Monotonic Alpha-divergence Variational Inference

Kamélia Daudel

University of Oxford
kamelia.daudel@gmail.com

16/09/2021

Joint work with Randal Douc, François Portier and François Roueff

Outline

Bayesian statistics

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

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

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

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

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

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

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

Bayesian statistics

- Compute / sample from the posterior density of the latent variables y given the data $\mathscr{D}$
$$p(y|\mathscr{D}) = \frac{p(\mathscr{D}, y)}{p(\mathscr{D})} \; .$$

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

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

1. Posit a variational family $\mathcal{Q}$, where $q \in \mathcal{Q}$.
2. Fit q to obtain the best approximation to the posterior density
$$q^\star = \operatorname{arginf}_{q \in \mathcal{Q}} D(\mathbb{Q}||\mathbb{P}_{|\mathscr{D}}) \; ,$$
where D is a measure of dissimilarity between the variational distribution $\mathbb{Q}$ and the posterior distribution $\mathbb{P}_{|\mathscr{D}}$ (typically the KL divergence)

Bayesian statistics

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

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

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

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

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

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

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

Important aspects in Variational Inference

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

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

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

- 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

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

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

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

- 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

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

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

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

- 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

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

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

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

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

- Choice of the measure of dissimilarity D
  $\rightarrow$ Alternative/more general D beyond the KL
- Choice of the approximating family $\mathcal{Q}$
  $\rightarrow$ 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

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

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

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

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

- Choice of the measure of dissimilarity D
  $\rightarrow$ Alternative/more general D beyond the KL
- Choice of the approximating family $\mathcal{Q}$
  $\rightarrow$ 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

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

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

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

- Choice of the measure of dissimilarity D
  $\rightarrow$ Alternative/more general D beyond the KL
- Choice of the approximating family $\mathcal{Q}$
  $\rightarrow$ 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

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

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

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

- Choice of the measure of dissimilarity D
  $\rightarrow$ Alternative/more general D beyond the KL
- Choice of the approximating family $\mathcal{Q}$
  $\rightarrow$ 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

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

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

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

where

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

Variational Inference with the α-divergence family

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

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

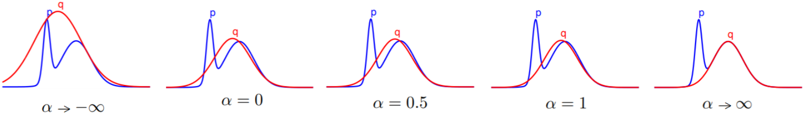
$$D_\alpha(\mathbb{Q}||\mathbb{P}) = \int_{\mathsf{Y}} f_\alpha\left(\frac{q(y)}{p(y)}\right) p(y)\nu(\mathrm{d}y) \ ,$$

where

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

1. A flexible family of divergences...

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



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

Variational Inference with the α-divergence family

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

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

where

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

❶ A flexible family of divergences...

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

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

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

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

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

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

Variational Inference with the α-divergence family

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

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

where

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

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

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

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

Black-box alpha divergence minimization. J. Hernandez-Lobato et al. (2016). ICML
Rényi divergence variational inference. Y. Li and R. E Turner (2016). NeurIPS
Variational inference via χ-upper bound minimization A. Dieng et al. (2017). NeurIPS

Outline

Infinite-dimensional α-divergence minimisation

Infinite-dimensional gradient-based descent for alpha-divergence minimisation.
K. Daudel, R. Douc and F. Portier (2020). To appear in the Annals of Statistics.

Idea : Extend the traditional variational parametric family

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

by putting a prior on the variational parameter θ

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

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

- Hierarchical Variational Inference

Hierarchical variational models. R. Ranganath, D. Tran, and D. Blei (2016). ICML
Semi-Implicit Variational Inference. M. Yin and M. Zhou (2018). ICML

$\rightarrow$ 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 (2020). To appear in the Annals of Statistics.

