PhD Defense

Aleksei Sholokhov

Tuesday 30th May, 2023

Plan of the Defense

Show topics and published papers. Mention covid

$\mathcal{MSR}3$ – Sparse Relaxed Regularized Regression for Linear Mixed-Effect Models

Linear Mixed-Effect (LME) Models

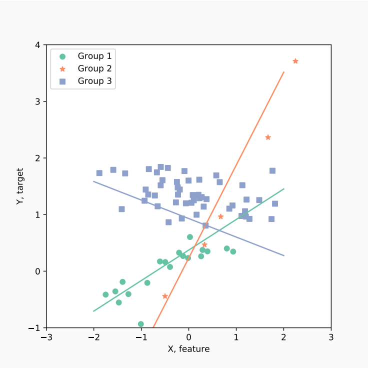
Dataset: m groups (X_i, Z_i, y_i), $\quad i = 1, \dots m$, each has n_i observations

- $X_i \in \mathbb{R}^{n_i \times p}$ – group i design matrix for fixed features
- $Z_i \in \mathbb{R}^{n_i \times q}$ – group i design matrix for random features
- $y_i \in \mathbb{R}^{n_i}$ – group i observations
- $u_i \in \mathbb{R}^{q}$ – random effects
- $\Gamma \in \mathbb{R}^{q \times q}$ – covariance matrix of random effects, often $\Gamma = \mathrm{diag}\left((\gamma)\right)$
- $\Lambda_i \in R^{n_i \times n_i}$ – covariance matrix for observation noise

Model:

$$y_i = X_i\beta + Z_i u_i + \varepsilon_i$$
$$\varepsilon_i \sim \mathcal{N}(0, \Lambda_i)$$
$$u_i \sim \mathcal{N}(0, \Gamma)$$

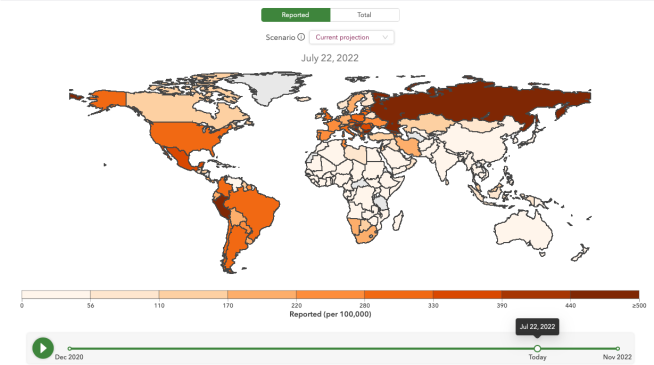
Unknowns: β, u_i, γ, sometimes Λ_i.



Mixed-Effect Models

Mixed-effect models

- Used for analyzing **combined data** across a range of **groups**.
- Use covariates to separate the **population variability** from the **group variability**.
- **Borrow strength** across groups to estimate key statistics.



[1]

[1] Picture is taken from covid19.healthdata.org

Feature Selection for Mixed-Effect Models

Practitioners:

- Often seek **sparse models** which only use **most informative** covariates.

Feature Selection for Mixed-Effect Models

Practitioners:

- Often seek **sparse models** which only use **most informative** covariates.
- Want the algorithm to be **efficient** but also **flexible** in using various regularizers.

Feature Selection for Mixed-Effect Models

Practitioners:

- Often seek **sparse models** which only use **most informative** covariates.
- Want the algorithm to be **efficient** but also **flexible** in using various regularizers.
- Want a library to be **universal and compatible** with e.g. `scikit-learn`.

Feature Selection for Mixed-Effect Models

Practitioners:

- Often seek **sparse models** which only use **most informative** covariates.
- Want the algorithm to be **efficient** but also **flexible** in using various regularizers.
- Want a library to be **universal and compatible** with e.g. `scikit-learn`.

Optimization problem:

$$\mathcal{FS} - \mathcal{LME} \qquad \min_{\beta \in \mathbb{R}^p,\ \gamma \in \mathbb{R}^q_+} \mathcal{L}(\beta, \gamma) + R(\beta, \gamma) \tag{1}$$

Where $\mathcal{L}$:

$$\begin{aligned} \mathcal{L}(\beta, \gamma) = \sum_{i=1}^{m} \frac{1}{2}(y_i - X_i\beta)^T (Z_i \Gamma Z_i^T + \Lambda_i)^{-1}(y_i - X_i\beta) + \\ + \frac{1}{2} \log\det(Z_i \Gamma Z_i^T + \Lambda_i), \quad \Gamma = \mathrm{diag}((\gamma)) \end{aligned} \tag{2}$$

Feature Selection for Mixed-Effect Models

Practitioners:

- Often seek **sparse models** which only use **most informative** covariates.
- Want the algorithm to be **efficient** but also **flexible** in using various regularizers.
- Want a library to be **universal and compatible** with e.g. `scikit-learn`.

Optimization problem:

$$\mathcal{FS} - \mathcal{LME} \qquad \min_{\beta \in \mathbb{R}^p,\, \gamma \in \mathbb{R}^q_+} \mathcal{L}(\beta, \gamma) + R(\beta, \gamma) \tag{1}$$

Where $\mathcal{L}$:

$$\begin{aligned} \mathcal{L}(\beta, \gamma) = \sum_{i=1}^{m} \frac{1}{2}(y_i - X_i\beta)^T(Z_i\Gamma Z_i^T + \Lambda_i)^{-1}(y_i - X_i\beta) + \\ + \frac{1}{2}\log\det\left(Z_i\Gamma Z_i^T + \Lambda_i\right), \quad \Gamma = \mathrm{diag}\left((\gamma)\right) \end{aligned} \tag{2}$$

- $\mathcal{L}(\beta, \gamma)$ is smooth on its domain, quadratic w.r.t. β and $\bar{\eta}$-weakly-convex w.r.t. γ.
- $R(\beta, \gamma)$ is closed, proper, with easily computed *prox operator*

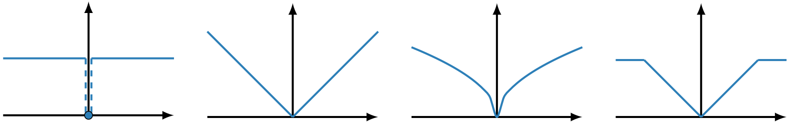
Regularization

- $R(\beta, \gamma)$ is closed, proper, with easily computed *prox operator*

$$\operatorname{prox}_{\alpha R + \delta_{\mathcal{C}}}(\tilde{\beta}, \tilde{\gamma}) := \operatorname*{argmin}_{(\beta,\gamma)\in\mathcal{C}} R(\beta, \gamma) + \frac{1}{2\alpha}\|(\beta, \gamma) - (\tilde{\beta}, \tilde{\gamma})\|_2^2, \quad \text{where } \mathcal{C} := \mathbb{R}^p \times R_+^q \tag{3}$$

Examples:

- $R(x) = \lambda \sum_{j=1}^{p} w_j \|x_j\|_1$ – LASSO and Adaptive LASSO penalties[2]
- $R(x) = \lambda \|x\|_0$ – ℓ_0 penalty[3]
- $R(x)$ – SCAD penalty[4]



(a) ℓ_0 (b) ℓ_1 (c) $\ell_p,\ p = 1/2$ (d) SCAD

Figure: Four commonly-used regularizers which promote sparsity

[2] Bondell, Krishna, and Ghosh, "Joint Variable Selection for Fixed and Random Effects in Linear Mixed-Effects Models"; Lin, Pang, and Jiang, "Fixed and random effects selection by REML and pathwise coordinate optimization".

[3] Jones, "Bayesian information criterion for longitudinal and clustered data".

[4] Fan and Li, "Variable selection in linear mixed effects models".

