,

$$\mathcal{I}_{\alpha}(\mu)(d\theta) = \frac{\mu(d\theta) \cdot \Gamma(b_{\mu, \alpha}(\theta) + \kappa)}{\mu(\Gamma(b_{\mu, \alpha} + \kappa))} \quad \text{with} \quad b_{\mu, \alpha}(\theta) = \int_{\mathbf{Y}} k(\theta, y) f'_{\alpha} \left(\frac{\mu k(y)}{p(y)} \right) \nu(dy)$$

The (α, Γ) -descent algorithm

Optimisation problem

$$\inf_{\mu \in \mathcal{M}} \Psi_{\alpha}(\mu k; p) \quad \text{with} \quad \Psi_{\alpha}(\mu k; p) := \int_{\mathcal{Y}} f_{\alpha} \left(\frac{\mu k(y)}{p(y)} \right) p(y) \nu(dy)$$

- p is a **nonnegative measurable function** defined on $(\mathcal{Y}, \mathcal{Y})$
- \mathcal{M} is a subset of $\mathcal{M}_1(\mathcal{T})$, the space of probability measures on \mathcal{T}
- $K : (\theta, A) \mapsto \int_A k(\theta, y) \nu(dy)$ is a Markov transition kernel defined on $\mathcal{T} \times \mathcal{Y}$ with density k

The (α, Γ) -descent algorithm

Let $\mu_1 \in \mathcal{M}_1(\mathcal{T})$ be such that $\Psi_{\alpha}(\mu_1 k) < \infty$. The sequence of probability measures $(\mu_n)_{n \geq 1}$ is defined iteratively by

$$\mu_{n+1} = \mathcal{I}_{\alpha}(\mu_n), \quad n \geq 1$$

where for all $\mu \in \mathcal{M}_1(\mathcal{T})$ and all $\theta \in \mathcal{T}$,

$$\mathcal{I}_{\alpha}(\mu)(d\theta) = \frac{\mu(d\theta) \cdot \Gamma(b_{\mu, \alpha}(\theta) + \kappa)}{\mu(\Gamma(b_{\mu, \alpha} + \kappa))} \quad \text{with} \quad b_{\mu, \alpha}(\theta) = \int_{\mathcal{Y}} k(\theta, y) f'_{\alpha} \left(\frac{\mu k(y)}{p(y)} \right) \nu(dy)$$

The (α, Γ) -descent algorithm

Optimisation problem

$$\inf_{\mu \in \mathcal{M}} \Psi_{\alpha}(\mu k; p) \quad \text{with} \quad \Psi_{\alpha}(\mu k; p) := \int_{\mathcal{Y}} f_{\alpha} \left(\frac{\mu k(y)}{p(y)} \right) p(y) \nu(dy)$$

- p is a **nonnegative measurable function** defined on $(\mathcal{Y}, \mathcal{Y})$
- \mathcal{M} is a subset of $\mathcal{M}_1(\mathcal{T})$, the space of probability measures on \mathcal{T}
- $K : (\theta, A) \mapsto \int_A k(\theta, y) \nu(dy)$ is a Markov transition kernel defined on $\mathcal{T} \times \mathcal{Y}$ with density k

The (α, Γ) -descent algorithm

Let $\mu_1 \in \mathcal{M}_1(\mathcal{T})$ be such that $\Psi_{\alpha}(\mu_1 k) < \infty$. The sequence of probability measures $(\mu_n)_{n \geq 1}$ is defined iteratively by

$$\mu_{n+1} = \mathcal{I}_{\alpha}(\mu_n), \quad n \geq 1$$

where for all $\mu \in \mathcal{M}_1(\mathcal{T})$ and all $\theta \in \mathcal{T}$,

$$\mathcal{I}_{\alpha}(\mu)(d\theta) = \frac{\mu(d\theta) \cdot \Gamma(b_{\mu, \alpha}(\theta) + \kappa)}{\mu(\Gamma(b_{\mu, \alpha} + \kappa))} \quad \text{with} \quad b_{\mu, \alpha}(\theta) = \int_{\mathcal{Y}} k(\theta, y) f'_{\alpha} \left(\frac{\mu k(y)}{p(y)} \right) \nu(dy)$$

The (α, Γ) -descent algorithm

Optimisation problem

$$\inf_{\mu \in \mathcal{M}} \Psi_{\alpha}(\mu k; p) \quad \text{with} \quad \Psi_{\alpha}(\mu k; p) := \int_{\mathcal{Y}} f_{\alpha} \left(\frac{\mu k(y)}{p(y)} \right) p(y) \nu(dy)$$

- p is a **nonnegative measurable function** defined on $(\mathcal{Y}, \mathcal{Y})$
- \mathcal{M} is a subset of $\mathcal{M}_1(\mathcal{T})$, the space of probability measures on \mathcal{T}
- $K : (\theta, A) \mapsto \int_A k(\theta, y) \nu(dy)$ is a Markov transition kernel defined on $\mathcal{T} \times \mathcal{Y}$ with density k

The (α, Γ) -descent algorithm

Let $\mu_1 \in \mathcal{M}_1(\mathcal{T})$ be such that $\Psi_{\alpha}(\mu_1 k) < \infty$. The sequence of probability measures $(\mu_n)_{n \geq 1}$ is defined iteratively by

$$\mu_{n+1} = \mathcal{I}_{\alpha}(\mu_n), \quad n \geq 1$$

where for all $\mu \in \mathcal{M}_1(\mathcal{T})$ and all $\theta \in \mathcal{T}$,

$$\mathcal{I}_{\alpha}(\mu)(d\theta) = \frac{\mu(d\theta) \cdot \Gamma(b_{\mu, \alpha}(\theta) + \kappa)}{\mu(\Gamma(b_{\mu, \alpha} + \kappa))} \quad \text{with} \quad b_{\mu, \alpha}(\theta) = \int_{\mathcal{Y}} k(\theta, y) f'_{\alpha} \left(\frac{\mu k(y)}{p(y)} \right) \nu(dy)$$

Conditions for a monotonic decrease

(A1) For all $(\theta, y) \in \mathcal{T} \times \mathcal{Y}$, $k(\theta, y) > 0$, $p(y) \geq 0$ and $\int_{\mathcal{Y}} p(y) \nu(dy) < \infty$.

(A2) The function $\Gamma : \text{Dom}_{\alpha} \rightarrow \mathbb{R}_{>0}$ is decreasing, continuously differentiable and satisfies the inequality

$$[(\alpha - 1)(v - \kappa) + 1] (\log \Gamma)'(v) + 1 \geq 0 .$$

Theorem

Assume (A1) and (A2). Let $\mu \in \mathcal{M}_1(\mathcal{T})$ be such that $\Psi_{\alpha}(\mu k) < \infty$ and $\mu(\Gamma(b_{\mu, \alpha} + \kappa)) < \infty$. Then,

- ① $\Psi_{\alpha}(\mathcal{I}_{\alpha}(\mu)k) \leq \Psi_{\alpha}(\mu k)$
- ② $\Psi_{\alpha}(\mathcal{I}_{\alpha}(\mu)k) = \Psi_{\alpha}(\mu k)$ if and only if $\mu = \mathcal{I}_{\alpha}(\mu)$

Conditions for a monotonic decrease

(A1) For all $(\theta, y) \in \mathcal{T} \times \mathcal{Y}$, $k(\theta, y) > 0$, $p(y) \geq 0$ and $\int_{\mathcal{Y}} p(y) \nu(dy) < \infty$.

(A2) The function $\Gamma : \text{Dom}_{\alpha} \rightarrow \mathbb{R}_{>0}$ is decreasing, continuously differentiable and satisfies the inequality

$$[(\alpha - 1)(v - \kappa) + 1] (\log \Gamma)'(v) + 1 \geq 0 .$$

Theorem

Assume (A1) and (A2). Let $\mu \in \mathcal{M}_1(\mathcal{T})$ be such that $\Psi_{\alpha}(\mu k) < \infty$ and $\mu(\Gamma(b_{\mu, \alpha} + \kappa)) < \infty$. Then,

- ① $\Psi_{\alpha}(\mathcal{I}_{\alpha}(\mu)k) \leq \Psi_{\alpha}(\mu k)$
- ② $\Psi_{\alpha}(\mathcal{I}_{\alpha}(\mu)k) = \Psi_{\alpha}(\mu k)$ if and only if $\mu = \mathcal{I}_{\alpha}(\mu)$

Conditions for a monotonic decrease

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

(A2) The function $\Gamma : \text{Dom}_{\alpha} \rightarrow \mathbb{R}_{>0}$ is decreasing, continuously differentiable and satisfies the inequality

$$[(\alpha - 1)(v - \kappa) + 1] (\log \Gamma)'(v) + 1 \geq 0 .$$

Theorem

Assume (A1) and (A2). Let $\mu \in \mathsf{M}_1(\mathsf{T})$ be such that $\Psi_{\alpha}(\mu k) < \infty$ and $\mu(\Gamma(b_{\mu, \alpha} + \kappa)) < \infty$. Then,

- ❶ $\Psi_{\alpha}(\mathcal{I}_{\alpha}(\mu)k) \leq \Psi_{\alpha}(\mu k)$
- ❷ $\Psi_{\alpha}(\mathcal{I}_{\alpha}(\mu)k) = \Psi_{\alpha}(\mu k)$ if and only if $\mu = \mathcal{I}_{\alpha}(\mu)$

Examples satisfying (A2)

(A2) The function $\Gamma : \text{Dom}_\alpha \rightarrow \mathbb{R}_{>0}$ is decreasing, continuously differentiable and satisfies the inequality

$$[(\alpha - 1)(v - \kappa) + 1] (\log \Gamma)'(v) + 1 \geq 0 .$$

- Entropic Mirror Descent : $\eta \in (0, 1]$, $\kappa \in \mathbb{R}$ and $\alpha = 1$

$$\Gamma(v) = e^{-\eta v}$$

$$\mu_{n+1}(\mathrm{d}\theta) \propto \mu_n(\mathrm{d}\theta) \exp \left[-\eta \int_Y k(\theta, y) \log \left(\frac{\mu_n k(y)}{p(y)} \right) \nu(\mathrm{d}y) \right]$$

