Shannon Inspired Approach to Limits of Learning

Contract No: W911NF-16-1-0561

Quarterly Report for Sep 1, 2016-Nov 30, 2016

PI: Vahid Tarokh

1 Executive Summary

- We develop a new theoretical formulation of the limits of supervised learning when the objective of the learner is given by a parametric loss function and her goal is to assess the generalization capability of her inferred parametric function.

- We present a numerical method for estimating the limits of learning in terms of generalization performance. We tailor the method to linear regression and logistic regression and calculate the learning limits in these scenarios. We compare the developed numerical framework with the existing widely used cross validation technique and demonstrate superior performance with significantly less computation.

- We are currently studying the statistical properties of our developed generalization performance estimator, such as its variance, and its asymptotic distribution, in addition to extending the calculations to more complex problems, such as deep neural networks.

- We are investigating neural circuits in the exacting setting that (i) knowledge acquisition can occur from single interactions, (ii) the results of these acquisitions are rapidly evaluble subcircuits, and (iii) recall in response to an external input can be in the form of a rapid evaluation of a composition of subcircuits that have been acquired at arbitrary different earlier times.

- We are developing efficient algorithms for discovering multiple optima in Bayesian non-negative matrix factorization (NMF). We are currently focused on the question of characterizing the space of non-identifiability within a connected region using rapidly-exploring random trees.

2 Technical Overview

We consider the general supervised learning problem, where the objective of the learner is to build a predictor based on historical data that can make predictions about the future samples. Let $z^n = \{(x_i, y_i)\}_{i=1}^n$ denote the observed data samples that are assumed to be drawn i.i.d. from an unknown joint density function $p(z) = p(x, y)$. We assume that each $x_i \in \mathcal{X} \subseteq \mathbb{R}^p$ is a p-dimensional feature vector and $y_i \in \mathcal{Y}$ is a one-dimensional response (also called label in classification problems), which can be real-valued in regression problems and discrete in classification problems.

We assume that the objective of the learner could be expressed in terms of a parametric loss function $l(z;\theta)$, which is smooth in the parameter vector θ. The learner solves for the unknown parameter vector $\theta \in \Theta$ where $\Theta \subseteq \mathbb{R}^k$. Further, let $L(\theta) := E\{l(z;\theta)\}$ be the risk associated with the parameter vector θ, where the expectation is with respect to the density $p(\cdot)$. Ideally, the goal of the learner is to choose the parameter vector θ^* such that

$$\theta^* = \arg\min_{\theta\in\Theta} L(\theta) = \arg\min_{\theta\in\Theta} E\{l(z;\theta)\}. \tag{1}$$

Since the density function $p(\cdot)$ is not known, the learner cannot compute θ^* and hence cannot achieve $L(\theta^*)$. Instead, she usually solves the empirical version of the problem and chooses $\hat{\theta}(z^n)$ as follows:

$$\hat{\theta}(z^n) = \arg\min_{\theta\in\Theta} \hat{L}(\theta) = \arg\min_{\theta\in\Theta} \hat{E}\{l(z_i;\theta)\}, \tag{2}$$

where $\hat{E}$ is the empirical mean operator with respect to the samples z^n. That is

$$\hat{E}\{f(z_i)\} = \frac{1}{n}\sum_{i=1}^{n} f(z_i). \tag{3}$$

Hence,

$$\hat{L}(\theta) = \hat{E}\{l(z_i;\theta)\} = \frac{1}{n}\sum_{i=1}^{n} l(z_i;\theta). \tag{4}$$

Observe that for any $\theta \in \Theta$, we have $E\left\{\hat{L}(\theta)\right\} = L(\theta)$. We consider the empirical risk corresponding to $\hat{\theta}(z^n)$, i.e., $\hat{L}(\hat{\theta}(z^n))$, which is given by

$$\hat{L}(\hat{\theta}(z^n)) := \hat{E}\left\{l(z_i;\hat{\theta}(z^n))\right\}. \tag{5}$$

This problem formulation encompasses a variety of the machine learning tasks, where the learner solves an optimization problem to minimize some loss function. Examples include solving for the weights in a deep neural network and finding the regressors in a linear/logistic regression problem. Note that so far we have not made explicit assumptions about the unknown density function $p(\cdot)$ and the objective of the learner expressed through l. Our goal is to understand the generalization performance of the learner. That is we assess the performance of the inferred loss function on a new sample z_{new} that has not been used in training.

While the learner can evaluate her performance on the observed data sample (also called the in-sample empirical risk, i.e., $\hat{L}(\hat{\theta}(z^n))$), it is imperative to assess the average performance of the learner on unobserved samples, i.e., $L(\hat{\theta}(z^n))$, which is referred to as the out-of-sample risk. Notice that L is not computable by the learner and hence the out-of-sample loss is

also not computable by the learner. Although the in-sample loss converges to the out-of-sample loss for large n, for any finite n the expected out-of-sample loss is larger than that of the in-sample loss, and the excess loss is called the generalization loss. To this end, the learner can incorporate a regularization term into her loss function so as to obtain a solution that incurs smaller generalization loss. Our goal, on the other hand, is to understand the limits of learning performance on a new sample. In particular, the limits of learning using a regularizer could also be assessed through the same framework that we develop here.

To measure the out-of-sample loss, it is a common practice to partition the data set into two parts, one part used for training the learning machine (training set), and the other used for testing the performance of the learning machine (test set). The learner would estimate the out-of-sample error by evaluating the performance of the learning machine (trained on the training set) on the test set, whose samples were not used in training. Among such validation methods, cross validation (where the training and testing are performed several times on the data set) has gained significant interest as it works very well in many practical situations to estimate the out-of-sample loss.