Proximal Gradient Descent for Feature Selection in MLE

Algorithm: PGD for standard LMEs

$\beta^{+} \leftarrow \beta_{0}, \quad \gamma^{+} \leftarrow \gamma_{0}, \quad \alpha \leftarrow 1/L$ `// Initialization`

$x^{+} = [\beta^{+}, \gamma^{+}]$;

while making progress **do**

  $x^{+} \leftarrow \operatorname{prox}_{\alpha^{-1}R+\delta_{\mathcal{C}}}(x^{+} - \alpha \nabla_{x} \mathcal{L}(x^{+}))$ `// PGD iterations`

end

return $\underline{x^{+} = [\beta^{+}, \gamma^{+}]}$

Proximal Gradient Descent for Feature Selection in MLE

Algorithm: PGD for standard LMEs

$\beta^+ \leftarrow \beta_0, \quad \gamma^+ \leftarrow \gamma_0, \quad \alpha \leftarrow 1/L$ `// Initialization`

$x^+ = [\beta^+, \gamma^+]$;

while making progress **do**

  $x^+ \leftarrow \mathrm{prox}_{\alpha^{-1}R+\delta_{\mathcal{C}}}(x^+ - \alpha \nabla_x \mathcal{L}(x^+))$ `// PGD iterations`

end

return $x^+ = [\beta^+, \gamma^+]$

Basic Assumptions for the PGD Algorithm (Theorem 10.15 from[5])

1. R is a closed proper convex function
2. $\mathcal{L}$ is closed and proper, $\mathrm{dom}\,\mathcal{L}$ convex, $\mathrm{dom}\,R \subset \mathit{int}(\mathrm{dom}\,\mathcal{L})$, and $\mathcal{L}$ is L-smooth over $\mathit{int}(\mathrm{dom}\,\mathcal{L})$.
3. The problem has an optimal solution with an optimal value $\mathcal{L}^*$

[5] Beck, First-Order Methods in Optimization.

Proximal Gradient Descent for Feature Selection in MLE

Algorithm: PGD for standard LMEs

$\beta^+ \leftarrow \beta_0, \quad \gamma^+ \leftarrow \gamma_0, \quad \alpha \leftarrow 1/L$ `// Initialization`

$x^+ = [\beta^+, \gamma^+]$;

while making progress **do**

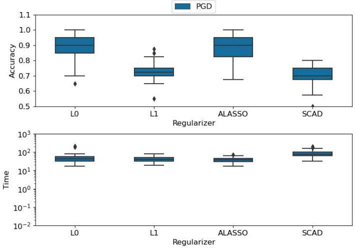
  $x^+ \leftarrow \mathrm{prox}_{\alpha^{-1}R+\delta_{\mathcal{C}}}(x^+ - \alpha \nabla_x \mathcal{L}(x^+))$ `// PGD iterations`

end

return $\underline{x^+ = [\beta^+, \gamma^+]}$

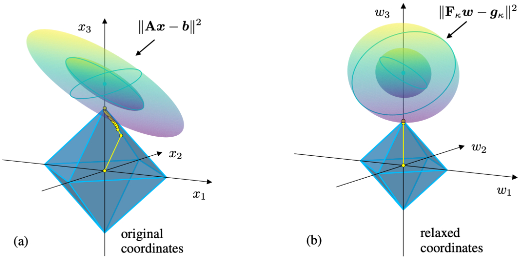
Synthetic Benchmark:

- 100 randomly-generated problems.
- $p = q = 20$.
- $\beta = \gamma = \frac{1}{2}[1, 2, 3, \dots, 10, 0 \dots, 0]$
- 9 groups from 3 to 15 observations
- $X_i \sim \mathcal{N}(0, I)^p$, $Z_i = X_i$,
  $\varepsilon_i \sim \mathcal{N}(0, 0.3^2 I)$
- Golden search for $\lambda \in [0, 10^5]$
- Final model is chosen to maximize BIC



Sparse Relaxed Regularized Regression ($\mathcal{SR}3$)[8]

$$\min_{x} f(x) + R(x) \quad \to \quad \min_{x,w} f(x) + \frac{\eta}{2}\|x - w\|_2^2 + R(w) \tag{4}$$



Objectives:

- Extend $\mathcal{SR}3$ relaxation to linear mixed-effects models - $\mathcal{MSR}3$ (see[5]).
- Develop theoretical foundations for it (see[6]).
- Implement it as a `scikit-learn`-compatible Python package – `pysr3` (see[7]).

[5] Sholokhov, Burke, et al., A Relaxation Approach to Feature Selection for Linear Mixed Effects Models.
[6] Aravkin et al., Analysis of Relaxation Methods for Feature Selection in Mixed Effects Models.
[7] Sholokhov, Zheng, and Aravkin, "pysr3: A Python Package for Sparse Relaxed Regularized Regression".
[8] Zheng and Aravkin, "Relax-and-split method for nonconvex inverse problems".

SR3-Relaxation for Mixed-Effect Models ($\mathcal{MSR}3$)

Original problem $\mathcal{FS}-\mathcal{LME}$:

$$\min_{\beta\in\mathbb{R}^p,\,\gamma\in\mathbb{R}^q_+} \mathcal{L}(\beta,\gamma) + R(\beta,\gamma) \tag{5}$$

Relaxed problem $\mathcal{MSR}3$:

$$\min_{\beta,\tilde{\beta}\in\mathbb{R}^p,\,\gamma,\tilde{\gamma}\in\mathbb{R}^q_+} \mathcal{L}(\beta,\gamma) + \phi_\mu(\gamma) + \kappa_\eta(\beta-\tilde{\beta},\gamma-\tilde{\gamma}) + R(\tilde{\beta},\tilde{\gamma}) \tag{6}$$

where the *relaxation* κ_η decouples the likelihood and the regularizer

$$\kappa_\eta(\beta-\tilde{\beta},\gamma-\tilde{\gamma}) := \frac{\eta}{2}\|\beta-\tilde{\beta}\|_2^2 + \frac{\eta}{2}\|\gamma-\tilde{\gamma}\|_2^2, \quad \eta > \bar{\eta} \tag{7}$$

and the *perspective mapping* ϕ_μ replaces $\gamma \geq 0$ with a log-barrier

$$\phi_\mu(\gamma) := \begin{cases} -\mu\sum_{i=1}^q \ln(\gamma_i/\mu), & \mu > 0 \\ \delta_{\mathbb{R}^q_+}(\gamma), & \mu = 0 \\ +\infty, & \mu < 0 \end{cases} \tag{8}$$

Value Function of $\mathcal{MSR}3$

$\mathcal{MSR}3$-relaxation replaces the original likelihood $\mathcal{L}$ with a *value function* $u_{\eta,\mu}$:

$$\begin{aligned} v_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) &:= \min_{(\beta,\gamma)} \mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) \\ &:= \min_{(\beta,\gamma)} \mathcal{L}(\beta,\gamma) + \phi_{\mu}(\gamma) + \kappa_{\eta}(\beta-\tilde{\beta},\gamma-\tilde{\gamma}) \end{aligned} \tag{9}$$

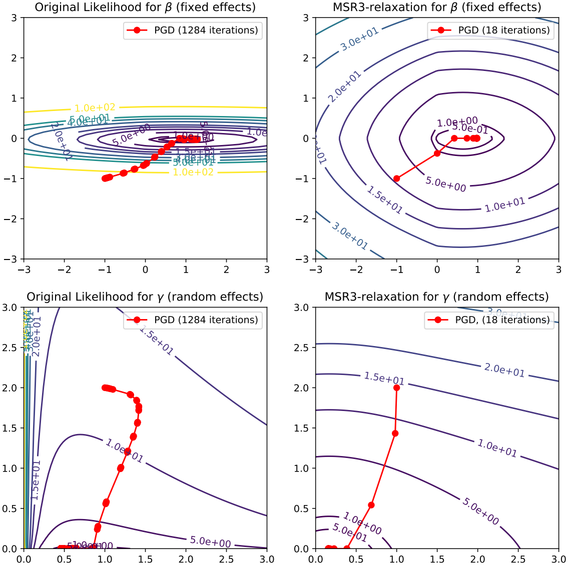
so $\mathcal{MSR}3$-formulation (6) becomes

$$\min_{\tilde{\beta},\tilde{\gamma}\in\mathcal{C}} v_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) + R(\tilde{\beta},\tilde{\gamma}) \tag{10}$$

NB: $v_{\eta,\mu}(\tilde{\beta},\tilde{\gamma})$ is smooth on $\mathcal{C}$ and can be evaluated using Interior Point (IP) method