Idea : Extend the traditional variational parametric family

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

by putting a prior on the variational parameter θ

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

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

- Hierarchical Variational Inference

Hierarchical variational models. R. Ranganath, D. Tran, and D. Blei (2016). ICML
Semi-Implicit Variational Inference. M. Yin and M. Zhou (2018). ICML

$\rightarrow$ 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 (2020). To appear in the Annals of Statistics.

Idea : Extend the traditional variational parametric family

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

by putting a prior on the variational parameter θ

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

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

- Hierarchical Variational Inference

Hierarchical variational models. R. Ranganath, D. Tran, and D. Blei (2016). ICML
Semi-Implicit Variational Inference. M. Yin and M. Zhou (2018). ICML

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

The (α, Γ)-descent algorithm

Optimisation problem

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

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

The (α, Γ)-descent algorithm

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

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

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

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

The (α, Γ)-descent algorithm

Optimisation problem

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

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

The (α, Γ)-descent algorithm

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

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

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

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

The (α, Γ)-descent algorithm

Optimisation problem

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

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

The (α, Γ)-descent algorithm

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

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

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

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

The (α, Γ)-descent algorithm

Optimisation problem

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

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

The (α, Γ)-descent algorithm

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

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

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

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

The (α, Γ)-descent algorithm

Optimisation problem

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

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

The (α, Γ)-descent algorithm

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

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

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

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

The (α, Γ)-descent algorithm

Optimisation problem

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

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

The (α, Γ)-descent algorithm

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

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

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

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

Conditions for a monotonic decrease

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

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

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

Theorem

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

1. $\Psi_\alpha(\mathcal{I}_\alpha(\mu)k) \leqslant \Psi_\alpha(\mu k)$
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 $(\theta, y) \in \mathsf{T} \times \mathsf{Y}$, $k(\theta, y) > 0$, $p(y) \geqslant 0$ and $\int_{\mathsf{Y}} p(y)\nu(\mathrm{d}y) < \infty$.

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

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

Theorem

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

1. $\Psi_\alpha(\mathcal{I}_\alpha(\mu)k) \leqslant \Psi_\alpha(\mu k)$
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 $(\theta, y) \in \mathsf{T} \times \mathsf{Y}$, $k(\theta, y) > 0$, $p(y) \geqslant 0$ and $\int_{\mathsf{Y}} p(y)\nu(\mathrm{d}y) < \infty$.

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

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

Theorem

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

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

Examples satisfying (A2)

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

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

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

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

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

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

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

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

Examples satisfying (A2)

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

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

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

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

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

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

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

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

Examples satisfying (A2)

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

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

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

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

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

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

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

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

Convergence results

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

Convergence results

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

Convergence results

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

Convergence results

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

Convergence results

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

$\rightarrow$ Minimal assumptions ensuring a systematic decrease

Convergence results

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

→ Minimal assumptions ensuring a systematic decrease

→ No β-smoothness assumption

The special case of mixture models

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

Let $\theta_1, ..., \theta_J \in \mathsf{T}$ be fixed and denote

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

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

$$\begin{cases} \boldsymbol{\lambda}_1 = \boldsymbol{\lambda} \\ \lambda_{j,n+1} = \frac{\lambda_{j,n} \Gamma(b_{\mu_n,\alpha}(\theta_j) + \kappa)}{\sum_{i=1}^J \lambda_{i,n} \Gamma(b_{\mu_n,\alpha}(\theta_i) + \kappa)} \ . \end{cases}$$

$\rightarrow$ 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 $Y_{1,n}, ..., Y_{M,n} \overset{\text{i.i.d}}{\sim} \mu_n k$.

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

The special case of mixture models

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

Let $\theta_1, ..., \theta_J \in \mathsf{T}$ be fixed and denote

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

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

$$\begin{cases} \boldsymbol{\lambda}_1 = \boldsymbol{\lambda} \\ \lambda_{j,n+1} = \frac{\lambda_{j,n} \Gamma(b_{\mu_n,\alpha}(\theta_j) + \kappa)}{\sum_{i=1}^J \lambda_{i,n} \Gamma(b_{\mu_n,\alpha}(\theta_i) + \kappa)} \ . \end{cases}$$

$\rightarrow$ 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 $Y_{1,n}, ..., Y_{M,n} \overset{\text{i.i.d}}{\sim} \mu_n k$.

