Mini-project 2

Advanced Machine Learning (02460)
Technical University of Denmark
Søren Hauberg

March 2024
(Version 2.0)

1 Formalities

This is the project description for the second mini project in *Advanced Machine Learning* (02460). The project is part of the course exam and will count towards your final grade.

Deadline You must submit your report as a group electronically via DTU Learn by 3 April 2024 at 12:00 (noon).

Groups You must do the project in groups of 3–4 people. You need an exception to deviate from this group size. You do not need to document individual contributions.

What should be handed in? You must hand in a single report in PDF format and your code in a single file (either a zip or tar archive).

Length The report must follow the published LaTeXtemplate and consist of:

1. A single page with the main text, including figures and tables. This page must include names, student numbers, course number and the title "Mini-project 2" (so do not include a front page).
2. Unlimited pages of references.
3. A single page of well-formatted code snippets.

You must use at least font size 10pt and margins of at least 2cm. Any content violating the space limitation will not be evaluated.

Code You may use all code you were given during weeks 1–7 in the course. If you use code from elsewhere, it must be documented in the report.

2 Project description

In this project, you will estimate geometries using variational autoencoders (VAEs) on a subset of (non-binarized) MNIST. The project is divided into three parts, where the last is optional. You must document your work in the report and provide relevant plots (example plots generated with my implementation are available in the report template for reference).

Part A: Fisher-Rao geodesics

We will consider a subset of MNIST containing 3 classes and a total of 2048 observations. You should train a VAE of a standard Gaussian prior and a product of either continuous Bernoulli or Bernoulli likelihood, $p(\mathbf{x}|\mathbf{z})$. The latent space should be two-dimensional to ease plotting. The code hand-out provides a starting point. Note that the hand-out training loop adds a bit of noise to the data as this improves the ensemble studied in Part B.

You must implement an algorithm to compute geodesics under the Fisher-Rao metric. The report should include:

- A plot of the latent variables alongside geodesics between, at least, 50 random latent variable pairs.
- A code snippet of the algorithm for computing the Fisher-Rao energy of a curve.
- A code snippet of the algorithm used for computing geodesics.
- A discussion of the qualitative behavior of the computed geodesics, e.g. are they reliable across training runs?

Note that code-snippets can be shortened in the report to save space as long as they preserve the main points of the code.

Part B: Ensemble VAE geometry

In the second part, we will take model uncertainty into account to increase the robustness of the learned geometry. You must train a VAE with an ensemble decoder with 10 ensemble members. You must further implement an algorithm for computing the *model-average Fisher-Rao curve energy*, and compute geodesics by minimizing this. Specifically, the model-average Fisher-Rao curve energy can be computed with the following approximation

$$\mathcal{E}(c) \approx \sum_{i=0}^{N} \mathbb{E}_{l,k} \mathrm{KL}\big(f_l(c(t_i)) \parallel f_k(c(t_{i+1}))\big), \tag{1}$$

where f_l and f_k denotes decoder ensemble members drawn uniformly. This expectation can be approximated using Monte Carlo.

The report should include:

- A plot of the latent variables alongside geodesics between the same random latent variable pairs as in Part A computed using the ensemble decoder VAE.
- A plot of the average proximity between the computed geodesics and the latent variables as a function of the number of ensemble members (from 1 to 10). The proximity between a curve is defined as the largest distance to the nearest data neighbor along the curve, i.e.

$$\text{proximity} = \max_t \min_n \|c(t) - \mathbf{x}_n\|. \tag{2}$$

  This is implemented in the `proximity` function part of the code hand-out. Note that to make this plot as a function of number of ensemble members, you only need to train a single ensemble with 10 members (e.g. there is no need to train an ensemble with 9 members as that can be extracted from the 10-member model without retraining). **Update:** the original hand-out code had a bug in the `proximity` code: the minimum should be with respect to dimension 1 rather than 0. The code has been updated.
- Relevant code snippets.

Part C: Impact of initalization

Optimizing geodesics can be a difficult problem, and initialization techniques can speed up convergence and yield better optima. If time allows, you should initialize the geodesic curves by computing geodesics under an *abstract density metric* (see Eqs. 6.1 and 6.2 in the DGGM book, and exercise 6.6). For the density estimate, you can e.g. using a kernel density estimator.

- A discussion of the impact of the initialization strategy.
- Relevant code snippets.