Article Synthesis
An Optimal Transport View on Generalization [ZLT18]

Nemo Fournier

Contents

Introduction

In class, we have studied the PAC learning framework, whose aim is essentially to study how well a learning algorithm working upon some given hypothesis class can generalize with high probability. In this article, the authors study the expected generalization of a learning algorithm. Their idea is that the analysis similar to the ones conducted in class are based on some "worst case scenario" when it comes to considering the underlying data distribution, and that we could obtain more refined bounds if we take into account some properties of this underlying distribution.

They introduce the notion of algorithmic transport cost of a learning algorithm, which is closely related to the notion of Wasserstein distance between two probability distributions. They derive a bound on the expected generalization of a given learning algorithm as a function of its algorithmic transport cost with respect to the underlying distribution, and then use this result to derive bounds in term of others quantites, each trying to provide an informative measure of some aspect of the underlying distribution. Finally, they show that the framework that they propose can shed a new light on the still puzzling abilities of Deep Neural Networks.

In this synthesis, I will briefly introduce the framework that the authors adopt in section 1, present their main theorem and give an overview of the many results that they derive in section 2. I will then try to give some intuition about their framework and theorem on a toy learning setting (in section 3, which is the *personnal work* that I tried to conduct for this synthesis) and we will see that even on a very simple example, this leads to quite convoluted computations. I will then mention some theoretical contribution that they provide for the analysis of deep neural networks' generalization abilities (section 4).

1 General framework

We are working on the traditionnal statistical learning paradigm. That is we have a *instance space* $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, a *hypothesis space* $\mathcal{W}$, and a *loss function* $\ell : \mathcal{Z} \times \mathcal{W} \to \mathbf{R}^+$. We consider that we have access to a *training sample* $\mathcal{S}_n \in \mathcal{Z}^n$ of size n, drawn as a realization of $\{Z_1, \dots, Z_n\} \sim D^{\otimes n}$, where D is the *underlying data* distribution on $\mathcal{X} \times \mathcal{Y}$, and each of the Z_i is thus draw i.i.d. from D. A *learning algorithm* $\mathcal{A}$ can thus be seen as a (possibly randomized) mapping $\mathcal{A} : \bigcup_{n=1}^{\infty} \mathcal{Z}^n \to \mathcal{W}$ that takes as input a training sample and returns an hypothesis. Such an algorithm can be caracterized by its Markov kernel $P_{W|S_n}$.

Classically we define the *risk* of a given hypothesis $w \in \mathcal{W}$ as

$$R(w) = \mathbb{E}_{z \sim D}[\ell(z, w)]$$

As usual, computing the risk as we have just defined is not really in reach of learning algorithm since the distribution D is unknown, and we rather use the *empirical risk* defined for an hypothesis w and a training as

$$R_{S_n}(w) = \mathbb{E}_{z \sim S_n}[\ell(z, h)] = \frac{1}{n} \sum_{i=1}^{n} \ell(z_i, w)$$

We are interested in the expected generalization error of a learning algorithm $\mathcal{A}$ under distribution D is defined as

$$G\left(D, P_{W|S_n}\right) = \mathbb{E}\Big[R(W) - R_{S_n}(W)\Big]$$

Where the expectation is taken over the joint distribution of both the training sample S_n and the hypothesis W, whose probability density function factorizes as $P_{S_n,W} = P_{S_n} \times P_{W|S_n}$. The goal of this paper is to provide new upper bounds on this quantity.

2 Wasserstein distance and main theorem

One of the achievements of Optimal Transport theory is to have devised a meaningfull notion of distance between probability distributions.

In our particular setting, we will consider that we have a distance $d_{\mathcal{W}}$ on the hypothesis space $\mathcal{W}$. Given two measure μ and ν on $\mathcal{W}$, we define a coupling T between μ and ν as a measure on $\mathcal{W} \times \mathcal{W}$ having marginal μ and ν on its first and second factor (*i.e.* $T(X, \mathcal{W}) = \mu(X)$ and $T(\mathcal{W}, X) = \nu(X)$ for $X \subset \mathcal{W}$). We denote $\Gamma(\mu, \nu)$ the set of all couplings of μ and ν.