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

The special case of mixture models

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

Let $\theta_1, ..., \theta_J \in \mathsf{T}$ be fixed and denote

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

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

$$\begin{cases} \boldsymbol{\lambda}_1 = \boldsymbol{\lambda} \\ \lambda_{j,n+1} = \frac{\lambda_{j,n} \Gamma(b_{\mu_n,\alpha}(\theta_j) + \kappa)}{\sum_{i=1}^J \lambda_{i,n} \Gamma(b_{\mu_n,\alpha}(\theta_i) + \kappa)}\ . \end{cases}$$

$\rightarrow$ 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 $Y_{1,n}, ..., Y_{M,n} \overset{\text{i.i.d}}{\sim} \mu_n k$.

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

The special case of mixture models

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

Let $\theta_1, ..., \theta_J \in \mathsf{T}$ be fixed and denote

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

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

$$\begin{cases} \boldsymbol{\lambda}_1 = \boldsymbol{\lambda} \\ \lambda_{j,n+1} = \frac{\lambda_{j,n}\Gamma(b_{\mu_n,\alpha}(\theta_j)+\kappa)}{\sum_{i=1}^J \lambda_{i,n}\Gamma(b_{\mu_n,\alpha}(\theta_i)+\kappa)}\ . \end{cases}$$

$\rightarrow$ 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 $Y_{1,n}, ..., Y_{M,n} \overset{\text{i.i.d}}{\sim} \mu_n k$.

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

Numerical experiments

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

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

Full algorithm

1. Exploitation step : optimise $\boldsymbol{\lambda}$ using the (α, Γ)-descent.
2. Exploration step : update Θ (e.g. by sampling under $\mu_{\boldsymbol{\lambda},\Theta} k_h$, $h \propto J^{-1/(4+d)}$)

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

- Bayesian Logistic Regression Covertype dataset ($581,012$ data points and 54 features)

Numerical experiments

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

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

Full algorithm

1. Exploitation step : optimise $\boldsymbol{\lambda}$ using the (α, Γ)-descent.
2. Exploration step : update Θ (e.g. by sampling under $\mu_{\boldsymbol{\lambda},\Theta} k_h$, $h \propto J^{-1/(4+d)}$)

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

- Bayesian Logistic Regression Covertype dataset ($581,012$ data points and 54 features)

Numerical experiments

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

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

Full algorithm

1. Exploitation step : optimise $\boldsymbol{\lambda}$ using the (α, Γ)-descent.
2. Exploration step : update Θ (e.g. by sampling under $\mu_{\boldsymbol{\lambda},\Theta} k_h$, $h \propto J^{-1/(4+d)}$)

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

- Bayesian Logistic Regression Covertype dataset ($581,012$ data points and 54 features)

Numerical experiments

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

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

<u>Full algorithm</u>

1. Exploitation step : optimise $\boldsymbol{\lambda}$ using the (α, Γ)-descent.
2. Exploration step : update Θ (e.g. by sampling under $\mu_{\boldsymbol{\lambda},\Theta} k_h$, $h \propto J^{-1/(4+d)}$)

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

- Bayesian Logistic Regression Covertype dataset ($581,012$ data points and 54 features)

Numerical experiments

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

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

<u>Full algorithm</u>

❶ Exploitation step : optimise $\boldsymbol{\lambda}$ using the (α, Γ)-descent.
❷ Exploration step : update Θ (e.g. by sampling under $\mu_{\boldsymbol{\lambda},\Theta} k_h$, $h \propto J^{-1/(4+d)}$)

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

- Bayesian Logistic Regression Covertype dataset ($581,012$ data points and 54 features)

Numerical experiments

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

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

Full algorithm

1. Exploitation step : optimise $\boldsymbol{\lambda}$ using the (α, Γ)-descent.
2. Exploration step : update Θ (e.g. by sampling under $\mu_{\boldsymbol{\lambda},\Theta} k_h$, $h \propto J^{-1/(4+d)}$)

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

- Bayesian Logistic Regression Covertype 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 $\alpha = 0.5$.