- Power descent : $\eta \in (0, 1]$, $(\alpha - 1)\kappa \geq 0$ and $\alpha \neq 1$

$$\Gamma(v) = [(\alpha - 1)v + 1]^{\frac{\eta}{1-\alpha}}$$

$$\mu_{n+1}(\mathrm{d}\theta) \propto \mu_n(\mathrm{d}\theta) \left[\int_Y k(\theta, y) \left(\frac{\mu_n k(y)}{p(y)} \right)^{\alpha-1} \nu(\mathrm{d}y) + (\alpha - 1)\kappa \right]^{\frac{\eta}{1-\alpha}}$$

Examples satisfying (A2)

(A2) The function $\Gamma : \text{Dom}_\alpha \rightarrow \mathbb{R}_{>0}$ is decreasing, continuously differentiable and satisfies the inequality

$$[(\alpha - 1)(v - \kappa) + 1] (\log \Gamma)'(v) + 1 \geq 0 .$$

- Entropic Mirror Descent : $\eta \in (0, 1]$, $\kappa \in \mathbb{R}$ and $\alpha = 1$

$$\Gamma(v) = e^{-\eta v}$$

$$\mu_{n+1}(\mathrm{d}\theta) \propto \mu_n(\mathrm{d}\theta) \exp \left[-\eta \int_Y k(\theta, y) \log \left(\frac{\mu_n k(y)}{p(y)} \right) \nu(\mathrm{d}y) \right]$$

- Power descent : $\eta \in (0, 1]$, $(\alpha - 1)\kappa \geq 0$ and $\alpha \neq 1$

$$\Gamma(v) = [(\alpha - 1)v + 1]^{\frac{\eta}{1-\alpha}}$$

$$\mu_{n+1}(\mathrm{d}\theta) \propto \mu_n(\mathrm{d}\theta) \left[\int_Y k(\theta, y) \left(\frac{\mu_n k(y)}{p(y)} \right)^{\alpha-1} \nu(\mathrm{d}y) + (\alpha - 1)\kappa \right]^{\frac{\eta}{1-\alpha}}$$

Examples satisfying (A2)

(A2) The function $\Gamma : \text{Dom}_\alpha \rightarrow \mathbb{R}_{>0}$ is decreasing, continuously differentiable and satisfies the inequality

$$[(\alpha - 1)(v - \kappa) + 1] (\log \Gamma)'(v) + 1 \geq 0 .$$

- Entropic Mirror Descent : $\eta \in (0, 1]$, $\kappa \in \mathbb{R}$ and $\alpha = 1$

$$\Gamma(v) = e^{-\eta v}$$

$$\mu_{n+1}(\mathrm{d}\theta) \propto \mu_n(\mathrm{d}\theta) \exp \left[-\eta \int_Y k(\theta, y) \log \left(\frac{\mu_n k(y)}{p(y)} \right) \nu(\mathrm{d}y) \right]$$

- Power descent : $\eta \in (0, 1]$, $(\alpha - 1)\kappa \geq 0$ and $\alpha \neq 1$

$$\Gamma(v) = [(\alpha - 1)v + 1]^{\frac{\eta}{1-\alpha}}$$

$$\mu_{n+1}(\mathrm{d}\theta) \propto \mu_n(\mathrm{d}\theta) \left[\int_Y k(\theta, y) \left(\frac{\mu_n k(y)}{p(y)} \right)^{\alpha-1} \nu(\mathrm{d}y) + (\alpha - 1)\kappa \right]^{\frac{\eta}{1-\alpha}}$$

Convergence results

Algorithm	Convergence results
<i>Entropic Mirror Descent</i>	
$\eta \in (0, \frac{1}{ \alpha-1 b _{\infty, \alpha+1}}), \kappa \in \mathbb{R}$	
<i>Power Descent</i>	
$\eta \in (0, 1], (\alpha - 1)\kappa \geq 0$	

Convergence results

Algorithm	Convergence results
<i>Entropic Mirror Descent</i> $\eta \in (0, \frac{1}{ \alpha-1 b _{\infty, \alpha+1}}), \kappa \in \mathbb{R}$	$O(1/N)$ convergence rates
<i>Power Descent</i> $\eta \in (0, 1], (\alpha - 1)\kappa \geq 0$	

Convergence results

Algorithm	Convergence results
<i>Entropic Mirror Descent</i> $\eta \in (0, \frac{1}{ \alpha-1 b _{\infty, \alpha+1}}), \kappa \in \mathbb{R}$	$O(1/N)$ convergence rates
<i>Power Descent</i> $\eta \in (0, 1], (\alpha - 1)\kappa \geq 0$	$\alpha > 1 : O(1/N)$ convergence rates

Convergence results

Algorithm	Convergence results
<i>Entropic Mirror Descent</i> $\eta \in (0, \frac{1}{ \alpha-1 b _{\infty, \alpha+1}}), \kappa \in \mathbb{R}$	$O(1/N)$ convergence rates
<i>Power Descent</i> $\eta \in (0, 1], (\alpha - 1)\kappa \geq 0$	$\alpha > 1$: $O(1/N)$ convergence rates $\alpha < 1$: convergence toward the optimum

Convergence results

Algorithm	Convergence results
<i>Entropic Mirror Descent</i> $\eta \in (0, \frac{1}{ \alpha-1 b _{\infty, \alpha}+1}), \kappa \in \mathbb{R}$	$O(1/N)$ convergence rates
<i>Power Descent</i> $\eta \in (0, 1], (\alpha - 1)\kappa \geq 0$	$\alpha > 1$: $O(1/N)$ convergence rates $\alpha < 1$: convergence toward the optimum

→ **Minimal** assumptions ensuring a systematic decrease

Convergence results

Algorithm	Convergence results
<i>Entropic Mirror Descent</i> $\eta \in (0, \frac{1}{ \alpha-1 b _{\infty, \alpha+1}}), \kappa \in \mathbb{R}$	$O(1/N)$ convergence rates
<i>Power Descent</i> $\eta \in (0, 1], (\alpha - 1)\kappa \geq 0$	$\alpha > 1$: $O(1/N)$ convergence rates $\alpha < 1$: convergence toward the optimum

→ **Minimal** assumptions ensuring a systematic decrease

→ No β -smoothness assumption

Convergence results

Algorithm	Convergence results
<i>Entropic Mirror Descent</i> $\eta \in (0, \frac{1}{ \alpha-1 b _{\infty, \alpha}+1}), \kappa \in \mathbb{R}$	$O(1/N)$ convergence rates
<i>Power Descent</i> $\eta \in (0, 1], (\alpha - 1)\kappa \geq 0$	$\alpha > 1$: $O(1/N)$ convergence rates $\alpha < 1$: convergence toward the optimum

→ **Minimal** assumptions ensuring a systematic decrease

→ No β -smoothness assumption

Mixture weights optimisation for Alpha-Divergence Variational Inference.

K. Daudel and R. Douc (2021). To appear in NeurIPS2021

Convergence results

Algorithm	Convergence results
<i>Entropic Mirror Descent</i> $\eta \in (0, \frac{1}{ \alpha-1 b _{\infty, \alpha+1}}), \kappa \in \mathbb{R}$	$O(1/N)$ convergence rates
<i>Power Descent</i> $\eta \in (0, 1], (\alpha - 1)\kappa \geq 0$	$\alpha > 1$: $O(1/N)$ convergence rates $\alpha < 1$: convergence toward the optimum

→ **Minimal** assumptions ensuring a systematic decrease

→ No β -smoothness assumption

Mixture weights optimisation for Alpha-Divergence Variational Inference.

K. Daudel and R. Douc (2021). To appear in NeurIPS2021

→ **Extension** of the Power Descent to the case $\alpha = 1$

Convergence results

Algorithm	Convergence results
<i>Entropic Mirror Descent</i> $\eta \in (0, \frac{1}{ \alpha-1 b _{\infty, \alpha}+1}), \kappa \in \mathbb{R}$	$O(1/N)$ convergence rates
<i>Power Descent</i> $\eta \in (0, 1], (\alpha - 1)\kappa \geq 0$	$\alpha > 1$: $O(1/N)$ convergence rates $\alpha < 1$: convergence toward the optimum

→ **Minimal** assumptions ensuring a systematic decrease

→ No β -smoothness assumption

Mixture weights optimisation for Alpha-Divergence Variational Inference.

K. Daudel and R. Douc (2021). To appear in NeurIPS2021

→ **Extension** of the Power Descent to the case $\alpha = 1$

→ **Full proof** of convergence for mixture models ($\alpha < 1$)

Convergence results

Algorithm	Convergence results
<i>Entropic Mirror Descent</i> $\eta \in (0, \frac{1}{ \alpha-1 b _{\infty, \alpha}+1}), \kappa \in \mathbb{R}$	$O(1/N)$ convergence rates
<i>Power Descent</i> $\eta \in (0, 1], (\alpha - 1)\kappa \geq 0$	$\alpha > 1$: $O(1/N)$ convergence rates $\alpha < 1$: convergence toward the optimum

→ **Minimal** assumptions ensuring a systematic decrease

→ No β -smoothness assumption

Mixture weights optimisation for Alpha-Divergence Variational Inference.

