If machine learning is to be of any use, can it…

Sai Ravela

Learn Physics?

- Let's re-examine the Poisson Equation problem

$$\nabla^2 \underline{\phi} = \underline{f}$$

On a periodic domain
In 1D

$$\underline{\phi} = \boldsymbol{D}^{-1}\underline{f}$$

$$\underline{\phi} = \boldsymbol{L}\underline{f}$$

Machine Learning problem:
Estimate $\boldsymbol{D}, \boldsymbol{L}$
Given: $\{\underline{\phi}_n, \underline{f}_n\}, the\ training\ data$

Sound Easy?

Step 1

How many samples?

Why L_2?

- $J(\boldsymbol{L}^{\#}) := \frac{1}{2N}\sum_{i=1}^{N}\left\|\underline{\phi} - \boldsymbol{L}^{\#}\underline{f}\right\|_2^2$

Matrix form of operator $\boldsymbol{L}$

$\boldsymbol{L}^{\#}$ degrees of freedom: n^2

Step 2, OLS

- $J(\boldsymbol{L}^o) := \frac{1}{2N}\sum_{i=1}^{N}\left\|\underline{\phi}_i - \boldsymbol{L}^o \underline{f}_i\right\|_2^2$

Oracle:
Let's invoke *invariance* as a principle
$\boldsymbol{L}^{\#} \Rightarrow \boldsymbol{L}^o$

- $\boldsymbol{L}^o$ has *n* degrees of freedom;
  block circulant matrix for our problem

- $\boldsymbol{L}^o = \begin{pmatrix} l_1 & \cdots & l_n \\ \vdots & \ddots & \vdots \\ l_2 & \cdots & l_1 \end{pmatrix}$ -- *special Toeplitz matrix*

Equivalently

- $J(\underline{l}) \stackrel{\text{def}}{=} \frac{1}{N}\sum_{i=1}^{N} J_i(\underline{l})$;

$$\Rightarrow \hat{\underline{l}} = \left(\sum_{i=1}^{N} \boldsymbol{F}_i^T \boldsymbol{F}_i\right)^{-1} \left(\sum_{j=1}^{N} \boldsymbol{F}_j^T \underline{\phi}_j\right)$$

- Where, $\underline{l} = \begin{bmatrix} l_1 \\ \vdots \\ l_n \end{bmatrix}$ $and\ \boldsymbol{F}_i = \begin{pmatrix} f_{i,1} & \cdots & f_{i,n} \\ \vdots & \ddots & \vdots \\ f_{i,2} & \cdots & f_{i,1} \end{pmatrix}$

- $J_i(\underline{l}) \stackrel{\text{def}}{=} \frac{1}{2}\left\|\underline{\phi}_i - \boldsymbol{F}_i \underline{l}\right\|_2^2$

- Local solution is: $\hat{\underline{l}}_i = (\boldsymbol{F}_i^T \boldsymbol{F}_i)^{-1} \boldsymbol{F}_i^T \underline{\phi}_i$

- Can estimate $\hat{\underline{l}} = \overline{(\boldsymbol{F}_i^T \boldsymbol{F}_i)^{-1} \boldsymbol{F}_i^T \underline{\phi}_i}$ work?

What is the approximation?
What kind of training data?
How many samples N/n ?

Regularization

- $J_i(\underline{l}) \stackrel{\text{def}}{=} \frac{1}{2}\left\|\underline{\phi}_i - \boldsymbol{F}_i\underline{l}\right\|_2^2 + \frac{\lambda}{2}\left\|\underline{l}\right\|_2^2$
- $\hat{\underline{l}}_i = (\boldsymbol{F}_i^T\boldsymbol{F}_i + \lambda\boldsymbol{I})^{-1}\boldsymbol{F}_i^T\underline{\phi}_i$
- $\boldsymbol{F}_i = \boldsymbol{U}_i\boldsymbol{S}_i\boldsymbol{U}_i^T$
- $\boldsymbol{U}_{\#i} \stackrel{\text{def}}{=} \boldsymbol{U}_i(:, 1{:}S < N \ll n)$
  - *Leading S directions*
- $\hat{\underline{l}}_i = \boldsymbol{U}_{\#i}[\boldsymbol{S}_{\#i}^2 + \lambda\boldsymbol{I}]^{-1}\boldsymbol{S}_{\#i}\boldsymbol{U}_{\#i}^T\underline{\phi}_i$

Why L_2 ?