Value Function of $\mathcal{MSR}3$

$$\min_{\beta,\gamma\in\mathcal{C}} \mathcal{L}(\beta,\gamma) + R(\beta,\gamma) \quad \text{vs} \quad \min_{\tilde\beta,\tilde\gamma\in\mathcal{C}} v_{\eta,\mu}(\tilde\beta,\tilde\gamma) + R(\tilde\beta,\tilde\gamma)$$



$\mathcal{MSR}3$: Algorithm

Algorithm: PGD for $\mathcal{MSR}3$

$\tilde\beta^+ \leftarrow \tilde\beta_0, \quad \tilde\gamma^+ \leftarrow \tilde\gamma_0, \quad \alpha \leftarrow 1/\eta, \quad \eta > \bar\eta$ `// Initialization`

$\tilde w^+ := [\tilde\beta^+, \tilde\gamma^+], \quad x^+ := [\beta, \gamma]$

while <u>making progress in $\tilde w$</u> **do**

- $x^+ \leftarrow$ IP solution on $\mathcal{L}_{\eta,\mu}(x^+, \tilde w^+)$ s.t. $x^+ \in \mathcal{C}$ `// IP Iterations`
- $\nabla_{\tilde w} v_{\eta,0}(\tilde w^+) \leftarrow \nabla_{\tilde w} \mathcal{L}_{\eta,0}(x^+, \tilde w^+)$ `// Evaluate Gradient`
- $\tilde w^+ \leftarrow \mathrm{prox}_{\alpha^{-1}R+\delta_{\mathcal{C}}}(\tilde w^+ - \alpha\nabla_{\tilde w} v_{\eta,0}(\tilde w^+))$ `// PGD on Value Function`

end

return $\underline{\tilde w^+ = [\tilde\beta^+, \tilde\gamma^+]}$

$\mathcal{MSR}3$: Algorithm

Algorithm: PGD for $\mathcal{MSR}3$

$\tilde{\beta}^+ \leftarrow \tilde{\beta}_0, \quad \tilde{\gamma}^+ \leftarrow \tilde{\gamma}_0, \quad \alpha \leftarrow 1/\eta, \quad \eta > \bar{\eta}$ // Initialization

$\tilde{w}^+ := [\tilde{\beta}^+, \tilde{\gamma}^+], \quad x^+ := [\beta, \gamma]$

while making progress in $\tilde{w}$ **do**

  $x^+ \leftarrow$ IP solution on $\mathcal{L}_{\eta,\mu}(x^+, \tilde{w}^+)$ s.t. $x^+ \in \mathcal{C}$ // IP Iterations

  $\nabla_{\tilde{w}} v_{\eta,0}(\tilde{w}^+) \leftarrow \nabla_{\tilde{w}} \mathcal{L}_{\eta,0}(x^+, \tilde{w}^+)$ // Evaluate Gradient

  $\tilde{w}^+ \leftarrow \mathrm{prox}_{\alpha^{-1}R+\delta_{\mathcal{C}}}(\tilde{w}^+ - \alpha \nabla_{\tilde{w}} v_{\eta,0}(\tilde{w}^+))$ // PGD on Value Function

end

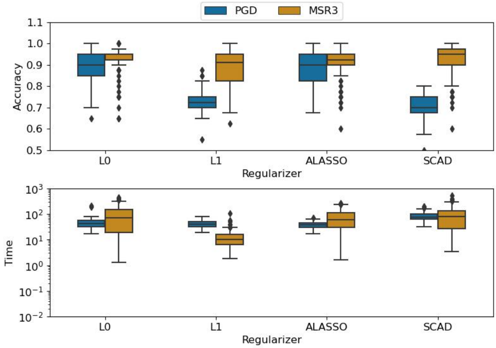
return $\tilde{w}^+ = [\tilde{\beta}^+, \tilde{\gamma}^+]$

Theoretical Results[9]:

1. The problem has an optimal solution with an optimal value Φ^* (**Theorem 5**)
2. $v_{\eta,\mu}$ is well-defined (**Theorem 5**) and continuously differentiable (**Theorem 10**)
3. $\nabla v_{\eta,\mu}$ is locally $\widetilde{L}$-continuous when R is 1-coercive (**Theorem 14**)
4. As $\mu \to 0$ (**Theorem 6**) or $\eta \to \infty$ (**Theorem 7**), cluster points of solutions to $\mathcal{MSR}3$ are FOSPs for $\mathcal{FS} - \mathcal{LME}$

[9] Aravkin et al., Analysis of Relaxation Methods for Feature Selection in Mixed Effects Models.

$\mathcal{MSR}3$: Results



$\mathcal{MSR}3$: Algorithm

Algorithm: PGD for $\mathcal{MSR}3$

$\tilde{\beta}^+ \leftarrow \tilde{\beta}_0, \quad \tilde{\gamma}^+ \leftarrow \tilde{\gamma}_0, \quad \alpha \leftarrow 1/\eta, \quad \eta > \bar{\eta}$ `// Initialization`

$\tilde{w}^+ := [\tilde{\beta}^+, \tilde{\gamma}^+], \quad x^+ := [\beta, \gamma]$

while making progress in $\tilde{w}$ **do**

- $x^+ \leftarrow$ IP solution on $\mathcal{L}_{\eta,\mu}(x^+, \tilde{w}^+)$ s.t. $x^+ \in \mathcal{C}$ `// IP Iterations`
- $\nabla_{\tilde{w}} v_{\eta,0}(\tilde{w}^+) \leftarrow \nabla_{\tilde{w}} \mathcal{L}_{\eta,0}(x^+, \tilde{w}^+)$ `// Evaluate Gradient`
- $\tilde{w}^+ \leftarrow \mathrm{prox}_{\alpha^{-1}R + \delta_{\mathcal{C}}}(\tilde{w}^+ - \alpha \nabla_{\tilde{w}} v_{\eta,0}(\tilde{w}^+))$ `// PGD on Value Function`

end

return $\tilde{w}^+ = [\tilde{\beta}^+, \tilde{\gamma}^+]$

$\mathcal{MSR}3$: Algorithm

Algorithm: PGD for $\mathcal{MSR}3$

$\tilde{\beta}^+ \leftarrow \tilde{\beta}_0, \quad \tilde{\gamma}^+ \leftarrow \tilde{\gamma}_0, \quad \alpha \leftarrow 1/\eta, \quad \eta > \bar{\eta}$ `// Initialization`

$\tilde{w}^+ := [\tilde{\beta}^+, \tilde{\gamma}^+], \quad x^+ := [\beta, \gamma]$

while making progress in $\tilde{w}$ **do**

  $x^+ \leftarrow$ IP solution on $\mathcal{L}_{\eta,\mu}(x^+, \tilde{w}^+)$ s.t. $x^+ \in \mathcal{C}$ `// IP Iterations`

  $\nabla_{\tilde{w}} v_{\eta,0}(\tilde{w}^+) \leftarrow \nabla_{\tilde{w}} \mathcal{L}_{\eta,0}(x^+, \tilde{w}^+)$ `// Evaluate Gradient`

  $\tilde{w}^+ \leftarrow \mathrm{prox}_{\alpha^{-1}R+\delta_{\mathcal{C}}}(\tilde{w}^+ - \alpha \nabla_{\tilde{w}} v_{\eta,0}(\tilde{w}^+))$ `// PGD on Value Function`

end

return $\tilde{w}^+ = [\tilde{\beta}^+, \tilde{\gamma}^+]$

Key Observation: $\nabla_{\tilde{w}} v(\tilde{w})$ does not need to be evaluated exactly. We only need to come close enough to the central path.