K. Daudel and R. Douc (2021). To appear in NeurIPS2021

→ **Extension** of the Power Descent to the case $\alpha = 1$

→ **Full proof** of convergence for mixture models ($\alpha < 1$)

→ Closely-related algorithm : **Rényi Descent**

The special case of mixture models

$$\mathcal{S}_J = \left\{ \boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_J) \in \mathbb{R}^J : \forall j \in \{1, \dots, J\}, \lambda_j \geq 0 \text{ and } \sum_{j=1}^J \lambda_j = 1 \right\}$$

Let $\theta_1, \dots, \theta_J \in \mathcal{T}$ be **fixed** and denote

$$\mu_{\boldsymbol{\lambda}} = \sum_{j=1}^J \lambda_j \delta_{\theta_j} \quad \text{where } \boldsymbol{\lambda} \in \mathcal{S}_J.$$

Then, $\mu_n = \underbrace{\mathcal{I}_{\alpha} \circ \dots \circ \mathcal{I}_{\alpha}}_{n \text{ times}}(\mu_{\boldsymbol{\lambda}})$ is of the form $\mu_n = \sum_{j=1}^J \lambda_{j,n} \delta_{\theta_j}$ with

$$\begin{cases} \lambda_1 = \lambda \\ \lambda_{j,n+1} = \frac{\lambda_{j,n} \Gamma(b_{\mu_n, \alpha}(\theta_j) + \kappa)}{\sum_{i=1}^J \lambda_{i,n} \Gamma(b_{\mu_n, \alpha}(\theta_i) + \kappa)}. \end{cases}$$

→ In practice, we use Monte Carlo approximations to estimate $b_{\mu_n, \alpha}(\theta_j)$, e.g.

$$\hat{b}_{\mu_n, \alpha, M}(\theta_j) = \frac{1}{M} \sum_{m=1}^M \frac{k(\theta_j, Y_{m,n})}{\mu_n k(Y_{m,n})} f'_{\alpha} \left(\frac{\mu_n k(Y_{m,n})}{p(Y_{m,n})} \right),$$

with $Y_{1,n}, \dots, Y_{M,n} \stackrel{\text{i.i.d.}}{\sim} \mu_n k$.

→ NB : **Exploitation step** that does not require any information on the distribution of $\{\theta_1, \dots, \theta_J\}$

The special case of mixture models

$$\mathcal{S}_J = \left\{ \boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_J) \in \mathbb{R}^J : \forall j \in \{1, \dots, J\}, \lambda_j \geq 0 \text{ and } \sum_{j=1}^J \lambda_j = 1 \right\}$$

Let $\theta_1, \dots, \theta_J \in \mathcal{T}$ be **fixed** and denote

$$\mu_{\boldsymbol{\lambda}} = \sum_{j=1}^J \lambda_j \delta_{\theta_j} \quad \text{where } \boldsymbol{\lambda} \in \mathcal{S}_J.$$

Then, $\mu_n = \underbrace{\mathcal{I}_{\alpha} \circ \dots \circ \mathcal{I}_{\alpha}}_{n \text{ times}}(\mu_{\boldsymbol{\lambda}})$ is of the form $\mu_n = \sum_{j=1}^J \lambda_{j,n} \delta_{\theta_j}$ with

$$\begin{cases} \lambda_1 = \boldsymbol{\lambda} \\ \lambda_{j,n+1} = \frac{\lambda_{j,n} \Gamma(b_{\mu_n, \alpha}(\theta_j) + \kappa)}{\sum_{i=1}^J \lambda_{i,n} \Gamma(b_{\mu_n, \alpha}(\theta_i) + \kappa)}. \end{cases}$$

→ In practice, we use Monte Carlo approximations to estimate $b_{\mu_n, \alpha}(\theta_j)$, e.g.

$$\hat{b}_{\mu_n, \alpha, M}(\theta_j) = \frac{1}{M} \sum_{m=1}^M \frac{k(\theta_j, Y_{m,n})}{\mu_n k(Y_{m,n})} f'_{\alpha} \left(\frac{\mu_n k(Y_{m,n})}{p(Y_{m,n})} \right),$$

with $Y_{1,n}, \dots, Y_{M,n} \stackrel{\text{i.i.d.}}{\sim} \mu_n k$.

→ NB : **Exploitation step** that does not require any information on the distribution of $\{\theta_1, \dots, \theta_J\}$

The special case of mixture models

$$S_J = \left\{ \boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_J) \in \mathbb{R}^J : \forall j \in \{1, \dots, J\}, \lambda_j \geq 0 \text{ and } \sum_{j=1}^J \lambda_j = 1 \right\}$$

Let $\theta_1, \dots, \theta_J \in \mathcal{T}$ be **fixed** and denote

$$\mu_{\boldsymbol{\lambda}} = \sum_{j=1}^J \lambda_j \delta_{\theta_j} \quad \text{where } \boldsymbol{\lambda} \in S_J.$$

Then, $\mu_n = \underbrace{\mathcal{I}_{\alpha} \circ \dots \circ \mathcal{I}_{\alpha}}_{n \text{ times}}(\mu_{\boldsymbol{\lambda}})$ is of the form $\mu_n = \sum_{j=1}^J \lambda_{j,n} \delta_{\theta_j}$ with

$$\begin{cases} \lambda_1 = \boldsymbol{\lambda} \\ \lambda_{j,n+1} = \frac{\lambda_{j,n} \Gamma(b_{\mu_n, \alpha}(\theta_j) + \kappa)}{\sum_{i=1}^J \lambda_{i,n} \Gamma(b_{\mu_n, \alpha}(\theta_i) + \kappa)}. \end{cases}$$

→ In practice, we use Monte Carlo approximations to estimate $b_{\mu_n, \alpha}(\theta_j)$, e.g.

$$\hat{b}_{\mu_n, \alpha, M}(\theta_j) = \frac{1}{M} \sum_{m=1}^M \frac{k(\theta_j, Y_{m,n})}{\mu_n k(Y_{m,n})} f'_{\alpha} \left(\frac{\mu_n k(Y_{m,n})}{p(Y_{m,n})} \right),$$

with $Y_{1,n}, \dots, Y_{M,n} \stackrel{\text{i.i.d.}}{\sim} \mu_n k$.

→ NB : **Exploitation step** that does not require any information on the distribution of $\{\theta_1, \dots, \theta_J\}$

The special case of mixture models

$$S_J = \left\{ \boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_J) \in \mathbb{R}^J : \forall j \in \{1, \dots, J\}, \lambda_j \geq 0 \text{ and } \sum_{j=1}^J \lambda_j = 1 \right\}$$

Let $\theta_1, \dots, \theta_J \in \mathcal{T}$ be **fixed** and denote

$$\mu_{\boldsymbol{\lambda}} = \sum_{j=1}^J \lambda_j \delta_{\theta_j} \quad \text{where } \boldsymbol{\lambda} \in S_J.$$

Then, $\mu_n = \underbrace{\mathcal{I}_{\alpha} \circ \dots \circ \mathcal{I}_{\alpha}}_{n \text{ times}}(\mu_{\boldsymbol{\lambda}})$ is of the form $\mu_n = \sum_{j=1}^J \lambda_{j,n} \delta_{\theta_j}$ with

$$\begin{cases} \lambda_1 = \boldsymbol{\lambda} \\ \lambda_{j,n+1} = \frac{\lambda_{j,n} \Gamma(b_{\mu_n, \alpha}(\theta_j) + \kappa)}{\sum_{i=1}^J \lambda_{i,n} \Gamma(b_{\mu_n, \alpha}(\theta_i) + \kappa)}. \end{cases}$$

→ In practice, we use Monte Carlo approximations to estimate $b_{\mu_n, \alpha}(\theta_j)$, e.g.

$$\hat{b}_{\mu_n, \alpha, M}(\theta_j) = \frac{1}{M} \sum_{m=1}^M \frac{k(\theta_j, Y_{m,n})}{\mu_n k(Y_{m,n})} f'_{\alpha} \left(\frac{\mu_n k(Y_{m,n})}{p(Y_{m,n})} \right),$$

with $Y_{1,n}, \dots, Y_{M,n} \stackrel{\text{i.i.d}}{\sim} \mu_n k$.

→ NB : **Exploitation step** that does not require any information on the distribution of $\{\theta_1, \dots, \theta_J\}$

Numerical experiments

- Gaussian kernel with density k_h and bandwidth h , $\mathsf{T} = \mathbb{R}^d$

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

Full algorithm

- ① **Exploitation step** : optimise λ using the (α, Γ) -descent.
 - ② **Exploration step** : update Θ (e.g. by sampling under $\mu_{\lambda, \Theta} k_h$, $h \propto J^{-1/(4+d)}$)
- Toy example
 $p(y) = Z \times [0.5\mathcal{N}(y; -2\mathbf{u}_d, \mathbf{I}_d) + 0.5\mathcal{N}(y; 2\mathbf{u}_d, \mathbf{I}_d)], Z = 2$
 - Bayesian Logistic Regression Covertypes dataset (581,012 data points and 54 features)

Numerical experiments

- Gaussian kernel with density k_h and bandwidth h , $\mathsf{T} = \mathbb{R}^d$

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

Full algorithm

- ① **Exploitation step** : optimise λ using the (α, Γ) -descent.
 - ② **Exploration step** : update Θ (e.g. by sampling under $\mu_{\lambda, \Theta} k_h$, $h \propto J^{-1/(4+d)}$)
- Toy example
 $p(y) = Z \times [0.5\mathcal{N}(y; -2\mathbf{u}_d, \mathbf{I}_d) + 0.5\mathcal{N}(y; 2\mathbf{u}_d, \mathbf{I}_d)]$, $Z = 2$
 - Bayesian Logistic Regression Covertypes dataset (581,012 data points and 54 features)