The estimates derived from cross validation become more accurate for prediction when the size of the training set is close to the size of the entire data set. The best of such estimates is derived from leave-one-out cross validation, in which $(n-1)$ of the samples are used for training while one sample is left out for testing. If this procedure is repeated on the n samples in a round-robin fashion, we will end up with n estimates for the out-of-sample loss. Let us formalize this notion. Let us exclude z_i for testing. Let $z^{n\backslash i} := (z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)$ denote the set of the training examples excluding z_i. Hence, the learner would obtain $\hat{\theta}(z^{n\backslash i})$. Thus, let the cross validation performance be defined as

$$\mathrm{CV}(z^n) := \hat{E}\{l(z_i; \hat{\theta}(z^{n\backslash i}))\}.$$

The empirical mean and the empirical variance of the cross validation performance are usually used by the practitioners as surrogates on assessing the out-of-sample performance of a learning method.

While leave-one-out cross validation provides a simple yet powerful tool to estimate the out-of-sample performance, it brings about an additional factor of n computational cost because the learner has to perform the learning task n times on each $z^{n\backslash i}$ while keeping z_i to test the performance for all $1 \leq i \leq n$. This can make the learning task intractable, particularly when the number of samples is large and the optimization problem that is solved in (2) is costly. For instance, when the learner is optimizing for the weights in a deep neural network with hundreds of neurons and millions of data samples, leave-one-out cross validation

is practically impossible. One common solution to this problem is to perform validation on fewer number of samples, where the downslide is that learner would suffer from an increase in the variance of the estimate of the out-of-sample performance. In this report, we provide an estimator for the limits of learning, which can be computed with small additional cost.

3 Technical Progress

3.1 Limits of Prediction in Supervised Learning

Next, we present the formal setup of the problem. While we give our results in the context of model selection, where the learner is going to choose the proper model complexity based on the samples that she has at hand. We suppose the standard model selection problem where the learner wants to choose a model in the filtration $\{\Theta_k\}_{k=1}^{\infty}$, and the goal is to choose k such that the inferred set of parameters $\hat{\theta}_k(z^n)$ would provide the smallest out-of-sample loss.

Let θ_k^* be

$$\theta_k^* := \arg\min_{\theta_k \in \Theta_k} L_k(\theta_k). \tag{6}$$

The objective of the learner is to choose k^*, which is the solution of the following problem:

$$k^*(z^n) := \arg\min_k L_k(\hat{\theta}_k(z^n)) = \arg\min_k E\{l_k(z_{\text{new}}, \hat{\theta}_k(z^n))\}, \tag{7}$$

where z_{new} is a sample drawn from $p(\cdot)$ and is independent of z^n and the expectation is over z_{new} with respect to $p(\cdot)$. Note that $\hat{\theta}_k(z^n)$ is the solution to (2). The learner cannot compute the expectation with respect to the unknown density $p(\cdot)$. Hence, she would compute the empirical version of the above as

$$\hat{L}_k(\hat{\theta}_k(z^n)) := \frac{1}{n}\sum_{i=1}^{n} l_k(z_i, \hat{\theta}_k(z^n)).$$

We assume that $p(\cdot)$ is sub-Gaussian, $l_k(z;\theta_k)$ is twice differentiable with respect to θ_k for all k, and θ_k^* is in the interior of Θ_k. These set of regulatory assumptions ensure that $\hat{\theta}_k(z^n) \to \theta_k^*$ and $\hat{L}(\hat{\theta}_k(z^n)) \to L(\theta_k^*)$ in expectation (and with high probability) as the sample size n grows. Finally note that in the case where the loss function $l_k(z, \theta_k)$ is non-convex in θ, we assume that the solver converges to a local minimum of (2), which in turn is asymptotically convergent to a local minimum of (1) when the sample size grows large.

Let the expected in-sample loss be denoted by L_k^{IS} and be given by

$$L_k^{\text{IS}} := E\left\{\hat{L}_k(\hat{\theta}_k(z^n))\right\}, \tag{8}$$

where expectation is on z^n with respect to $p(\cdot)$. Recall that the goal of the learner is to estimate the out-of-sample loss $L_k(\hat{\theta}(z^n))$. Let the expected out-of-sample loss be denoted by L_k^{OS} and be given by

$$L_k^{\text{OS}} := E\left\{L_k(\hat{\theta}_k(z^n))\right\}. \tag{9}$$

We are interested in understanding the excess risk (also called the generalization risk) defined as

$$R_k(z^n) = L_k(\hat{\theta}_k(z^n)) - \hat{L}_k(\hat{\theta}_k(z^n)). \tag{10}$$

Note that the generalization risk is not computable by the learner because L_k is not available to the learner. The expected generalization risk (where expectation is with respect to training samples z^n) is denoted by $\overline{R}_k$ and defined as

$$\overline{R}_k := E\left\{R_k(z^n)\right\} = L_k^{\text{OS}} - L_k^{\text{IS}}, \tag{11}$$

In other words, $\overline{R}_k$ quantifies the limits of learning within class Θ_k. Recall that the alternative approach to the estimation of the generalization risk (and out-of-sample risk) is cross validation, where the out-of-sample loss could be directly estimated using the empirical mean of the cross validation, i.e., $\text{CV}_k(z^n)$. Inspired from information theoretic techniques, we will provide an estimator for $\overline{R}_k$, which requires significantly less computational resources.

