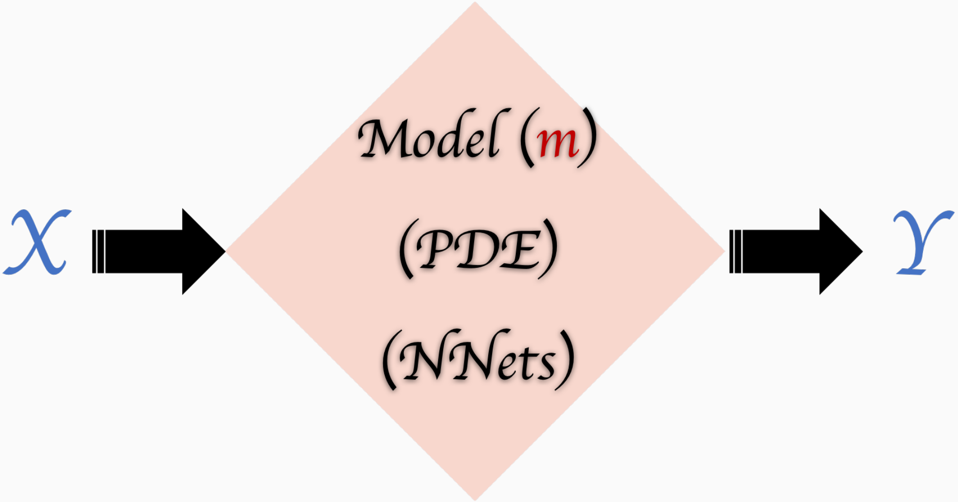
A Generalized Weighted Optimization Method for Computational Learning and Inversion

Björn Engquist[1], Kui Ren[2], **Yunan Yang**[3]

[1]Department of Mathematics and Oden Institute, The University of Texas at Austin, USA
[2]Department of Applied Physics and Mathematics, Columbia University, USA
[3]Institute for Theoretical Studies, ETH Zürich, Switzerland

General Inverse Problems



Aim at finding m such that the predicted outputs $(X, F(m))$ match given measured data (X, Y).

Inverse Problem $\to$ Inverse Data Matching Problem

Most of such inverse problems are solved as data-matching problems:

$$m^* = \underset{m\in(\mathcal{M},d_m)}{\operatorname{argmin}} J(m) = \underset{m\in(\mathcal{M},d_m)}{\operatorname{argmin}} d_g^2(f,g),\ f = F(m),\ f,g \in (\mathcal{D},d_g).$$

(The map $f = F(m)$ is often *implicitly* given through a system of constraints.)

Two Metric Spaces

- $(\mathcal{M}, d_m)$ is the metric space for the **parameters**.
- $(\mathcal{D}, d_g)$ is the metric space for the **data**.

Inverse Problem → Inverse Data Matching Problem

Most of such inverse problems are solved as data-matching problems:

$$m^* = \underset{m\in(\mathcal{M},d_m)}{\operatorname{argmin}} J(m) = \underset{m\in(\mathcal{M},d_m)}{\operatorname{argmin}} d_g^2(f,g),\ f = F(m),\ f,g \in (\mathcal{D},d_g).$$

(The map $f = F(m)$ is often *implicitly* given through a system of constraints.)

Two Metric Spaces

- $(\mathcal{M}, d_m)$ is the metric space for the **parameters**.
- $(\mathcal{D}, d_g)$ is the metric space for the **data**.

How to choose these two spaces may affect the learning process greatly!

In this work, we propose to use **weighted** metrics for both spaces.

Zoom into Feature Regression

Given training data $\{x_j, y_j\}_{j=1}^{N}$, where $x_j \in \mathbb{R}$, $y_j \in \mathbb{C}$, we are interested in learning a random Fourier feature (RFF) model

$$f_{\boldsymbol{\theta}}(x) = \sum_{k=0}^{P-1} \theta_k e^{\mathrm{i}kx}, \quad x \in [0, 2\pi],$$

where we aim to find $\boldsymbol{\theta} := (\theta_0, \cdots, \theta_{P-1})^\top$.

Zoom into Feature Regression

Given training data $\{x_j, y_j\}_{j=1}^N$, where $x_j \in \mathbb{R}$, $y_j \in \mathbb{C}$, we are interested in learning a random Fourier feature (RFF) model

$$f_{\boldsymbol{\theta}}(x) = \sum_{k=0}^{P-1} \theta_k e^{ikx}, \quad x \in [0, 2\pi],$$

where we aim to find $\boldsymbol{\theta} := (\theta_0, \cdots, \theta_{P-1})^\top$.