Numerical experiments

- Gaussian kernel with density k_h and bandwidth h , $\mathsf{T} = \mathbb{R}^d$

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

Full algorithm

- ① **Exploitation step** : optimise λ using the (α, Γ) -descent.
- ② **Exploration step** : update Θ (e.g. by sampling under $\mu_{\lambda, \Theta} k_h$, $h \propto J^{-1/(4+d)}$)
- Toy example
 $p(y) = Z \times [0.5\mathcal{N}(y; -2u_d, I_d) + 0.5\mathcal{N}(y; 2u_d, I_d)]$, $Z = 2$
- Bayesian Logistic Regression Covertypes dataset (581,012 data points and 54 features)

Numerical experiments

- Gaussian kernel with density k_h and bandwidth h , $\mathsf{T} = \mathbb{R}^d$

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

Full algorithm

- 1 **Exploitation step** : optimise λ using the (α, Γ) -descent.
 - 2 **Exploration step** : update Θ (e.g. by sampling under $\mu_{\lambda, \Theta} k_h$, $h \propto J^{-1/(4+d)}$)
- Toy example
 $p(y) = Z \times [0.5\mathcal{N}(y; -2u_d, I_d) + 0.5\mathcal{N}(y; 2u_d, I_d)]$, $Z = 2$
 - Bayesian Logistic Regression Covertypes dataset (581,012 data points and 54 features)

Numerical experiments

- Gaussian kernel with density k_h and bandwidth h , $\mathsf{T} = \mathbb{R}^d$

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

Full algorithm

- ❶ **Exploitation step** : optimise λ using the (α, Γ) -descent.
 - ❷ **Exploration step** : update Θ (e.g. by sampling under $\mu_{\lambda, \Theta} k_h$, $h \propto J^{-1/(4+d)}$)
- Toy example
 $p(y) = Z \times [0.5\mathcal{N}(y; -2\mathbf{u}_d, \mathbf{I}_d) + 0.5\mathcal{N}(y; 2\mathbf{u}_d, \mathbf{I}_d)], Z = 2$
 - Bayesian Logistic Regression Covertypes dataset (581,012 data points and 54 features)

Numerical experiments

- Gaussian kernel with density k_h and bandwidth h , $\mathsf{T} = \mathbb{R}^d$

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

Full algorithm

- ① **Exploitation step** : optimise λ using the (α, Γ) -descent.
- ② **Exploration step** : update Θ (e.g. by sampling under $\mu_{\lambda, \Theta} k_h$, $h \propto J^{-1/(4+d)}$)
- Toy example
 $p(y) = Z \times [0.5\mathcal{N}(y; -2\mathbf{u}_d, \mathbf{I}_d) + 0.5\mathcal{N}(y; 2\mathbf{u}_d, \mathbf{I}_d)], Z = 2$
- Bayesian Logistic Regression Covertypes dataset (581,012 data points and 54 features)

Toy example : Entropic Mirror Descent vs Power Descent

Comparison between

- 0.5-Mirror descent : $\Gamma(v) = e^{-\eta v}$ and $\alpha = 0.5$,
- 0.5-Power descent : $\Gamma(v) = [(\alpha - 1)v + 1]^{\eta/(1-\alpha)}$ and $\alpha = 0.5$.

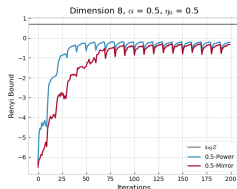
$J = M = 100$, initial mixture weights : $[1/J, \dots, 1/J]$, $N = 10$, $T = 20$
 $\eta_n = \eta_0/\sqrt{n}$, $\eta_0 = 0.5$, cv criterion : VR-Bound averaged over 100 trials

Toy example : Entropic Mirror Descent vs Power Descent

Comparison between

- 0.5-Mirror descent : $\Gamma(v) = e^{-\eta v}$ and $\alpha = 0.5$,
- 0.5-Power descent : $\Gamma(v) = [(\alpha - 1)v + 1]^{\eta/(1-\alpha)}$ and $\alpha = 0.5$.

$J = M = 100$, initial mixture weights : $[1/J, \dots, 1/J]$, $N = 10$, $T = 20$
 $\eta_n = \eta_0/\sqrt{n}$, $\eta_0 = 0.5$, cv criterion : VR-Bound averaged over 100 trials

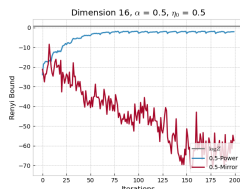
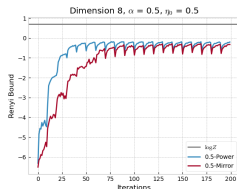


Toy example : Entropic Mirror Descent vs Power Descent

Comparison between

- 0.5-Mirror descent : $\Gamma(v) = e^{-\eta v}$ and $\alpha = 0.5$,
- 0.5-Power descent : $\Gamma(v) = [(\alpha - 1)v + 1]^{\eta/(1-\alpha)}$ and $\alpha = 0.5$.

$J = M = 100$, initial mixture weights : $[1/J, \dots, 1/J]$, $N = 10$, $T = 20$
 $\eta_n = \eta_0/\sqrt{n}$, $\eta_0 = 0.5$, cv criterion : VR-Bound averaged over 100 trials

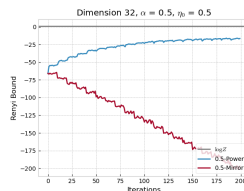
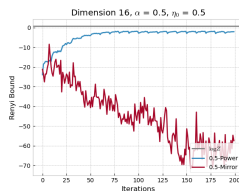
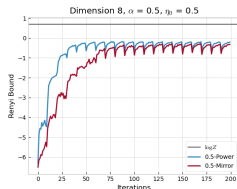


Toy example : Entropic Mirror Descent vs Power Descent

Comparison between

- 0.5-Mirror descent : $\Gamma(v) = e^{-\eta v}$ and $\alpha = 0.5$,
- 0.5-Power descent : $\Gamma(v) = [(\alpha - 1)v + 1]^{\eta/(1-\alpha)}$ and $\alpha = 0.5$.

$J = M = 100$, initial mixture weights : $[1/J, \dots, 1/J]$, $N = 10$, $T = 20$
 $\eta_n = \eta_0/\sqrt{n}$, $\eta_0 = 0.5$, cv criterion : VR-Bound averaged over 100 trials



Toy example : the case $\alpha = 1$

Comparison between:

- 1-Mirror descent : $\Gamma(v) = e^{-\eta v}$ with $\alpha = 1$,
- 0.5-Power descent : $\Gamma(v) = [(\alpha - 1)v + 1]^{\eta/(1-\alpha)}$ with $\alpha = 0.5$.

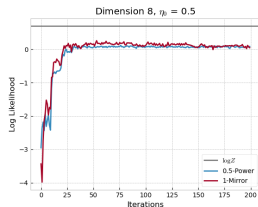
$J = M = 100$, initial mixture weights : $[1/J, \dots, 1/J]$, $N = 10$, $T = 20$
 $\eta_n = \eta_0/\sqrt{n}$, $\eta_0 = 0.5$, cv criterion : llh averaged over 100 trials

Toy example : the case $\alpha = 1$

Comparison between:

- 1-Mirror descent : $\Gamma(v) = e^{-\eta v}$ with $\alpha = 1$,
- 0.5-Power descent : $\Gamma(v) = [(\alpha - 1)v + 1]^{\eta/(1-\alpha)}$ with $\alpha = 0.5$.

$J = M = 100$, initial mixture weights : $[1/J, \dots, 1/J]$, $N = 10$, $T = 20$
 $\eta_n = \eta_0/\sqrt{n}$, $\eta_0 = 0.5$, cv criterion : llh averaged over 100 trials

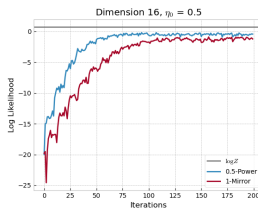
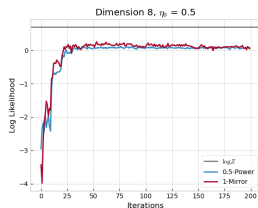


Toy example : the case $\alpha = 1$

Comparison between:

- 1-Mirror descent : $\Gamma(v) = e^{-\eta v}$ with $\alpha = 1$,
- 0.5-Power descent : $\Gamma(v) = [(\alpha - 1)v + 1]^{\eta/(1-\alpha)}$ with $\alpha = 0.5$.

$J = M = 100$, initial mixture weights : $[1/J, \dots, 1/J]$, $N = 10$, $T = 20$
 $\eta_n = \eta_0/\sqrt{n}$, $\eta_0 = 0.5$, cv criterion : llh averaged over 100 trials

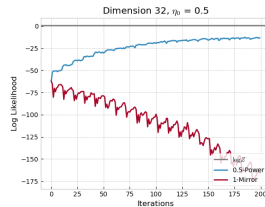
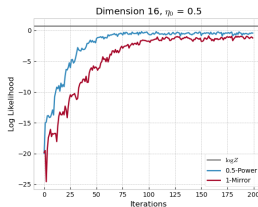
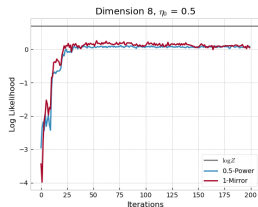


Toy example : the case $\alpha = 1$

Comparison between:

- 1-Mirror descent : $\Gamma(v) = e^{-\eta v}$ with $\alpha = 1$,
- 0.5-Power descent : $\Gamma(v) = [(\alpha - 1)v + 1]^{\eta/(1-\alpha)}$ with $\alpha = 0.5$.

