Computing f-Divergences and Distances of High-Dimensional Probability Density Functions

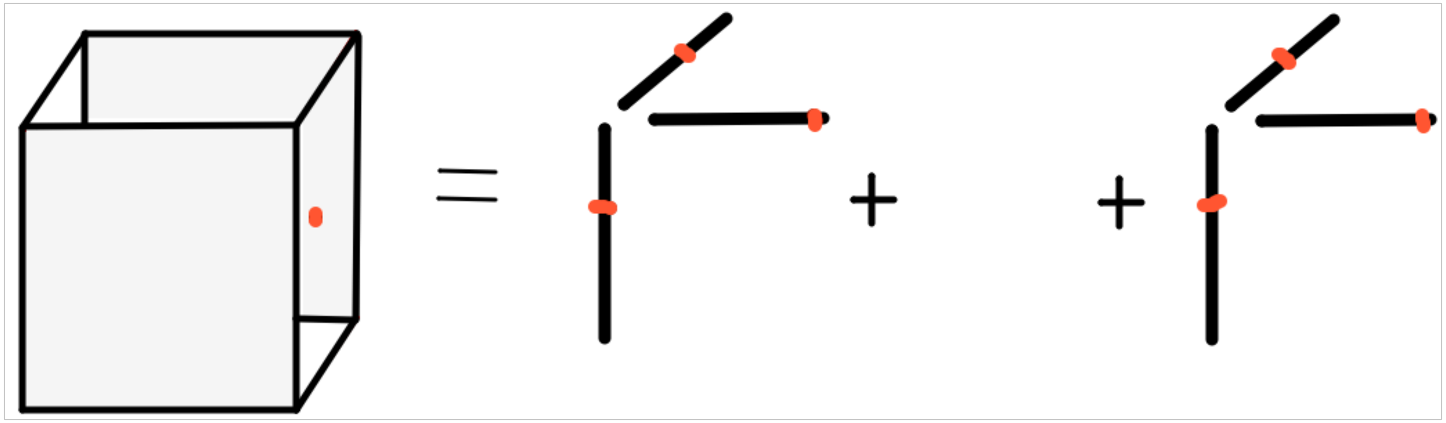
Alexander Litvinenko (RWTH Aachen, Germany),
joint work with
Youssef Marzouk, Hermann G. Matthies, Marco Scavino, Alessio Spantini

Plan

1. CP Tensors
2. Motivating examples, working diagram
3. Basics: pdf, pcf, FFT
4. Theoretical background
5. Computation of moments and divergences
6. Tensor formats
7. Algorithms
8. Numerics

Representation of a 3D tensor in the CP tensor format

A full tensor $\boldsymbol{w} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is represented as a sum of tensor products.
The lines on the right denote vectors $\boldsymbol{w}_{i,k} \in \mathbb{R}^{n_k}$, $i = 1, \ldots, r$, $k = 1, 2, 3$.



CP tensor format linear algebra

$$\alpha \cdot \boldsymbol{w} = \sum_{j=1}^{r} \alpha \bigotimes_{\nu=1}^{d} \boldsymbol{w}_{j,\nu} = \sum_{j=1}^{r} \bigotimes_{\nu=1}^{d} \alpha_\nu \boldsymbol{w}_{j,\nu}$$

where $\alpha_\nu := \sqrt[d]{|\alpha|}$ for all $\nu > 1$. and $\alpha_1 := \mathrm{sign}(\alpha)\sqrt[d]{|\alpha|}$.
The sum of two tensors costs only $\mathcal{O}(1)$:

$$\boldsymbol{w} = \boldsymbol{u} + \boldsymbol{v} = \left(\sum_{j=1}^{r_u} \bigotimes_{\nu=1}^{d} \boldsymbol{u}_{j,\nu}\right) + \left(\sum_{k=1}^{r_v} \bigotimes_{\mu=1}^{d} \boldsymbol{v}_{k,\mu}\right) = \sum_{j=1}^{r_u+r_v} \bigotimes_{\nu=1}^{d} \boldsymbol{w}_{j,\nu}$$

where $\boldsymbol{w}_{j,\nu} := \boldsymbol{u}_{j,\nu}$ for $j \leqslant r_u$ and $\boldsymbol{w}_{j,\nu} := \boldsymbol{v}_{j,\nu}$ for $r_u < j \leqslant r_u + r_v$.
The sum $\boldsymbol{w}$ generally has rank $r_u + r_v$.

CP properties: Hadamard product

$$\boldsymbol{u} \odot \boldsymbol{v} = \left(\sum_{j=1}^{r_u} \bigotimes_{\nu=1}^{d} \boldsymbol{u}_{j,\nu} \right) \odot \left(\sum_{k=1}^{r_v} \bigotimes_{\nu=1}^{d} \boldsymbol{v}_{k,\nu} \right) = \sum_{j=1}^{r_u} \sum_{k=1}^{r_v} \bigotimes_{\nu=1}^{d} \left(\boldsymbol{u}_{j,\nu} \odot \boldsymbol{v}_{k,\nu} \right)$$

The new rank can increase till $r_u r_v$, and the computational cost is $\mathcal{O}(r_u\, r_v n\, d)$.

CP properties: scalar product and norm

The scalar product can be computed as follows:

$$\langle \boldsymbol{u} | \boldsymbol{v} \rangle_{\mathcal{T}} = \langle \sum_{j=1}^{r_u} \bigotimes_{\nu=1}^{d} \boldsymbol{u}_{j,\nu} | \sum_{k=1}^{r_v} \bigotimes_{\nu=1}^{d} \boldsymbol{v}_{k,\nu} \rangle_{\mathcal{T}} = \sum_{j=1}^{r_u} \sum_{k=1}^{r_v} \prod_{\nu=1}^{d} \langle \boldsymbol{u}_{j,\nu} | \boldsymbol{v}_{k,\nu} \rangle$$

Cost is $\mathcal{O}(r_u \, r_v \, n \, d)$.
Rank truncation via the ALS-method or Gauss-Newton-method.

The scalar product above helps to compute the Frobenius norm

$$\|\boldsymbol{u}\|_2 := \sqrt{\langle \boldsymbol{u} | \boldsymbol{v} \rangle_{\mathcal{T}}}$$

.

Motivation 1: How to compute entropy in high dimensions?

Let $\boldsymbol{\xi} \in \mathbb{R}^d$ be a random vector $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_d)$ with pdf $p_{\boldsymbol{\xi}}$.
Entropy is the expectation of logarithm of pdf :

$$h(p_{\boldsymbol{\xi}}) := \mathbb{E}\left(-\ln(p_{\boldsymbol{\xi}}(\mathbf{y}))\right) := \int_{\mathbb{R}^d} -\ln(p_{\boldsymbol{\xi}}(\mathbf{y}))p_{\boldsymbol{\xi}}(\mathbf{y})\ \mathrm{d}\mathbf{y}. \tag{1}$$

Discretise:

$\operatorname{supp} p_{\boldsymbol{\xi}} := \mathrm{cl}\{\mathbf{y} \in \mathbb{R}^d \mid p_{\boldsymbol{\xi}}(\mathbf{y}) \neq \mathbf{0}\} \subseteq \times_{\nu=1}^{d}[\xi_\nu^{(\min)}, \xi_\nu^{(\max)}] \subset \mathbb{R}^d.$

