Taming numerical imprecision by adapting the KL divergence to negative probabilities

Simon Pfahler[1], Peter Georg[1], Rudolf Schill[2], Maren Klever[3], Lars Grasedyck[3], Rainer Spang[4], Tilo Wettig[1]

[1]Department of Physics, University of Regensburg
[2]Department of Biosystems and Engineering, ETH Zürich
[3]Institute for Geometry and Applied Mathematics, RWTH Aachen University
[4]Departement of Informatics and Data Science, University of Regensburg

Scan for digital version

Summary

- When working on high-dimensional data, approximations are often necessary to keep calculations tracktable
- This can be problematic when probabilities are involved, as these approximations can lead to small negative entries in the approximation of probability vectors
- Existing approaches are either problem-specific or computationally expensive
- Our method [1] improves on this by providing a generic approach to this problem that does not come with a computational overhead

Problem statement

- The Kullback-Leibler (KL) divergence for two discrete probability distributions $\mathbf{p}$ and $\mathbf{q}$ is defined as [2]

$$D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}$$

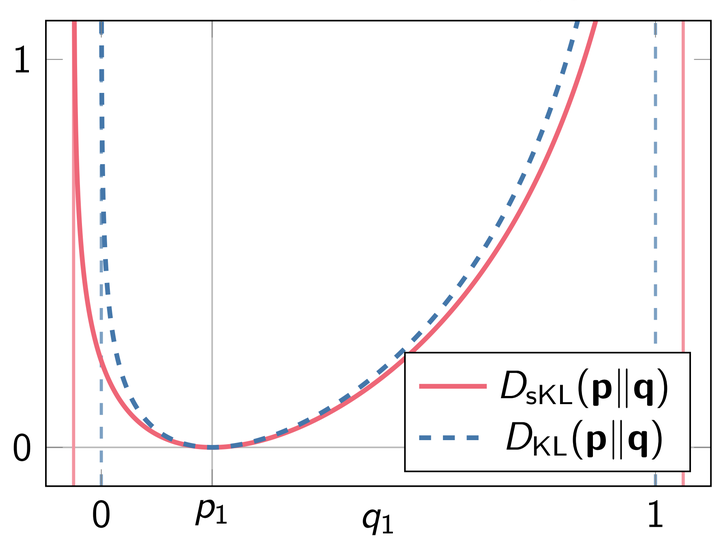
- $\mathbf{p}$ and $\mathbf{q}$ are often given by the probability distributions of the data and the model
- Approximations in the calculation of $\mathbf{q}$ can lead to negative entries $q_i < 0$
  $\Rightarrow$ KL divergence is no longer well-defined

shifted KL divergence

- Idea: shift the entries of $\mathbf{q}$ such that the shifted entries are positive and the logarithm is well-defined
- Many important properties of the KL divergence have to be preserved, in particular the resulting function still has to be a statistical divergence
  $\Rightarrow$ To achieve this, the probability vector $\mathbf{p}$ also has to be shifted
- Definition of the shifted Kullback-Leibler (sKL) divergence:

$$D_{\mathrm{sKL}}(\mathbf{p}\|\mathbf{q}) = \sum_{i=1}^{n} (p_i + \varepsilon_i) \log \frac{p_i + \varepsilon_i}{q_i + \varepsilon_i}$$

- This introduces a parameter vector $\boldsymbol{\varepsilon} \in \mathbb{R}^n_{\geq 0}$
- The sKL divergence is now well-defined for $q_i > -\varepsilon_i$
- Regardless of the choice of $\boldsymbol{\varepsilon}$, the sKL divergence satisfies important properties:
  - D_{sKL} is a statistical divergence for $\mathbf{p}$ and $\mathbf{q}$ that satisfy $\sum_{i=1}^n p_i = \sum_{i=1}^n q_i$
  - D_{sKL} is convex in the pair of its arguments
- Usefulness highly depends on the choice of the parameters
- Simplest choice: constant vector $\boldsymbol{\varepsilon}$
  - Makes usage of D_{sKL} in higher-order optimizers possible
  - Not useable in most realistic cases (e.g. Gaussian noise)



Plot of the KL and sKL divergence of probability vector $\mathbf{p} = (0.2, 0.8)$ from $\mathbf{q} = (q_1, 1 - q_1)$. For the sKL divergence, $\varepsilon_i = 0.05$ was chosen for $i = 1, 2$.