$J = M = 100$, initial mixture weights : $[1/J, \dots, 1/J]$, $N = 10$, $T = 20$
 $\eta_n = \eta_0/\sqrt{n}$, $\eta_0 = 0.5$, cv criterion : llh averaged over 100 trials



Bayesian Logistic Regression

→ $\mathcal{D} = \{\mathbf{c}, \mathbf{x}\} : I$ binary class labels, $c_i \in \{-1, 1\}$, L covariates for each datapoint, $\mathbf{x}_i \in \mathbb{R}^L$

→ Model : L regression coefficients $w_l \in \mathbb{R}$, precision parameter $\beta \in \mathbb{R}^+$

$$p_0(\beta) = \text{Gamma}(\beta; a, b) ,$$

$$p_0(w_l | \beta) = \mathcal{N}(w_l; 0, \beta^{-1}) , \quad 1 \leq l \leq L$$

$$p(c_i = 1 | \mathbf{x}_i, \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}_i}} , \quad 1 \leq i \leq I$$

where $a = 1$ and $b = 0.01$

Nonparametric variational inference S. Gershman, M. Hoffman, and D. Blei (2012). ICML

→ Quantity of interest : $p(y | \mathcal{D})$ with $y = [\mathbf{w}, \log \beta]$

Comparison between

- 0.5-Power descent
- Typical AIS

$N = 1$, $T = 500$, $J_0 = M_0 = 20$, $J_{t+1} = M_{t+1} = J_t + 1$

initial mixture weights : $[1/J_t, \dots, 1/J_t]$, $\eta_n = \eta_0 / \sqrt{n}$ with $\eta_0 = 0.05$

Bayesian Logistic Regression

→ $\mathcal{D} = \{\mathbf{c}, \mathbf{x}\}$: I binary class labels, $c_i \in \{-1, 1\}$, L covariates for each datapoint, $\mathbf{x}_i \in \mathbb{R}^L$

→ Model : L regression coefficients $w_l \in \mathbb{R}$, precision parameter $\beta \in \mathbb{R}^+$

$$p_0(\beta) = \text{Gamma}(\beta; a, b) ,$$

$$p_0(w_l | \beta) = \mathcal{N}(w_l; 0, \beta^{-1}) , \quad 1 \leq l \leq L$$

$$p(c_i = 1 | \mathbf{x}_i, \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}_i}} , \quad 1 \leq i \leq I$$

where $a = 1$ and $b = 0.01$

Nonparametric variational inference S. Gershman, M. Hoffman, and D. Blei (2012). ICML

→ Quantity of interest : $p(y | \mathcal{D})$ with $y = [\mathbf{w}, \log \beta]$

Comparison between

- 0.5-Power descent
- Typical AIS

$N = 1$, $T = 500$, $J_0 = M_0 = 20$, $J_{t+1} = M_{t+1} = J_t + 1$

initial mixture weights : $[1/J_t, \dots, 1/J_t]$, $\eta_n = \eta_0 / \sqrt{n}$ with $\eta_0 = 0.05$

Bayesian Logistic Regression

→ $\mathcal{D} = \{\mathbf{c}, \mathbf{x}\}$: I binary class labels, $c_i \in \{-1, 1\}$, L covariates for each datapoint, $\mathbf{x}_i \in \mathbb{R}^L$

→ Model : L regression coefficients $w_l \in \mathbb{R}$, precision parameter $\beta \in \mathbb{R}^+$

$$p_0(\beta) = \text{Gamma}(\beta; a, b) ,$$

$$p_0(w_l | \beta) = \mathcal{N}(w_l; 0, \beta^{-1}) , \quad 1 \leq l \leq L$$

$$p(c_i = 1 | \mathbf{x}_i, \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}_i}} , \quad 1 \leq i \leq I$$

where $a = 1$ and $b = 0.01$

Nonparametric variational inference S. Gershman, M. Hoffman, and D. Blei (2012). ICML

→ Quantity of interest : $p(y | \mathcal{D})$ with $y = [\mathbf{w}, \log \beta]$

Comparison between

- 0.5-Power descent
- Typical AIS

$N = 1$, $T = 500$, $J_0 = M_0 = 20$, $J_{t+1} = M_{t+1} = J_t + 1$

initial mixture weights : $[1/J_t, \dots, 1/J_t]$, $\eta_n = \eta_0 / \sqrt{n}$ with $\eta_0 = 0.05$

Bayesian Logistic Regression

→ $\mathcal{D} = \{\mathbf{c}, \mathbf{x}\} : I$ binary class labels, $c_i \in \{-1, 1\}$, L covariates for each datapoint, $\mathbf{x}_i \in \mathbb{R}^L$

→ Model : L regression coefficients $w_l \in \mathbb{R}$, precision parameter $\beta \in \mathbb{R}^+$

$$p_0(\beta) = \text{Gamma}(\beta; a, b),$$

$$p_0(w_l | \beta) = \mathcal{N}(w_l; 0, \beta^{-1}), \quad 1 \leq l \leq L$$

$$p(c_i = 1 | \mathbf{x}_i, \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}_i}}, \quad 1 \leq i \leq I$$

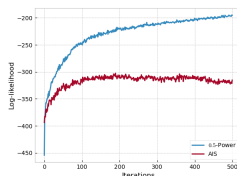
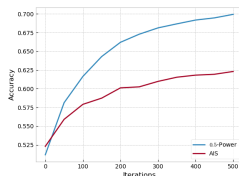
where $a = 1$ and $b = 0.01$

Nonparametric variational inference S. Gershman, M. Hoffman, and D. Blei (2012). ICML

→ Quantity of interest : $p(y | \mathcal{D})$ with $y = [\mathbf{w}, \log \beta]$

Comparison between

- 0.5-Power descent
- Typical AIS



$N = 1$, $T = 500$, $J_0 = M_0 = 20$, $J_{t+1} = M_{t+1} = J_t + 1$

initial mixture weights : $[1/J_t, \dots, 1/J_t]$, $\eta_n = \eta_0 / \sqrt{n}$ with $\eta_0 = 0.05$

At this point...

General framework for infinite-dimensional α -divergence minimisation over

$$\mathcal{Q} = \left\{ q : y \mapsto \int_{\mathcal{T}} \mu(d\theta) k(\theta, y) : \mu \in \mathcal{M} \right\}$$

- recovers the Entropic Mirror Descent algorithm
- novel Power Descent algorithm
- conditions for a systematic decrease + convergence results
- applicable to mixture models :

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

→ Exploitation - Exploration algorithm

- ① No constraint on how to update Θ
- ② Empirical advantages of using the Power Descent algorithm

At this point...

General framework for infinite-dimensional α -divergence minimisation over

$$\mathcal{Q} = \left\{ q : y \mapsto \int_{\mathcal{T}} \mu(d\theta) k(\theta, y) : \mu \in \mathcal{M} \right\}$$

- recovers the **Entropic Mirror Descent** algorithm
- novel **Power Descent** algorithm
- conditions for a **systematic decrease + convergence** results
- applicable to **mixture models** :

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

→ Exploitation - Exploration algorithm

- ① **No constraint** on how to update Θ
- ② Empirical advantages of using the **Power Descent** algorithm

At this point...

General framework for infinite-dimensional α -divergence minimisation over

$$\mathcal{Q} = \left\{ q : y \mapsto \int_{\mathcal{T}} \mu(d\theta) k(\theta, y) : \mu \in \mathcal{M} \right\}$$

- recovers the **Entropic Mirror Descent** algorithm
- novel **Power Descent** algorithm
- conditions for a **systematic decrease + convergence** results
- applicable to **mixture models** :

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

→ Exploitation - Exploration algorithm

- ① **No constraint** on how to update Θ
- ② Empirical advantages of using the **Power Descent** algorithm

At this point...

General framework for infinite-dimensional α -divergence minimisation over

$$\mathcal{Q} = \left\{ q : y \mapsto \int_{\mathcal{T}} \mu(d\theta) k(\theta, y) : \mu \in \mathcal{M} \right\}$$

- recovers the **Entropic Mirror Descent** algorithm
- novel **Power Descent** algorithm
- conditions for a **systematic decrease** + **convergence** results
- applicable to **mixture models** :

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

→ Exploitation - Exploration algorithm

- ① **No constraint** on how to update Θ
- ② Empirical advantages of using the **Power Descent** algorithm

At this point...

General framework for infinite-dimensional α -divergence minimisation over

$$\mathcal{Q} = \left\{ q : y \mapsto \int_{\mathcal{T}} \mu(d\theta) k(\theta, y) : \mu \in \mathcal{M} \right\}$$

- recovers the **Entropic Mirror Descent** algorithm
- novel **Power Descent** algorithm
- conditions for a **systematic decrease** + **convergence** results
- applicable to **mixture models** :

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

→ Exploitation - Exploration algorithm

- ① **No constraint** on how to update Θ
- ② Empirical advantages of using the **Power Descent** algorithm

At this point...

General framework for infinite-dimensional α -divergence minimisation over

$$\mathcal{Q} = \left\{ q : y \mapsto \int_{\mathcal{T}} \mu(d\theta) k(\theta, y) : \mu \in \mathcal{M} \right\}$$

- recovers the **Entropic Mirror Descent** algorithm
- novel **Power Descent** algorithm
- conditions for a **systematic decrease** + **convergence** results
- applicable to **mixture models** :

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

→ Exploitation - Exploration algorithm

- ① **No constraint** on how to update Θ
- ② Empirical advantages of using the **Power Descent** algorithm

At this point...

General framework for infinite-dimensional α -divergence minimisation over

$$\mathcal{Q} = \left\{ q : y \mapsto \int_{\mathcal{T}} \mu(d\theta) k(\theta, y) : \mu \in \mathcal{M} \right\}$$

- recovers the **Entropic Mirror Descent** algorithm
- novel **Power Descent** algorithm
- conditions for a **systematic decrease** + **convergence** results
- applicable to **mixture models** :

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

→ Exploitation - Exploration algorithm

- ❶ **No constraint** on how to update Θ
- ❷ Empirical advantages of using the **Power Descent** algorithm

At this point...