3.2 Technical Derivation of the Limits of Prediction

In what follows, we derive an estimator for $\overline{R_k}$, which quantifies the limit of learning within class Θ_k. To do so, let us examine $\overline{R}_k$ more closely:

$$\overline{R}_k = E\left\{L_k(\hat{\theta}_k(z^n)) - L_k(\theta_k^*)\right\} \tag{12}$$

$$+ E\left\{\hat{L}_k(\theta_k^*) - \hat{L}_k(\hat{\theta}_k(z^n))\right\}, \tag{13}$$

where the expectations are taken with respect to the training samples z^n, and the equality holds because $L_k(\theta_k^*) = E\left\{\hat{L}_k(\theta_k^*)\right\}$. Note that although for sufficiently large n we have $\hat{L}_k(\hat{\theta}_k(z^n)) \approx L_k(\theta_k^*)$, our interest is in deriving the first significant term in their difference in the asymptote as n grows. For this approximation to hold, we would require that $k = o(n)$, i.e., we could only consider model complexities whose number of parameters are much smaller than n. The extension to the high-dimensional case where the number of parameters in the model can be even larger than the number of samples is part of our ongoing investigation. We will analyze (12) and (13) in more detail below.

Considering (12), we notice that when sufficient data has been acquired, i.e., as $n \to \infty$, we have $\hat{\theta}_k(z^n) \to \theta_k^*$. Further notice that $L_k(\theta_k) - L_k(\theta_k^*)$ is non-negative . Further, as a function of θ_k, it is minimized at $\theta_k = \theta_k^*$. Hence, we can apply Taylor's theorem to $L_k(\theta_k) - L_k(\theta_k^*)$ to derive an expansion around $\theta_k = \theta_k^*$. Note that the first two terms are zero, and hence we only keep the second order term.

$$L_k(\hat{\theta}_k(z^n)) - L_k(\theta_k^*) \approx \frac{1}{2}\left((\hat{\theta}_k(z^n) - \theta_k^*)^\top \nabla^2_{\theta_k} L_k(\theta_k^*)\ (\hat{\theta}_k(z^n) - \theta_k^*)\right), \tag{14}$$

where $\nabla^2_{\theta_k}$ is the Hessian operator with respect to θ_k.

Next, we consider (13). For any z^n, by definition $\hat{\theta}_k(z^n)$ is the minimizer of $\hat{L}_k(\theta_k)$ with respect to θ_k. Hence, for any z^n, we know that $\hat{L}_k(\theta_k) - \hat{L}_k(\hat{\theta}_k(z^n)) > 0$ and is minimized by $\hat{\theta}_k(z^n)$. Thus, we apply Taylor's theorem this time around $\theta_k = \hat{\theta}_k(z^n)$ to obtain

$$\hat{L}_k(\theta_k^*) - \hat{L}_k(\hat{\theta}_k(z^n)) \approx \frac{1}{2}\left((\hat{\theta}_k(z^n) - \theta_k^*)^\top \nabla^2_{\theta_k} \hat{L}_k(\hat{\theta}_k(z^n))\ (\hat{\theta}_k(z^n) - \theta_k^*)\right). \tag{15}$$

Note that $\nabla^2_{\theta_k} \hat{L}_k(\hat{\theta}_k(z^n))$ is the empirical Hessian matrix given by

$$\nabla^2_{\theta_k} \hat{L}_k(\hat{\theta}_k(z^n)) = \frac{1}{n}\sum_{i=1}^{n} \nabla^2_{\theta_k} l_k(z_i; \hat{\theta}_k(z^n)).$$

For sufficiently large n, we have

$$\nabla^2_{\theta_k} \hat{L}_k(\hat{\theta}_k(z^n)) \approx \nabla^2_{\theta_k} L_k(\theta_k^*).$$

We further define the short-hand notations:

$$H_k(\theta_k) := \nabla^2_{\theta_k} L_k(\theta_k) \tag{16}$$

$$\hat{H}_k(\theta_k) := \nabla^2_{\theta_k} \hat{L}_k(\theta_k) \tag{17}$$

Observe that for any matrix H and vector $\mathbf{a}$,

$$\mathbf{a}^\top H \mathbf{a} = \mathrm{tr}\left\{H \mathbf{a}\mathbf{a}^\top\right\},$$

where $\mathrm{tr}\{\cdot\}$ denotes the trace operator, and hence,

$$E\left\{(\hat{\theta}_k(z^n) - \theta_k^*)^\top\ H_k(\theta_k)\ (\hat{\theta}_k(z^n) - \theta_k^*)\right\} = \mathrm{tr}\left\{H_k(\theta_k)\ E\left\{(\hat{\theta}_k(z^n) - \theta_k^*)(\hat{\theta}_k(z^n) - \theta_k^*)^\top\right\}\right\}.$$

Finally, note that applying Taylor's expansion on $L_k(\theta_k^*)$, we can get

$$E\left\{(\hat{\theta}_k(z^n) - \theta_k^*)(\hat{\theta}_k(z^n) - \theta_k^*)^\top\right\} \approx [H_k(\theta_k^*)]^{-1} V_k(\theta_k^*) [H_k(\theta_k^*)]^{-1},$$

where $V_k(\theta_k^*)$ is defined as

$$\begin{aligned} V_k(\theta_k) :&= \operatorname{cov}(\nabla_{\theta_k} l_k(z;\theta_k)) \\ &= E\left\{ \left(\nabla_{\theta_k} l_k(z;\theta_k) - \nabla_{\theta_k} L_k(\theta_k)\right) \left(\nabla_{\theta_k} l_k(z;\theta_k) - \nabla_{\theta_k} L_k(\theta_k)\right)^\top \right\}, \end{aligned}$$

where ∇_{θ_k} denotes the gradient operator with respect to θ_k, and $\operatorname{cov}\{\cdot\}$ is the covariance operator. Note that $\nabla_{\theta_k} L_k(\theta_k^*) = 0$, and hence $V_k(\theta_k^*)$ could be simplified to

$$V_k(\theta_k^*) = E\left\{ \left(\nabla_{\theta_k} l_k(z;\theta_k^*)\right) \left(\nabla_{\theta_k} l_k(z;\theta_k^*)\right)^\top \right\}.$$