Definition. (*Wasserstein distance*) The (1−)Wasserstein distance between two measures μ and ν over $\mathcal{W}$, under the condition that they have a finite expectation, is defined as

$$\mathbb{W}_1(\mu, \nu) = \inf_{T \in \Gamma(\mu,\nu)} \mathbb{E}_{(W,W') \sim T}\left[d_{\mathcal{W}}(W, W')\right]$$

This notion of distance is often referred to as the *Earth Mover Distance*, because it quantifies the cost of moving the mass of μ toward the mass of ν following the best coupling. This quantity is heavily studied and has been shown to provide an insightfull notion of distance between probability measures. The authors then introduce the following notion

Definition. *(Algorithmic Transport Cost)* The algorithmic transport cost of a learning algorithm $\mathcal{A}$ caracterized by $P_{W|S_n}$, under the underlying data distribution D is defined as

$$Opt\left(D, P_{W|S_n}\right) = \mathbb{E}_{z \sim D}\Big[\mathbb{W}_1\left(P_W, P_{W|z}\right)\Big]$$

Intuitively, this quantity encodes how much the learning algorithm will be sensitive to data points. A limit case might be when the learning algorithm does not take into account the learning data, and thus $P_W = P_{W|S_n}$. Such an algorithm should provide a low generalization error, since it has the same performances on both training data and test data.

Note that this cost is indeed dependant on n since P_W is the distribution of the output of the learning algorithm marginalized over all training samples of size n as its input. The authors then proceed to show the core theorem of their paper, which states the following:

Theorem. Assuming that $W \mapsto \ell(z, W)$ is K-Lipschitz continuous for any $z \in \mathcal{Z}$, we have the following upper bound on the expected generalization error of a learning algorithm $\mathcal{A}$ caracterized by $P_{W|S_n}$:

$$G\left(D, P_{W|S_n}\right) = \mathbb{E}\Big[R(W) - R_{S_n}(W)\Big] \leq K \times Opt\left(D, P_{W|S_n}\right)$$

This theorem formalizes the intuition that we discussed above when the output of the algorithm $\mathcal{A}$ is independant from the training sample.

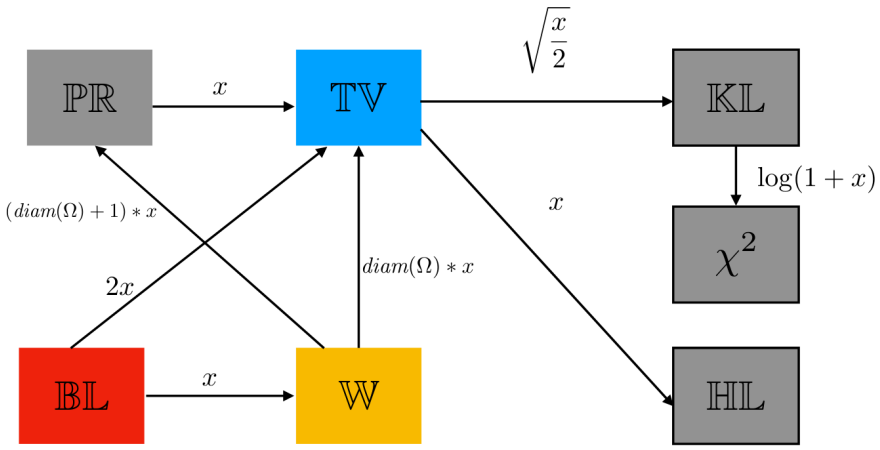

Figure 1: Relationships between probability metrics on the same mesurable space. If an arrows links $\mathbb{A}$ to $\mathbb{B}$ anotated by the function g, it means that $\mathbb{A}(\cdot,\cdot) \leq g(\mathbb{B}(\cdot,\cdot))$. (Image from [ZLT18])

The strength of this bound is that it relates the data distribution and the generalization, and does not involve a *worst-case* analysis over the actual distribution of the data.

They then use this theorem as a starting point to derive a long serie of bounds of the same kind, which can be summed up in figure 1. In particular they manage to link generalization with some information theoretic metrics.