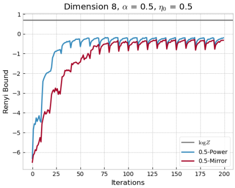
$J = M = 100$, initial mixture weights : $[1/J, ..., 1/J]$, $N = 10$, $T = 20$
$\eta_n = \eta_0/\sqrt{n}$, $\eta_0 = 0.5$, cv criterion : VR-Bound averaged over 100 trials

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

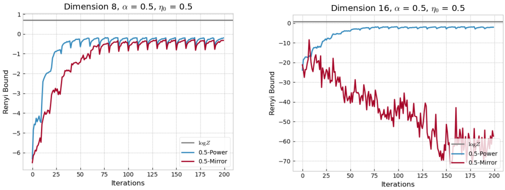


Toy example : Entropic Mirror Descent vs Power Descent

Comparison between

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

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

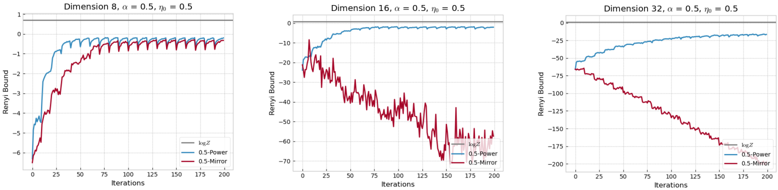


Toy example : Entropic Mirror Descent vs Power Descent

Comparison between

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

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



Toy example : the case $\alpha = 1$

Comparison between:

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

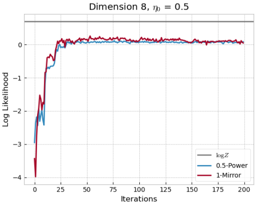
$J = M = 100$, initial mixture weights : $[1/J, ..., 1/J]$, $N = 10$, $T = 20$
$\eta_n = \eta_0/\sqrt{n}$, $\eta_0 = 0.5$, cv criterion : llh averaged over 100 trials

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

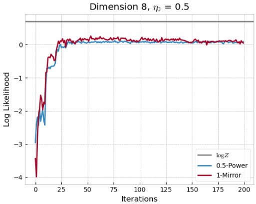


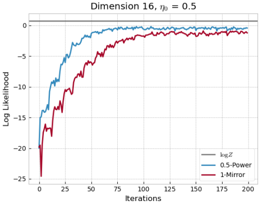
Toy example : the case $\alpha = 1$

Comparison between:

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

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



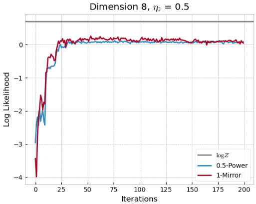


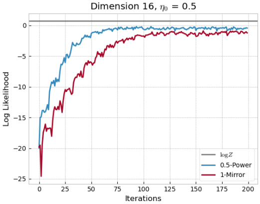
Toy example : the case $\alpha = 1$

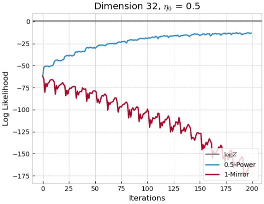
Comparison between:

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

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







Bayesian Logistic Regression

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

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

$$p_0(\beta) = \mathrm{Gamma}(\beta; a, b) \ ,$$
$$p_0(w_l|\beta) = \mathcal{N}(w_l; 0, \beta^{-1}) \ , \quad 1 \leqslant l \leqslant L$$
$$p(c_i = 1|\boldsymbol{x}_i, \boldsymbol{w}) = \frac{1}{1 + e^{-\boldsymbol{w}^T \boldsymbol{x}_i}} \ , \quad 1 \leqslant i \leqslant I$$