What does λ *do*?

Why L_2 ? aka Ridge

Why this form? What are they?

How many "leading" dimensions?

Try Matlab Code

Recursion

- $J(\underline{l}) \stackrel{\text{def}}{=} \frac{1}{2}\sum_{i=1}^{N} \left\| \underline{\phi}_i - \boldsymbol{F}_i \underline{l} \right\|_2^2$
- $\boldsymbol{Q}_1 = \boldsymbol{F}_1^T \boldsymbol{F}_1$ and $\hat{\underline{l}}_1 = [\boldsymbol{F}_1^T \boldsymbol{F}_1]^{-1} \boldsymbol{F}_1^T \left[\underline{\phi}_1\right]$
- $\hat{\underline{l}}_{i+1} = \hat{\underline{l}}_i + \left[\boldsymbol{Q}_i + \boldsymbol{F}_{i+1}^T \boldsymbol{F}_{i+1}\right]^{-1} \boldsymbol{F}_{i+1}^T \left[\underline{\phi}_{i+1} - \boldsymbol{F}_{i+1} \hat{\underline{l}}_i\right]$
- $\boldsymbol{Q}_{i+1} = \boldsymbol{Q}_i + \boldsymbol{F}_{i+1}^T \boldsymbol{F}_{i+1}$ (cumulant;)

- Assignment 1: Derive this result.
- Compare with batch solution
- Compare with naïve average
- Optional: Compare with Kalman Filter

Estimation Beyond OLS

- Maximum Likelihood Estimation (MLE)
- Let $J_i(\underline{l}) \stackrel{\text{def}}{=} \frac{1}{2}\left\|\underline{\phi}_i - \boldsymbol{F}_i\underline{l}\right\|_{\boldsymbol{R}}^2 = \frac{1}{2}\left[\underline{\phi}_i - \boldsymbol{F}_i\underline{l}\right]^T \boldsymbol{R}^{-1}\left[\underline{\phi}_i - \boldsymbol{F}_i\underline{l}\right]$
- then $\hat{\underline{l}}_i = (\boldsymbol{F}_i^T\boldsymbol{R}^{-1}\boldsymbol{F}_i)^{-1}\boldsymbol{F}_i^T\boldsymbol{R}^{-1}\underline{\phi}_i$
- Maximum A Posteriori Estimation (MAP):
- Let $J_i(\underline{l}) \stackrel{\text{def}}{=} \frac{1}{2}\left\|\underline{\phi}_i - \boldsymbol{F}_i\underline{l}\right\|_R^2 + \frac{1}{2}\left\|\underline{l} - \bar{\underline{l}}\right\|_C^2$
- $\hat{\underline{l}}_i = (\boldsymbol{F}_i^T\boldsymbol{R}^{-1}\boldsymbol{F}_i + \boldsymbol{C}^{-1})^{-1}[\boldsymbol{F}_i^T\boldsymbol{R}^{-1}\underline{\phi}_i + \boldsymbol{C}^{-1}\bar{\underline{l}}]$
- Can you combine this with regularization? Truncation?
- Why are these called MAP and MLE? – Next class from a Bayesian perspective.

Notes

- The best result did not come by
  - Increasing sample size
  - Using L2 regularization
- It immediately emerged when we chose a “subspace”
- The Kernel (∇^2) is sparse, so L^o is non-sparse. U is the Fourier Transform (in the continuous case), and therefore truncating $\boldsymbol{U}_{\#}$ projects the residual onto very low frequencies, promoting sparse D.
- What we assumed was
  - Linearity, Invariance, small magnitude perturbations.