$\mathcal{MSR}3$-fast: Algorithm

Algorithm: $\mathcal{MSR}3$-fast

$\tilde{\beta}^+ \leftarrow \tilde{\beta}_0, \quad \tilde{\gamma}^+ \leftarrow \tilde{\gamma}_0, \quad \alpha \leftarrow 1/\eta, \quad \eta > \bar{\eta}$ `// Initialization`

$\tilde{w}^+ := [\tilde{\beta}^+, \tilde{\gamma}^+], \quad x^+ := [\beta, \gamma]$

while making progress **do**

  while not close enough to the central path **do**

    $x^+ \leftarrow$ IP iteration on $\mathcal{L}_{\eta,\mu}(x^+, \tilde{w}^+)$ s.t. $x^+ \in \mathcal{C}$ `// IP Iterations`

  end

  Decrease μ

  $\nabla_{\tilde{w}} v_{\eta,\mu}(\tilde{w}^+) \leftarrow \nabla_{\tilde{w}} \mathcal{L}_{\eta,\mu}(x^+, \tilde{w}^+)$ `// Evaluate Gradient`

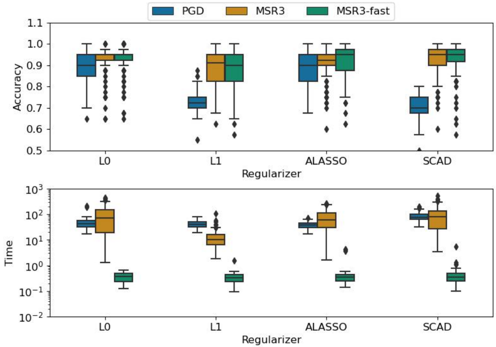
  $\tilde{w}^+ \leftarrow \mathrm{prox}_{\alpha^{-1}R+\delta_{\mathcal{C}}}(\tilde{w}^+ - \alpha \nabla_{\tilde{w}} v_{\eta,\mu}(\tilde{w}^+))$ `// PGD on Value Function`

end

return $\underline{\tilde{w}^+ = [\tilde{\beta}^+, \tilde{\gamma}^+]}$

$\mathcal{MSR}$3-fast: Results

- The number of fixed effects p and random effects q is 20.
- $\beta = \gamma = \frac{1}{2}[1, 2, 3, \ldots, 10, 0 \ldots, 0]$
- 9 groups with sizes [10, 15, 4, 8, 3, 5, 18, 9, 6]
- $X_i \sim \mathcal{N}(0, I)^p$, $Z_i = X_i$, $\varepsilon_i \sim \mathcal{N}(0, 0.3^2 I)$
- Each experiment is repeated 100 times.
- Grid-search for $\eta \in [10^{-4}, 10^2]$, golden search for $\lambda \in [0, 10^5]$
- Final model is chosen to maximize BIC



\+ $\mathcal{MSR}$3-relaxation improves feature selection performance of the original likelihood.
\+ $\mathcal{MSR}$3-fast optimization accelerates the compute time by $\sim 10^2$.
− Initialization of η is problem-specific

Comparison to Other Libraries

Algorithm	MSR3-Fast (ℓ_1)	`glmmLasso`[10][11]	`lmmLasso`[12][13]	PGD (ℓ_1)
Accuracy, %	**88**	48	66	73
FE Accuracy, %	**86**	52	47	56
RE Accuracy, %	**91**	45	84	**91**
Time, sec	**0.19**	1.37	11.51	38.39
Iterations, num	34	50	-	7693

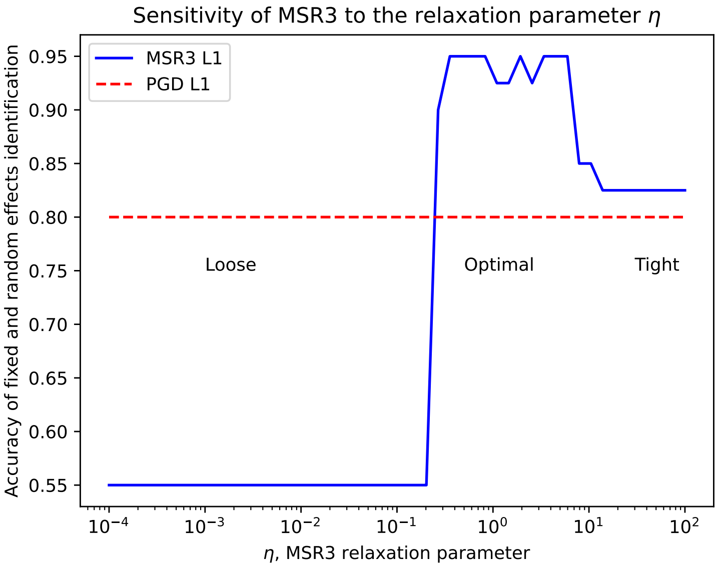
[10] https://rdrr.io/cran/glmmLasso/man/glmmLasso.html

[11] Groll and Tutz, "Variable selection for generalized linear mixed models by L 1-penalized estimation".

[12] https://rdrr.io/cran/lmmlasso/

[13] Schelldorfer, Bühlmann, and DE GEER, "Estimation for high-dimensional linear mixed-effects models using l1-penalization".

Choice of η



ℓ_0-based Covariate Selection for Bullying Study from GBD

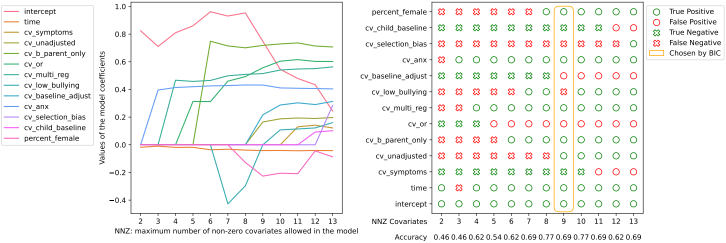
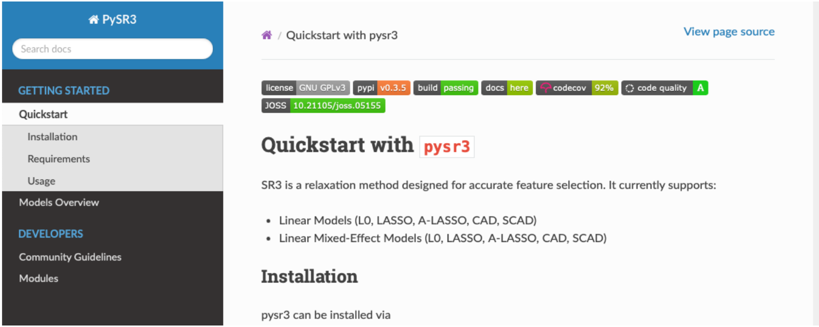


Figure: Fixed and random covariate selection for Bullying dataset[14]. The model selected 9 covariates, 7 of which were historically significant, and did not select 4 covariates, 1 of which was historically significant.

[14] Institute for Health Metrics and Evaluation (IHME). Bullying Victimization Relative Risk Bundle GBD 2020. Seattle, United States of America (USA), 2021.

Software



The code is available on GitHub: https://github.com/aksholokhov/pysr3[15]

- All estimators are fully compatible to `scikit-learn` library.
- Implements SR3 for linear, generalized-linear, and linear mixed-effect models.
- Has tutorials, tests, and documentation.

[15] Aleksei Sholokhov, Peng Zheng, and Aleksandr Aravkin. "pysr3: A Python Package for Sparse Relaxed Regularized Regression". In: Journal of Open Source Software 8.84 (2023), p. 5155.

