Machine Learning on Curved Space

Gim Seng Ng

February 2020

1 Introduction

1.1 Warm-up: Linear regression on curved space

1.1.1 Flat space

Consider a set data in a 1d regression problem:

$$D = \{x^i : y^i\} \quad , \quad i = 1, \dots, N. \tag{1}$$

Here we pair them up as a object similar to a Python dict, and the number of data is given by N.

The linear regression model seeks to fit a linear function

$$f_\theta(x) = \theta^0 + \theta^1 x \equiv \theta^\mu x_\mu \quad , \quad \mu = 0, \dots, N_f \tag{2}$$

where N_f labels the number of features, which in this simple case is just unity. It is conventional to use the redundant label x^0 and set it to unity, i.e. $x^0 = 1$. The goal is to seek in a space of parameters (i.e. θ_μ), such that they minimize a cost/loss function. In "flat space", we will use some version of the L-norm. For our purpose, we will stick to the L_2-norm, i.e. Euclidean distance in flat space:

$$V(D;\theta) \equiv \frac{1}{N} \sum_{i=1}^{N} \left[f_\theta(x^i) - y^i\right]^2 . \tag{3}$$

The convention of $1/N$ in front simplifies the derivative in the next step. The goal is to find optimum set of θ to minimize V:

$$\Theta \equiv \{\theta | V(D;\theta) \text{ is minimized over } D\} . \tag{4}$$

Could we model steepest descent as a discrete version of a dynamical model seeking to find minimum? A easy guest is to promote J to be a potential for Θ with the D providing the locations of "defects" or something. The (continous) steepest descent equation is then

$$\frac{d\theta^\mu}{dt} = -\eta \partial_\mu V = -\eta \times \frac{2}{N} \sum_{i=1}^{N} \left(f_\theta(x^i) - y^i\right) x^i_\mu \tag{5}$$

where η is the learning rate. This can be uplifted to a dynamical Lagrangian for θ:

$$L(D) = -\frac{1}{2}\left(\frac{d\theta}{dt}\right)^2 + \eta V(D;\theta). \tag{6}$$

Now that the interpretation is that: Given the set D (which we should think of as boundary conditions for defects or something), that fixes a particular potentials for θ. The steepest descendat implements a discrete version of the equation of motion.

A few questions and generations: Should we promote general/special relativistic invariant to mix t and x's? Do we gain anything? Or it it sufficient to promote the x-space to be Euclidean curved space. Lastly, should the model f_θ be a geodesic in θ-space, or is that somewhat arbitrary on curved space ? It is probably most interesting if f_θ is geodesic, and the goal is to find parameters of the geodesic such that it approximates the data (which we assume to look like geodesics on curved space).

Also, stochastic descent, is this some quantum / path integral way to think about this? If so, QM on curved space would be fun.

Lastly, in the case where $N_f > 1$, then we potentially have somethinglike $O(N_f)$ symmetry and the like, could we make use of that? Taking large N and/or N_f limits would be fun to think about.

1.1.2 Towards curved space

To generalize, suppose we are still in the regime of linear/free-field model, but however on curved space. This is akin to free field theory on curved space. One straight forward generalization is to simply change the cost function to be

$$V(g;D;\theta) \equiv \frac{1}{N}\sum_{i,j=1}^{N} g_{ij}(x^i)\left[f_\theta(x^i) - y^i\right]\left[f_\theta(x^j) - y^j\right] \tag{7}$$

where g_{ij} is the curved space metric.

What is the use of this generalization? This could potentially be equivalent to polynomial regression in the case of polynomial model, however, suppose the data seems to be something like $y = 1/x$, then doing a coordinate transformation to map $x \to 1/x$ seems to be a promising thing to do. However, the right measure to use is then the flat measure on $1/x$, which will translates into non-flat measure in the original y vs x problem.

So far, it seems to me that if I knew how to do the transformation, then I could gain something. However, is there a way to "stochastically" learn the appropriate metric or coordinate transformation? Otherwise, I won't have an algorithm way to do so. Maybe I could combine this with a bit of deep-learning to figure out the coordinate transformation in some simple model

1.1.3 An example: modelling $y = x^2$

Show the classic example that linear regression does badly on x^2. However, by transforming the metric such that $y = x^2 = \tilde{x}$ in $\tilde{x}$ coordinates, we get a flat metric there which makes steepest descent there nice. We could reformula that ”nice” problem into the x-space but we need a curved metric for the loss function.

$$ds^2 = dx^2 = d\tilde{x}^2 \tag{8}$$

1.1.4 An example: modelling $y = 1/x$

1.1.5 2d example: two-sphere vs R^2