General framework for infinite-dimensional α -divergence minimisation over

$$\mathcal{Q} = \left\{ q : y \mapsto \int_{\mathcal{T}} \mu(d\theta) k(\theta, y) : \mu \in \mathcal{M} \right\}$$

- recovers the **Entropic Mirror Descent** algorithm
- novel **Power Descent** algorithm
- conditions for a **systematic decrease** + **convergence** results
- applicable to **mixture models** :

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

→ Exploitation - Exploration algorithm

- ❶ **No constraint** on how to update Θ
- ❷ Empirical advantages of using the **Power Descent** algorithm

At this point...

General framework for infinite-dimensional α -divergence minimisation over

$$\mathcal{Q} = \left\{ q : y \mapsto \int_{\mathcal{T}} \mu(d\theta) k(\theta, y) : \mu \in \mathcal{M} \right\}$$

- recovers the **Entropic Mirror Descent** algorithm
- novel **Power Descent** algorithm
- conditions for a **systematic decrease** + **convergence** results
- applicable to **mixture models** :

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

→ Exploitation - Exploration algorithm

- ❶ **No constraint** on how to update Θ
- ❷ Empirical advantages of using the **Power Descent** algorithm



Outline

- 1 Introduction
- 2 Infinite-dimensional α -divergence minimisation
- 3 Monotonic α -divergence minimisation**
- 4 Conclusion

Monotonic α -divergence minimisation

Monotonic Alpha-divergence Minimisation.

K. Daudel, R. Douc and F. Roueff (2021). <https://arxiv.org/abs/2103.05684>

Idea : Consider the variational family

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

and propose an update formula for (λ, Θ) that ensures a systematic decrease in the α -divergence at each step

→ Novelty : optimisation w.r.t Θ !

Monotonic α -divergence minimisation

Monotonic Alpha-divergence Minimisation.

K. Daudel, R. Douc and F. Roueff (2021). <https://arxiv.org/abs/2103.05684>

Idea : Consider the variational family

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

and propose an update formula for (λ, Θ) that ensures a systematic decrease in the α -divergence at each step

→ Novelty : optimisation w.r.t Θ !

Conditions for a monotonic decrease

Optimisation problem

$$\inf_{\lambda \in \mathcal{S}_J, \Theta \in \mathcal{T}^J} \Psi_\alpha(\mu_{\lambda, \Theta} k; p) \quad \text{with} \quad \Psi_\alpha(\mu k; p) := \int_Y f_\alpha \left(\frac{\mu k(y)}{p(y)} \right) p(y) \nu(dy)$$

(A1) For all $(\theta, y) \in \mathcal{T} \times Y$, $k(\theta, y) > 0$, $p(y) \geq 0$ and $\int_Y p(y) \nu(dy) < \infty$.

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\lambda_n, \Theta_n)_{n \geq 1}$ so that:
 $\Psi_\alpha(\mu_{\lambda_1, \Theta_1} k) < \infty$ and $\forall n \geq 1$,

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}} \right) \nu(dy) \geq 0 \quad (\text{Weights})$$

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(dy) \geq 0 \quad (\text{Components})$$

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

Conditions for a monotonic decrease

Optimisation problem

$$\inf_{\lambda \in \mathcal{S}_J, \Theta \in \mathcal{T}^J} \Psi_\alpha(\mu_{\lambda, \Theta} k; p) \quad \text{with} \quad \Psi_\alpha(\mu k; p) := \int_Y f_\alpha \left(\frac{\mu k(y)}{p(y)} \right) p(y) \nu(dy)$$

(A1) For all $(\theta, y) \in \mathcal{T} \times Y$, $k(\theta, y) > 0$, $p(y) \geq 0$ and $\int_Y p(y) \nu(dy) < \infty$.

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\lambda_n, \Theta_n)_{n \geq 1}$ so that:
 $\Psi_\alpha(\mu_{\lambda_1, \Theta_1} k) < \infty$ and $\forall n \geq 1$,

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}} \right) \nu(dy) \geq 0 \quad (\text{Weights})$$

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(dy) \geq 0 \quad (\text{Components})$$

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

Conditions for a monotonic decrease

Optimisation problem

$$\inf_{\lambda \in \mathcal{S}_J, \Theta \in \mathcal{T}^J} \Psi_\alpha(\mu_{\lambda, \Theta} k) \quad \text{with} \quad \Psi_\alpha(\mu k) := \int_Y f_\alpha \left(\frac{\mu k(y)}{p(y)} \right) p(y) \nu(dy)$$

(A1) For all $(\theta, y) \in \mathcal{T} \times Y$, $k(\theta, y) > 0$, $p(y) \geq 0$ and $\int_Y p(y) \nu(dy) < \infty$.

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\lambda_n, \Theta_n)_{n \geq 1}$ so that:
 $\Psi_\alpha(\mu_{\lambda_1, \Theta_1} k) < \infty$ and $\forall n \geq 1$,

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}} \right) \nu(dy) \geq 0 \quad (\text{Weights})$$

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(dy) \geq 0 \quad (\text{Components})$$

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

Conditions for a monotonic decrease (2)

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\lambda_n, \Theta_n)_{n \geq 1}$ so that:
 $\Psi_\alpha(\mu_{\lambda_1, \Theta_1} k) < \infty$ and $\forall n \geq 1$,

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}} \right) \nu(dy) \geq 0 \quad (\text{Weights})$$

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(dy) \geq 0 \quad (\text{Components})$$

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

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

② The dependency is simpler in (Weights)

→ (Weights) holds for λ_{n+1} such that

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

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

Conditions for a monotonic decrease (2)

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\lambda_n, \Theta_n)_{n \geq 1}$ so that:
 $\Psi_\alpha(\mu_{\lambda_1, \Theta_1} k) < \infty$ and $\forall n \geq 1$,

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}} \right) \nu(dy) \geq 0 \quad (\text{Weights})$$

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(dy) \geq 0 \quad (\text{Components})$$

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

❶ (Weights) and (Components) permit **simultaneous** updates

❷ The dependency is simpler in (Weights)

→ (Weights) holds for λ_{n+1} such that

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

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

Conditions for a monotonic decrease (2)

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\lambda_n, \Theta_n)_{n \geq 1}$ so that:
 $\Psi_\alpha(\mu_{\lambda_1, \Theta_1} k) < \infty$ and $\forall n \geq 1$,

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}} \right) \nu(dy) \geq 0 \quad (\text{Weights})$$

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(dy) \geq 0 \quad (\text{Components})$$

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

❶ (Weights) and (Components) permit **simultaneous** updates

❷ The dependency is simpler in (Weights)

→ (Weights) holds for λ_{n+1} such that

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

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

Conditions for a monotonic decrease (2)

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\lambda_n, \Theta_n)_{n \geq 1}$ so that:
 $\Psi_\alpha(\mu_{\lambda_1, \Theta_1} k) < \infty$ and $\forall n \geq 1$,

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{\lambda_{j,n+1}}{\lambda_{j,n}} \right) \nu(dy) \geq 0 \quad (\text{Weights})$$

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(dy) \geq 0 \quad (\text{Components})$$

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

❶ (Weights) and (Components) permit **simultaneous** updates

❷ The dependency is simpler in (Weights)

→ (Weights) holds for λ_{n+1} such that

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

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

Understanding the mixture weights update

(Weights) and (Components) hold for λ_{n+1} and Θ_{n+1} such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_Y \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_Y \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\Theta_{n+1} = \Theta_n$$

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

→ We recover the **Power Descent** algorithm from

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

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

<https://doi.org/10.1214/20-AOS2035>.

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 λ_{n+1} and Θ_{n+1} such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_Y \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_Y \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\Theta_{n+1} = \Theta_n$$

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

→ We recover the **Power Descent** algorithm from

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

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

<https://doi.org/10.1214/20-AOS2035>.

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 λ_{n+1} and Θ_{n+1} such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_Y \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_Y \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\Theta_{n+1} = \Theta_n$$

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

→ We recover the **Power Descent** algorithm from

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

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

<https://doi.org/10.1214/20-AOS2035>.

Core insight :

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

Towards simultaneous updates

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(dy) \geq 0 \quad (\text{Components})$$

- Maximisation approach

$$\theta_{j,n+1} = \operatorname{argmax}_{\theta_j \in \mathcal{T}} \int_Y \gamma_{j,\alpha}^n(y) \log(k(\theta_j, y)) \nu(dy), \quad j = 1 \dots J$$

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

$$g_{j,n}(\theta) = c_{j,n} \int_Y \frac{\gamma_{j,\alpha}^n(y)}{\alpha - 1} \log \left(\frac{k(\theta, y)}{k(\theta_{j,n}, y)} \right) \nu(dy).$$

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

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

Towards simultaneous updates

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(dy) \geq 0 \quad (\text{Components})$$

- Maximisation approach

$$\theta_{j,n+1} = \operatorname{argmax}_{\theta_j \in \mathcal{T}} \int_Y \gamma_{j,\alpha}^n(y) \log(k(\theta_j, y)) \nu(dy) , \quad j = 1 \dots J$$

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

$$g_{j,n}(\theta) = c_{j,n} \int_Y \frac{\gamma_{j,\alpha}^n(y)}{\alpha - 1} \log \left(\frac{k(\theta, y)}{k(\theta_{j,n}, y)} \right) \nu(dy) .$$

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

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

Towards simultaneous updates

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(dy) \geq 0 \quad (\text{Components})$$

- Maximisation approach

$$\theta_{j,n+1} = \operatorname{argmax}_{\theta_j \in \mathcal{T}} \int_Y \gamma_{j,\alpha}^n(y) \log(k(\theta_j, y)) \nu(dy), \quad j = 1 \dots J$$

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

$$g_{j,n}(\theta) = c_{j,n} \int_Y \frac{\gamma_{j,\alpha}^n(y)}{\alpha - 1} \log \left(\frac{k(\theta, y)}{k(\theta_{j,n}, y)} \right) \nu(dy).$$