Dynamic parameter choice

- Dynamically choosing $\boldsymbol{\varepsilon}$ after $\mathbf{p}$ and $\mathbf{q}$ are known is usually a better option
- Define a function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ and choose

$$\varepsilon_i = \begin{cases} 0\,, & p_i = 0 \text{ or } q_i > 0\,, \\ |q_i| + f(|q_i|)\,, & \text{else} \end{cases}$$

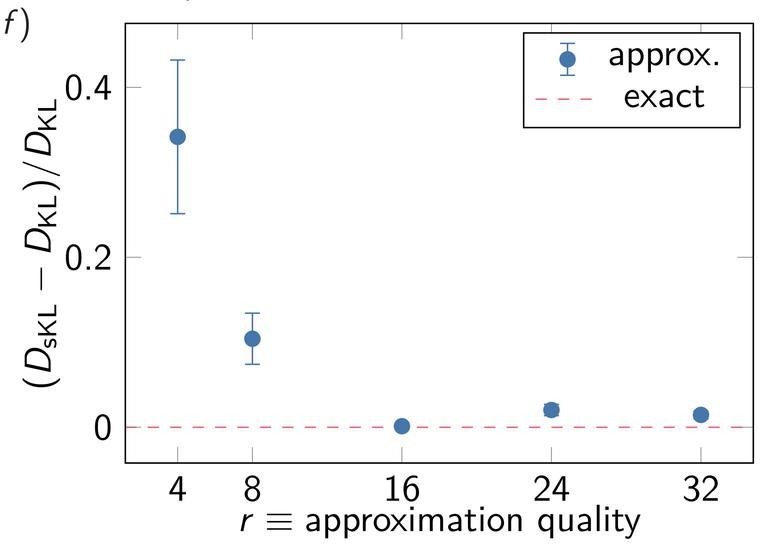
- Ensures that $D_{\mathrm{sKL}}(\mathbf{p}\|\mathbf{q})$ is well-defined regardless of the values in $\mathbf{q}$
- A nonzero ε_i is only introduced when needed, resulting in a small difference between the KL and sKL divergences
- For exact probability distributions $\mathbf{p}$ and $\mathbf{q}$ and a vector $\mathbf{x}$ of i.i.d. Gaussian random variables with mean 0 and standard deviation σ, the average of the sKL divergence is given by

$$\left\langle D_{\mathrm{sKL}}(\mathbf{p}\|\mathbf{q} + \mathbf{x}) \right\rangle_{\mathbf{x}} = D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) + \sigma^2 \sum_{i=1}^{n} \frac{p_i}{2q_i^2} + \mathcal{O}(\sigma^4)$$

  This formula remains true for a large class of different noise distributions
- In the following, we use the simple choice

$$f(x) = \delta \cdot x$$

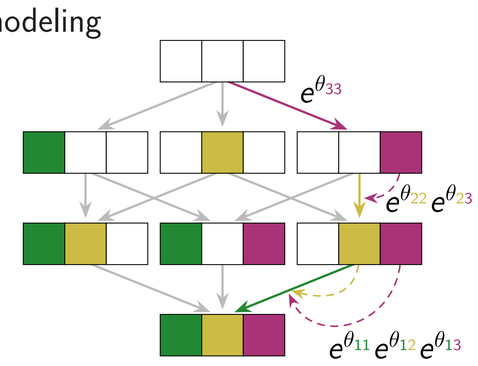
with parameter $\delta = 10$ (arbitrary choice, similar results are obtained with other choices of f)



Application: Mutual Hazard Networks [3]