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

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

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

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 : $[1/J_t, ..., 1/J_t]$, $\eta_n = \eta_0/\sqrt{n}$ with $\eta_0 = 0.05$

Bayesian Logistic Regression

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

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

$$
\begin{aligned}
p_0(\beta) &= \mathrm{Gamma}(\beta; a, b) \;, \\
p_0(w_l|\beta) &= \mathcal{N}(w_l; 0, \beta^{-1}) \;, \quad 1 \leqslant l \leqslant L \\
p(c_i = 1|\boldsymbol{x}_i, \boldsymbol{w}) &= \frac{1}{1 + e^{-\boldsymbol{w}^T \boldsymbol{x}_i}} \;, \quad 1 \leqslant i \leqslant I
\end{aligned}
$$

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

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

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

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 : $[1/J_t, ..., 1/J_t]$, $\eta_n = \eta_0/\sqrt{n}$ with $\eta_0 = 0.05$

Bayesian Logistic Regression

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

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

$$
\begin{aligned}
p_0(\beta) &= \mathrm{Gamma}(\beta; a, b) \ , \\
p_0(w_l|\beta) &= \mathcal{N}(w_l; 0, \beta^{-1}) \ , \quad 1 \leqslant l \leqslant L \\
p(c_i = 1|\boldsymbol{x}_i, \boldsymbol{w}) &= \frac{1}{1 + e^{-\boldsymbol{w}^T \boldsymbol{x}_i}} \ , \quad 1 \leqslant i \leqslant I
\end{aligned}
$$

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

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

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

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 : $[1/J_t, ..., 1/J_t]$, $\eta_n = \eta_0/\sqrt{n}$ with $\eta_0 = 0.05$

Bayesian Logistic Regression

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

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

$$p_0(\beta) = \mathrm{Gamma}(\beta; a, b) \ ,$$
$$p_0(w_l|\beta) = \mathcal{N}(w_l; 0, \beta^{-1}) \ , \quad 1 \leqslant l \leqslant L$$
$$p(c_i = 1|\boldsymbol{x}_i, \boldsymbol{w}) = \frac{1}{1 + e^{-\boldsymbol{w}^T \boldsymbol{x}_i}} \ , \quad 1 \leqslant i \leqslant I$$

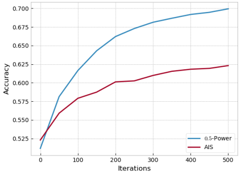
where $a = 1$ and $b = 0.01$

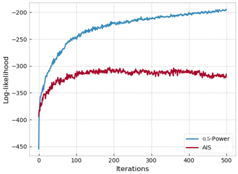
Nonparametric variational inference S. Gershman, M. Hoffman, and D. Blei (2012). ICML

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

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 : $[1/J_t, \ldots, 1/J_t]$, $\eta_n = \eta_0/\sqrt{n}$ with $\eta_0 = 0.05$

At this point...

General framework for infinite-dimensional α-divergence minimisation over

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

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

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

$\rightarrow$ 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

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

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

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

$\to$ 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

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

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

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

$\rightarrow$ 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

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

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

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

$\rightarrow$ 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

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

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

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

$\rightarrow$ 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

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

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

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

$\rightarrow$ 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

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

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

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

$\rightarrow$ 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

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

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

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

$\rightarrow$ 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

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

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

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

$\rightarrow$ Exploitation - Exploration algorithm

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

Outline

Monotonic α-divergence minimisation

Monotonic Alpha-divergence Minimisation.
K. Daudel, R. Douc and F. Roueff (2021). Submitted.

Idea : Consider the variational family

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

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

$\rightarrow$ Novelty : optimisation w.r.t Θ !