Thus,

$$\operatorname{tr}\left\{ H_k(\theta_k^*)\ E\left\{ (\hat{\theta}_k(z^n) - \theta_k^*)(\hat{\theta}_k(z^n) - \theta_k^*)^\top \right\} \right\} \approx \operatorname{tr}\left\{ V_k(\theta_k^*) \left[H_k(\theta_k^*)\right]^{-1} \right\}.$$

Putting all of these together, we propose the following estimator for $\overline{R}_k$:

$$\hat{R}_k(z^n) := \operatorname{tr}\left\{ \hat{V}_k(\hat{\theta}_k(z^n)) \left[\hat{H}_k(\hat{\theta}_k(z^n))\right]^{-1} \right\}. \tag{18}$$

where $\hat{H}$ denotes the empirical Hessian matrix (17) and $\hat{V}$ is the empirical covariance of the score function given by

$$\hat{V}_k(\theta) := \hat{E}\left\{ \left(\nabla_{\theta_k} l_k(z_i;\theta_k)\right) \left(\nabla_{\theta_k} l_k(z_i;\theta_k)\right)^\top \right\}, \tag{19}$$

where $\hat{E}$ denotes the empirical mean operator defined in (3).

The following is our main result about the proposed estimator.

Theorem 1 *The estimator $\hat{R}_k(z^n)$ satisfies:*

$$E\left\{ \hat{R}_k(z^n) \right\} = \bar{R}_k \left(1 + o_n(1)\right).$$

In other words, this is an asymptotically consistent estimator of the generalization risk. Note that since $\mathrm{CV}_k(z^n)$ is an asymptotically consistent estimator of the out-of-sample loss, then $\hat{L}_k(z^n) + \hat{R}_k(z^n)$ and $\mathrm{CV}_k(z^n)$ are asymptotically equivalent in expectation for estimating the out-of-sample loss. Our ongoing work also considers other properties of the estimator $\hat{R}_k(z^n)$, such as its variance as well as its distribution, which we will also compare against cross validation.

3.3 Examples of Calculation of Limits of Prediction

Next, we calculate the limits of learning in two specific supervised learning problems.

3.3.1 Limits of prediction in linear regression with weighted ℓ_2 loss

First, we consider a regression problem. Let $\mathcal{X} = [-1, 1]$, i.e., there is only one single feature vector. Further, let $\mathcal{Y} = \mathbb{R}$. Let $f : [-1, 1] \to \mathbb{R}$ be an analytic function such that it could be expanded via a Taylor's series:

$$f(x) = \sum_{l=0}^{\infty} a_l x^l. \tag{20}$$

We assume that the learner observes data points $z^n = \{(x_i, y_i)\}_{i=1}^n$, where the joint distribution $p(x_i, y_i)$ is such that x_i is uniformly and independently drawn on $[-1, 1]$. Further, $y_i = f(x_i) + \epsilon_i$ such that ϵ_i is i.i.d. and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. The learner does not know $f(x)$ and σ. The learner then receives a new sample x_{new}, and is supposed to predict y_{new}.

Consider the set of all polynomial functions of order at most k, i.e., $\{g_{\theta_k}(x) : \theta_k \in \Theta_k\}$, where $\Theta_k = \{(\theta_{k,0}, \ldots, \theta_{k,k}) : \forall i \in [k]\ \theta_{k,i} \in \mathbb{R}\}$, and for any $\theta_k \in \Theta_k$ we have $g_{\theta_k}(x) = \sum_{l=0}^k \theta_{k,l} x^l$. We assume that the learner solves a regression problem to find the best k-th order fit to the data samples z^n presented to her. We further assume that the her objective is such that the fit is non-uniformly weighted for $x \in [-1, 1]$ according to a weight function $w : [-1, 1] \to \mathbb{R}_{>0}$. Thus, the learner chooses her loss function as follows:

$$l_k(z, \theta_k) = l_k((x, y), \theta) = w(x) \left(y - \sum_{l=0}^{k} \theta_{k,l} x^l \right)^2, \tag{21}$$

If the joint distribution $p(\cdot)$ was known to the learner or alternatively $f(\cdot)$ was known, then the learner could compute θ_k^* by the projection of f onto the set of Legendre polynomials of order at most k. It is clear that by increasing k, the fit on data will improve, i.e., $L(\theta_k^*)$ is a non-increasing function of k.

On the other hand, $p(\cdot)$ is not available to the learner. We assume that the learner solves the minimization problem stated in (2), and will calculate the loss $\hat{L}_k(\hat{\theta}_k(z^n))$ as in (5). It is also clear that $\hat{L}_k(\hat{\theta}_k(z^n))$ is also a non-increasing function of k as the learner's ability to adapt to the data increases with the additional degrees of freedom provided by the model. On the other hand, the out-of-sample loss will suffer from overfitting, i.e., $L_k(\hat{\theta}_k(z^n))$ will initially decrease and then will increase as k grows large. We use the proposed estimator to obtain the generalization risk in this example in order to select the best k that minimizes $L_k(\hat{\theta}_k(z^n))$.