Physics-Informed Neural ODE (PINODE): Embedding Physics into Models using Collocation Points

Data-Driven Modeling of Physical Systems

Data-Driven Modeling of Physical Systems

First-Principle Models:

- Require extensive knowledge of the phenomenon
- Require a lot of compute for simulating large-scale phenomena

$$\begin{cases} \rho \dfrac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, p) = \mathbf{f} & \text{in } \Omega \times (0, T) \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \times (0, T) \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma_D \times (0, T) \\ \boldsymbol{\sigma}(\mathbf{u}, p)\hat{\mathbf{n}} = \mathbf{h} & \text{on } \Gamma_N \times (0, T) \\ \mathbf{u}(0) = \mathbf{u}_0 & \text{in } \Omega \times \{0\} \end{cases}$$

Data-Driven Modeling of Physical Systems

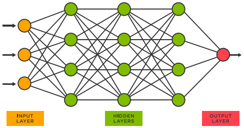
First-Principle Models:

- Require extensive knowledge of the phenomenon
- Require a lot of compute for simulating large-scale phenomena

$$
\begin{cases}
\rho \dfrac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, p) = \mathbf{f} & \text{in } \Omega \times (0, T) \\
\nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \times (0, T) \\
\mathbf{u} = \mathbf{g} & \text{on } \Gamma_D \times (0, T) \\
\boldsymbol{\sigma}(\mathbf{u}, p)\hat{\mathbf{n}} = \mathbf{h} & \text{on } \Gamma_N \times (0, T) \\
\mathbf{u}(0) = \mathbf{u}_0 & \text{in } \Omega \times \{0\}
\end{cases}
$$

Data-Driven Models:

- Require a lot of data
- Often struggle to extrapolate



Data-Driven Modeling of Physical Systems

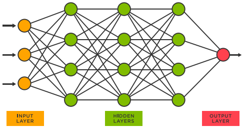
First-Principle Models:

- Require extensive knowledge of the phenomenon
- Require a lot of compute for simulating large-scale phenomena

$$\begin{cases} \rho \dfrac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, p) = \mathbf{f} & \text{in } \Omega \times (0, T) \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \times (0, T) \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma_D \times (0, T) \\ \boldsymbol{\sigma}(\mathbf{u}, p)\hat{\mathbf{n}} = \mathbf{h} & \text{on } \Gamma_N \times (0, T) \\ \mathbf{u}(0) = \mathbf{u}_0 & \text{in } \Omega \times \{0\} \end{cases}$$

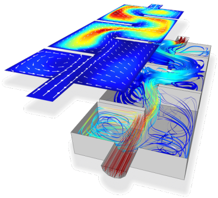
Data-Driven Models:

- Require a lot of data
- Often struggle to extrapolate



Hybrid Models:

- Incorporate elements of both approaches
- Supplement data with knowledge or priors



Incorporating Knowledge of Physics into Neural Networks

[16] Geiger and Smidt, "e3nn: Euclidean neural networks".

[17] Finzi et al., "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data".

[18] Chidester, Do, and Ma, Rotation Equivariance and Invariance in Convolutional Neural Networks.

[19] Champion et al., "Data-driven discovery of coordinates and governing equations".

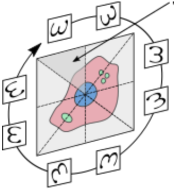
[20] Schmidt and Lipson, "Distilling free-form natural laws from experimental data".

[21] Raissi, Perdikaris, and Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations".

[22] Rackauckas et al., "Universal differential equations for scientific machine learning".

Incorporating Knowledge of Physics into Neural Networks

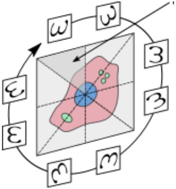
1. **Symmetry Based:** incorporate symmetries and conservation laws as hard constraints into a network: E3NNs[16], LieConv[17], RiCNN[18]

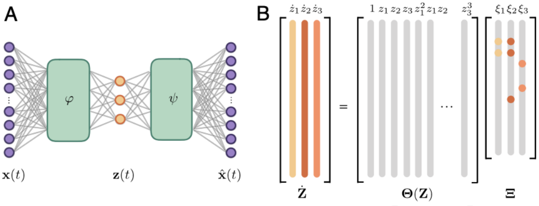


[16] Geiger and Smidt, "e3nn: Euclidean neural networks".
[17] Finzi et al., "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data".
[18] Chidester, Do, and Ma, Rotation Equivariance and Invariance in Convolutional Neural Networks.
[19] Champion et al., "Data-driven discovery of coordinates and governing equations".
[20] Schmidt and Lipson, "Distilling free-form natural laws from experimental data".
[21] Raissi, Perdikaris, and Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations".
[22] Rackauckas et al., "Universal differential equations for scientific machine learning".

Incorporating Knowledge of Physics into Neural Networks

1. **Symmetry Based:** incorporate symmetries and conservation laws as hard constraints into a network: E3NNs[16], LieConv[17], RiCNN[18]
2. **Model Discovery Based:** use a library of terms to discover simple equations: SINDy[19], Genetic Programming[20]





[16] Geiger and Smidt, "e3nn: Euclidean neural networks".

[17] Finzi et al., "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data".

[18] Chidester, Do, and Ma, Rotation Equivariance and Invariance in Convolutional Neural Networks.

[19] Champion et al., "Data-driven discovery of coordinates and governing equations".

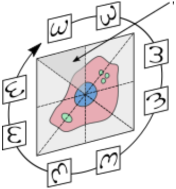
[20] Schmidt and Lipson, "Distilling free-form natural laws from experimental data".

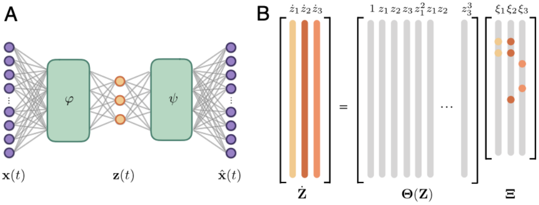
[21] Raissi, Perdikaris, and Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations".

[22] Rackauckas et al., "Universal differential equations for scientific machine learning".

Incorporating Knowledge of Physics into Neural Networks

1. **Symmetry Based:** incorporate symmetries and conservation laws as hard constraints into a network: E3NNs[16], LieConv[17], RiCNN[18]
2. **Model Discovery Based:** use a library of terms to discover simple equations: SINDy[19], Genetic Programming[20]
3. **Equations Based:** incorporate first-principle models to aid training of networks: PINNs[21], UDEs[22]





[16] Geiger and Smidt, "e3nn: Euclidean neural networks".

[17] Finzi et al., "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data".

[18] Chidester, Do, and Ma, Rotation Equivariance and Invariance in Convolutional Neural Networks.

[19] Champion et al., "Data-driven discovery of coordinates and governing equations".

[20] Schmidt and Lipson, "Distilling free-form natural laws from experimental data".

[21] Raissi, Perdikaris, and Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations".

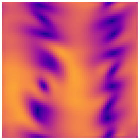
[22] Rackauckas et al., "Universal differential equations for scientific machine learning".

Reduced-Order Models (ROMs)

$x \in \mathbb{R}^n$

$$\frac{dx}{dt} = f(x)$$

x_0



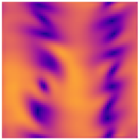
Reduced-Order Models (ROMs)

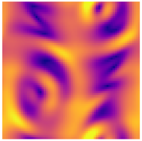
$x \in \mathbb{R}^n$

$$\frac{dx}{dt} = f(x)$$

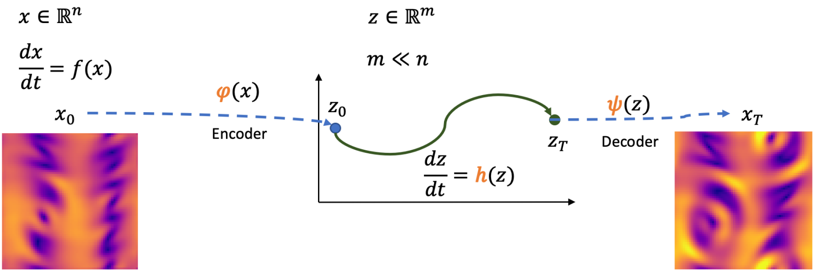
$$x_T = x_0 + \int_0^T f(x)dt$$

$x_0 \longrightarrow x_T$

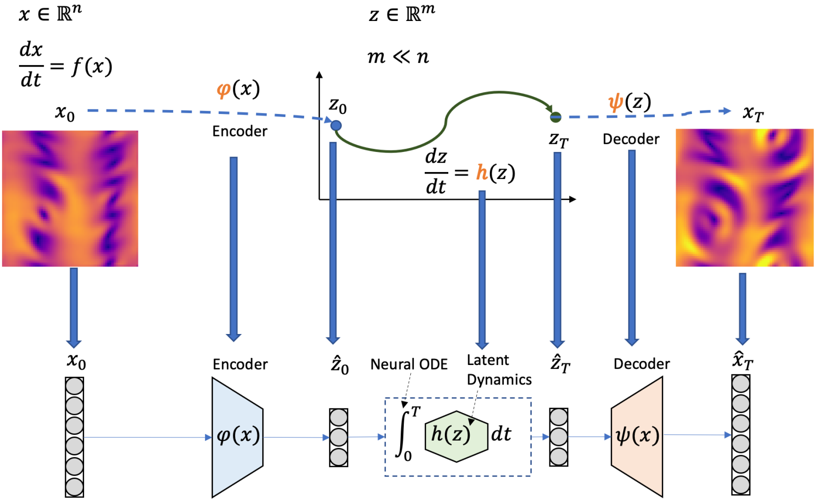




Reduced-Order Models (ROMs)