Monotonic α-divergence minimisation

Monotonic Alpha-divergence Minimisation.
K. Daudel, R. Douc and F. Roueff (2021). Submitted.

Idea : Consider the variational family

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

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

$\rightarrow$ Novelty : optimisation w.r.t Θ !

Conditions for a monotonic decrease

Optimisation problem

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

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

Theorem

Assume (A1) and let $\alpha \in [0,1)$. Then, choosing $(\boldsymbol{\lambda}_n, \Theta_n)_{n\geqslant 1}$ so that: $\forall n \geqslant 1$,

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

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

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

Conditions for a monotonic decrease

Optimisation problem

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

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

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\boldsymbol{\lambda}_n, \Theta_n)_{n \geqslant 1}$ so that: $\forall n \geqslant 1$,

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

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

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

Conditions for a monotonic decrease

Optimisation problem

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

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

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\boldsymbol{\lambda}_n, \Theta_n)_{n\geqslant 1}$ so that: $\forall n \geqslant 1$,

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

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

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

Conditions for a monotonic decrease (2)

Theorem

Assume (A1) and let $\alpha \in [0,1)$. Then, choosing $(\boldsymbol{\lambda}_n, \Theta_n)_{n\geqslant 1}$ so that: $\forall n \geqslant 1$,

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

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

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

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

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

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

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

Conditions for a monotonic decrease (2)

Theorem

Assume (A1) and let $\alpha \in [0, 1)$. Then, choosing $(\boldsymbol{\lambda}_n, \Theta_n)_{n \geqslant 1}$ so that: $\forall n \geqslant 1$,

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

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

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

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

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

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

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

Conditions for a monotonic decrease (2)

Theorem

Assume (A1) and let $\alpha \in [0,1)$. Then, choosing $(\boldsymbol{\lambda}_n, \Theta_n)_{n \geqslant 1}$ so that: $\forall n \geqslant 1$,

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

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

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

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

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

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

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

Conditions for a monotonic decrease (2)

Theorem

Assume (A1) and let $\alpha \in [0,1)$. Then, choosing $(\boldsymbol{\lambda}_n, \Theta_n)_{n \geqslant 1}$ so that: $\forall n \geqslant 1$,

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

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

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

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

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

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

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

Understanding the mixture weights update

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

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

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

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

$\rightarrow$ We recover the Power Descent algorithm from

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

K. Daudel, R. Douc and F. Portier (2021). To appear in the Annals of Statistics.

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

Understanding the mixture weights update

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

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

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

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

$\rightarrow$ We recover the Power Descent algorithm from

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

K. Daudel, R. Douc and F. Portier (2021). To appear in the Annals of Statistics.

Core insight :

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

Understanding the mixture weights update

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

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

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

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

$\rightarrow$ We recover the Power Descent algorithm from

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

K. Daudel, R. Douc and F. Portier (2021). To appear in the Annals of Statistics.

Core insight :