Let $\Phi_{n\times(k+1)} = \Phi(z^n)$ be an $n \times (k+1)$ dimensional matrix such that $\Phi_{ij} = (x_i)^j$. Further, let $W_{n\times n} = W(z^n) = \text{diag}(w(x_1), \ldots, w(x_n))$ and $\mathbf{y} = (y_1, \ldots, y_n)^\top$. Thus, the learner solves

$$\hat{\theta}_k(z^n) = \arg\min_{\theta_k} \left\{ (\mathbf{y} - \Phi\theta_k)^\top W (\mathbf{y} - \Phi\theta_k) \right\}.$$

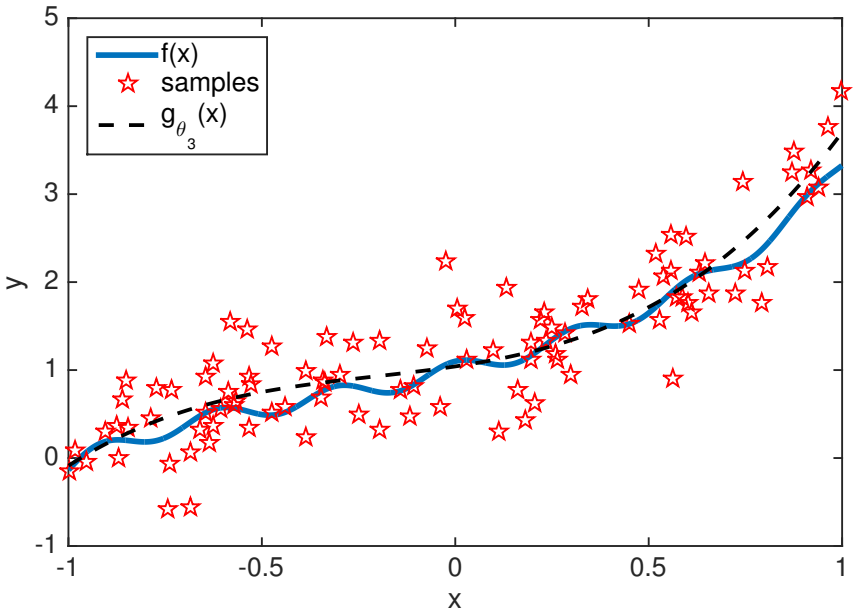

Figure 1: The data samples are generated from $y_i = f(x_i) + \epsilon_i$ where ϵ_i is i.i.d. $\mathcal{N}(0, \sigma^2)$.

The solution to the above could be readily obtained in closed form:

$$\hat{\theta}_k(z^n) = (\Phi^\top W \Phi)^{-1} \Phi^\top W \mathbf{y}.$$

Further, $\hat{L}_k(\hat{\theta}_k(z^n))$ could be calculated in closed form as well:

$$\hat{L}_k(\hat{\theta}_k(z^n)) = \frac{1}{n} \mathbf{y}^\top W (\mathbf{y} - \Phi \hat{\theta}_k).$$

To compute our estimator of the generalization risk, we need to calculate the Hessian matrix, which is readily given by $\nabla^2_{\theta_k} l_k(z_i, \theta_k) = 2\Phi^\top W \Phi$. Thus,

$$\hat{H}_k(\hat{\theta}_k(z^n)) = \frac{1}{n} \sum_{i=1}^{n} \nabla^2_{\theta_k} l_k(z_i, \theta_k) = 2\Phi^\top W \Phi.$$

Next, we compute $\hat{V}_k(\hat{\theta}_k(z^n))$ to be:

$$\hat{V}_k(\hat{\theta}_k(z^n)) = 4\hat{E}\left\{ w(x_i)^2 (y_i - \Phi^\top \hat{\theta}_k(z^n))^2 \Phi_i \Phi_i^\top \right\},$$

where Φ_i is the i-th row of the matrix Φ.

Next, we consider a numerical example. Let

$$p(x, y) = \frac{1}{2\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(f(x)-y)^2},$$

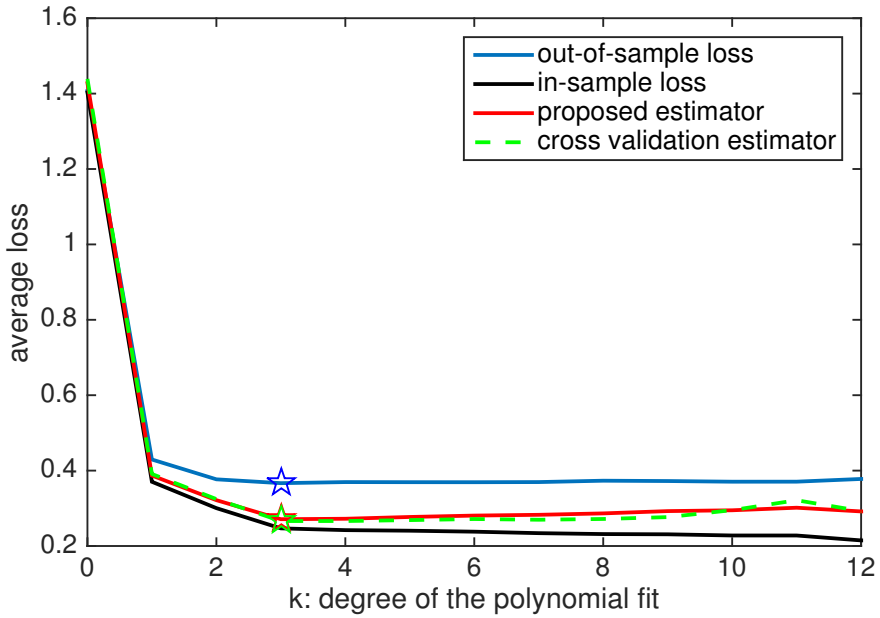

Figure 2: The y-axis is the loss, and the x-axis is k.

where $\sigma = 0.5$ and

$$f(x) = \cosh(\frac{\pi}{3}x) + \tan(\frac{\pi}{3}x) + 0.1\cos(20.48x).$$

Fig. 1 depicts $f(x)$. The learner observes $n = 100$ i.i.d. samples from $p(\cdot)$, which are depicted as red stars. The learner's loss function is given by (21) where $w(x) = 1 + x^2$, i.e., more weight is given to the corners of the domain $\mathcal{X}$. In this case, it is straightforward to show that as $k \to \infty$, $g_{\theta^*_k(x)} \to f(x)$. Thus, $f(x)$ is the target function that a learner with unlimited noiseless observations would be able to obtain. The dashed black curve depicts $g_{\hat{\theta}_3(z^n)}(x)$.