Let $\Psi \in \mathbb{C}^{N \times P}$ be the feature matrix

$$(\Psi)_{jk} = e^{ikx_j}, \quad 0 \le j \le N-1, \;\; 0 \le k \le P-1\,.$$

The learning problem is recast as an optimization problem

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\Psi\boldsymbol{\theta} - \mathbf{y}\|_2^2, \quad \mathbf{y} = (y_0, \cdots, y_{N-1})^\top.$$

Previous Work on Weighted Optimization [Xie et al., 2020]

Prior assumption about the coefficient vector θ:

$$\mathbb{E}_{\boldsymbol{\theta}}[\theta] = \mathbf{0}, \;\; \mathbb{E}_{\boldsymbol{\theta}}[\theta\theta^*] = c_\gamma \boldsymbol{\Lambda}_{[P]}^{-2\gamma}, \quad \gamma > 0$$

where $\boldsymbol{\Lambda}_{[P]} = \mathrm{diag}\{t_0, \ldots, t_k, \ldots, t_{P-1}\}, \quad , t_k := 1 + k.$

The normalization constant c_γ enforces $\mathbb{E}_{\boldsymbol{\theta}}[\|\theta\|^2] = 1.$

This is assuming that the forward model is smoothing!

Previous Work on Weighted Optimization [Xie et al., 2020]

Prior assumption about the coefficient vector θ:

$$\mathbb{E}_{\boldsymbol{\theta}}[\theta] = \mathbf{0}, \;\; \mathbb{E}_{\boldsymbol{\theta}}[\theta\theta^*] = c_\gamma \boldsymbol{\Lambda}_{[P]}^{-2\gamma}, \quad \gamma > 0$$

where $\boldsymbol{\Lambda}_{[P]} = \mathrm{diag}\{t_0,\ \dots,\ t_k,\ \dots,\ t_{P-1}\}, \quad , t_k := 1 + k.$

The normalization constant c_γ enforces $\mathbb{E}_{\boldsymbol{\theta}}[\|\theta\|^2] = 1$.

This is assuming that the forward model is smoothing!

[Xie et al., 2020] proposed to learn a model with $p \leq P$ features through the *weighted* least-squares formulation

$$\widehat{\theta}_p = \boldsymbol{\Lambda}_{[p]}^{-\beta}\widehat{\mathbf{w}}, \;\; \text{with} \;\; \widehat{\mathbf{w}} = \mathrm{argmin}_{\boldsymbol{\theta}} \|\Psi_{[N\times p]}\boldsymbol{\Lambda}_{[p]}^{-\beta}\mathbf{w} - \mathbf{y}\|_2^2,$$

when the learning problem is overparameterized, i.e., $p > N$.

Main Proposal of Our Work

A **generalized weighted** least-squares formulation for feature regression:

$$\widehat{\boldsymbol{\theta}}_p^\delta = \boldsymbol{\Lambda}_{[p]}^{-\beta}\widehat{\mathbf{w}}, \quad \text{with} \quad \widehat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}} \, \|\boldsymbol{\Lambda}_{[N]}^{-\alpha}\left(\Psi_{[N\times p]}\boldsymbol{\Lambda}_{[p]}^{-\beta}\mathbf{w} - \mathbf{y}^\delta\right)\|_2^2,$$

where δ denotes that the training data is polluted with noise.

[Engquist-Ren-Y.,2022]

Main Proposal of Our Work

A **generalized weighted** least-squares formulation for feature regression:

$$\widehat{\theta}_p^{\delta} = \mathbf{\Lambda}_{[p]}^{-\beta}\widehat{\mathbf{w}}, \quad \text{with} \quad \widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \left\|\mathbf{\Lambda}_{[N]}^{-\alpha}\left(\Psi_{[N\times p]}\mathbf{\Lambda}_{[p]}^{-\beta}\mathbf{w} - \mathbf{y}^{\delta}\right)\right\|_2^2,$$

where δ denotes that the training data is polluted with noise.