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

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

Towards simultaneous updates

$$\int_Y \sum_{j=1}^J \lambda_{j,n} \gamma_{j,\alpha}^n(y) \log \left(\frac{k(\theta_{j,n+1}, y)}{k(\theta_{j,n}, y)} \right) \nu(dy) \geq 0 \quad (\text{Components})$$

- Maximisation approach

$$\theta_{j,n+1} = \operatorname{argmax}_{\theta_j \in \mathcal{T}} \int_Y \gamma_{j,\alpha}^n(y) \log(k(\theta_j, y)) \nu(dy), \quad j = 1 \dots J$$

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

$$g_{j,n}(\theta) = c_{j,n} \int_Y \frac{\gamma_{j,\alpha}^n(y)}{\alpha - 1} \log \left(\frac{k(\theta, y)}{k(\theta_{j,n}, y)} \right) \nu(dy).$$

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

→ **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 λ_{n+1} and Θ_{n+1} such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_Y \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_Y \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\theta_{j,n+1} = \operatorname{argmax}_{\theta_j \in \mathcal{T}} \int_Y \gamma_{j,\alpha}^n(y) \log(k(\theta_j, y)) \nu(dy), \quad j = 1 \dots J$$

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

→ 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 λ_{n+1} and Θ_{n+1} such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_Y \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_Y \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\theta_{j,n+1} = \operatorname{argmax}_{\theta_j \in \mathcal{T}} \int_Y \gamma_{j,\alpha}^n(y) \log(k(\theta_j, y)) \nu(dy), \quad j = 1 \dots J$$

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

→ 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 λ_{n+1} and Θ_{n+1} such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_Y \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_Y \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\theta_{j,n+1} = \operatorname{argmax}_{\theta_j \in \mathcal{T}} \int_Y \gamma_{j,\alpha}^n(y) \log(k(\theta_j, y)) \nu(dy), \quad j = 1 \dots J$$

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

→ 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 λ_{n+1} and Θ_{n+1} such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_Y \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_Y \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\theta_{j,n+1} = \operatorname{argmax}_{\theta_j \in \mathcal{T}} \int_Y \gamma_{j,\alpha}^n(y) \log(k(\theta_j, y)) \nu(dy), \quad j = 1 \dots J$$

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

→ 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 λ_{n+1} and Θ_{n+1} such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_Y \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_Y \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\theta_{j,n+1} = \operatorname{argmax}_{\theta_j \in \mathcal{T}} \int_Y \gamma_{j,\alpha}^n(y) \log(k(\theta_j, y)) \nu(dy), \quad j = 1 \dots J$$

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

→ 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 λ_{n+1} and Θ_{n+1} such that

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_Y \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_Y \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\theta_{j,n+1} = \operatorname{argmax}_{\theta_j \in \mathcal{T}} \int_Y \gamma_{j,\alpha}^n(y) \log(k(\theta_j, y)) \nu(dy), \quad j = 1 \dots J$$

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

→ 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

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

Algorithm 1: α -divergence minimisation for GMMs

At iteration n ,

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

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

→ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

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

Application to GMMs

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

Algorithm 1: α -divergence minimisation for GMMs

At iteration n ,

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

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

→ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

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

Application to GMMs

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

Algorithm 1: α -divergence minimisation for GMMs

At iteration n ,

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

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

→ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

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

Improving on the M-PMC algorithm

Target : $p(y) = 2 \times [0.5\mathcal{N}(\mathbf{y}; -2\mathbf{u}_d, \mathbf{I}_d) + 0.5\mathcal{N}(\mathbf{y}; 2\mathbf{u}_d, \mathbf{I}_d)]$, $d = 16$

Parameters

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

$$M = 200, J = 100$$

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

→ varying η and κ

Improving on the M-PMC algorithm

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

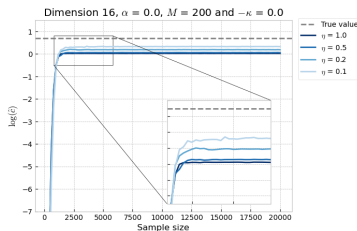
Parameters

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

$$M = 200, J = 100$$

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

→ varying η and κ



Improving on the M-PMC algorithm

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

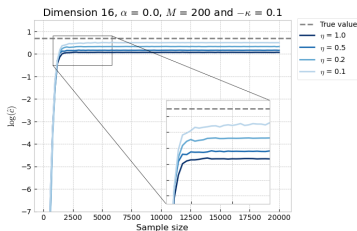
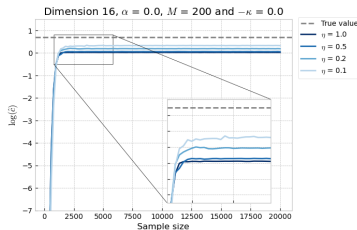
Parameters

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

$$M = 200, J = 100$$

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

→ varying η and κ



Improving on the M-PMC algorithm

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

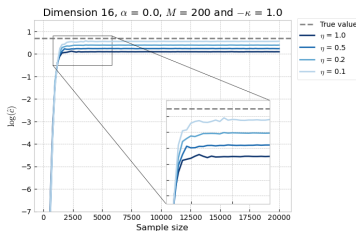
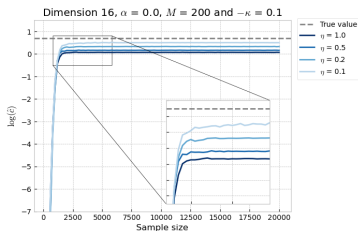
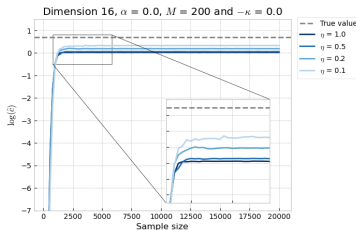
Parameters

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

$$M = 200, J = 100$$

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

→ varying η and κ



Gradient-based approach

Gradient-based approach and Gradient Descent

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_{\mathcal{Y}} \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\theta_{j,n+1} = \theta_{j,n} - \frac{\gamma_{j,n}}{\beta_{j,n}} \nabla g_{j,n}(\theta)|_{\theta=\theta_{j,n}}, \quad j = 1 \dots J$$

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

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

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

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

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

- Rényi's α -divergence minimisation :

$$c_{j,n} = \lambda_{j,n} \left(\int_{\mathcal{Y}} \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy) \right)^{-1}$$

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

→ Solution enabled by our framework : $c_{j,n} = \left(\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy) \right)^{-1}$

Gradient-based approach and Gradient Descent

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_{\mathcal{Y}} \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\theta_{j,n+1} = \theta_{j,n} - \frac{\gamma_{j,n}}{\beta_{j,n}} \nabla g_{j,n}(\theta)|_{\theta=\theta_{j,n}}, \quad j = 1 \dots J$$

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

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

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

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

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

- Rényi's α -divergence minimisation :

$$c_{j,n} = \lambda_{j,n} \left(\int_{\mathcal{Y}} \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy) \right)^{-1}$$

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

→ Solution enabled by our framework : $c_{j,n} = \left(\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy) \right)^{-1}$

Gradient-based approach and Gradient Descent

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_{\mathcal{Y}} \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\theta_{j,n+1} = \theta_{j,n} - \frac{\gamma_{j,n}}{\beta_{j,n}} \nabla g_{j,n}(\theta)|_{\theta=\theta_{j,n}}, \quad j = 1 \dots J$$

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

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

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

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

- **α -divergence** minimisation : $c_{j,n} = \lambda_{j,n}$

- **Rényi's α -divergence** minimisation :

$$c_{j,n} = \lambda_{j,n} (\int_{\mathcal{Y}} \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy))^{-1}$$

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

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

Gradient-based approach and Gradient Descent

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_{\mathcal{Y}} \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\theta_{j,n+1} = \theta_{j,n} - \frac{\gamma_{j,n}}{\beta_{j,n}} \nabla g_{j,n}(\theta)|_{\theta=\theta_{j,n}}, \quad j = 1 \dots J$$

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

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

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

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

- **α -divergence** minimisation : $c_{j,n} = \lambda_{j,n}$

- **Rényi's α -divergence** minimisation :

$$c_{j,n} = \lambda_{j,n} (\int_{\mathcal{Y}} \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy))^{-1}$$

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

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

Gradient-based approach and Gradient Descent

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_{\mathcal{Y}} \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\theta_{j,n+1} = \theta_{j,n} - \frac{\gamma_{j,n}}{\beta_{j,n}} \nabla g_{j,n}(\theta)|_{\theta=\theta_{j,n}}, \quad j = 1 \dots J$$

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

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

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

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

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

- Rényi's α -divergence minimisation :

$$c_{j,n} = \lambda_{j,n} (\int_{\mathcal{Y}} \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy))^{-1}$$

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

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

Gradient-based approach and Gradient Descent

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_{\mathcal{Y}} \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1)\kappa \right]^{\eta_n}}, \quad j = 1 \dots J$$
$$\theta_{j,n+1} = \theta_{j,n} - \frac{\gamma_{j,n}}{\beta_{j,n}} \nabla g_{j,n}(\theta)|_{\theta=\theta_{j,n}}, \quad j = 1 \dots J$$

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

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

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

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

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

- Rényi's α -divergence minimisation :

$$c_{j,n} = \lambda_{j,n} (\int_{\mathcal{Y}} \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy))^{-1}$$

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

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

Application to GMMs (2)

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

- Case 1 : $c_{j,n} = \lambda_{j,n} (\int_Y \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy))^{-1}$ with $\beta_{j,n} = \sigma^{-2} (1 - \alpha)^{-1}$
- Case 2 : $c_{j,n} = (\int_Y \gamma_{j,\alpha}^n(y) \nu(dy))^{-1}$ with $\beta_{j,n} = \sigma^{-2} (1 - \alpha)^{-1}$

→ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

Application to GMMs (2)