In Fig. 2, we plot the in-sample and out-of-sample losses. The black curve is the in-sample loss observed by the learner, i.e., $\hat{L}_k(\hat{\theta}_k(z^n))$, which is a non-increasing function of k. The blue curve is the expected out-of-sample loss $L_k(\hat{\theta}_k(z^n))$. As can be seen, $L_k(\hat{\theta}_k(z^n))$ attains its minimum at $k^* = 3$. The red curve is the estimate of the out-of-sample loss obtained from our proposed estimator, i.e., $\hat{L}_k(\hat{\theta}_k(z^n)) + \hat{R}_k(z^n)$. The green curve is the cross validation estimate of the out-of-sample error $\mathrm{CV}(z^n)$. As can be seen, both the proposed estimator and cross validation correctly predict the best fit at $k^* = 3$ while cross validation requires n times more computations.

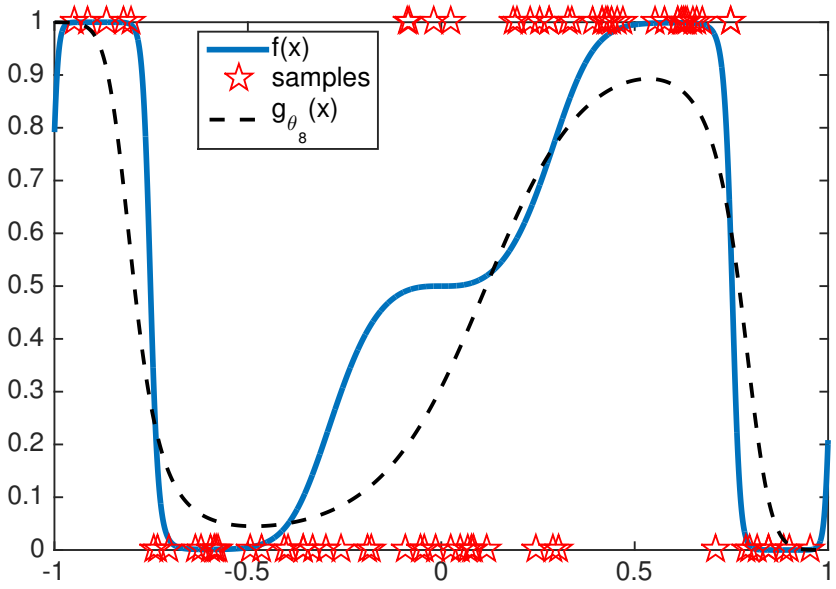

Figure 3: The samples are generated from $p(x, y) = \frac{1}{2}\text{sigmoid}f(x)$.

3.3.2 Limits of prediction in logistic regression

Next, we provide an example of the calculation of the limits of learning in a classification problem. As in the previous example, let $\mathcal{X} = [-1, 1]$, i.e., a single feature. For simplicity of the presentation we focus on binary classification where $\mathcal{Y} = \{0, 1\}$ while the exact same technique carries through for multi-class classification. The learner is presented with z^n, which consist of labeled samples and is supposed to learn a rule to classify future samples. If the learner knew $p(\cdot)$, her classification rule would only depend on the conditional distribution of y given x. In particular, the learner would choose the label to be 1 if $p(Y = 1|x) > p(Y = 0|x)$ and 0 otherwise. Since $p(\cdot)$ is unknown, the learner would seek to approximate the conditional distribution $p(Y = 1|x)$ by a parametric distribution p_{θ_k} defined as

$$p_{\theta_k}(Y = 1|x) = \text{sigmoid}\left(\sum_{l=0}^{k} \theta_{k,l} x^l\right), \tag{22}$$

where

$$\text{sigmoid}(t) = \frac{e^t}{1 + e^t}. \tag{23}$$

As it is the conventional assumption, the learner seeks to choose the parameters to maximize the likelihood of the observed samples, i.e.,

$$\hat{\theta}_k(z^n) = \arg\max_{\theta_k} \left\{ \prod_{i=1}^{n} \left(p_{\theta_k}(Y = 1|x_i)^{y_i} (1 - p_{\theta_k}(Y = 1|x_i))^{1-y_i} \right) \right\}.$$

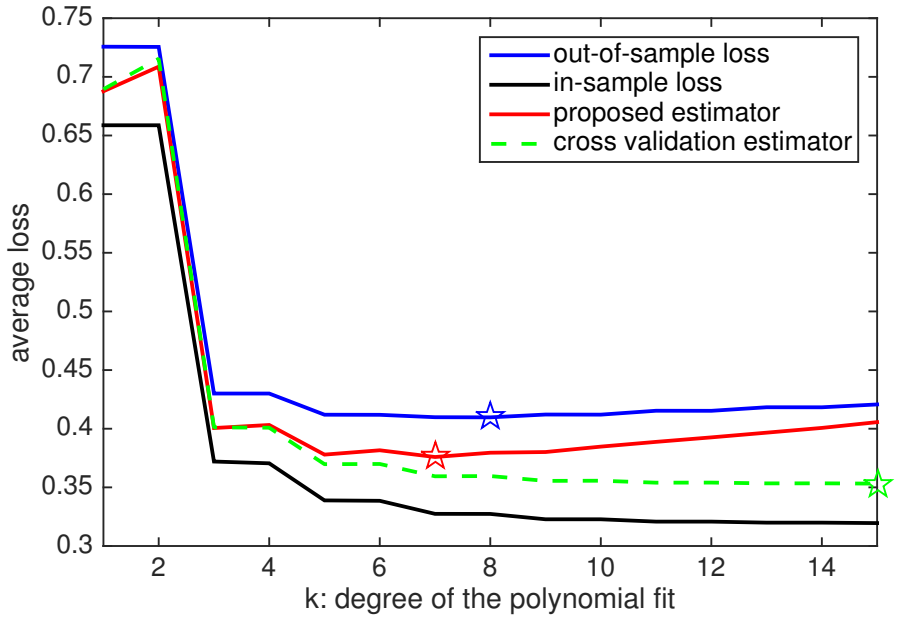

Figure 4: The y-axis is the loss, and the x-axis is k.

