Learning ridge functions from point queries
(Master thesis project)

Hemant Tyagi
INRIA Lille-Nord Europe (MODAL team)
E-mail: hemant.tyagi@inria.fr

Many problems in science and engineering can be typically modeled as that of learning an unknown function $f : \mathbb{R}^d \to \mathbb{R}$ from its samples $(x_i, y_i)_{i=1}^n$ where $x_i \in \mathcal{S} \subset \mathbb{R}^d$ ($\mathcal{S}$ is compact) and

$$y_i = f(x_i) + \eta_i; \quad i = 1, \dots, n$$

with η_i denoting noise. In particular, a common setting in many applications is freedom to obtain the value of f at any location $x \in \mathcal{S}$. Under suitable assumptions on the smoothness of f, one is interested in deriving efficient algorithms for learning f, with n small. It is well known that provided we only make smoothness assumptions on f (such as differentiability or Lipschitz continuity), then the problem is intractable, i.e., has exponential complexity (in the worst case) with respect to the dimension d. For instance if $f \in C^r(\mathcal{S})$, then any algorithm needs in the worst case $n = \Omega(\delta^{-d/r})$ samples to uniformly approximate f with error $\delta \in (0,1)$, cf. [1, 2]. Furthermore, the constants behind the Ω-notation may also depend on d. This exponential dependence on d is referred to as the *curse of dimensionality* and suggests that in order to get *tractable* algorithms in the high dimensional regime, one needs to make additional *structural* assumptions on f.

Ridge functions. A popular class of functions are so-called ridge functions of the form

$$f(x) = g(Ax + b) \tag{1}$$

where $A \in \mathbb{R}^{k \times d}$ and $b \in \mathbb{R}^k$ (with $k < d$). These functions have a rich history in mathematics and arise for instance in the areas of statistics [3, 4] and approximation theory [5, 6]. They are intrinsically k dimensional and so one could hope to derive algorithms which uniformly approximate f with the number of queries depending at most exponentially in k and polynomially in d. Thus in the setting where $k \ll d$, we would have bypassed the curse of dimensionality. Recent results in this regard confirm that this is possible [7, 8]. These algorithms consider f to be sufficiently smooth and are based on numerical approximation of the gradient of f at sufficiently many points. While this is a natural approach, such schemes are sensitive to the choice of the step-size for estimating the gradient – especially in the presence of noise. Moreover, these methods cannot be used for learning f which are not continuously differentiable, for eg., Hölder continuous functions.

Goal(s) of the project. Assuming freedom to sample f within (a compact subset of) its domain, we are primarily interested in answering the following question.

Can one tractably learn f via an approach which is not based on estimating its gradient?

Since answering this question in its full generality might be ambitious, we will begin with the following restricted problem where

$$f(x)=\sum_{i=1}^{k}\alpha_i \max\left\{w_i^T x+b_i,0\right\}; \quad \alpha_i,b_i\in\mathbb{R}; w_i\in\mathbb{R}^d, \tag{2}$$

and the goal is to estimate w_i, α_i, b_i from point queries of f. The function in (2) is a neural network (NN) with one hidden layer and with the activation function being a ReLU (Rectifiable linear unit)[1]. We would like to derive an efficient algorithm for estimating f, i.e., estimating w_i, α_i, b_i for each i, and understand the sample complexity for the same. As a warm-up, one can even make the assumption that the w_i's are orthogonal. At the end, we would ideally like a result that captures the relative geometry of the w_i's.

Depending on the progress made and the preference of the student, we will then start considering generalizations of (2). Some possibilities are the following.

- Replacing ReLU in (2) with a more general class of functions, leading to f of the form

$$f(x)=\sum_{i=1}^{k} g_i(w_i^T x+b_i); \quad g_i:\mathbb{R}\to\mathbb{R},\ i=1,\dots,k. \tag{3}$$

- A NN consisting of more than one hidden layer, but with the ReLU activation functions. While it would be ideal to capture the dependency on the number of layers, even a result for two hidden layers would be interesting.

The project is primarily theoretical with a focus on proofs. However, time permitting, it would be interesting to complement the theory with numerical simulations.

Pre-requisites. This project is suitable for a Masters thesis or as an internship for PhD students. The student is expected to have a strong mathematical background in linear algebra, probability theory (especially concentration of measure) and optimization. Some basic knowledge in approximation theory would be helpful, but is not necessary.

Logistics. The duration of the project will be around 4–6 months. The student will also receive a monthly stipend of roughly 550 Euros.

References

[1] E. Novak and H. Triebel. Function spaces in lipschitz domains and optimal rates of convergence for sampling. *Constr. Approx.*, 23(3):325–350, 2006.

[2] J. Vybíral. Sampling numbers and function spaces. *J. Compl.*, 23(4-6):773–792, 2007.

[3] Yingcun Xia. A multiple-index model and dimension reduction. *Journal of the American Statistical Association*, 103(484):1631–1640, 2008.

[4] David L. Donoho and Iain M. Johnstone. Projection-based approximation and a duality with kernel methods. *Ann. Statist.*, 17(1):58–106, 1989.

[5] Emmanuel J. Candès. Harmonic analysis of neural networks. *Applied and Computational Harmonic Analysis*, 6(2):197 – 218, 1999.

[1] The function $g(y)=\max\{y,0\}$ is a ReLU.

[6] Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1999.

[7] M. Fornasier, K. Schnass, and J. Vybíral. Learning functions of few arbitrary linear parameters in high dimensions. *Foundations of Computational Mathematics*, 12(2):229–262, 2012.

[8] H. Tyagi and V. Cevher. Learning non parametric basis independent models from point queries via low-rank methods. *Applied and Computational Harmonic Analysis*, 37(3):389 – 412, 2014.