Reduced-Order Models (ROMs)



Physics-Informed Loss

Using chain rule:

$$\frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = \nabla\varphi(x)^T f(x)$$

Physics-Informed Loss

Using chain rule:

$$\frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = \nabla\varphi(x)^T f(x) \qquad \frac{dz}{dt} = h(\varphi(x))$$

Physics-Informed Loss

Using chain rule:

$$\frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = \nabla\varphi(x)^T f(x) \qquad \frac{dz}{dt} = h(\varphi(x))$$

$$\mathcal{L}^{physics}(\tilde{x}) = \|\nabla\varphi(\tilde{x})^T f(\tilde{x}) - h(\varphi(\tilde{x}))\|_2^2 + \|\tilde{x} - \psi(\varphi(\tilde{x}))\|_2^2$$

Physics-Informed Loss = Latent Gradient Loss + Collocation Reconstruction Loss

Physics-Informed Loss

Using chain rule:

$$\frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = \nabla\varphi(x)^T f(x) \qquad \frac{dz}{dt} = h(\varphi(x))$$

1 2 3

$$\mathcal{L}^{physics}(\tilde{x}) = \|\nabla\varphi(\tilde{x})^T f(\tilde{x}) - h(\varphi(\tilde{x}))\|_2^2 + \|\tilde{x} - \psi(\varphi(\tilde{x}))\|_2^2$$

Physics-Informed Loss = Latent Gradient Loss + Collocation Reconstruction Loss

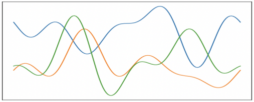
1

$$-u_t = u_{xx} + u_{xxxx} + \frac{1}{2}u_x^2 \Rightarrow \dot{x} = f(x)$$

$$u(x) = \frac{a}{1 + e^{-k(x-x_0)}} - \frac{a}{1 + e^{-k(x-x_1)}}, \quad x_0 < x_1$$

$$u(x) = \sum_{w=1}^{30} a(w)\sin(2\pi x) + b(w)\cos(2\pi x)$$

$$u(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-x_0)^2}{2\sigma^2}}$$



Physics-Informed Loss

Using chain rule:

$$\frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = \nabla\varphi(x)^T f(x) \qquad \frac{dz}{dt} = h(\varphi(x))$$

$$\mathcal{L}^{physics}(\tilde{x}) = \|\nabla\varphi(\tilde{x})^T f(\tilde{x}) - h(\varphi(\tilde{x}))\|_2^2 + \|\tilde{x} - \psi(\varphi(\tilde{x}))\|_2^2$$

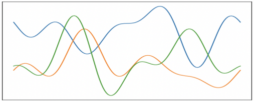
Physics-Informed Loss = Latent Gradient Loss + Collocation Reconstruction Loss

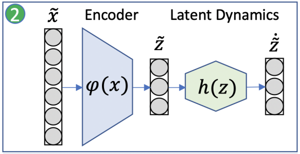
$$-u_t = u_{xx} + u_{xxxx} + \frac{1}{2}u_x^2 \Rightarrow \dot{x} = f(x)$$

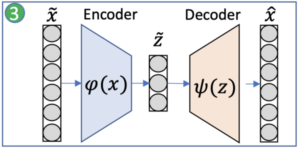
$$u(x) = \frac{a}{1+e^{-k(x-x_0)}} - \frac{a}{1+e^{-k(x-x_1)}}, \quad x_0 < x_1$$

$$u(x) = \sum_{w=1}^{30} a(w)\sin(2\pi x) + b(w)\cos(2\pi x)$$

$$u(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-x_0)^2}{2\sigma^2}}$$







Results: Extrapolation to Unknown Regions

Duffing Oscillator on a low-dimensional (2D) manifold:

$$\begin{aligned} \frac{dz_1}{dt} &= z_2 \\ \frac{dz_2}{dt} &= z_1 - z_1^3 \end{aligned} \tag{11}$$

Projection to a high-dimensional (128) space:

$$\boldsymbol{x} := \mathcal{A}(\boldsymbol{z}) = A\boldsymbol{z}^3, \quad A \in \mathbb{R}^{128\times 2}, \quad A_{ij} \sim_{i.i.d.} \mathcal{N}(0,1) \tag{12}$$

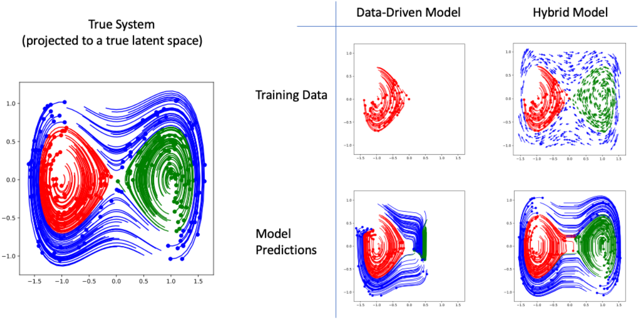
Results: Extrapolation to Unknown Regions

Duffing Oscillator on a low-dimensional (2D) manifold:

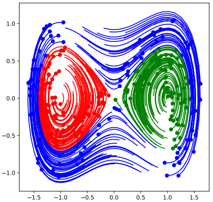
$$\begin{aligned} \frac{dz_1}{dt} &= z_2 \\ \frac{dz_2}{dt} &= z_1 - z_1^3 \end{aligned} \tag{11}$$

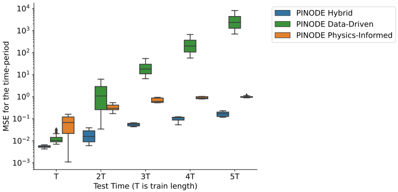
Projection to a high-dimensional (128) space:

$$\boldsymbol{x} := \mathcal{A}(\boldsymbol{z}) = A\boldsymbol{z}^3, \quad A \in \mathbb{R}^{128\times 2}, \quad A_{ij} \sim_{i.i.d.} \mathcal{N}(0,1) \tag{12}$$

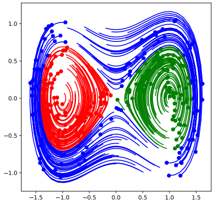


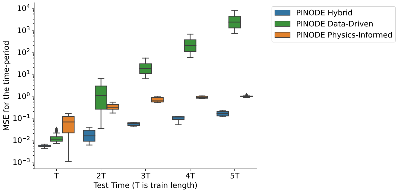
Results: Stable Long-Term Predictions





Results: Stable Long-Term Predictions

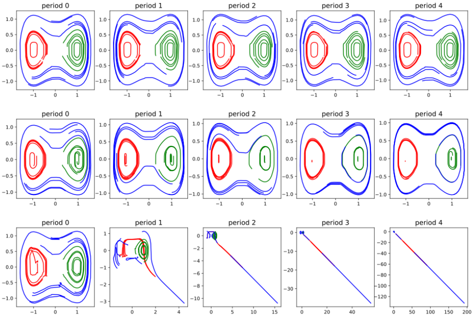




Hybrid Model

Physics-Informed Model

Data-Driven Model



Results: Burgers' Equation

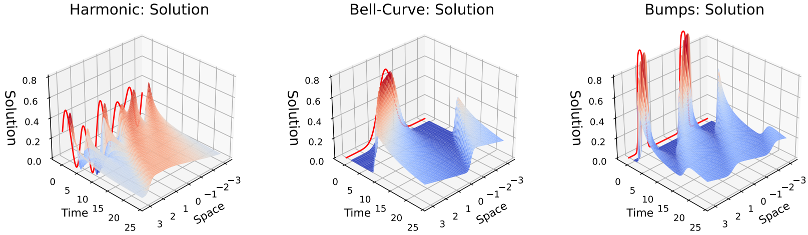
Burgers' Equation:

$$\begin{aligned} u_t + uu_x &= \nu u_{xx} \\ u(-\pi, t) &= u(\pi, t), \quad \forall t \in [0, T] \end{aligned} \tag{13}$$