Let the loss function be defined as

$$l_k(z,\theta_k) = l_k((x,y),\theta_k) = \left(y \log\left(\frac{1}{p_{\theta_k}(Y=1|x)}\right) + (1-y)\log\left(\frac{1}{1-p_{\theta_k}(Y=1|x)}\right)\right). \tag{24}$$

Hence,

$$\hat{\theta}_k(z^n) = \arg\min_{\theta_k} \hat{E}\left\{l_k(z_i,\theta_k)\right\}. \tag{25}$$

We assume that the learner chooses to solve the classification problem using logistic regression. Unlike the previous example, there is no closed form solution for the logistic regression problem. However, as the objective function is convex, it can be efficiently computed using convex optimization packages.

$\hat{H}_k$ and $\hat{V}_k$ could be calculated as

$$\hat{H}_k(\hat{\theta}_k(z^n)) = \hat{E}\left\{p_\theta(Y=1|x_i)(1-p_\theta(Y=1|x_i))\Phi_i\Phi_i^\top\right\},$$
$$\hat{V}_k(\hat{\theta}_k(z^n)) = \hat{E}\left\{\left[y_i(1-p_\theta(Y=1|x_i))^2 + (1-y_i)p_\theta(Y=1|x_i)^2\right]\Phi_i\Phi_i^\top\right\}.$$

Let us continue through a numerical example. Let x be distributed uniformly over $[-1,1]$. Further, let

$$p(Y=1|x) = \text{sigmoid}(f(x)),$$

where

$$f(x) = 8\sin(6.76x^3) - 5x^3.$$

Fig. 3 depicts $f(x)$. The learner observes $n = 80$ i.i.d. samples from $p(\cdot)$, which are depicted as red stars. The loss function is given by (24). Similar to the previous example, it is straightforward to show that as $k \to \infty$, $g_{\theta_k^*(x)} \to f(x)$. Thus, $f(x)$ is the target function that a learner with unlimited noiseless observations would be able to obtain as she increases the complexity of her model. The dashed black curve depicts $g_{\hat{\theta}_8(z^n)}(x)$.

In Fig. 4, we plot the in-sample and out-of-sample loss incurred to the learner. The black curve is the in-sample loss observed by the learner, i.e., $\hat{L}_k(\hat{\theta}_k(z^n))$, which is a non-increasing function of k. The blue curve is the expected out-of-sample loss $L_k(\hat{\theta}_k(z^n))$. As can be seen, $L_k(\hat{\theta}_k(z^n))$ attains its minimum at $k^* = 8$. The red curve is the estimate of the out-of-sample loss obtained from our proposed estimator, i.e., $\hat{L}_k(\hat{\theta}_k(z^n)) + \hat{R}_k(z^n)$. The green curve is the cross validation estimate of the out-of-sample error $\mathrm{CV}(z^n)$. As can be seen, the proposed estimator estimates $k^* = 7$ whereas cross validation fails at estimating the order correctly. Further, observe that for small k the proposed estimate and cross validation track each other closely.

3.4 Sequential Methods for Calculating the Limits of Learning

Thus far, we derived the limits of prediction for a learner whose objective is to predict future samples. Next, we provide a procedure to characterize model ambiguity under the general objective of the learner. Following the idea introduced in Burnham and Anderson to provide a Bayesian view of the Akaike information criterion, one would hope to extend such framework to derive model ambiguity under the general objective of the learner. This would require to first assume that there exists a prior distribution π on $\{\Theta_k\}_{k=0}^{\infty}$ (if no such prior exists, one can assume the non-informative improper uniform prior on $\mathbb{Z}^+$). In order to demonstrate the idea, we first assume that $l_k(z;\theta_k) = -\log p_{\theta_k}(z)$, i.e., the objective of the learner is to maximize the parametric likelihood function $p_{\theta_k}(z)$. The logistic regression example provided in the previous section falls into this framework. Assuming certain regularity conditions, one can compute the posterior distribution on model classes as follows:

$$\pi\left(\Theta_k|z^n\right) = \frac{e^{-\hat{L}_k(\hat{\theta}_k(z^n))-\hat{R}_k(z^n)}\pi(\Theta_k)}{\sum_{j=0}^{\infty} e^{-\hat{L}_k(\hat{\theta}_j(z^n))-\hat{R}_j(z^n)}\pi(\Theta_j)} \tag{26}$$

where $\hat{\theta}_k(z^n)$ is the maximum likelihood estimator (2), and $\hat{R}_k(z^n)$ is the generalization loss (18).

Recall that our technical derivations only hold for the case where $k = o(n)$, i.e., the number of parameters in the model is much smaller than the number of observed samples. Hence, we

would need an upper bound on k. On the other hand, our estimator for the generalization loss, i.e., $\hat{R}_k(z^n)$, grows large as k grows and it also ensures that $\hat{R}_k(z^n) = \infty$ for all $k > n$, which leads to $\pi(\Theta_k|z^n) = 0$ for all $k > n$. Hence, in practice one could use (26) without worrying about the relationship between k and n.