Equidistant grid $\hat{\boldsymbol{x}}_\nu := (\hat{x}_{1,\nu}, \ldots, \hat{x}_{M_\nu,\nu})$, $1 \leqslant \nu \leqslant d$, of size M_ν: $\forall \nu$ it holds that $\hat{x}_{i_\nu,\nu} \in [\xi_\nu^{(\min)}, \xi_\nu^{(\max)}]$,

$$\hat{\boldsymbol{X}} = \bigtimes_{\nu=1}^{d} \hat{\boldsymbol{x}}_\nu = (\hat{X})_{(\nu, i_1, \ldots, i_d)}$$

with $1 \leqslant i_\nu \leqslant M_\nu$.
The notation $\boldsymbol{P} := p_{\boldsymbol{\xi}}(\hat{\boldsymbol{X}})$ will denote the tensor $\boldsymbol{P} \in \bigotimes_{\nu=1}^{d} \mathbb{R}^{M_\nu}$.

$$\boldsymbol{P} := p_{\boldsymbol{\xi}}(\hat{\boldsymbol{X}}) := (P_{i_1,\ldots,i_d}) := (p_{\boldsymbol{\xi}}(\hat{x}_{i_1,1}, \ldots, \hat{x}_{i_d,d})) \tag{2}$$

And the entropy

$$h(p_{\xi}) \approx \sum_{i_1=1}^{M_1} \cdots \sum_{i_d=1}^{M_d} -\ln(P_{i_1,\ldots,i_d}) P_{i_1,\ldots,i_d} w_{i_1,\ldots,i_d}, \tag{3}$$

where $w_{i_1,\ldots,i_d}$ are weights.
Sometimes can

$$p_{\xi}(\mathbf{y}) \approx \tilde{p}_{\xi}(\mathbf{y}) = \sum_{\ell=1}^{R} \bigotimes_{\nu=1}^{d} p_{\ell,\nu}(y_\nu), \tag{4}$$

where each $p_{\ell,\nu}$ is a function of y_ν in dimension ν. The $p_{\ell,\nu}$ are evaluated on the grid vector $\hat{\mathbf{x}}_\nu$ for all ν and ℓ, giving

$$\boldsymbol{p}_{\ell,\nu} := (p_{\ell,\nu}(\hat{x}_{1,\nu}), \ldots, p_{\ell,\nu}(\hat{x}_{M_\nu,\nu})) \in \mathbb{R}^{M_\nu}$$

A possible low-rank CP representation of the tensor $\boldsymbol{P}$:

$$\boldsymbol{P} \approx \sum_{\ell=1}^{R} \bigotimes_{\nu=1}^{d} \boldsymbol{p}_{\ell,\nu}. \tag{5}$$

Motivation 2: stochastic PDEs

$$-\nabla \cdot (\kappa(x,\omega)\nabla u(x,\omega)) = f(x,\omega), \quad x \in \mathcal{G} \subset \mathbb{R}^d,\ \omega \in \Omega$$

Write first Karhunen-Loeve Expansion and then for uncorrelated random variables the Polynomial Chaos Expansion

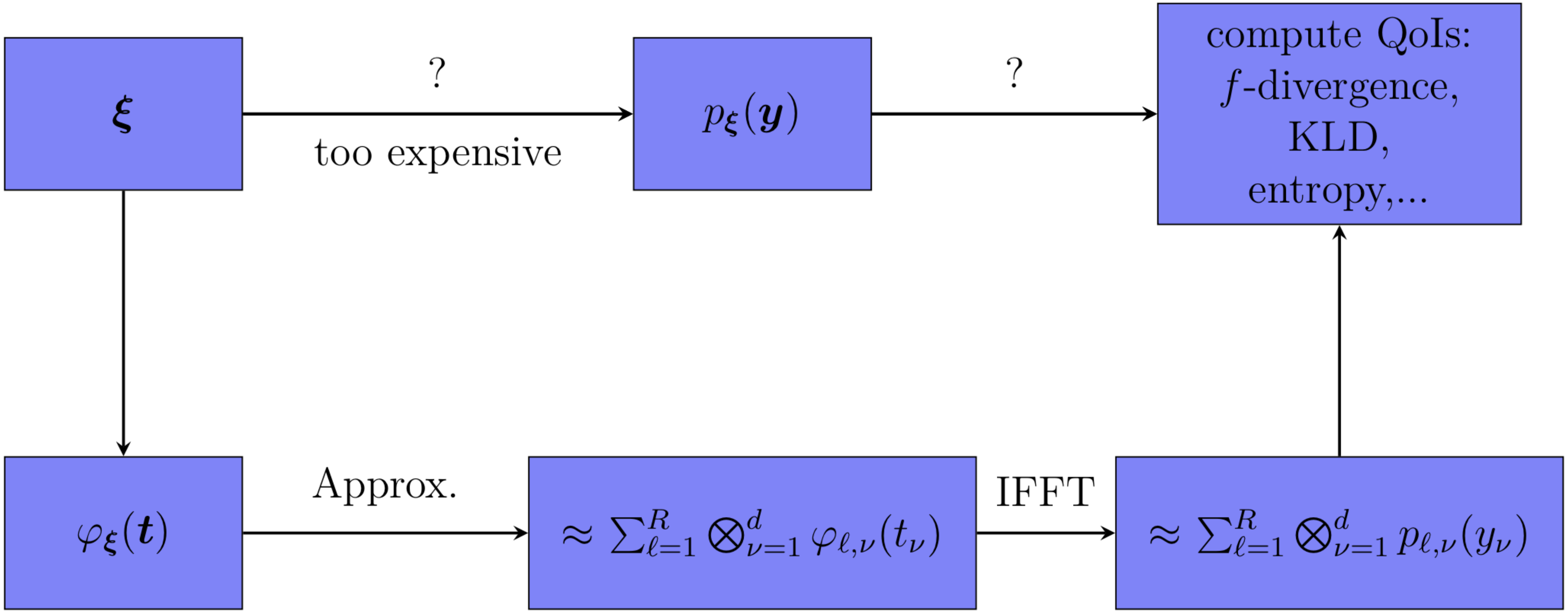
$$u(x,\omega) = \sum_{i=1}^{K} \sqrt{\lambda_i}\phi_i(x)\xi_i(\omega) = \sum_{i=1}^{K} \sqrt{\lambda_i}\phi_i(x) \sum_{\alpha\in\mathcal{J}} \xi_i^{(\alpha)} H_\alpha(\boldsymbol{\theta}(\omega)) \quad (6)$$

$$= \sum_{i=1}^{K} \sqrt{\lambda_i}\phi_i(x) \sum_{\alpha_1=1}^{p_1} \dots \sum_{\alpha_M=1}^{p_M} \xi_i^{(\alpha_1,\dots,\alpha_M)} \prod_{j=1}^{M} h_{\alpha_j}(\theta_j) \quad (7)$$

with multi-variate polynomials $H_\alpha(\boldsymbol{\theta}(\omega)) := \prod_{j=1}^{d} h_{\alpha_j}(\theta_j(\omega))$ in *iid* standard normalised Gaussians $\boldsymbol{\theta}(\omega) = (\theta_1(\omega), \dots, \theta_d(\omega))$, where $\alpha = (\alpha_1, \dots, \alpha_d)$ is a multi-index and the $h_{\alpha_j}(\theta_j(\omega))$ are uni-variate polynomials.

How to compute f-divergences from Eq. 6 ?

Idea: Computing f-divergences if pdfs are not available?



$\boldsymbol{\xi}$

?

too expensive

$p_{\boldsymbol{\xi}}(\boldsymbol{y})$

?

compute QoIs:
f-divergence,
KLD,
entropy,...

$\varphi_{\boldsymbol{\xi}}(\boldsymbol{t})$

Approx.