3 A toy learning problem to understand the authors theorem

I will try to evaluate this bound on a toy setting. I will consider the following binary classification problem over the space $\mathcal{X} = [0,1]$ and $\mathcal{Y} = \{0,1\}$. We will fix the following distribution on the data. First, we fix a point $a \in]0,1[$. $Z = (X,Y) \sim D$ is such that X is drawn uniformly over $\mathcal{X}$, and $Y = \mathbf{1}_{\{x \geq a\}}(X)$. The hypothesis class $\mathcal{W}$ that we consider is the set of all the indicator functions of the form $\mathbb{1}_{\{x \geq w\}}$ for $w \in [0,1]$; since we have a clear bijection between $\mathcal{W}$ and $[0,1]$, I will confuse those two sets, and even confuse w and $\mathbb{1}_{\{x \geq w\}}$ for the value of an element of W. We can easily equip $\mathcal{W}$ with the distance $d_{\mathcal{W}}(x,y) = |x-y|$.

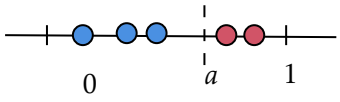

Figure 2: Simple learning setting

This setting is more or less one of those that we have studied in the PAC learning chapter. Let me define the following learning algorithm $\mathcal{A}$

$$\mathcal{A}: \begin{cases} \mathcal{Z}^n & \to & \mathcal{W} \\ S_n = \{(x_1, y_1), \dots, (x_n, y_n)\} & \mapsto & \begin{cases} \max\limits_{\substack{1 \le i \le n \\ \text{s.t. } y_i = 0}} x_i \text{ if } \{i \mid y_i = 0\} \neq \emptyset \\ 0 \text{ otherwise} \end{cases} \end{cases}$$

From the study that we have conducted in class, this yields an empirical risk minimizer for the classification problem and ℓ being the loss defined as

$$\ell((x, y), \mathbb{1}_{x \le w}) = |y - \mathbb{1}_{x \le w}(x)| \times |x - w| \tag{1}$$

I do not consider a regular $0-1$-loss because later on we will need a continuity property on this loss ℓ.

Now we will try to compute the algorithmic transport cost of such a learning algorithm under D. First we need to compute P_W, W being the random output of $\mathcal{A}$ on random input S_n. Start by observing that the support of W is [0,a] by construction of $\mathcal{A}$. Let $S_n^{(0)}$ be the random variable denoting the number of points (x_i, y_i) of S_n classified as 0 (*i.e.* $y_i = 0$ or equivalently $x_i \in [0, a]$). Let $w \in [0, a]$.

$$\mathbf{P}\{W \le w\} = \mathbf{P}\left\{S_n^{(0)} = 0\right\} \mathbf{P}\left\{W \le w \mid S_n^{(0)} = 0\right\} + \sum_{k=1}^{n} \mathbf{P}\left\{S_n^{(0)} = k\right\} \mathbf{P}\left\{W \le w \mid S_n^{(0)} = k\right\}$$

Since for $Z = (X, Y) \sim D$, $X \sim \mathcal{U}(0, 1)$ we obtain $Y = \mathbb{1}_{\{x \ge a\}}(X) \sim \mathcal{B}(1 - a)$, and thus $S_n^{(0)} \sim \mathcal{B}(n, a)$, hence

$$\mathbf{P}\left\{S_n^{(0)} = k\right\} = \binom{n}{k} a^k (1 - a)^{n-k}$$

Moreover $w \mapsto \mathbf{P}\left\{W \le w \mid S_n^{(0)} = k\right\}$ is the cumulative distribution function of the maximum of k random variables drawn uniformly between 0 and a, which equals $\left(\frac{w}{a}\right)^k$.

By definition of $\mathcal{A}$, we also have $\mathbf{P}\left\{W \leq w \mid S_n^{(0)} = 0\right\} = 1$ since if $S_n^{(0)} = 0$, then $W = \mathcal{A}(S_n) = 0$.