- Real-world application: Cancer progression modeling
- Cancer progresses by accumulating genetic events, so tumors are represented by binary vectors $x \in \{0, 1\}^d$
- Progression is modeled as a Markov chain with transition rates

$$\mathbf{Q}_{x^{+i},x} = e^{\theta_{ii}} \prod_{x_j=1}^{d} e^{\theta_{ij}}$$



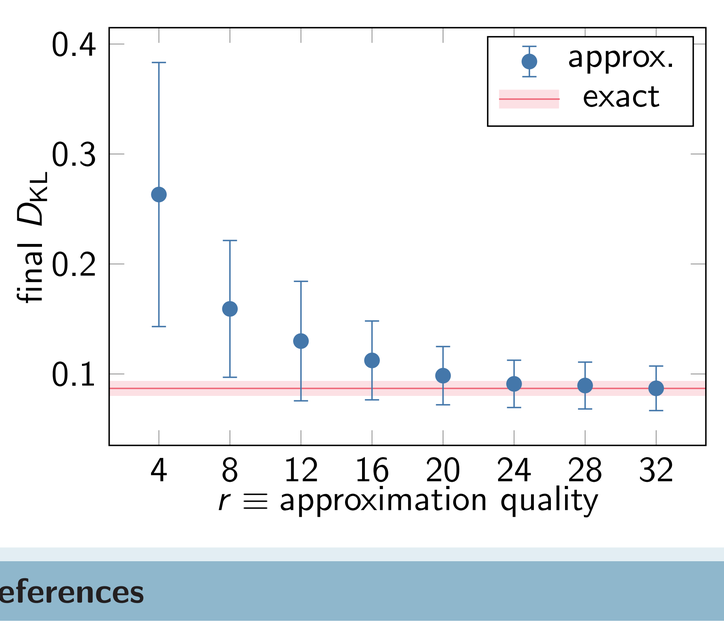
  parameterized by a matrix $\theta \in \mathbb{R}^{d \times d}$
- The time-marginalized probability distribution of tumors is given by

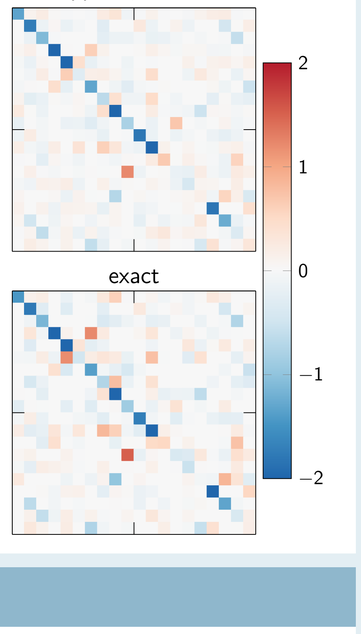
$$\mathbf{q}_\theta = (\mathbf{I} - \mathbf{Q})^{-1} \mathbf{q}_\varnothing$$

  with initial distribution $\mathbf{q}_\varnothing = (1, 0, \ldots)$
- The relationship (inhibition/promotion) between events is given by θ, which can be obtained by minimizing the distance between $\mathbf{p}_\theta$ and a patient data distribution $\mathbf{p}_D$
- Problem: For $\gtrsim 25$ possible events, optimization is prohibitively slow due to the exponential increase in size of x and $\mathbf{Q}$
  - Approximations are needed to enable calculations with more events
  - We use the tensor-train format to approximate the high-dimensional tensor $\mathbf{p}_\theta$
  - Approximation quality is controlled through the tensor-train rank r
  - Comparison to exact calculation is possible for small enough number of events

Application: Results

- We used $d = 20$ events, so results can be compared to the exact solution
- The sKL divergence with approximations is used during optimization, KL divergence is used for evaluation





References

[1] Simon Pfahler et al. "Taming Numerical Imprecision by Adapting the KL Divergence to Negative Probabilities". Dec. 20, 2023. preprint.

[2] Solomon Kullback and Richard A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), p. 79.

[3] Rudolf Schill et al. "Modelling Cancer Progression Using Mutual Hazard Networks". In: *Bioinformatics* 36.1 (Jan. 1, 2020), pp. 241–249.

Supported by the German Research Foundation (DFG)

Correspondence: simon.pfahler@ur.de