Further note that in practice the learner would like to be able to calculate the above sequentially as new data samples arrive. This can be done using sequential Monte Carlo methods, such as iterated batch importance sampling, to compute the posterior when the $(n+1)$-th sample arrives in a sequential manner. Within a model class, this can be done explicitly as follows. Let us define $\hat{R}_k(z^n;\theta_k)$ as follows.

$$\hat{R}_k(z^n;\theta_k) := \operatorname{tr}\left\{\hat{V}(\theta_k)\hat{H}^{-1}(\theta_k)\right\},$$

where $\hat{H}(\cdot)$ and $\hat{V}(\cdot)$ are the empirical Hessian and the empirical covariance defined in (17) and (19), respectively. Note that with the above definition we get that the generalization loss defined in (18) would be $\hat{R}_k(z^n) = \hat{R}_k(z^n;\hat{\theta}_k(z^n))$.

Let us consider the following problem: $\min_{\theta_k}\{\hat{L}_k(\theta_k) + \hat{R}_k(z^n;\theta_k)\}$, where we explicitly indicate the dependence of the dimension on the sample size n. By abuse of notation, assume that we have a prior π on θ_k. Then we obtain the posterior on θ_k by:

$$\pi_{n+1}(\theta_k|z^n) = \frac{e^{-\hat{L}_k(\theta_k)-\hat{R}_k(z^n;\theta_k)}\pi(\theta_k)}{\int e^{-\hat{L}_k(\theta_k)-\hat{R}_k(z^n;\theta_k)}\pi(\theta_k)d\theta_k}. \tag{27}$$

Thus, we can sequentially obtain

$$\pi_{n+1}\left(\theta_k|z^{n+1}\right) \propto \pi_n(\theta_k|z^n)\, e^{-l_k(\theta_k|z_{n+1})-\hat{R}_k(z^{n+1};\theta_k)-\hat{R}_k(z^n;\theta_k)}. \tag{28}$$

Once the distributions (26) and (28) are calculated, one could calculate the Shannon entropy of Θ_k to quantify the learner's ambiguity in choosing a model, as well as the Shannon entropy of θ_k for each k to understand the learner's ambiguity in choosing the parameters within each class. Such calculations would provide an alternative for the learner to choose a model class based on her ambiguity. We remark that Shannon entropy does not necessarily capture the general objective of the learner and we are investigating other objectives as part of our ongoing work.

Finally, note that the above approach can be extended to the case where one can sample from parameter of different dimensions, i.e., from different model classes, by using reversible-jump Monte Carlo Markov Chain (MCMC) techniques. These extensions are currently under development as part of our ongoing work. We also remark that although the prior information in the above derivation is used under the assumption that $l_k(z;\theta_k) = -\log p_{\theta_k}(z)$, we are investigating such calculations for the general objective of the learner.

3.5 References

H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, 19(6), 716-723, 1974.

K. P. Burnham and D. R. Anderson, "Multimodel inference understanding AIC and BIC in model selection," *Sociological Methods and Research*, 33(2), 261-304, 2004.

4 Next Steps

- In the first quarter, we derived an estimator for the limit of learning within each class Θ_k when the learner considers the filtration $\{\Theta_k\}_{k=0}^{\infty}$ for the general objective of the learner. We proved that the estimator is asymptotically consistent as the sample size grows for each finite k. We also applied this estimator to two learning problems in linear and logistic regression and derived the limits of learning in these problems. Our ongoing work includes assessing the statistical properties of the estimator of the limits of learning, such as its asymptotic distribution and its variance and comparing them to cross validation techniques. We also plan to tailor the proposed estimator to high-dimensional data where the number of features can be larger than the number of received samples, and to derive the limits of learning on existing datasets of interest. Finally, we derived model ambiguity when the objective of the learner is maximizing a likelihood function. We plan to provide numerical methods for efficient computation of the model ambiguity under a variety of relevant metrics.

- In neural circuits in the exacting setting, the most basic computational task is the cognitive function of association, defined here to mean that a subcircuit is to be set up so that the excitation of a set of neurons that represent A will in future cause the excitation of a set of neurons that represent B. The composability requirement constrains the model to ensure, for each association as realized in a subcircuit, that the excitation in the triggering set of neurons A is quantitatively similar to that in the triggered set B, and also that the unintended excitation in the rest of the system be negligible. These requirements ensure that arbitrary chains of associations can be executed. Previously studied neural models do not have this feature. The aim is to develop a complexity theory, in terms of asymptotically matching upper and lower bounds, on the capacity of a neural network for acquiring and executing, in this setting,

basic cognitive functions such as association. A model of computation is considered that is fully distributed and in which the quantitative resources n, the neuron numbers, d, the number of other neurons each neuron is connected to, and k, the inverse of the maximum synaptic strength, are all accounted for. A further parameter is the topology of the network. We consider sparse random graphs, both where the nodes representing the A and B can be directly connected, and where connections go through intermediate relay nodes.

- In Bayesian non-negative matrix factorization (NMF) using rapidly-exploring random trees (RRT), the core idea is to fill the space of possible solutions via a space-filling curve (the RRT), and then the nodes of that tree can be used as centers within a standard inference approach (such as nonparametric variational inference). So far, we have shown that for rank-2 decompositions, which we can visualize, the RRT fills the space for randomly-generated matrices. We have also observed that we seem to find more optima than standard methods (see `https://arxiv.org/abs/1610.08928`). With these initial promising results, we are working on determining whether these methods scale to higher rank decompositions (in r=50 to r=100).

5 Publications

M. A. Masood and F. Doshi-Velez, "Rapid Posterior Exploration in Bayesian Non-negative Matrix Factorization," arXiv preprint, `https://arxiv.org/abs/1610.08928`, 2016.