Hence, by differentiating with respect to w, we obtain

$$P_W(w) = (1-a)^{n-k} + \sum_{k=1}^{n} \binom{n}{k} a^k (1-a)^{n-k} k \frac{w^{k-1}}{a^k} = (1-a)^{n-k} + n(w+1-a)^n \quad (2)$$

Now that we have the expression of the density of W, we need to investigate $P_{W|z}$. This time, it is fairly easy, since our algorithm $\mathcal{A}$ is actually deterministic once given its input: those densities will be Dirac impulses. We thus have for $z = (x, y)$,

$$P_{W|z} = \delta_x \text{ if } x \leq a \qquad\qquad P_{W|z} = \delta_0 \text{ otherwise}$$

Now, the trick that will allow us to actually compute the algorithmic transport cost in this specific case, is that computing the Wasserstein distance between a general distribution and a Dirac impulse is actually tractable. We indeed have the following result

$$\mathbb{W}_1(\mu, \delta_t) = \mathbb{E}_{X \sim \mu}[d(X, t)]$$

Intuitively: there is only one way to move any distribution toward a single Dirac impulse. That being said, we can thus compute for $0 \leq x \leq a$,

$$\mathbb{W}_1(P_W, \delta_x) = \int_0^a |x - w| P_W(w) \mathrm{d}w \quad (3)$$

Now, using both equations (2) and (3), we can compute the integral and we obtain:

$$\begin{aligned}\mathbb{W}_1(P_W, \delta_x) = \frac{1}{2(an+a)} \Big(& a^2((-a+1)^n + 2)n + a^2(3(-a+1)^n + 2) + 2((-a+1)^n n + (-a+1)^n)x^2 \\ & -4\Big(a^2 - ax - a\Big)(-a+x+1)^n - 2a((-a+1)^n + 1) - 2\left(a(2(-a+1)^n + 1)n\right. \\ & \left. +a(2(-a+1)^n + 1)\right)x) \end{aligned} \quad (4)$$

Although the expression we obtain (using a CAS system) is quite unnice, we can already plot it for some values of n and a, and try to get the intuition of this quantity, as shown in figure 3.

I tried to give an interpretation of those plots, especially about their "valley" shape shifting toward the right as n grows: indeed, recall that what we plot is the distance (in the *Earth Mover Distance* sense) of P_W to δ_x (the Dirac impulse). P_W is essentially the distribution of the maximum of

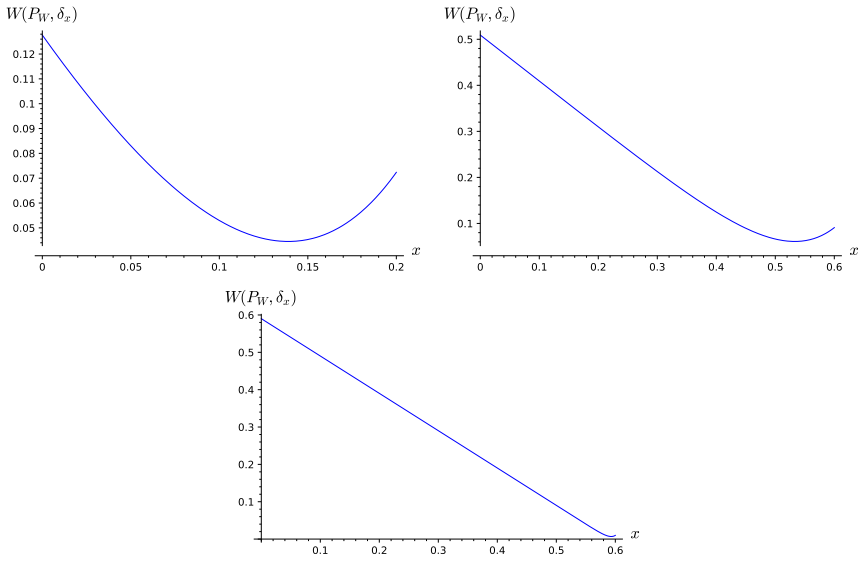

Figure 3: Graphs of the Wasserstein distance between P_W and δ_x for several value of a and n. First row: $n = 10$ with $a = 0.2$ (left) and $a = 0.6$ (right); second row: $n = 100$ and $a = 0.6$

random variables drawn uniformly and independantly. What those plots of the wasserstein distance tell us is that, as n grows, the output of the learning algorithm will be closer and closer to the (optimal) value of a (this is the shift to the right as n grows), but since the set S_n that we randomly draw has some randomness, it is unlikely that its maximum reaches exactly the value of a, and it will rather be a bit behind it: that is why we have this "valley shape".