Results: Burgers' Equation

Burgers' Equation:

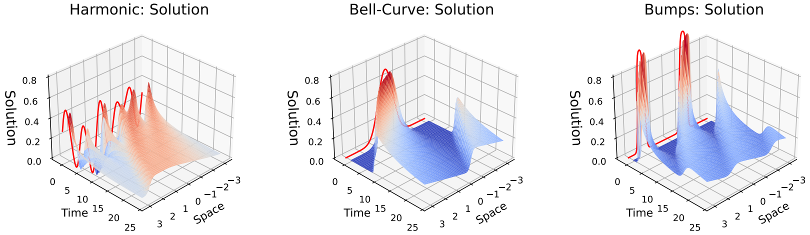
$$
\begin{aligned}
u_t + u u_x &= \nu u_{xx} \\
u(-\pi, t) &= u(\pi, t), \quad \forall t \in [0, T]
\end{aligned}
\tag{13}
$$

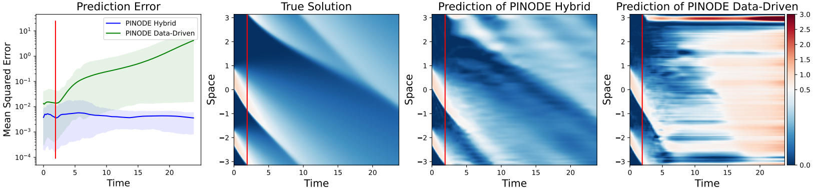


Results: Burgers' Equation

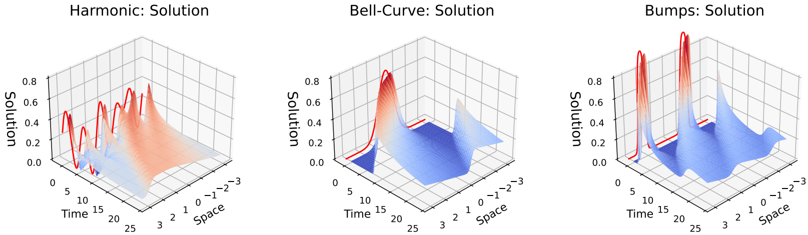
Burgers' Equation:

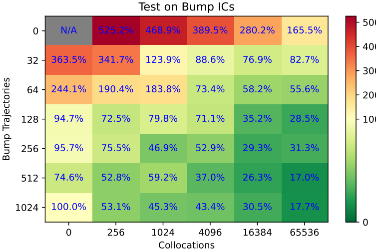
$$
\begin{aligned}
u_t + u u_x &= \nu u_{xx} \\
u(-\pi, t) &= u(\pi, t), \quad \forall t \in [0, T]
\end{aligned}
\tag{13}
$$

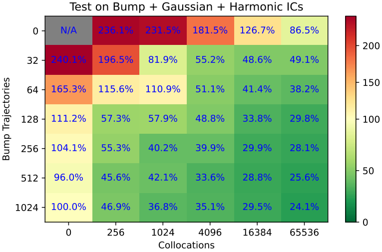




Results: Learning From Collocations







Discussion and Limitations

We showed that:

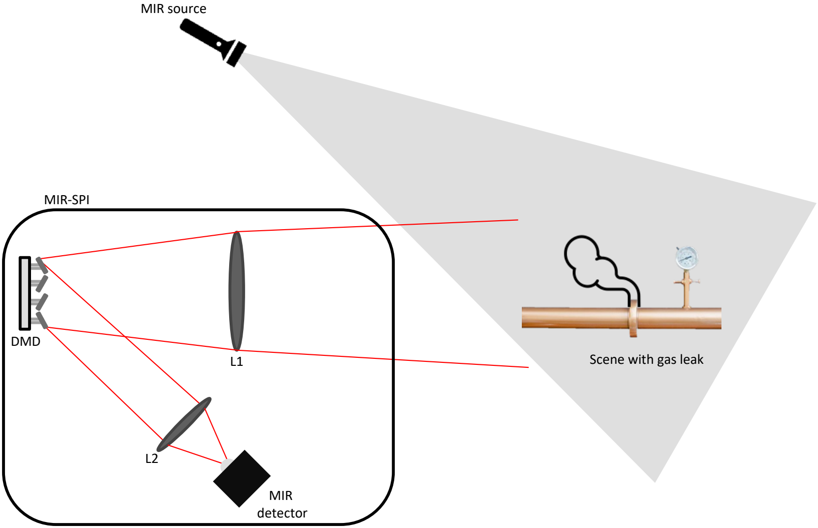
- Physics-informed loss improves accuracy and forecasting stability of ROMs
- Collocations can supplement data to improve model's performance for unseen initial conditions.

Limitations:

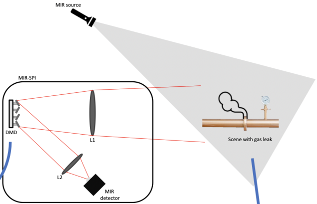
- Optimal choice of collocations is problem-specific
- Need a lot of collocations
  - *Could* be possible to overcome with smarter sampling techniques

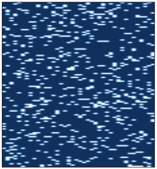
Single pixel imaging of spatio-temporal flows using differentiable latent dynamics

Single-Pixel Imaging



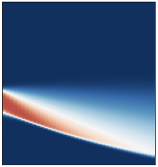
Single-Pixel Imaging





Compressive Sensing
matrix **A**

*



Complete Observation **x**

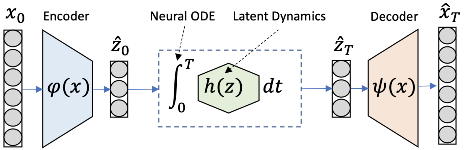
Sum

Collected Information

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$$

Compressive Sensing with Reduced-Order Models

Offline Step: Train a Data-Driven Reduced-Order Model



Online Step: Reconstruct Complete Observations by Optimizing in Latent Space

Reconstruction Loss

Compressive Sensing Loss

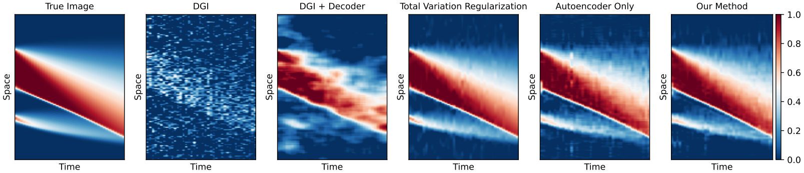
Loss for Prediction in Latent space

$$\mathcal{L}^{recon.}(z) = \underbrace{||y - A\psi(z)||}_{\text{"What the data tells us the trajectory should be"}} + \lambda \left\| z - \underbrace{(z_0 + \int_0^T h(z)dz)}_{\text{"What the model thinks the trajectory should be"}} \right\|$$

Latent-space representation of the trajectory

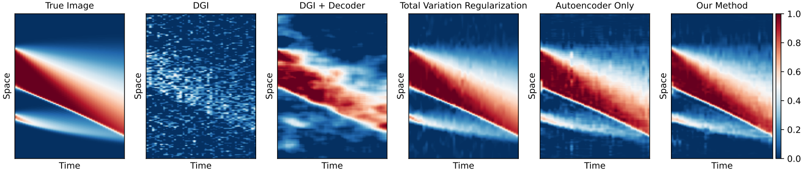
Results: Burger's Equation

When we capture 32 samples per frame:

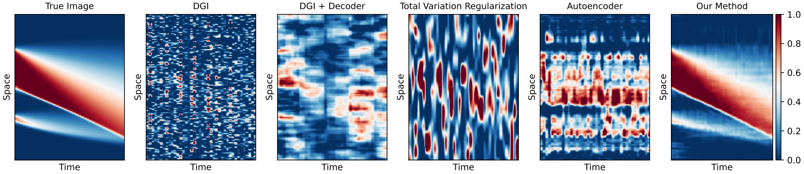


Results: Burger's Equation

When we capture 32 samples per frame:



