Probabilistic Inference

The probabilistic formulation of inference-conditioning probability measures encoding prior assumptions by multiplying with a likelihood describing the data given the generative process (refer to Figure 1.9)-remains one of the main research streams within machine learning, and therefore, of the EI Department, where we address the main aspects of probabilistic inference.

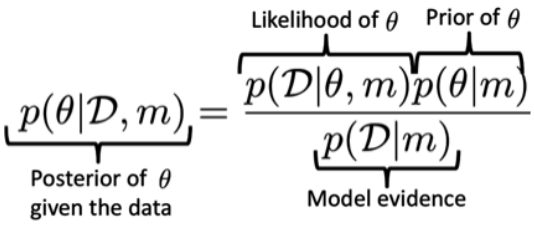

Figure 1.9: Illustration of probabilistic inference.

Nonparametric inference on function spaces Gaussian Process models allow for non-parametric inference in function spaces. They have a strong inductive bias specified by the covariance (kernel) function between observations, which allows for data-efficient learning. One of our main themes in this field has been non-parametric inference on function spaces using Gaussian process models, with its main practical challenge for inference being the cubic complexity in terms of the number of training points. In our work carried out on this topic, we provided a more thorough theoretical and practical analysis of two classes of approximations [222], highlighting both the theoretical merits but also the optimisation problems associated with variational inference. In another project dealing with Gaussian Processes, we have explored models that can account for invariances between data points [123]. To this end, we construct a covariance kernel that explicitly takes these invariances into account by integrating over the orbits of general symmetry transformations. We provide a tractable inference scheme that generalises recent advances in Convolutional Gaussian Processes.

In the work on the Mondrian kernel [223], we provide a new algorithm for approximating the Laplace kernel in any kernel method application (including in Gaussian processes) using random features. Attractive properties of the algorithm include that it allows finding the best kernel bandwidth hyperparameter efficiently and it is well-suited for online learning.

Variational Inference Variational inference is a popular technique to approximate a possibly intractable Bayesian posterior with a more tractable one. Recently, boosting variational inference has been proposed as a new paradigm to approximate the posterior using a mixture of densities by greedily adding components to the mixture. However, its theoretical properties were unclear. In our research [151], we study the convergence properties of this approach from a modern optimization viewpoint by establishing connections to the classic Frank-Wolfe algorithm. Our analysis yields novel theoretical insights regarding the sufficient conditions for convergence, explicit rates, and algorithmic simplifications.

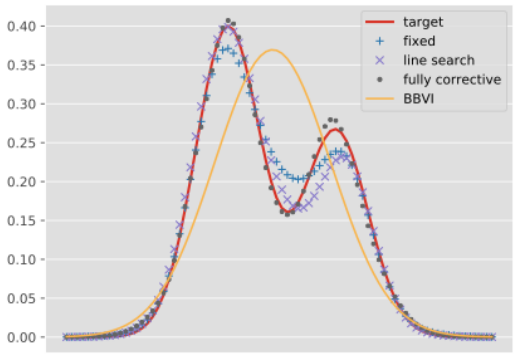

Figure 1.10: Comparison between black box variational inference (BBVI) and three variants of our boosting BBVI method on a mixture of Gaussians [124].

Unfortunately, the above work imposes stringent assumptions that require significant effort for practitioners. Specifically, they require a custom implementation of the greedy step (called the LMO) for every probabilistic model with respect to an unnatural variational family of truncated distributions. In a more recent work [124], we fix these issues with novel theoretical and algorithmic insights. On the theoretical side, we show that boosting variational inference (VI) satisfies a relaxed smoothness assumption which is sufficient for the convergence of the functional Frank-Wolfe (FW) algorithm. Furthermore, we rephrase the LMO problem and propose to maximize the Residual ELBO (RELBO) which replaces the standard ELBO optimization in VI.

These theoretical enhancements allow for black box implementation of the boosting subroutine. As a result, the proposed boosting black box variational inference algorithm can be readily implemented in any probabilistic programming framework based on variational inference. As shown in Figure 1.10, it leads to richer posterior approximations that standard black box variational inference (BBVI).

Inference on discrete graphical models Another open challenge is to perform efficient inference in discrete graphical models, where estimating normalising constants and sampling is often difficult. In this context, in [176], we show that the Gumbel trick – known to convert either the partition-function-estimation problem (an integration problem), or the sampling problem, into an optimisation problem – is just one method out of an entire family and that other methods of the family can sometimes work better. For partition function estimation this means that other members of the family can lead to estimators of the partition function that have lower variance, or lower mean-squared-error, than the estimator obtained from the standard Gumbel trick. This paper received a Best Paper Honourable Mention at ICML 2017.

In a more applied piece of work [12], a Bayesian inference algorithm on slice sampling and particle Gibbs with ancestor sampling is developed, to efficiently deal with the combinatorial number of states in the infinite factorial finite state machine model, which is here used to address the problem of joint channel parameter and data estimation in a multiuser communication channel in which the number of transmitters is unknown.

Probabilistic programming Recent probabilistic programming languages aim to enable data scientists to express sophisticated probabilistic models appropriate for her data as programs, without needing to worry about the inference step, since they come with the implementation of multiple algorithms for performing posterior inference for models and data sets expressed in these languages. In this sense, probabilistic programming languages will help pave the way to statistical data science and AI.

In [132], we present an architectural design of a library for Bayesian modeling and inference in modern functional programming languages. The novel aspect of the approach is modular implementations of existing state-of-the-art inference algorithms. The design relies on three inherently functional features: higher-order functions, inductive data-types, and support for either type-classes or an expressive module system. We provide a performant Haskell implementation of this architecture, demonstrating that high-level and modular probabilistic programming can be added as a library in sufficiently expressive languages.

Although in probabilistic programming languages, sophisticated inference algorithms are often explained in terms of the composition of smaller parts, neither their theoretical justification nor their implementation reflects this modularity. In [133], it is shown how to conceptualise and analyse such inference algorithms as manipulating intermediate representations of probabilistic programs using higher-order functions and inductive types, and their denotational semantics.

A theoretical study relevant to probabilistic programming [227] deals with the problem of representing the distribution of $f(X)$ for a random variable X and a function f. We use kernel mean embedding methods to construct consistent estimators of the mean embedding of $f(X)$. The method is applicable to arbitrary data types on which suitable kernels can be defined. It thus allows us to generalize (to the probabilistic case) functions in computer programming which are originally only defined on deterministic data types.

More information: https://ei.is.mpg.de/project/probabilistic-inference