Learning Algorithms

A *learning algorithm* is the backbone of machine learning that distinguishes it from traditional computer programming by allowing data-driven model building. In the past years, we have developed learning algorithms using a number and tools and for diverse application domains, as outlined below.

Learning with Kernels Kernel methods offer a mathematically elegant arsenal to help tackle several problems that arise in machine learning ranging from probabilistic inference to deep learning. Recently, a subfield of kernel methods known as *Hilbert space embedding of distributions* has gained increasing popularity [59], thanks to foundational work done in our department during the last 10+ years. For a probability distribution $\mathbb{P}$ over a measurable space $\mathcal{X}$, the kernel mean embedding of $\mathbb{P}$ can is defined as the mapping $\mu : \mathbb{P} \mapsto \int k(x, \cdot)\, \mathrm{d}\mathbb{P}(x)$ where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel function. Its applications include, but are not limited to, comparing real-world distributions on the basis of samples, differentially private learning, and determining goodness-of-fit of a model.

Our department has an ongoing history of contributions to the state-of-the-art in this area. In [155], we develop privacy-preserving algorithms based on the kernel mean embedding that allow one to release a database while guaranteeing the privacy of each record in the database.

In applications such as probabilistic programming, transforming a base random variable X with a function f forms a basic building block to manipulate a probabilistic model. It then becomes necessary to characterize the distribution of $f(X)$. In [227], we show that for any continuous function f, consistent estimators of the mean embedding of a random variable X lead to consistent estimators of the mean embedding of $f(X)$. For Matèrn kernels and sufficiently smooth functions, we also provide rates of convergence.

In [126], we address the problem of measuring the relative goodness of fit of two models using kernel mean embeddings. Given two candidate models, and a set of target observations, the goal is to produce a set of interpretable examples (so-called informative features) which indicate the regions in the data domain where one model fits better than the other. The task is formulated as a statistical test whose runtime complexity is linear in the sample size.

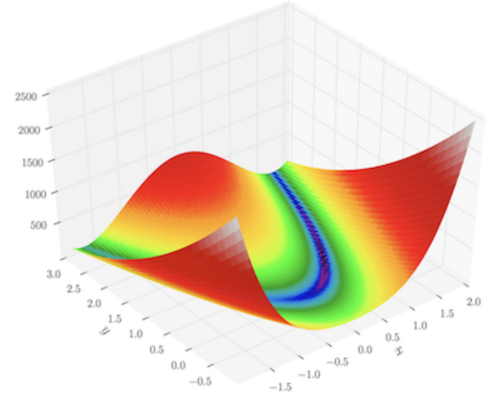

Figure 1.5: The Rosenbrock function, a non-convex function which serves as a test-bed for optimization algorithms (image credit: Wikipedia)

Optimization for Machine Learning Optimization lies at the heart of most machine learning algorithms. Characteristics of modern large-scale machine learning problems include: high-dimensional, noisy, and uncertain data; huge volumes of batch or streaming data; intractable models, low accuracy, and reliance on distributed computation or stochastic approximations. Optimizing under these settings with approaches such as coordinate descent and the Frank-Wolfe algorithm has shown promosing results in recent years. The high-level goal of research in optimization in our department is to understand the convergence property of coordinate descent as well as Frank-Wolfe optimization algorithms under different sampling schemes and constraints.

It is well known that greedy coordinate descent (CD) converges faster in practice than the randomized version, however the properties of greedy CD were less well understood. In [188], we provide a theoretical understanding of greedy coordinate descent for smooth functions. We also propose an approximate greedy CD approach which is computationally cheap and always provably better than the randomized version. Similarly, in [172] we propose an adaptive recursive sampling scheme based on the min-max optimal solution of the variance reduction problem to achieve faster convergence for CD. The proposed approach can also be applied to stochastic gradient descent.

Matching pursuit (MP), Frank-Wolfe (FW), and coordinate descent do have a similar structure of the optimization problem. A connection between MP and coordinate descent is explored in [144]. We also prove the rate for accelerated matching pursuit, which was not known previously. The MP algorithm for optimization over conic hulls is proposed in [170]. In [143], an easy-to-implement conditional gradient method is proposed for a composite minimization problem, which converges at the rate of $O(1/\sqrt{k})$. In a different line of work, we propose a Frank-Wolfe based approach to boost variational inference [151], which enables us to analyze the convergence of the proposed framework under suitable assumptions.

Extreme Classification Extreme multi-label classification refers to supervised multi-label learning involving hundreds of thousands or even millions of labels. It has been shown that machine learning problems arising in tasks such as recommendation, ranking, and web-advertising can be reduced to the framework of extreme classification. It had been long conjectured that a binary-relevance-based one-vs-rest scheme is not statistically and computationally tenable for such scenarios. Surprisingly, we have been able to show in our recent work [204], that a Hamming loss minimizing one-vs-rest paradigm is key to getting good prediction performance, as well as to efficient training (by enabling parallel training). DiSMEC [204], when published in 2016, surpassed the contemporary state-of-the-art methods by up to 10% points on various datasets consisting of up to a million labels. Since then, it has been a top-performing benchmark method in this domain for over two years now. The concurrent training coupled with model pruning paradigms in DiSMEC have motivated algorithms by Microsoft research which have been used in the Bing Search engine for dynamic search advertising and related searches.

Neural Networks Research interest in *deep neural networks*, especially in the generative adversarial network (GAN) approach [112, 135, 150, 200, 202], has increased substantially in recent years. In [150], we propose a simple module to improve a GAN by preprocessing samples with a network that initially makes the task of the discriminator harder (akin to a data smoothing), thus simplifying the generator's task. This leads to a tempered learning process for both generator and discriminator. In a number of experiments, the proposed method can improve quality, stability and/or convergence speed across a range of different GAN architectures (DCGAN, LSGAN, WGAN-GP). In [200], we propose the AdaGAN, a boosting style meta-algorithm which can be combined with various modern generative models (including GANs and VAEs) to improve their quality. We provide an optimal closed form solution for performing greedy updates to approximate an unknown distribution with a sequentially built mixture in any given f-divergence. The paper establishes a fruitful connection between learning theory and neural network research and has already attracted a large amount of follow-up work.

The work [201] develops a deep neural network that can learn to write programs from a corpus of program induction problems. The approach leads to an order of magnitude speedup over strong baselines and an approach based on a recurrent neural network (RNN).

Ideas from causality are beginning to influence our work on machine learning, and the notion of *independent causal mechanism* has been adopted in several areas including semi-supervised learning, domain adaptation, and transfer learning. From a deep learning perspective, we investigated whether a set of independent mechanisms can be recovered using deep neural networks [139]. We proposed an algorithm that enables a set of experts (i.e., deep neural networks) to recover independent (inverse) mechanisms from a data set that has undergone unlabelled transformations. Using a competitive training procedure, the experts specialize to different mechanisms. Not only can the mechanisms be learned successfully, but the system also generalizes to transformed data in other domains.

More information: https://ei.is.mpg.de/project/learning-algorithms