$\approx \sum_{\ell=1}^{R} \bigotimes_{\nu=1}^{d} \varphi_{\ell,\nu}(t_\nu)$

IFFT

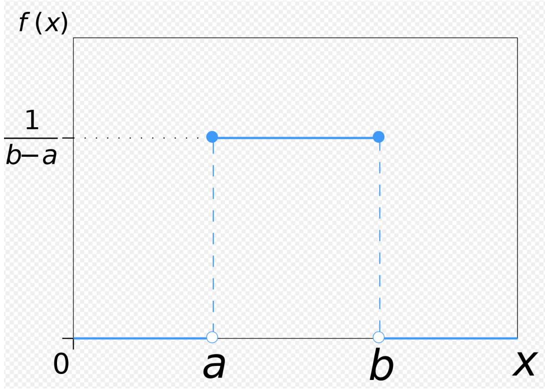
$\approx \sum_{\ell=1}^{R} \bigotimes_{\nu=1}^{d} p_{\ell,\nu}(y_\nu)$

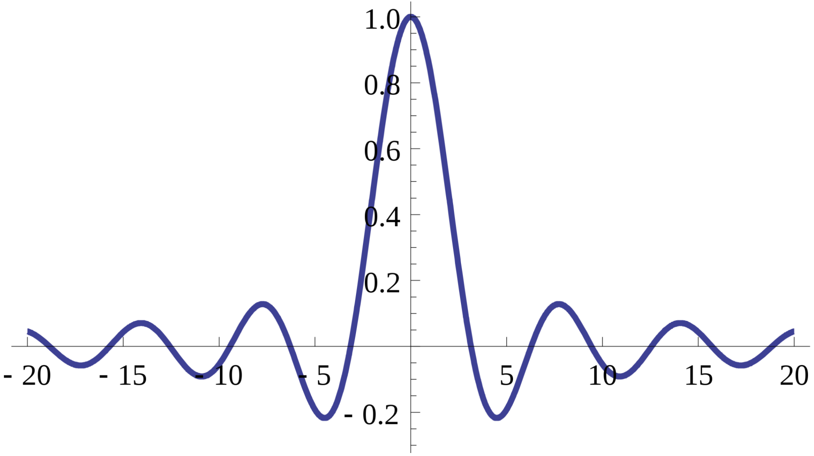
Two ways to compute f-divergence, KLD, entropy,...

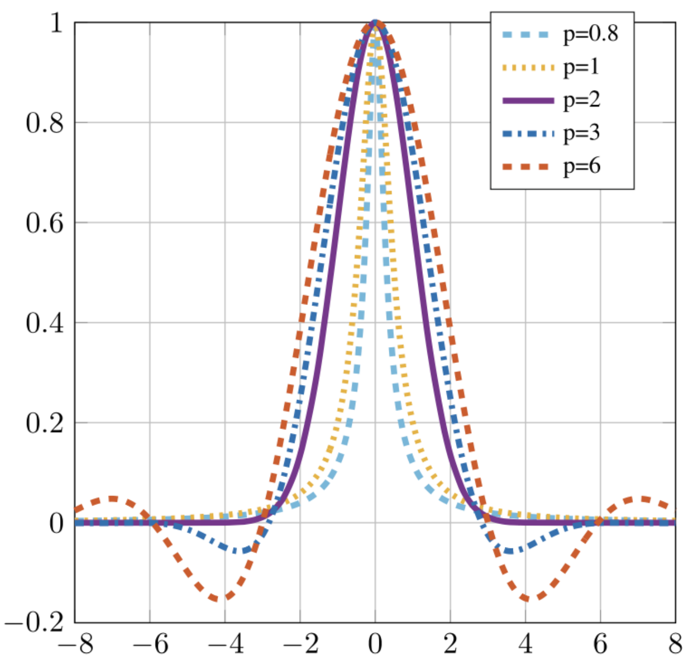
Examples of pcf

(left) Uniform pdf, $U[a, b]$; (right) pcf for $U(-1, 1)$

(bottom) pcf of generalized Gaussian, $f_\xi(x) \propto \exp \frac{-|x-\mu|^p}{2a^p}$







Some examples of the function f for the f-divergence

Many common divergences, such as the KL-divergence, the Hellinger distance, and the total variation distance, are special cases of the f-divergence, coinciding with a particular choice of f.

Name of the divergence	Corresponding $f(t)$
KL-divergence	$t \log(t)$
reverse KL-divergence	$-\log(t)$
squared Hellinger distance	$(\sqrt{t} - 1)^2$
total variation distance	$\lvert t - 1 \rvert / 2$
Pearson χ^2_P-divergence	$(t - 1)^2$
Neyman χ^2_N-divergence (reverse Pearson)	$t^{-1} - 1$
Pearson-Vajda χ^k_P-divergence	$(t - 1)^k$
Pearson-Vajda $\lvert\chi\rvert^k_P$-divergence	$\lvert t - 1 \rvert^k$
Jensen-Shannon-divergence	$t \log(t) - (t + 1) \log((t + 1$

Connection of pcf and pdf

The probability characteristic function (pcf) $\varphi_{\boldsymbol{\xi}}$ defined:

$$\varphi_{\boldsymbol{\xi}}(\boldsymbol{t}) := \mathbb{E}\left(\exp(\mathrm{i}\langle\boldsymbol{\xi}|\boldsymbol{t}\rangle)\right) := \int_{\mathbb{R}^d} p_{\boldsymbol{\xi}}(\mathbf{y}) \exp(\mathrm{i}\langle\mathbf{y}|\boldsymbol{t}\rangle)\ \mathrm{d}\mathbf{y} =: \mathcal{F}^{[d]}(p_{\boldsymbol{\xi}})(\boldsymbol{t}),$$

where $\boldsymbol{t} = (t_1, t_2, \ldots, t_d) \in \mathbb{R}^d$,

$\langle\mathbf{y}|\boldsymbol{t}\rangle = \sum_{j=1}^{d} y_j t_j$, and

$\mathcal{F}^{[d]}$ is the probabilist's d-dimensional Fourier transform.

$$p_{\boldsymbol{\xi}}(\mathbf{y}) = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} \exp(-\mathrm{i}\langle\boldsymbol{t}|\mathbf{y}\rangle)\varphi_{\boldsymbol{\xi}}(\boldsymbol{t})\,\mathrm{d}\boldsymbol{t} = \mathcal{F}^{[-d]}(\varphi_{\boldsymbol{\xi}})(\mathbf{y}), \qquad (8)$$

$\mathcal{F}^{[-d]}$ is the d-dimensional inverse Fourier transform.

Low-rank property of FFTd and iFFTd

We try to find an approximation

$$\varphi_{\boldsymbol{\xi}}(\boldsymbol{t}) \approx \widetilde{\varphi}_{\boldsymbol{\xi}}(\boldsymbol{t}) = \sum_{\ell=1}^{R} \bigotimes_{\nu=1}^{d} \varphi_{\ell,\nu}(t_\nu), \tag{9}$$

where the $\varphi_{\ell,\nu}(t_\nu)$ are one-dimensional functions.
Then we can get

$$p_{\xi}(\mathbf{y}) \approx \widetilde{p}_{\xi}(\mathbf{y}) = \mathcal{F}^{[-d]}(\widetilde{\varphi}_{\xi})\mathbf{y} = \sum_{\ell=1}^{R} \bigotimes_{\nu=1}^{d} \mathcal{F}_1^{-1}(\varphi_{\ell,\nu})(y_\nu),$$

where $\mathcal{F}_1^{-1}$ is the *one-dimensional* inverse Fourier transform.