When we capture 2 samples per frame:



Results: Burger's Equation

Aggregated results

Results: Interpretation

Results: Kolmogorov Flow OR Real Example

Conclusion

Results on Burgers Maybe results on a harder problem

Appendix: $\mathcal{MSR}3$

Designing an Algorithm

$G_{\nu,\eta}$ encodes both gradient of a Lagrangian (lines 1-2) and the complementarity condition (line 3):

$$G_{\nu,\eta}((\beta,\gamma,v),(\tilde{\beta},\tilde{\gamma})) := \begin{bmatrix} \nabla_\beta \mathcal{L}(\beta,\gamma) + \eta(\beta - \tilde{\beta}) \\ \nabla_\gamma \mathcal{L}(\beta,\gamma) + \eta(\gamma - \tilde{\gamma}) - v \\ v \bigodot \gamma - \mu \mathbf{1} \end{bmatrix} \tag{14}$$

We apply Newton method to G while geometrically decreasing μ.
Lemma: For every $(\mu,\eta) \in \mathbb{R}_+ \times \mathbb{R}_{++}$,

$$\begin{aligned} &(\hat{\beta},\hat{\gamma}) = \operatorname*{argmin}_{(\beta,\gamma)} \mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) \\ &\iff \\ &\exists \hat{v} \in \mathbb{R}^q_+ \text{ s.t. } G_{\nu,\eta}((\beta,\gamma,\hat{v}),(\tilde{\beta},\tilde{\gamma})) = 0 \end{aligned} \tag{15}$$

If $\mu > 0$, then $\hat{v} = -\nabla\phi_\mu(\hat{\gamma})$, and if $\mu = 0$, then $\hat{v}$ is the unique KKT multiplier associated with the constraint $0 \leq \gamma$.

$\mathcal{MSR}3$-fast Algorithm

progress $\leftarrow$ **True**; iter = 0;

$\beta^+, \tilde{\beta}^+ \leftarrow \beta_0$; $\gamma^+, \tilde{\gamma}^+ \leftarrow \gamma_0$; $v^+ \leftarrow 1 \in \mathbb{R}^q$; $\mu \leftarrow \frac{{v^+}^T \gamma^+}{10q}$

while iter $<$ max_iter and $\|G_\mu(\beta^+, \gamma^+, v^+)\| >$ tol and progress **do**

- $\beta \leftarrow \beta^+$; $\gamma \leftarrow \gamma^+$; $\tilde{\beta} \leftarrow \tilde{\beta}^+$; $\tilde{\gamma} \leftarrow \tilde{\gamma}^+$
- $[dv, d\beta, d\gamma] \leftarrow \nabla G_\mu((\beta, \gamma, v), (\tilde{\beta}, \tilde{\gamma}))^{-1} G_\mu((\beta, \gamma, v), (\tilde{\beta}, \tilde{\gamma}))$
  $\alpha \leftarrow 0.99 \times \min\left(1, -\frac{\gamma_i}{d\gamma_i}, \forall i: \; d\gamma_i < 0\right)$
- $\beta^+ \leftarrow \beta + \alpha d\beta$; $\gamma^+ = \gamma + \alpha d\gamma$; $v^+ \leftarrow v + \alpha dv$
- **if** $\|\gamma^+ \odot v^+ - q^{-1} {\gamma^+}^T v^+ \mathbf{1}\| > 0.5 q^{-1} {v^+}^T \gamma^+$ **then** continue;
- **else**
  - $\tilde{\beta}^+ = \mathrm{prox}_{\alpha R}(\beta^+)$; $\tilde{\gamma}^+ = \mathrm{prox}_{\alpha R + \delta_{\mathbb{R}_+}}(\gamma^+)$; $\mu = \frac{1}{10} \frac{{v^+}^T \gamma^+}{q}$
- **end**
- progress $= (\|\beta^+ - \beta\| \geq$ tol or $\|\gamma^+ - \gamma\| \geq$ tol or $\|\tilde{\beta}^+ - \tilde{\beta}\| \geq$ tol or $\|\tilde{\gamma}^+ - \tilde{\gamma}\| \geq$ tol$)$
- iter += 1

end

return $\tilde{\beta}^+, \tilde{\gamma}^+$

References I

Aravkin, Aleksandr et al. Analysis of Relaxation Methods for Feature Selection in Mixed Effects Models. 2022. arXiv: 2209.10575 [stat.ME].

Beck, Amir. First-Order Methods in Optimization. MOS-SIAM Series on Optimization. SIAM, 2017. ISBN: 9781611974980. DOI: 10.1137/1.9781611974997.

Bondell, Howard D., Arun Krishna, and Sujit K. Ghosh. "Joint Variable Selection for Fixed and Random Effects in Linear Mixed-Effects Models". In: Biometrics 66.4 (Dec. 2010), pp. 1069–1077. ISSN: 0006341X. DOI: 10.1111/j.1541-0420.2010.01391.x. arXiv: NIHMS150003. URL: http://doi.wiley.com/10.1111/j.1541-0420.2010.01391.x.

Champion, Kathleen et al. "Data-driven discovery of coordinates and governing equations". In: Proceedings of the National Academy of Sciences 116.45 (2019), pp. 22445–22451.

Chidester, Benjamin, Minh N. Do, and Jian Ma. Rotation Equivariance and Invariance in Convolutional Neural Networks. 2018. arXiv: 1805.12301 [stat.ML].

Fan, Yingying and Runze Li. "Variable selection in linear mixed effects models". In: The Annals of Statistics 40.4 (Aug. 2012), pp. 2043–2068. ISSN: 0090-5364. DOI: 10.1214/12-AOS1028. URL: http://projecteuclid.org/euclid.aos/1351602536.

References II

- Finzi, Marc et al. "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data". In: 37th International Conference on Machine Learning, ICML 2020 PartF16814 (2020), pp. 3146–3157. arXiv: 2002.12880.
- Geiger, Mario and Tess Smidt. "e3nn: Euclidean neural networks". In: arXiv preprint arXiv:2207.09453 (2022).
- Groll, Andreas and Gerhard Tutz. "Variable selection for generalized linear mixed models by L 1-penalized estimation". In: Statistics and Computing 24.2 (2014), pp. 137–154.
- Jones, Richard H. "Bayesian information criterion for longitudinal and clustered data". In: Statistics in Medicine 30.25 (Nov. 2011), pp. 3050–3056. ISSN: 02776715. DOI: 10.1002/sim.4323. URL: http://doi.wiley.com/10.1002/sim.4323.
- Lin, Bingqing, Zhen Pang, and Jiming Jiang. "Fixed and random effects selection by REML and pathwise coordinate optimization". In: Journal of Computational and Graphical Statistics 22.2 (2013), pp. 341–355. ISSN: 10618600. DOI: 10.1080/10618600.2012.681219.
- Rackauckas, Christopher et al. "Universal differential equations for scientific machine learning". In: arXiv preprint arXiv:2001.04385 (2020).
- Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis. "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations". In: arXiv preprint arXiv:1711.10561 (2017).

References III

Schelldorfer, Jürg, Peter Bühlmann, and SARA VAN DE GEER. "Estimation for high-dimensional linear mixed-effects models using l1-penalization". In: Scandinavian Journal of Statistics 38.2 (2011), pp. 197–214.

Schmidt, Michael and Hod Lipson. "Distilling free-form natural laws from experimental data". In: science 324.5923 (2009), pp. 81–85.

Sholokhov, Aleksei, James V. Burke, et al. A Relaxation Approach to Feature Selection for Linear Mixed Effects Models. 2022. arXiv: 2205.06925 [stat.ME].

Sholokhov, Aleksei, Peng Zheng, and Aleksandr Aravkin. "pysr3: A Python Package for Sparse Relaxed Regularized Regression". In: Journal of Open Source Software 8.84 (2023), p. 5155.

Zheng, Peng and Aleksandr Aravkin. "Relax-and-split method for nonconvex inverse problems". In: Inverse Problems 36.9 (2020). ISSN: 13616420. DOI: 10.1088/1361-6420/aba417.