Short summaries of what I have read so far.

Jaemin Oh

December 15, 2022

Contents

1 Causal Inference

1.1 Holland, 1986 [3]

The very basic of Rubin's model was explained, and the distinction between associational inference and the causal inference was provided.

Let U be the population and $u \in U$ be a unit. Three variables Y, W, and C are given, where Y is the variable to be analyzed (response variable), W is a (potential) cause of Y, and C is an attribute. They are functions from U to $\mathbb{R}$, and their distributions are given by the relative frequency on U. As a function, W and C are the same. However, they are different in the sense that we cannot do a randomized experiment with C and can do with W. For example, a clinical surgery can be W and the gender can be C. This property distinguishes the cause from the attribute.

In this setting, the associational inference is focused on $E(Y|W)$ or $E(Y|C)$. In other words, the discovery of the way that Y is related to W or C will be satisfactory. On the other hand, in causal inference, a direct comparison between treatment and control for each unit is required. This cannot be done in practice, since any unit cannot receive both treatments simultaneously. But we can do counterfactual imaginations that lead additional functions $\{Y_w\}_{w \in I}$ which is called potential outcomes. To overcome the practical issue and estimate the causal effect, the researcher should design the study to approximate randomized experiment, which is the simplest setting. Note that, in a randomized experiment, $Y_w =_d Y|W = w$ by consistency and ignorability (missing at random).

2 Spatial Data Analysis

2.1 Hierarchichal Modeling and Analysis for Spatial Data [2]

1. Overview of spatial data problems

   Spatial data has three possible different forms: point referenced data, areal data, and point pattern data. Let $D \subset \mathbb{R}^d$ be a set of locations. If the data can be described as $Y(s_i)$ where $s_i \in D$ and s_i is deterministic, then it belongs to the class of point referenced data. Instead of the exact location, imagine that the information in $B_i \in 2^D$ is given. We call this case as an areal data. When D is a random set, then it is a point pattern data.

   For point referenced data, it is natural to think that $Cov\left(Y(s_i), Y(s_j)\right)$ is a function of a distance between s_i and s_j. The most convenient approach is assuming

   $$\begin{aligned}(Y(s_i), \ldots, Y(s_m)) &\sim N_m\left(\mu, \Sigma\right)\\ (\Sigma)_{ij} &= \sigma^2 e^{-\phi d_{ij}^{\kappa}} + \tau^2 I(i=j)\end{aligned}$$

   where τ^2 is called a *nugget effect*.

3 Machine Learning

3.1 Ainsworth, 2021 [1]

Plateau phonemenon may be occured during GD-based l^2-loss optimization. Due to this, we don't know when to stop the iteration.

In this paper, the authors concentrated on the "activation pattern" of neurons and verified that the plateau corresponds to the period when the activation pattern remains constant. Specifically, the V_t component of the loss decays to the staionary point exponentially during the plateau interval. They proposed the method "active neuron least squares" (ANSL) that finds the best activation pattern among candidates generated by non-arbitrary manipulation and fits local least square linear lines to the data. This optimization procedure is done without any gradient information. It significantly reduces computation time compared to existing methods, such as Adam or GD.

3.2 Lu, 2021 [4]

Universal approximation theorem is a theoretical justification of NN's performance in function approximation. There is a similar theorem for not just a function, but an operator between two Banach spaces. With this theorem, the authors proposed Deep Operator Network (DeepONet) to approximate an operator.

DeepONet consists of two subnetworks, branch net and trunk net. Branch net eats discretized function with specified sensor $(x_1, \ldots, x_m)$, and trunk net eats y the point at which the output function will be evaluated. Specifically,

$$G^{NN}[u](y) = \sum_{i=1}^{p} b_i t_i$$

where $(b_1, \ldots, b_p) = B(u(x_1), \ldots, u(x_m))$, and $(t_i, \ldots, t_p) = T(y)$.

It performs quite well when there is large, high-fidelity dataset. One great advantage of this is, its prediction speed is very fast compared to exisiting simulation methods. Once the network is trained, we can exploit the speed and get a lot of simulation results within a small time interval. However, obtaining high-fidelity dataset is very expensive. Instead of using high-fidelity data only, we can use governing PDEs and low-fidelity data to lower the cost.

3.3 Zhu, 2022 [5]

NN's ability of interpolation is well known. However, in real world application, one cannot avoid the situation of extrapolation. The extrapolation problem of NN is easily identified, by training it to approximate $\sin(\pi x)$ on $x \in [0, 1]$.

In this paper, the extrapolation of Deep Operator Network is systemically studied. As inputs of branch net, discretized functions from Gaussian Random Field with Gaussian kernel are usually used. Here the extrapolation means, training DeepONet with GRF class of l_{train}, and predicting the output with GRF class of $l_{test} \neq l_{train}$. As W_2 distance of test function space and training function space increases, the test error increases polynomially. To obtain consistent result with extrapolation, the authors suggests to fine-tune the pre-trained DeepONet in two-ways. First, when governing PDEs are known, fine-tune the network with PINN loss

$$\frac{\lambda_1}{R} \sum_{R} \|\mathcal{F}[G^{NN}[v](\xi); v]\|^2 + \frac{\lambda_2}{B} \sum_{B} \|\mathcal{B}[G^{NN}[v](\xi); v]\|^2,$$

where the first term is residual part, and the second term is boundary and initial condition. Second, when sparse observations are given, fine-tune the network with those observations. However, in the second case, over-fitting and catastrophic forgetting may be happen. By fine-tuning together with training data points, forgetting can be avoided. Here the loss is

$$\frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \|G^{NN}[v](\xi_i) - G[v](\xi_i)\|^2 + \frac{\lambda}{N_{obs}} \sum_{j=1}^{N_{obs}} \|G^{NN}[v](\xi_j) - G[v](\xi_j)\|^2,$$

where i denotes index for training data, and j denotes index for sparse observation.

References

[1] Mark Ainsworth and Yeonjong Shin. Plateau phenomenon in gradient descent training of relu networks: Explanation, quantification, and avoidance. *SIAM Journal on Scientific Computing*, 43(5):A3438–A3468, 2021.

[2] Sudipto Banerjee, Bradley P. Carlin, and Alan E Gelfand. *Hierarchical Modeling and Analysis for Spatial Data, Second Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press, 2ed. edition, 2015.

[3] Paul W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

[4] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

[5] Min Zhu, Handi Zhang, Anran Jiao, George Em Karniadakis, and Lu Lu. Reliable extrapolation of deep neural operators informed by physics or sparse observations, 2022.