Discrete representation of the pdf

Discrete representation of pdf and the pcf is based on equidistant grid vectors

$$\hat{x}_{i_\nu,\nu} = \hat{x}_{1,\nu} + (i_\nu - 1)\varDelta_{x_\nu}$$

(with increment $\varDelta_{x_\nu}$) of size M_ν in each dimension $1 \leqslant \nu \leqslant d$ of $\mathbb{R}^d$.

$V = \prod_{\nu=1}^{d} M_\nu \varDelta_{x_\nu}$, trapezoidal integration rule with weights $\frac{V}{N}$.

The whole grid is

$$\hat{\boldsymbol{X}} = \bigtimes_{\nu=1}^{d} \hat{\boldsymbol{x}}_\nu$$

$\boldsymbol{P} := p_\xi(\hat{\boldsymbol{X}})$ denotes the tensor $\boldsymbol{P} \in \bigotimes_{\nu=1}^{d} \mathbb{R}^{M_\nu} =: \mathcal{T}$,

$\dim \mathcal{T} = \prod_{\nu=1}^{d} M_\nu =: N$,
the components of which are the evaluation of the pdf p_ξ on the grid $\hat{\boldsymbol{X}}$.

Discrete representation of the pcf

Dual grid

$$\hat{\boldsymbol{T}} = \bigtimes_{\nu=1}^{d} \hat{\boldsymbol{t}}_\nu$$

$\hat{\boldsymbol{t}}_\nu := (\hat{t}_{1,\nu}, \ldots, \hat{t}_{M_\nu,\nu})$, $\hat{t}_{M_\nu,\nu} = \pi/\Delta_{x_\nu}$, the equi-distant spacing of the dual grid in dimension ν is $2\pi/L_\nu$..
$\mathbf{0} \in \hat{\boldsymbol{T}}$, $\boldsymbol{j}^0 = (j_1^0, \ldots, j_d^0)$, i.e. $(\hat{t}_{j_1^0,1}, \ldots, \hat{t}_{j_d^0,d}) = \mathbf{0} = (0, \ldots, 0)$.

The pcf on the dual grid is represented through the tensor $\boldsymbol{\Phi} := \phi_{\boldsymbol{\xi}}(\hat{\boldsymbol{T}}) \in \mathcal{T}$.

Thus, we deal with discretisations

$$\boldsymbol{P} := p_{\boldsymbol{\xi}}(\hat{\boldsymbol{X}})$$
$$\boldsymbol{\Phi} := \phi_{\boldsymbol{\xi}}(\hat{\boldsymbol{T}})$$

Notation, Moments and Covariance

$\mathbf{x} \in \mathbb{R}^d, \quad \mathbf{x}^{\otimes k} := \bigotimes_{j=1}^{k} \mathbf{x}.$

Random variable (RV) $\boldsymbol{\xi} : \Omega \to \mathcal{V} = \mathbb{R}^d$.
Expectation operator is denoted by $\mathbb{E}\left(\cdot\right)$,

$\bar{\boldsymbol{\xi}} := \mathbb{E}\left(\boldsymbol{\xi}\right) = \int_{\Omega} \boldsymbol{\xi}(\omega)\, \mathbb{P}(\mathrm{d}\omega) \in \mathbb{R}^d$,
$\tilde{\boldsymbol{\xi}} := \boldsymbol{\xi} - \bar{\boldsymbol{\xi}}$.

The moments $\boldsymbol{X}_k$ and the central moments $\boldsymbol{\Xi}_k$ of $\boldsymbol{\xi}$ of order k:

$$\boldsymbol{X}_k = \mathbb{E}\left(\boldsymbol{\xi}^{\otimes k}\right) \in (\mathbb{R}^d)^{\otimes k}$$

$$\boldsymbol{\Xi}_k = \mathbb{E}\left(\tilde{\boldsymbol{\xi}}^{\otimes k}\right) \in (\mathbb{R}^d)^{\otimes k}.$$

Notation, Moments and Covariance

The covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\xi}} = \operatorname{cov} \boldsymbol{\xi} = \boldsymbol{\Xi}_2 = \boldsymbol{X}_2 - \bar{\boldsymbol{\xi}} \otimes \bar{\boldsymbol{\xi}} \in (\mathbb{R}^d)^{\otimes 2}$.

The mixed and mixed central moments are denoted by

$$\boldsymbol{Y}_{k,\ell} = \mathbb{E}\left(\boldsymbol{\xi}^{\otimes k} \otimes \boldsymbol{\eta}^{\otimes \ell}\right)$$

$$\boldsymbol{\Upsilon}_{k,\ell} = \mathbb{E}\left(\tilde{\boldsymbol{\xi}}^{\otimes k} \otimes \tilde{\boldsymbol{\eta}}^{\otimes \ell}\right) \in (\mathbb{R}^d)^{\otimes k} \otimes (\mathbb{R}^n)^{\otimes \ell}.$$

The covariance is also denoted as

$$\operatorname{cov}(\boldsymbol{\xi}, \boldsymbol{\eta}) = \boldsymbol{\Upsilon}_{1,1} = \boldsymbol{Y}_{1,1} - \bar{\boldsymbol{\xi}} \otimes \bar{\boldsymbol{\eta}}$$

Higher order moments

$$\left(-\mathrm{i}\,\partial_{t_k}\right)\varphi_{\xi}(\boldsymbol{t}) = \int_{\mathbb{R}^d} x_k \exp(\mathrm{i}\langle \boldsymbol{t}|\mathbf{x}\rangle) p_{\xi}(\mathbf{x})\ \mathrm{d}\mathbf{x} = \mathcal{F}^{[d]}\left(x_k p_{\xi}(\mathbf{x})\right)(\boldsymbol{t})$$

Further, denoting the tensor of k-th derivatives by

$$\mathrm{D}^k\varphi_{\xi}(\boldsymbol{t}) = \left(\frac{\partial^k}{\partial_{t_{i_1}}\dots\partial_{t_{i_k}}}\varphi_{\xi}(\boldsymbol{t})\right),$$

and

$$(-\mathrm{i})^k\,\mathrm{D}^k\varphi_{\xi}(\mathbf{0}) = \int_{\mathbb{R}^d} \mathbf{x}^{\otimes k}\, p_{\xi}(\mathbf{x})\ \mathrm{d}\mathbf{x} = \mathcal{F}^{[d]}\left(\mathbf{x}^{\otimes k}\, p_{\xi}(\mathbf{x})\right)(\mathbf{0}) = \boldsymbol{X}_k \tag{10}$$

Second characteristic function

(cumulant generating function) whose derivative tensors of order k are essentially the *cumulants* $\boldsymbol{K}_k$ of $\boldsymbol{\xi}$, is defined as the point-wise logarithm of the `pcf` :

$$\chi_{\boldsymbol{\xi}}(\boldsymbol{t}) := \log(\varphi_{\boldsymbol{\xi}}(\boldsymbol{t})) = \log\left(\mathbb{E}\left(\exp(\mathrm{i}\langle\boldsymbol{t}|\boldsymbol{\xi}\rangle)\right)\right), \qquad (11)$$

with

$$(-\mathrm{i})^k\,\mathrm{D}^k\chi_{\boldsymbol{\xi}}(\mathbf{0}) =: \boldsymbol{K}_k$$