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

- Case 1 : $c_{j,n} = \lambda_{j,n} (\int_{\mathcal{Y}} \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy))^{-1}$ with $\beta_{j,n} = \sigma^{-2} (1 - \alpha)^{-1}$
- Case 2 : $c_{j,n} = (\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy))^{-1}$ with $\beta_{j,n} = \sigma^{-2} (1 - \alpha)^{-1}$

→ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

Application to GMMs (2)

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

- Case 1 : $c_{j,n} = \lambda_{j,n} (\int_{\mathcal{Y}} \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy))^{-1}$ with $\beta_{j,n} = \sigma^{-2} (1 - \alpha)^{-1}$
- Case 2 : $c_{j,n} = (\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy))^{-1}$ with $\beta_{j,n} = \sigma^{-2} (1 - \alpha)^{-1}$

→ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

Application to GMMs (2)

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

- Case 1 : $c_{j,n} = \lambda_{j,n} (\int_{\mathcal{Y}} \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy))^{-1}$ with $\beta_{j,n} = \sigma^{-2} (1 - \alpha)^{-1}$
- Case 2 : $c_{j,n} = (\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy))^{-1}$ 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

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1) \kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_{\mathcal{Y}} \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1) \kappa \right]^{\eta_n}}$$
$$\theta_{j,n+1} = \begin{cases} \theta_{j,n} + \gamma_n \frac{\int_{\mathcal{Y}} \lambda_{j,n} \gamma_{j,\alpha}^n(y) (y - \theta_{j,n}) \nu(dy)}{\int_{\mathcal{Y}} \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy)} & \text{(Case 1)} \\ (1 - \gamma_n) \theta_{j,n} + \gamma_n \frac{\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) y \nu(dy)}{\int_{\mathcal{Y}} \gamma_{j,\alpha}^n(y) \nu(dy)} & \text{(Case 2)} \end{cases}$$

→ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

Application to GMMs (2)

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

- Case 1 : $c_{j,n} = \lambda_{j,n} (\int_{\mathbf{Y}} \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy))^{-1}$ with $\beta_{j,n} = \sigma^{-2} (1 - \alpha)^{-1}$
- Case 2 : $c_{j,n} = (\int_{\mathbf{Y}} \gamma_{j,\alpha}^n(y) \nu(dy))^{-1}$ 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

$$\lambda_{j,n+1} = \frac{\lambda_{j,n} \left[\int_{\mathbf{Y}} \gamma_{j,\alpha}^n(y) \nu(dy) + (\alpha - 1) \kappa \right]^{\eta_n}}{\sum_{\ell=1}^J \lambda_{\ell,n} \left[\int_{\mathbf{Y}} \gamma_{\ell,\alpha}^n(y) \nu(dy) + (\alpha - 1) \kappa \right]^{\eta_n}}$$
$$\theta_{j,n+1} = \begin{cases} \theta_{j,n} + \gamma_n \frac{\int_{\mathbf{Y}} \lambda_{j,n} \gamma_{j,\alpha}^n(y) (y - \theta_{j,n}) \nu(dy)}{\int_{\mathbf{Y}} \mu_{\lambda_n, \Theta_n} k(y)^\alpha p(y)^{1-\alpha} \nu(dy)} & \text{(Case 1)} \\ (1 - \gamma_n) \theta_{j,n} + \gamma_n \frac{\int_{\mathbf{Y}} \gamma_{j,\alpha}^n(y) y \nu(dy)}{\int_{\mathbf{Y}} \gamma_{j,\alpha}^n(y) \nu(dy)} & \text{(Case 2)} \end{cases}$$

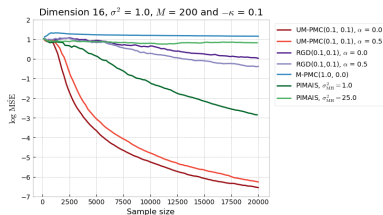
→ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

Improving on Gradient Descent updates

Target : $p(y) = 2 \times [0.5\mathcal{N}(\mathbf{y}; -2\mathbf{u}_d, \mathbf{I}_d) + 0.5\mathcal{N}(\mathbf{y}; 2\mathbf{u}_d, \mathbf{I}_d)]$, $d = 16$

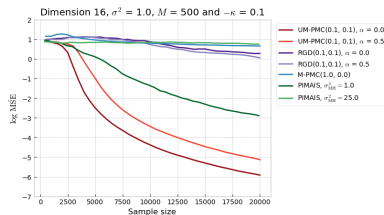
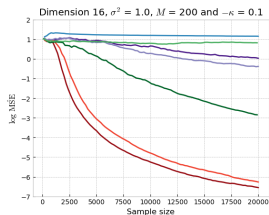
Improving on Gradient Descent updates

Target : $p(y) = 2 \times [0.5\mathcal{N}(\mathbf{y}; -2\mathbf{u}_d, \mathbf{I}_d) + 0.5\mathcal{N}(\mathbf{y}; 2\mathbf{u}_d, \mathbf{I}_d)]$, $d = 16$



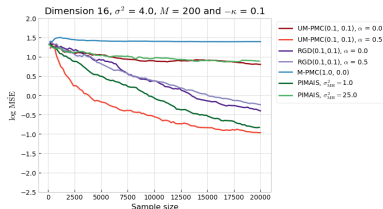
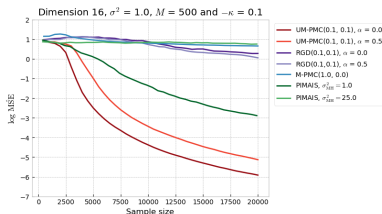
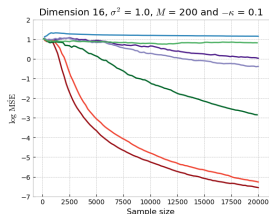
Improving on Gradient Descent updates

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



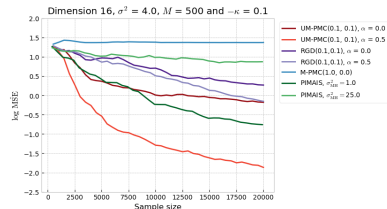
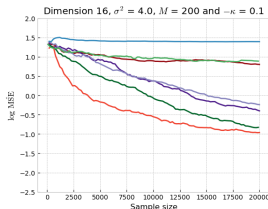
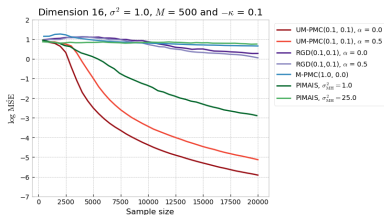
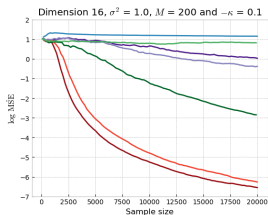
Improving on Gradient Descent updates

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



Improving on Gradient Descent updates

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



Outline

- 1 Introduction
- 2 Infinite-dimensional α -divergence minimisation
- 3 Monotonic α -divergence minimisation
- 4 Conclusion**

Conclusion

Novel framework for **monotonic α -divergence minimisation**

- applicable to **mixture model** optimisation
- enables **simultaneous** updates for mixture weights and mixture components parameters
- **empirical benefits** compared to Entropic Mirror Descent, Gradient Descent and Integrated EM algorithms

Perspectives

- ML applications (suggestions?)
- Additionnal convergence results

Conclusion

Novel framework for **monotonic α -divergence minimisation**

- applicable to **mixture model** optimisation
- enables **simultaneous** updates for mixture weights and mixture components parameters
- **empirical benefits** compared to Entropic Mirror Descent, Gradient Descent and Integrated EM algorithms

Perspectives

- ML applications (suggestions?)
- Additionnal convergence results

Conclusion

Novel framework for **monotonic α -divergence minimisation**

- applicable to **mixture model** optimisation
- enables **simultaneous** updates for mixture weights and mixture components parameters
- **empirical benefits** compared to Entropic Mirror Descent, Gradient Descent and Integrated EM algorithms

Perspectives

- ML applications (suggestions?)
- Additionnal convergence results

Conclusion

Novel framework for **monotonic α -divergence minimisation**

- applicable to **mixture model** optimisation
- enables **simultaneous** updates for mixture weights and mixture components parameters
- **empirical benefits** compared to Entropic Mirror Descent, Gradient Descent and Integrated EM algorithms

Perspectives

- ML applications (suggestions?)
- Additionnal convergence results

Conclusion

Novel framework for **monotonic α -divergence minimisation**

- applicable to **mixture model** optimisation
- enables **simultaneous** updates for mixture weights and mixture components parameters
- **empirical benefits** compared to Entropic Mirror Descent, Gradient Descent and Integrated EM algorithms

Perspectives

- ML applications (suggestions?)
- Additionnal convergence results

Conclusion

Novel framework for **monotonic α -divergence minimisation**

- applicable to **mixture model** optimisation
- enables **simultaneous** updates for mixture weights and mixture components parameters
- **empirical benefits** compared to Entropic Mirror Descent, Gradient Descent and Integrated EM algorithms

Perspectives

- ML applications (suggestions?)
- Additionnal convergence results

Conclusion

Novel framework for **monotonic α -divergence minimisation**

- applicable to **mixture model** optimisation
- enables **simultaneous** updates for mixture weights and mixture components parameters
- **empirical benefits** compared to Entropic Mirror Descent, Gradient Descent and Integrated EM algorithms

Perspectives

- ML applications (suggestions?)
- Additionnal convergence results

Thank you for your attention!

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

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

<https://doi.org/10.1214/20-AOS2035>.

Mixture weights optimisation for Alpha-Divergence Variational Inference.

K. Daudel and R. Douc (2021). To appear in NeurIPS2021

Monotonic Alpha-divergence Minimisation.

K. Daudel, R. Douc and F. Roueff (2021). <https://arxiv.org/abs/2103.05684>