A short informal summary:

$$\widehat{\theta} = \operatorname*{argmin}_{\theta} \|\Psi\theta - \mathbf{y}\|_2^2, \quad \text{(unweighted)} \tag{1}$$

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\theta} \|\Psi\mathbf{\Lambda}_{[p]}^{-\beta}\mathbf{w} - \mathbf{y}\|_2^2, \quad \text{([Xie et al., 2020])} \tag{2}$$

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \left\|\mathbf{\Lambda}_{[N]}^{-\alpha}\left(\Psi\mathbf{\Lambda}_{[p]}^{-\beta}\mathbf{w} - \mathbf{y}\right)\right\|_2^2, \quad \text{(our proposal)} \tag{3}$$

$\widehat{\theta}_p = \mathbf{\Lambda}_{[p]}^{-\beta}\widehat{\mathbf{w}}$ for the last two.

[Engquist-Ren-Y.,2022]

The Benefits of Weighting

$$\widehat{\boldsymbol{\theta}}_p^\delta = \boldsymbol{\Lambda}_{[p]}^{-\beta}\widehat{\mathbf{w}}, \quad \text{with} \quad \widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \|\boldsymbol{\Lambda}_{[N]}^{-\alpha}\left(\Psi_{[N\times p]}\boldsymbol{\Lambda}_{[p]}^{-\beta}\mathbf{w} - \mathbf{y}^\delta\right)\|_2^2$$

Define the generalization error (thanks to RFF):

$$\mathcal{E}_{\alpha,\beta}^{\delta}(P,p,N) := \mathbb{E}_{\boldsymbol{\theta}}\left[\left\|f_{\boldsymbol{\theta}}(x) - f_{\widehat{\boldsymbol{\theta}}_p^\delta}(x)\right\|_{L^2([0,2\pi])}^2\right] = \mathbb{E}_{\boldsymbol{\theta}}\left[\left\|\widehat{\boldsymbol{\theta}}_p^\delta - \boldsymbol{\theta}\right\|_2^2\right].$$

[Engquist-Ren-Y.,2022]

The Benefits of Weighting

$$\widehat{\boldsymbol{\theta}}_p^\delta = \boldsymbol{\Lambda}_{[p]}^{-\beta}\widehat{\mathbf{w}}, \quad \text{with} \quad \widehat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \|\boldsymbol{\Lambda}_{[N]}^{-\alpha}\left(\Psi_{[N\times p]}\boldsymbol{\Lambda}_{[p]}^{-\beta}\mathbf{w} - \mathbf{y}^\delta\right)\|_2^2$$

Define the generalization error (thanks to RFF):

$$\mathcal{E}_{\alpha,\beta}^\delta(P,p,N) := \mathbb{E}_{\boldsymbol{\theta}}\left[\left\|f_{\boldsymbol{\theta}}(x) - f_{\widehat{\boldsymbol{\theta}}_p^\delta}(x)\right\|_{L^2([0,2\pi])}^2\right] = \mathbb{E}_{\boldsymbol{\theta}}\left[\left\|\widehat{\boldsymbol{\theta}}_p^\delta - \boldsymbol{\theta}\right\|_2^2\right].$$

In this paper, we analyze the **improved** generalization error in three cases:

1. Training with noise-free data in RFF model (Section 3)
2. Training with noisy data in RFF model (Section 4)
3. Beyond Random Fourier Feature (RFF) model (Section 5)

[Engquist-Ren-Y.,2022]

Summary

We solve an inverse problem through the optimization format.

$$m^* = \underset{m\in(\mathcal{M},d_m)}{\operatorname{argmin}} J(m) = \underset{m\in(\mathcal{M},d_m)}{\operatorname{argmin}} d_g^2(f,g),\ f = F(m),\ f,g \in (\mathcal{D}, d_g).$$

Two Metric Spaces

- $(\mathcal{M}, d_m)$ is the metric space for the **parameters**.
- $(\mathcal{D}, d_g)$ is the metric space for the **data**.

Two Weight Matrices

$$\widehat{\theta}_p^\delta = \mathbf{\Lambda}_{[p]}^{-\beta}\widehat{\mathbf{w}}, \quad \text{with} \quad \widehat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \| \mathbf{\Lambda}_{[N]}^{-\alpha} \left(\Psi_{[N\times p]} \mathbf{\Lambda}_{[p]}^{-\beta} \mathbf{w} - \mathbf{y}^\delta \right) |_2^2.$$

The Impact of the Two Metric Spaces/Weighting Matrices

1. Improve Generalization Error/Resolution
2. Robustness to Noise (Mitigate Overfitting)
3. Improve Optimization Landscape
4. Change Convergence Speed/Trajectory