We can confirm this behaviour by checking analytically using equation (4) that for $x \in [0, a]$,

$$\lim_{n\to\infty} \mathbb{W}_1 (P_W, \delta_x) = a - x$$

Now, let us go back on our initial objective, which was to compute the *algorithmic transport cost* of the learning algorithm $\mathcal{A}$ that we have described. This quantity is defined as:

$$Opt\left(D, P_{W_n}\right) = \mathbb{E}_{z\sim D}\Big[\mathbb{W}_1\left(P_W, P_{W|z}\right)\Big] = \int_0^a \mathbb{W}_1 (P_W, \delta_x)\,\mathrm{d}x + \underbrace{(1-a)\mathbb{W}_1 (P_W, \delta_0)}_{\substack{\text{because every } x \text{ classified} \\ \text{as } y=1 \text{ (equiv. to } x>a) \\ \text{yields } \delta_0 \text{ through } \mathcal{A}}}$$

Once again, we can compute this quantity and obtain

$$Opt\left(D, P_{W_n}\right) = \frac{1}{6(n^2+3n+2)}\Big(2a^2(2(-a+1)^n-3) - \Big(a^2(4(-a+1)^n+3) - 3a((-a+1)^n+2)\Big)n^2 \\ -3\Big(3a^2+3a((-a+1)^n-2) - 2(-a+1)^n+2\Big)n\Big) \\ -6a((-a+1)^n-2) \tag{5}$$

We can easily see that the loss function defined as in equation (1) is 1-Lipschitz continuous in W, *i.e.* for any $w, w' \in \mathcal{W}$ and $z \in \mathcal{X} \times \mathcal{Y}$,

$$\left|\ell(z,w) - \ell(z,w')\right| \leq 1 \times d_{\mathcal{W}}(w,w') = |w - w'|$$

Hence we can apply the theorem that we introduced in section 2 and use the expression (5) to obtain the explicit bound in this very particular setting:

$$G\left(D, P_{W|S_n}\right) \leq 1 \times Opt\left(D, P_{W_n}\right)$$

4 A result on Deep Neural Networks

Thanks to the information theoretic metric based bound that the authors have derived, they manage to derive a bound on the generalization error that a deep neural network neural network can achieve, using the following remark. They take advantage of the peculiar structure of a DNN that is its organization as a succession of layers, as presented in figure 4, and using an adequate data processing inequality layer by layer (because the information flow inside a DNN can be seen as a Markov chain), they manage to link the average generalization error of a DNN, obtaining a bound of the form

$$\mathbb{E}\Big[R(W) - R_{S_n}(W)\Big] \leq \exp\left(-\frac{H}{2}\log\frac{1}{\eta}\right)\sqrt{\frac{K^2R^2I(S_n;W)}{2n}}$$

Where η is the constant corresponding to the contractive property of the mutual information in a data processing setting (from the data processing inequality), K is a Lipschitz constant for the loss function, and R a constant coming from the structure of the inequality space.

This bound provides insights as to why DNN are able to provide such good generalization properties, and do not fall into the *overfitting* trap even though they have a very strong expressiveness.

This latter analysis thus shows the strength of their approach that can thus

- Disregard the *worst case* approach for the data distribution and focus on the very link between this data distribution and the learning process.

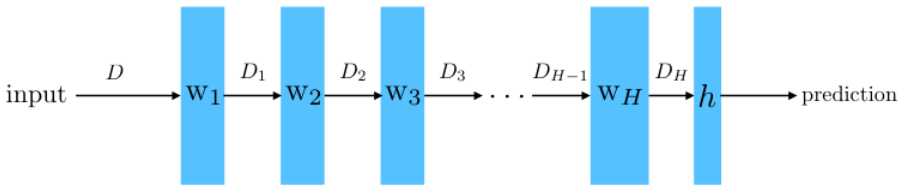

Figure 4: Hierarchical structure of a DNN (image from [ZLT18])

- Take into account some structure of the hypothesis space (here a succession of layers forming a markov chain processing data)

Conclusion

We have seen that the authors propose a novel approach to study the generalization learning algorithms, and that this approach can be successful to analyse the generalization properties of some general models (in this paper, the DNNs). Yet, even though theoretically powerful, this model is not really convenient to perform precise and concrete bound derivation, as we can see in section 3 where even for a very simple learning setting, the relative untractability of optimal transport based metrics yields convoluted computations.

References

[ZLT18] Jingwei Zhang, Tongliang Liu, and Dacheng Tao. An optimal transport view on generalization, 11 2018.