Moment generating function

Is defined as:

$$M_{\boldsymbol{\xi}}(\boldsymbol{t}) := \mathbb{E}\left(\exp(\langle \boldsymbol{t} | \boldsymbol{\xi} \rangle)\right) = \int_{\mathbb{R}^d} \exp(\langle \boldsymbol{t} | \mathbf{x} \rangle) p_{\boldsymbol{\xi}}(\mathbf{x}) \, \mathrm{d}\mathbf{x}$$
$$= \mathcal{L}_d(p_{\boldsymbol{\xi}})(-\boldsymbol{t}) = \varphi_{\boldsymbol{\xi}}(-\mathrm{i}\, \boldsymbol{t}),$$

where $\mathcal{L}_d(p_{\boldsymbol{\xi}})(\boldsymbol{t}) = \int \exp(\langle -\boldsymbol{t} | \mathbf{x} \rangle) p_{\boldsymbol{\xi}}(\mathbf{x}) \, \mathrm{d}\mathbf{x}$ is the two-sided d-dimensional *Laplace* transform of $p_{\boldsymbol{\xi}}$. Then

$$\mathrm{D}^k M_{\boldsymbol{\xi}}(\mathbf{0}) = \int_{\mathbb{R}^d} \mathbf{x}^{\otimes k} \, p_{\boldsymbol{\xi}}(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \boldsymbol{X}_k, \quad k \in \mathbb{N}_0. \tag{12}$$

Cumulant generating function is the point-wise logarithm of the moment generating function $M_{\boldsymbol{\xi}}$:

$$K_{\boldsymbol{\xi}}(\boldsymbol{t}) := \log(M_{\boldsymbol{\xi}}(\boldsymbol{t})) = \log\left(\mathbb{E}\left(\exp(\langle \boldsymbol{t} | \boldsymbol{\xi} \rangle)\right)\right), \tag{13}$$

with $\mathrm{D}^k K_{\boldsymbol{\xi}}(\mathbf{0}) = \boldsymbol{K}_k$.

Computation of QoIs

For tensors $\boldsymbol{P}$, $\boldsymbol{F}$ representing pdf $p(\mathbf{x})$ and a function $f(\cdot)$ evaluated on the grid, obtain

$$\int p(\mathbf{x})\ \mathrm{d}\mathbf{x} \approx \mathcal{S}(\boldsymbol{P}) := \frac{V}{N}\langle \boldsymbol{P}|\mathbf{1}\rangle_{\mathcal{T}}, \tag{14}$$

where $\mathbf{1} = \bigotimes_{\nu=1}^{d} \mathbf{1}_{\nu}$; — the tensor with all ones — satisfying $\boldsymbol{r} \odot \mathbf{1} = \boldsymbol{r}$ for any $\boldsymbol{r}$,
$\mathbf{1}_{\nu} := (1, \ldots, 1) \in \mathbb{R}^{M_{\nu}}$.

If $\boldsymbol{F}$ is a tensor which represents the grid-values of a function $f(\mathbf{x})$, i.e. $\boldsymbol{F} = f(\hat{\boldsymbol{X}})$, then

$$\mathbb{E}\left(f(\boldsymbol{\xi})\right) = \int_{\mathbb{R}^d} f(\mathbf{x}) p_{\boldsymbol{\xi}}(\mathbf{x})\ \mathrm{d}\mathbf{x}\ \approx\ \mathcal{S}(\boldsymbol{F} \odot \boldsymbol{P}) = \frac{V}{N}\langle \boldsymbol{F}|\boldsymbol{P}\rangle_{\mathcal{T}}$$

Computation of QoIs

Differential entropy, requiring the point-wise logarithm of $\boldsymbol{P}$:

$$h(p) := \mathbb{E}\left(-\log(p)\right)_p \approx \mathbb{E}\left(-\log(\boldsymbol{P})\right)_{\boldsymbol{P}} = -\frac{V}{N}\langle \log(\boldsymbol{P}) | \boldsymbol{P}\rangle,$$

Then the f-divergence of p from q and its discrete approximation is defined as

$$D_f(p\|q) := \mathbb{E}\left(f\left(\frac{p}{q}\right)\right)_q \approx \mathbb{E}\left(f(\boldsymbol{P}\odot\boldsymbol{Q}^{\odot-1})\right)_{\boldsymbol{Q}}$$

$$= \frac{V}{N}\langle f(\boldsymbol{P}\odot\boldsymbol{Q}^{\odot-1}) | \boldsymbol{Q}\rangle.$$

List of some typical divergences and distances.

Divergence	$D_{\bullet}(p \Vert q)$
KLD — D_{KL}:	$\int (\log(p(\mathbf{x})/q(\mathbf{x})))\, p(\mathbf{x}) \; \mathrm{d}\mathbf{x} = \mathbb{E}_p(\log(p/q))$
Hellinger, $(D_H)^2$:	$\frac{1}{2} \int \left(\sqrt{p(\mathbf{x})} - \sqrt{q(\mathbf{x})} \right)^2 \; \mathrm{d}\mathbf{x}$
Bregman, D_ϕ:	$\int \left[(\phi(p(\mathbf{x})) - \phi(q(\mathbf{x}))) - (p(\mathbf{x}) - q(\mathbf{x}))\phi'(q(\mathbf{x})) \right] \; \mathrm{d}\mathbf{x}$
Bhattach., D_{Bh}:	$-\log \left(\int \sqrt{(p(\mathbf{x}) q(\mathbf{x}))} \; \mathrm{d}\mathbf{x} \right)$

Discrete approximations for divergences above

Divergence	Approx. $D_\bullet(p \| q)$
KLD	$\frac{V}{N}(\langle \log(\boldsymbol{P}) \vert \boldsymbol{P} \rangle - \langle \log(\boldsymbol{Q}) \vert \boldsymbol{P} \rangle)$
$(D_H)^2$:	$\frac{V}{2N}\langle \boldsymbol{P}^{\odot 1/2} - \boldsymbol{Q}^{\odot 1/2} \vert \boldsymbol{P}^{\odot 1/2} - \boldsymbol{Q}^{\odot 1/2} \rangle$
D_ϕ:	$\mathcal{S}\left((\phi(\boldsymbol{P}) - \phi(\boldsymbol{Q})) - (\boldsymbol{P} - \boldsymbol{Q}) \odot \phi'(\boldsymbol{Q})\right)$
D_{Bh}:	$-\log\left(\frac{V}{N}\langle \boldsymbol{P}^{\odot 1/2} \vert \boldsymbol{Q}^{\odot 1/2} \rangle\right)$

Algorithms

Below we will list algorithms, which approximate $f(p_\xi(\mathbf{y}))$ by $f(\boldsymbol{P})$, where the f's considered are

$$f(\cdot) = \{\mathrm{sign}(\cdot), (\cdot)^{-1}, \sqrt{\cdot}, \sqrt[m]{\cdot}, (\cdot)^k, \log(\cdot), \exp(\cdot), (\cdot)^2, |\cdot|\}, \qquad (15)$$

$k > 0$,
$\boldsymbol{P} = p_\xi(\hat{\boldsymbol{X}}) = \sum_{j=1}^{r_p} \bigotimes_{\nu=1}^{d} \boldsymbol{p}_{j,\nu}$.

Available methods:

1. TT-cross
2. iterative methods (e.g., Newton algorithm)
3. power series
4. quadrature rule to compute the Dunford-Cauchy contour integral
5. others (like sinc quadrature)

Iterative methods

We want to compute $f(\boldsymbol{w})$ for some function $f : \mathcal{T} \rightarrow \mathcal{T}$.
We have an iteration function Ψ_f,
which only uses operations from the Hadamard algebra on $\mathcal{T}$, and which is iterated,

$$\boldsymbol{v}_{i+1} = \Psi_f(\boldsymbol{v}_i)$$

and converges to a fixed point

$$\Psi_f(\boldsymbol{v}_*) = \boldsymbol{v}_*$$

When started with a $\boldsymbol{v}_0$ depending on $\boldsymbol{w}$,
the fixed point is

$$\lim_{i \rightarrow \infty} \boldsymbol{v}_i = \boldsymbol{v}_* = \Psi_f(\boldsymbol{v}_*) = f(\boldsymbol{w})$$

Computing pointwise inverse $w^{\odot -1}$.

