Diffusion models for probabilistic programming

Semester project

June, 2022

Project description

Diffusion probabilistic models (DPMs) have recently seen increased attention in machine learning for generative modelling of images, text or graphs. DPMs are simple Markovian models that are trained to gradually remove noise from a variable to match a data set after finite time (see Fig. 1). Training DPMs boils down to optimizing a variational objective which suggests their application for approximate inference, i.e., for automated inference of latent variables in probabilistic programming languages (PPLs).

![Diagram of generative and diffusion processes](image)

Figure 1: The generative process (solid lines) and diffusion process (dashed lines) of diffusion models (adopted from Ho, Jain, and Abbeel 2020).

Probabilistic programming is concerned with performing Bayesian inference using the tools of computer science. PPLs allow to automatically infer the posterior over latent variables and learn parameters of interest of a probabilistic model, for instance, via Markov chain Monte Carlo or variational methods. Hence, PPLs lie at the interface of computer science and statistics, since they incorporate ideas from programming language theory, approximate inference and probabilistic machine learning. Currently, the two main approaches in automated probabilistic programming with variational inference are either to naively factorize variational guides [1] or via normalizing flows [6]. Both of these can suffer from poor performance in probabilistic programs if the prior model has a complex structure.

In this project, we explore diffusion probabilistic models for use in probabilistic programming. Here, we implement and validate denoising diffusion probabilistic models [4] and variations thereof [7, 8, 3] and compare them in terms of inferential accuracy with conventional methods that are currently supported in probabilistic programming languages. For this project, the student will read the recent literature on diffusion models, develop a method for automatic inference of posterior distributions, implement it using the PPL NumPyro (see Fig. 2., Phan, Pradhan, and Jankowiak [5]), and evaluate the developed methodology on complex probabilistic models. If successful, the project can be continued by exploring alternative latent structures, e.g., following the ideas in Caterini et al. [2], and

![Code snippet for the eight schools model](image)

Figure 2: A probabilistic program of the the eight schools model written in NumPyro.
published in appropriate journals or conferences.

**Additional Information**

- **What will you learn?**
  - Good understanding of diffusion probabilistic models, probabilistic programming, variational inference
  - Python packaging and tools of the Python ecosystem

- **Requirements**
  - Python experience (!)
  - Experience with TensorFlow/JAX helpful
  - Basics of statistical machine learning and deep learning

- **Supervisor** Dr Simon Dirmeier (simon.dirmeier@sdsc.ethz.ch) and Prof Dr Fernando Perez Cruz

**References**


