AIML - CS 337

Lecture 9: Trainability Stability and Regularization

4th september, 2022

Lecturer: Abir De *Scribe: Shiv kiran,Teja Varanasi,Ansh,Arya*

Focus of this class in on defining and proving stability for regularized loss minimization algorithms. Some light has also been shed on modelling non-linearity.

1 Definitions

1.1 Trainability

We define trainabilty in order to emphasize on uses of regularization. Being said that, this has not been directly discussed in class. Trainable loss functions requires us to get a bound on expected value of test loss on any test data derived from same distribution as of train data.

Let A be the algorithm, trained on set S(which is and i.i.d from a distribution P), D denotes the test set derived from same distribution.
For all $D, \epsilon > 0$ there exists M such that for all $m > M$, the following condition holds true

$$\underset{s \in P^m}{E}[L_D(A(s))] \leq \min_w L_D(w) + \epsilon$$

1.2 Stability

Stability is a measure of how the algorithm responds on adding a new data point to the training set. Idea is that we should not give excessive priority to a single data point, more importantly in the cases where the algorithm has already been trained on enough points. This property is also closely related to over-fitting(generating a complex model which reduces loss on train-data, but not on test-data) on data. In some cases stability is also viewed in the terms of drift in model if one of the points is tampered at random. We are going to consider the following metric to view stability.

$$\|A(S \cup x) - A(S)\| \;\; where\; S \in D^m, x \in D$$

We aim at showing that this quantity is bounded and the bound becomes tighter on increasing m tending to 0 for large values of m. and we can increasingly tightly bound this value on expectation.

It is not entirely true that more stable algorithms are better than the less stable ones, consider constant output algorithms for example. A useful algorithm should find a hypothesis that on one hand fits the training set and on the other hand does not over-fit (low structural risk)

1.3 Useful properties of loss functions

There are lot of properties that a loss function can satisfy which can probably boost performance of algorithm depending on problem requirements. The following properties are found in most loss functions and are going to be used in the proof that comes up in the next section.

1.3.1 Convexity

Convexity of a function helps us to reach its minimum using gradient descent. Convexity of differentiable functions from $R \mapsto R$ is defined by using double derivatives whereas for functions from R^d, we define using eigen-values of $\frac{\partial^2 f}{\partial w^2}$.

Condition for a function to be convex given $\frac{\partial^2 f}{\partial w^2}$ exits is that its eigen-values must be non-negative.

Que:- Find all eigen values of $XX^T,\ X \in m \times 1$
Solution:-
$Rank(XX^T) \leq Rank(X) \leq 1$
$Rank(XX^T) \leq 1$
$XX^T\ has\ all\ real\ eigen\ values \geq 0$
{$||x||^2$, 0 {with geometric multiplicity m-1}} (observe that sum of eigen-values is $\|x\|^2$)

1.3.2 Lipschitzness

Let $C \in R^d$, A function $f : R^d \to R^k$ is p-Lipschitz over C if for every $w_1, w_2 \in C$ we have

$$\|f(w_1) - f(w_2)\| \leq p||w_1 - w_2||.$$

Note that Lipschitzness does not guarantee differentiability but differentiable function with bounded derivative is Lipschitz.

1.3.3 Smoothness

A differentiable function is β - smooth if its gradient is β lipschitz.

$$\nabla f(w) = (\frac{\partial f(w)}{\partial w_1}, ..., \frac{\partial f(w)}{\partial w_m})$$
$$\|\nabla f(v) - \nabla f(w)\| \leq \beta \|v - w\|$$

Note: It can be proven that if f is β-smooth, following condition holds

$$f(v) \leq f(w) + \langle \nabla f(w), v - w \rangle + \frac{\beta}{2}\|v - w\|^2$$

2 Regularization

2.1 Definition

Most RLM algorithms are in the form given below. We have two terms, one defining fitting of model to given examples and the other representing complexity of the model.

$$\arg\min_{w} \; L_s(w) + R(w)$$

In most scenarios, regularization is used as a tool to reduce over-fitting and provide stability, more on which will be discussed in the section named Regularisation and Stability.

2.2 Uses

There are many uses of introducing regularization but most prominent ones are the following :

- Balance between fitting and stability
- Only some of the functions in the convex smooth domain can be proved to be trainable. But with correct regularization, all smooth convex functions can be shown to be trainable.

Hence, RLM can be used as a general learning rule for convex smooth learning problems.
You may have already intuitively noticed this in previous classes where $XX^T + \lambda$I ($\lambda > 0$) is shown to always have an inverse .