Let $F(\boldsymbol{x}) := \boldsymbol{w} - \boldsymbol{x}^{\odot -1}$.
Applying Newton's method to $F(\boldsymbol{x})$ for approximating the inverse of a given tensor $\boldsymbol{w}$, one obtains the following iteration function $\Psi_{\odot -1}$ with the i.c. $\boldsymbol{v}_0 = \alpha \cdot \boldsymbol{w}$ to bring $\boldsymbol{v}_0$ close to $\boldsymbol{v}_a = \mathbf{1}$:

$$\Psi_{\odot -1}(\boldsymbol{v}) = \boldsymbol{v} \odot (2 \cdot \mathbf{1} - \boldsymbol{w} \odot \boldsymbol{v}).$$

The iteration converges if the initial iterate $\boldsymbol{v}_0$ satisfies $\|\mathbf{1} - \boldsymbol{w} \odot \boldsymbol{v}_0\|_\infty < 1$.
A possible candidate for the starting value is $\boldsymbol{v}_0 = \alpha \boldsymbol{w}$ with $\alpha < (1/\|\boldsymbol{w}\|_\infty)^2$.
For such a $\boldsymbol{v}_0$, the convergence initial condition $\|\mathbf{1} - \alpha \boldsymbol{w}^{\odot 2}\|_\infty < 1$ is always satisfied.

Computing pointwise $\sqrt{w}$ via Newton iteration

Let $F(\boldsymbol{x}) := \boldsymbol{x}^{\odot 2} - \boldsymbol{w} = 0$.

The Newton iteration

$$\Psi_{\sqrt{}}(\boldsymbol{v}) = \frac{1}{2} \cdot (\boldsymbol{v} + \boldsymbol{v}^{\odot -1} \odot \boldsymbol{w}). \tag{16}$$

with i.c. $\boldsymbol{v}_0 = (\boldsymbol{w} + \mathbf{1})/2$.

Computing pointwise $\sqrt{w}$ via Newton-Schulz iteration

Let $F(\boldsymbol{x}) := \boldsymbol{x}^{\odot 2} - \boldsymbol{w} = 0$.

Newton-Schulz iteration computes
$\boldsymbol{v}_*^+ = \sqrt{\boldsymbol{w}} = \boldsymbol{w}^{\odot 1/2}$ and $\boldsymbol{v}_*^- = (\sqrt{\boldsymbol{w}})^{\odot -1} = \boldsymbol{w}^{\odot -1/2}$.

We set $\boldsymbol{V}_0 = [\boldsymbol{y}_0, \boldsymbol{z}_0] = [\alpha \cdot \boldsymbol{w}, \mathbf{1}] \in \mathcal{T}^2$, and the auxiliary function $A(\boldsymbol{y}, \boldsymbol{z}) = 3 \cdot \mathbf{1} - \boldsymbol{z} \odot \boldsymbol{y}$:

$$\Psi_{\sqrt{}}\left(\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix}\right) = \frac{1}{2}\begin{bmatrix} \boldsymbol{y} \odot A(\boldsymbol{y}, \boldsymbol{z}) \\ A(\boldsymbol{y}, \boldsymbol{z}) \odot \boldsymbol{z} \end{bmatrix}. \qquad (17)$$

The iteration converges to

$$\boldsymbol{V}_* = [\boldsymbol{v}_*^+, \boldsymbol{v}_*^-] = [\sqrt{\boldsymbol{y}_0}, (\sqrt{\boldsymbol{y}_0})^{\odot -1}]$$

if $\|\mathbf{1} - \boldsymbol{y}_0\|_\infty < 1$, and $\alpha < 1/\|\boldsymbol{w}\|_\infty$.
Fixed point of the iteration is $\boldsymbol{v}_*^+ = \sqrt{\alpha} \cdot \sqrt{\boldsymbol{w}}$ and
$\boldsymbol{v}_*^- = (1/\sqrt{\alpha}) \cdot (\sqrt{\boldsymbol{w}})^{\odot -1}$.
Obtain: $\sqrt{\boldsymbol{w}} = (1/\sqrt{\alpha}) \cdot \boldsymbol{v}_*^+$ and $(\sqrt{\boldsymbol{w}})^{\odot -1} = \sqrt{\alpha} \cdot \boldsymbol{v}_*^-$.

Computing $\log(\boldsymbol{w})$

Assume $\boldsymbol{w} > 0$.
See [Higham'01, Higham'12].
For the algorithms to work well $\boldsymbol{w}$ has to be close to the identity $\mathbf{1}$, which can be achieved by taking roots: for $\lambda > 0$ one has $\log\left(\boldsymbol{w}^{\lambda}\right) = \lambda \log \boldsymbol{w}$.

Truncated Taylor series (radius of convergence $\|\boldsymbol{x}\|_\infty < 1$):

$$\log(\mathbf{1} - \boldsymbol{x}) = -\sum_{n=1}^{\infty} \frac{1}{n} \cdot \boldsymbol{x}^{\odot n}$$

where $\boldsymbol{x} := \mathbf{1} - \boldsymbol{w}$. If $\boldsymbol{w}$ is not near to the identity, then one may use the relation $\log(\boldsymbol{w}) = 2^k \log(\boldsymbol{w}^{\odot 1/2^k})$, where $\boldsymbol{w}^{\odot 1/2^k} \to \mathbf{1}$ as k increases.

Computing power function by $w \mapsto w^{\odot m}$

Power function by

$$\boldsymbol{w} \mapsto \boldsymbol{w}^{\odot m} =: \Psi_{\mathrm{pow}}(m, \boldsymbol{w})$$

For $m < 0$ this is simply

$$\Psi_{\mathrm{pow}}(m, \boldsymbol{w}) = \Psi_{\mathrm{pow}}(-m, \boldsymbol{w}^{\odot -1})$$

The recursive formula:

$$\Psi_{\mathrm{pow}}(m, \boldsymbol{w}) = \begin{cases} m > 1 \text{ and odd}: & \boldsymbol{w} \odot \Psi_{\mathrm{pow}}(m-1, \boldsymbol{w}); \\ m \text{ even}: & \Psi_{\mathrm{pow}}(\frac{m}{2}, \boldsymbol{w}) \odot \Psi_{\mathrm{pow}}(\frac{m}{2}, \boldsymbol{w}); \\ m = 1: & \boldsymbol{w}; \end{cases} \tag{18}$$

Computing $\boldsymbol{w}^{\odot \frac{1}{m}}$

See Section 7 in N.Higham's book.

Assume $\boldsymbol{w} \geqslant \mathbf{o}$
Newton's method for $F(\boldsymbol{x}) = \boldsymbol{x}^{\odot m} - \boldsymbol{w} = \mathbf{o}$.
The iteration function with $\boldsymbol{v}_0 = \boldsymbol{w}$ looks like

$$\Psi_{m-\mathrm{root}}(\boldsymbol{v}) = \frac{1}{m}\left((m-1)\cdot \boldsymbol{v} + \Psi_{\mathrm{pow}}(1-m, \boldsymbol{v}) \odot \boldsymbol{v}_0\right). \qquad (19)$$

If $m \geqslant 2$, this involves a negative power

$$\boldsymbol{v}^{\odot(1-m)} = \Psi_{\mathrm{pow}}(1-m, \boldsymbol{v})$$

.