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

Towards simultaneous updates

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

- Maximisation approach

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

- Gradient-based approach

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

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

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

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

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

Towards simultaneous updates

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

- Maximisation approach

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

- Gradient-based approach

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

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

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

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

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

Towards simultaneous updates

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

- Maximisation approach

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

- Gradient-based approach

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

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

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

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

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

Towards simultaneous updates

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

- Maximisation approach

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

- Gradient-based approach

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

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

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

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

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

Maximisation approach

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

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

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

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

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

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

We have generalised an integrated EM algorithm for mixture models optimisation

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

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

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

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

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

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

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

We have generalised an integrated EM algorithm for mixture models optimisation

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

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

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

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

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

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

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

We have generalised an integrated EM algorithm for mixture models optimisation

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

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

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

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

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

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

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

We have generalised an integrated EM algorithm for mixture models optimisation

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

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

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

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

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

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

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

We have generalised an integrated EM algorithm for mixture models optimisation

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

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

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

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

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

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

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

We have generalised an integrated EM algorithm for mixture models optimisation

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

Application to GMMs

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

Algorithm 1: α-divergence minimisation for GMMs

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

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

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

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

$\rightarrow$ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

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

Application to GMMs

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

Algorithm 1: α-divergence minimisation for GMMs

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

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

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

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

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

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

Application to GMMs

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

Algorithm 1: α-divergence minimisation for GMMs

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

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

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

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

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

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

Improving on the M-PMC algorithm

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

<u>Parameters</u>
$\alpha = 0$, $\eta_n = \eta$
$M = 200$, $J = 100$
$q_n(y) = \sum_{j=1}^{J} \lambda_{j,n} k(\theta_{j,n}, y)$
$\rightarrow$ varying η and κ

Improving on the M-PMC algorithm

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

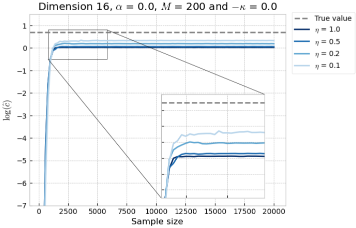
<u>Parameters</u>

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

$M = 200$, $J = 100$

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

$\rightarrow$ varying η and κ



Improving on the M-PMC algorithm

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

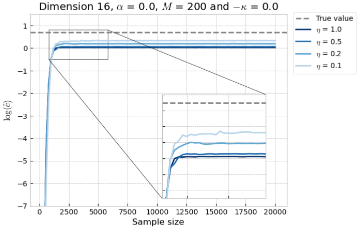
<ins>Parameters</ins>

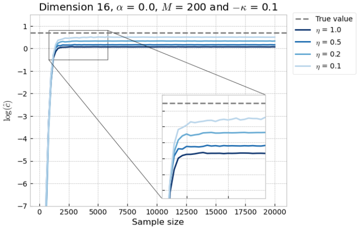
$\alpha = 0, \ \eta_n = \eta$

$M = 200, \ J = 100$

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

$\rightarrow$ varying η and κ





Improving on the M-PMC algorithm

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

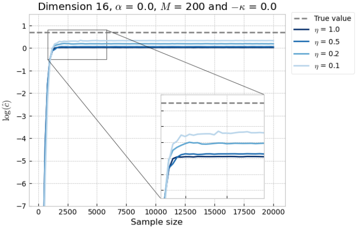
<u>Parameters</u>

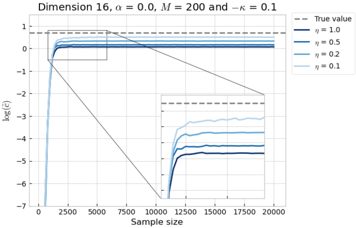
$\alpha = 0,\ \eta_n = \eta$

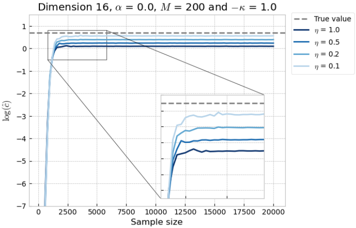
$M = 200,\ J = 100$

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

$\rightarrow$ varying η and κ







Gradient-based approach

Gradient-based approach and Gradient Descent

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

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

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

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

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

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

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

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

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

Gradient-based approach and Gradient Descent

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

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

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

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

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

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

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

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

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

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

Gradient-based approach and Gradient Descent

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

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

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

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

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

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

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

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

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

Gradient-based approach and Gradient Descent

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

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

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

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

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

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

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

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

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

Gradient-based approach and Gradient Descent

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

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

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

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

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

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

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

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

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

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

Gradient-based approach and Gradient Descent

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

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

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

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

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

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

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

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

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

Application to GMMs (2)

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

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

$\rightarrow$ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

Application to GMMs (2)

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

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

$\rightarrow$ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

Application to GMMs (2)

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

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

$\rightarrow$ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

Application to GMMs (2)

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

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

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

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

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

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

$\rightarrow$ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

Application to GMMs (2)