Tikhonov Regularization : $R(w) := \lambda \|w\|^2$

2.3 Regularization and Stability

Stability, as defined earlier is measured by the quantity

$$\|A(S \cup x) - A(S)\|$$

where x is new data point added to S.
proving the stability requires us to prove that $\|_{S \in D^m} A(S \cup x) - A(S)\|$ is bounded and the bound becomes tighter as m increases, converging to 0. We prove this result considering bounds of

$$E = L(A, S(A \cup x)) - L(A, S(A))$$

Note that this quantity is greater than 0 for a good algorithm and loss function i.e $E > 0$.

2.3.1 Proving upper bound

Using the Smoothness/ lipschitz

$$L(A, S(A \cup x)) - L(A, S(A)) < B(m)||S(A \cup m) - S(A)||$$

This sure is a bound but not a good one as the factor m destroys the purpose of the bound, which will be clear by the end of this proof .

$$L(A, S(A \cup x)) - L(A, S(A)) = L(A \cup x, S(A \cup x)) - L(x, S(A \cup x)) - (L(A \cup x, S(A)) - L(x, S(A))) \tag{1}$$

$$let\ G(M) = L(A \cup x, S(M)) - L(x, S(M)) \tag{2}$$

Observe that eq 1 can be rewritten as

$$G(A \cup x) - G(A) \tag{3}$$

using smoothness of eq 3 we can write

$$G(S(A \cup x)) - G(S(A)) \leq B||S(A \cup x) - S(x)|| \tag{4}$$

2.3.2 Proving lower bound

This proof uses a non-trivial result from taylor series which will be stated as lemma.

Lemma-1:-

$$f(h_1) \geq f(h_2) + < \nabla f(h_1, h_1 - h_2) > + \frac{1}{2}(h_1 - h_2)^T \nabla^2 f(h^{'}, h_1 - h_2^{'})$$

for some $h^{'}$.

Apply lemma-1 on eq 1, we get

$$\begin{aligned} L(A, S(A \cup x)) - L(A, S(A)) \geq & < \nabla L(A, S(A)), S(A \cup x) - S(A) > \\ & + \frac{1}{2}(S(A \cup x) - S(A))(\nabla^2 l(A, h^{'}) * m)(S(A \cup x) - S(A)) \end{aligned} \tag{5}$$

Observe that $\nabla L(A, S(A)) = 0$, by definition

$$L(A, S(A \cup x)) - L(A, S(A)) \geq \frac{m}{2}(S(A \cup x) - S(A))^T \nabla^2 l(A, h^{'})(S(A \cup x) - S(A)) \tag{6}$$

considering the case of Tiknow regularisation $eig(\nabla^2 L(A, h)) \geq \lambda$ giving

$$L(A, S(A \cup x)) - L(A, S(A)) \geq \frac{m}{2}\lambda||S(A \cup x) - S(A)||^2 \tag{7}$$

using Eq 7 and 4

$$B||S(A \cup x) - S(A)|| \geq \frac{\lambda}{2}\boldsymbol{m}||S(A \cup x) - S(A)||^2 \tag{8}$$

$$\frac{2B}{\lambda m} \geq ||S(A \cup x) - S(A)|| \tag{9}$$

Question:-
On similar lines, derive a bound for following case $||S(A^{'}) - S(A)||$ where $A^{'}$ is set with one case (z_i) tampered to ($z_i^{'}$).

HINT

$$||\overline{S}(A^{'} \cup Z_i) + S(A^{'}) + S(A \cup Z_i^{'}) - S(A)|| = ||S(A^{'}) - S(A)||$$

3 Non-Linearity

3.1 Kernels-Mappings

Kernel-mappings represent transformation of one space to another, which is primarily done to help in solving the given problem, i.e d - dimensional ($u_1, ...u_d$) to ($\delta_1..\delta_m$)

Before directly using polynomial regression, different kernels were used for different scenarios. For general cases, following idea is used :

$$x \mapsto \mathrm{e}^{-\|x\|_1}, e^{-\|x\|_2}, ..., e^{-\|x\|_n}$$

3.1.1 Example

Consider the following kernel,

$$x,y \mapsto \mathrm{x}^2,y^2$$

This kernel now has power to represent ellipses in (x,y) space just by using linear operator on the new feature vector.

3.2 Polynomial Regression

Most continuous curves can be modeled by using polynomials, taylor series also allows us to estimate degree of such polynomials required to achieve the desired accuracy for a given range of values. This fact is used in designing a polynomial kernel whose factors can be found out using regression on the new space

$$x \mapsto 1 \quad \mathrm{x} \quad \mathrm{x}^2 ... \ x^n$$