Algorithm converges for all $\boldsymbol{w} \geqslant \mathbf{o}$.

Computing $w^{\odot \frac{1}{m}}$

Auxiliary function $A(\boldsymbol{y}, \boldsymbol{z}) = (1/m) \cdot ((m+1) \cdot \mathbf{1} - \boldsymbol{z})$:

$$\Psi_{m\text{-root}} = \left(\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} \right) = \begin{bmatrix} \boldsymbol{y} \odot A(\boldsymbol{y}, \boldsymbol{z}) \\ \Psi_{\text{pow}}(m, A(\boldsymbol{y}, \boldsymbol{z})) \odot \boldsymbol{z} \end{bmatrix}, \tag{20}$$

where $\boldsymbol{y}_i \to \boldsymbol{w}^{\odot -\frac{1}{m}}$ and $\boldsymbol{z}_i \to \boldsymbol{w}^{\odot \frac{1}{m}}$.

The starting values are

$$\boldsymbol{V}_0 = [\boldsymbol{y}_0, \boldsymbol{z}_0] = [\alpha \cdot \mathbf{1}, (\alpha)^m \boldsymbol{w}] \in \mathcal{T}^2$$

with $\alpha < (\|\boldsymbol{w}\|_\infty / \sqrt{2})^{-\frac{1}{m}}$.

For scaling purposes it is best used with $m = 2^k$.

Computing $w^{\odot \frac{1}{m}}$

Another way of computing the m-th root is Tsai's algorithm(Tsai'88, Lorin'21), which uses the auxiliary function

$$B(\boldsymbol{y}) = (2 \cdot \mathbf{1} + (m-2) \cdot \boldsymbol{y}) \odot (\mathbf{1} + (m-1) \cdot \boldsymbol{y})^{\odot -1}$$

:

$$\Psi_{Tsai} = \left(\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} \right) = \begin{bmatrix} \boldsymbol{y} \odot \Psi_{\text{pow}}(m, B(\boldsymbol{y})) \\ \boldsymbol{z} \odot (B(\boldsymbol{y})) \end{bmatrix}, \tag{21}$$

with starting value $\boldsymbol{V}_0 = [\boldsymbol{w}, \mathbf{1}]$.

Then $\boldsymbol{z}_i \longrightarrow \boldsymbol{w}^{\odot \frac{1}{m}}$.

Computing $\log(w)$ via Gregory's series

Converges for all $\boldsymbol{w} > 0$.
Setting $\boldsymbol{z} = (\mathbf{1} - \boldsymbol{w}) \odot (\mathbf{1} + \boldsymbol{w})^{\odot -1}$, one has

$$\log \boldsymbol{w} = -2 \sum_{k=0}^{\infty} \frac{1}{2k+1} \cdot \boldsymbol{z}^{\odot(2k+1)}. \tag{22}$$

Computing $\exp \boldsymbol{w}$.

See book of N. Higham, Chapter 10:

$$\boldsymbol{u}_{r,s} = \left(\sum_{k=0}^{r} \frac{1}{k! s^k} \boldsymbol{w}^{\odot k} \right)^{\odot s} . \tag{23}$$

Here $\lim_{r\to\infty} \boldsymbol{u}_{r,s} = \lim_{s\to\infty} \boldsymbol{u}_{r,s} = \exp \boldsymbol{w}$.

It is of advantage to use s from the series of powers of 2,
$s = 1, 2, 4, \ldots, 2^k$,
then the s-th power can be computed by squaring.

For the scaling the best choice is $\alpha > \|\boldsymbol{w}\|_\infty$.

Four numerical tests:

1. KLD is computed with the analytical formula and the amen_cross algorithm from TT-toolbox
2. Hellinger distances is computed with well-known analytical formulas and the amen_cross algorithm.
3. (`pdf` is not known analytically), the d-variate elliptically contoured α-stable distributions are chosen and accessed via their `pcfs` ,
4. KLD and Hellinger distances for different value of d, n and the parameter α.

Example 1: KLD for two Gaussian distributions

$\mathcal{N}_1 := \mathcal{N}(\boldsymbol{\mu}_1, \mathbf{C}_1)$ and $\mathcal{N}_2 := \mathcal{N}(\boldsymbol{\mu}_2, \mathbf{C}_2)$, where
$\mathbf{C}_1 := \sigma_1^2 \boldsymbol{I}$, $\mathbf{C}_2 := \sigma_2^2 \boldsymbol{I}$,
$\boldsymbol{\mu}_1 = (1.1 \ldots, 1.1)$ and $\boldsymbol{\mu}_2 = (1.4, \ldots, 1.4) \in \mathbb{R}^d$,
$d = \{16, 32, 64\}$, and $\sigma_1 = 1.5$, $\sigma_2 = 22.1$.

The well-known analytical formula is

$$2D_{\mathrm{KL}}(\mathcal{N}_1 \| \mathcal{N}_2) = tr(\mathbf{C}_2^{-1}\mathbf{C}_1) + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \mathbf{C}_2^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) - d + \log \frac{|\mathbf{C}_2}{|\mathbf{C}_1}$$

Comparison of KLDs computed via two methods

D_{KL} computed via TT tensors (AMEn algorithm) and the analytical formula for various values of d.
TT tolerance $= 10^{-6}$, the stopping difference between consecutive iterations.

d	16	32	64
n	2048	2048	2048
D_{KL} (exact)	35.08	70.16	140.32
$\widetilde{D}_{\mathrm{KL}}$	35.08	70.16	140.32
err_a	4.0e-7	2.43e-5	1.4e-5
err_r	1.1e-8	3.46e-8	8.1e-8
comp. time, sec.	1.0	5.0	18.7

Example 2 — Hellinger distance

(for Gaussian distributions)

$$D_H(\mathcal{N}_1, \mathcal{N}_2)^2 = 1 - K_{\frac{1}{2}}(\mathcal{N}_1, \mathcal{N}_2), \qquad \text{where}$$

$$K_{\frac{1}{2}}(\mathcal{N}_1, \mathcal{N}_2) = \frac{\det(\mathbf{C}_1)^{\frac{1}{4}} \det(\mathbf{C}_2)^{\frac{1}{4}}}{\det\left(\frac{\mathbf{C}_1 + \mathbf{C}_2}{2}\right)^{\frac{1}{2}}} \cdot \\ \cdot \exp\left(-\frac{1}{8} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \left(\frac{\mathbf{C}_1 + \mathbf{C}_2}{2} \right)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \right)$$

Hellinger distance D_H

is computed via TT tensors (AMEn) and the analytical formula. TT tolerance $= 10^{-6}$.

d	16	32	64
n	2048	2048	2048
D_H (exact)	0.99999	0.99999	0.99999
$\widetilde{D}_H$	0.99992	0.99999	0.99999
err_a	3.5e-5	7.1e-5	1.4e-4
err_r	2.5e-5	5.0e-5	1.0e-4
comp. time, sec.	1.7	7.5	30.5

The AMEn algorithm is able to compute the Hellinger distance D_H between two multiv. Gaussian distribes for $d = \{16, 32, 64\}$, and $n = 2048$. The exact and approximate values are almost identical.

Example 3: α-stable distribution

The pcf of a d-variate elliptically contoured α-stable distribution is given by

$$\varphi_{\boldsymbol{\xi}}(\boldsymbol{t}) = \exp\left(\mathrm{i}\langle \boldsymbol{t} | \boldsymbol{\mu} \rangle - \langle \boldsymbol{t} | \mathbf{C}\boldsymbol{t} \rangle^{\frac{\alpha}{2}}\right).$$

AMEn tol.$= 10^{-9}$.