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

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

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

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

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

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

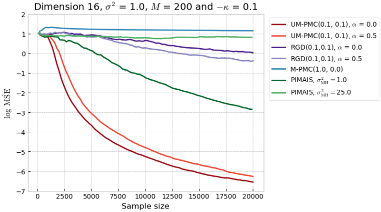
$\rightarrow$ NB : Monte Carlo approximations e.g. M i.i.d samples generated from q_n

Improving on Gradient Descent updates

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

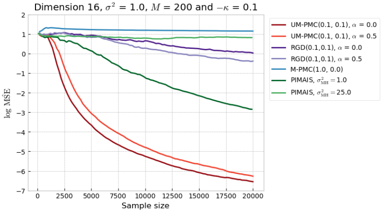
Improving on Gradient Descent updates

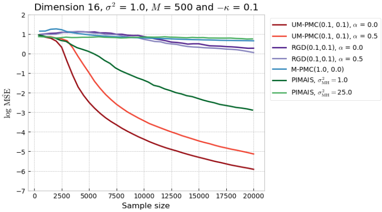
$$\text{Target :}\quad p(y) = 2 \times \left[0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})\right]\ ,\ d = 16$$



Improving on Gradient Descent updates

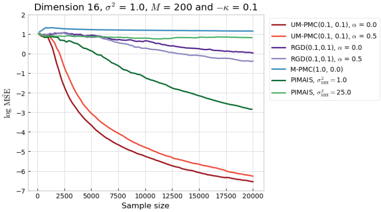
Target : $p(y) = 2 \times [0.5\mathcal{N}(\boldsymbol{y}; -2\boldsymbol{u_d}, \boldsymbol{I_d}) + 0.5\mathcal{N}(\boldsymbol{y}; 2\boldsymbol{u_d}, \boldsymbol{I_d})]$, $d = 16$

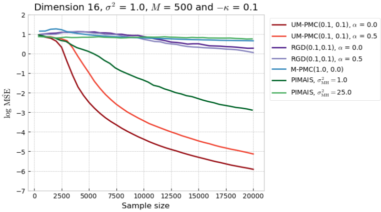


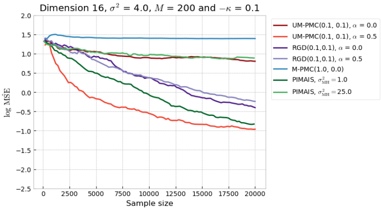


Improving on Gradient Descent updates

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

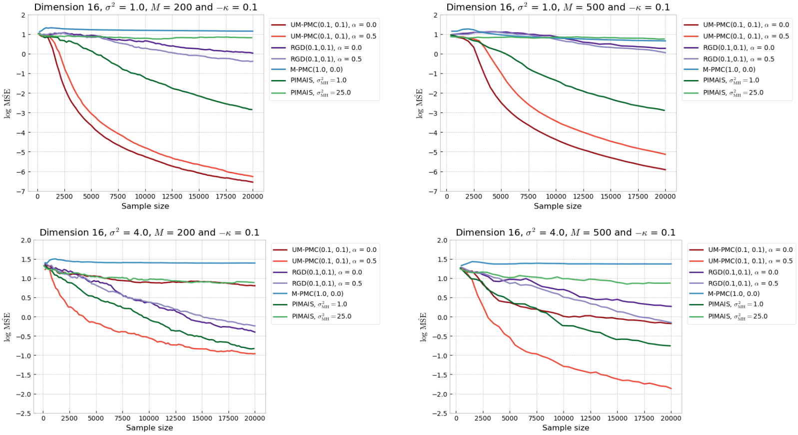






Improving on Gradient Descent updates

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



Outline

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

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

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

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

Thank you for your attention!

Infinite-dimensional gradient-based descent for alpha-divergence minimisation.
K. Daudel, R. Douc and F. Portier (2020). To appear in the Annals of Statistics.

Monotonic Alpha-divergence Minimisation.
K. Daudel, R. Douc and F. Roueff (2021). *Submitted*