Example 3: KLD between two α-stable distributions

with $\alpha_1 = 2.0$, $\alpha_2 = 1.9$ ($\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = 0$, $\mathbf{C}_1 = \mathbf{C}_2 = \boldsymbol{I}$).

d	16	16	16	16	16	16	16	32	32	32
n	8	16	32	64	128	256	512	64	128	256
$D_{\mathrm{KL}}(2.0, 1.9)$	0.016	0.06	0.06	0.062	0.06	0.06	0.06	0.09	0.14	0.12
time, sec.	0.8	3	8.9	14	22	61	207	46	100	258
maxTT rank	40	57	79	79	59	79	77	80	78	79
mem., MB	1.8	7	34	54	73	158	538	160	313	626

AMEn tol.$= 10^{-9}$.

Full storage vs low-rank

For $d = 32$ and $n = 256$ the amount of data in full storage mode would be
$N = n^d = 265^{32} \approx 1.16\mathrm{E}77 \approx 1\mathrm{E}78$ bytes.
In TT-low-rank approximation: $626\mathrm{MB}$, and fits on a laptop.
Assuming $1\mathrm{GHz}$ notebook, the KLD computation in full mode would require ca. $1.2\mathrm{E}68\mathrm{sec}$,
or more than $3\mathrm{E}60$ years,
and even with a perfect speed-up on a parallel super-computer with say $1,000,000$ processors,
this would require still more than $3\mathrm{E}54$ years;
compare this with the estimated age of the universe of ca. $1.4\mathrm{E}10$ years.

Example 4: $D_{\mathbf{KL}}(\alpha_1, \alpha_2)$ between two α-stable distributions

for (α_1, α_2) and fixed $d = 8$ and $n = 64$.

(α_1, α_2)	$(2.0, 0.5)$	$(2.0, 1.0)$	$(2.0, 1.5)$	$(2.0, 1.9)$	$(1.5, 1.4)$	$(1.0, 0.4)$
$D_{\mathrm{KL}}(\alpha_1, \alpha_2)$	2.27	0.66	0.3	0.03	0.031	0.6
comp. time, sec.	8.4	7.8	7.5	8.5	11	8.7
max. TT rank	78	74	76	76	80	79
memory, MB	28.5	28.5	27.1	28.5	35	29.5

$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = 0$, $\mathbf{C}_1 = \mathbf{C}_2 = \boldsymbol{I}$.
AMEn tol.$= 10^{-12}$.

Example 3: Hellinger distance $D_H(\alpha_1, \alpha_2)$

for the *d*-variate elliptically contoured α-stable distribution for $\alpha = 1.5$ and $\alpha = 0.9$ for different *d* and *n*.
$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = 0$, $\mathbf{C}_1 = \mathbf{C}_2 = \boldsymbol{I}$.

d	16	16	16	16	16	16	32	32	32	32
n	8	16	32	64	128	256	16	32	64	128
$D_H(1.5, 0.9)$	0.218	0.223	0.223	0.223	0.219	0.223	0.180	0.176	0.175	0.176
comp. time, sec.	2.8	3.7	7.5	19	53	156	11	21	62	117
max. TT rank	79	76	76	76	79	76	75	71	75	74
memory, MB	7.7	17	34	71	145	283	34	66	144	285

AMEn tolerance is 10^{-9}.

Example 6: D_H vs. TT (AMEn) tolerances

TT(AMEn) tolerance	10^{-7}	10^{-8}	10^{-9}	10^{-10}	10^{-14}
$D_H(1.5, 0.9)$	0.1645	0.1817	0.176	0.1761	0.1802
comp. time, sec.	43	86	103	118	241
max. TT rank	64	75	75	78	77
memory, MB	126	255	270	307	322

Computation of $D_H(\alpha_1, \alpha_2)$ between two α-stable distributions ($\alpha = 1.5$ and $\alpha = 0.9$) for different AMEn tolerances.
$n = 128$, $d = 32$, $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = 0$, $\mathbf{C}_1 = \mathbf{C}_2 = \boldsymbol{I}$.

Conclusion

Demonstrated that high-dim. `pdfs` , `pcfs` , and some functions of them can be approximated and represented in a low-rank tensor data format.

Provided numerical methods to compute

1. entropy, KLD, and f-divergences in low-rank tensor format
2. functions
$$f(\cdot) = \{\mathrm{sign}(\cdot), (\cdot)^{-1}, \sqrt{\cdot}, \sqrt[m]{\cdot}, (\cdot)^k, \log(\cdot), \exp(\cdot), (\cdot)^2, |\cdot|\},$$
of `pcf` and `pdf`
3. low-rank approximations reduce the complexity and storage from exponential $\mathcal{O}(n^d)$ to linear in n.

Literature

1. A. Litvinenko, Y. Marzouk, H.G. Matthies, M. Scavino, A. Spantini, Computing f-Divergences and Distances of High-Dimensional Probability Density Functions – Low-Rank Tensor Approximations, Numer Linear Algebra Appl. 2022;e2467. https://doi.org/10.1002/nla.2467
2. M. Espig, W. Hackbusch, A. Litvinenko, H.G. Matthies, E Zander, Iterative algorithms for the post-processing of high-dimensional data, JCP 410, 109396, 2020, https://doi.org/10.1016/j.jcp.2020.109396
3. S. Dolgov, A. Litvinenko, D. Liu, KRIGING IN TENSOR TRAIN DATA FORMAT, Conf. Proc., 3rd Int. Conf. on Uncertainty Quantification in CSE, https://files.eccomasproceedia.org/papers/e-books/uncecomp_2019.pdf, pp 309-329, 2019

Literature

4. A. Litvinenko, D. Keyes, V. Khoromskaia, B.N. Khoromskij, H.G. Matthies, Tucker tensor analysis of Matérn functions in spatial statistics, Computational Methods in Applied Mathematics, vol. 19, no 1, 2019, pp 101-122, https://doi.org/10.1515/cmam-2018-0022
5. A. Litvinenko, R. Kriemann, M.G. Genton, Y. Sun, D.E. Keyes, HLIBCov: Parallel hierarchical matrix approximation of large covariance matrices and likelihoods with applications in parameter identification, MethodsX 7, 100600, 2020, https://doi.org/10.1016/j.mex.2019.07.001
6. A. Litvinenko, Y. Sun, M.G. Genton, D.E. Keyes, Likelihood approximation with hierarchical matrices for large spatial datasets, Computational Statistics & Data Analysis 137, pp 115-132, 2019, https://doi.org/10.1016/j.csda.2019.02.002
7. A. Litvinenko, Application of hierarchical matrices for solving multiscale problems, Leipzig University, Germany, 2006

Acknowledgement

Funding information:
Alexander von Humboldt-Stiftung, Deutsche Forschungsgemeinschaft, Gay-Lussac Humboldt Research Award

How to compute KLD and other divergences? Classical result is given only for pdfs, which are usually unknown.

Divergence	$D_\bullet(p\|q)$
KLD	$\int (\log(p(\mathbf{x})/q(\mathbf{x})))\, p(\mathbf{x})\ \mathrm{d}\mathbf{x}$
Hellinger dist.	$\frac{1}{2}\int \left(\sqrt{p(\mathbf{x})} - \sqrt{q(\mathbf{x})}\right)^2\ \mathrm{d}\mathbf{x}$
Bhattacharyya	$-\log\left(\int \sqrt{(p(\mathbf{x})q(\mathbf{x}))}\ \mathrm{d}\mathbf{x}\right)$