Dimensionality Reduction in the Parameter Space

Active Subspace and Nonlinear Level Set Learning Techniques

Malcolm Gaynor and Cade Stanley

Advisor: Zhu Wang

Graduate Student: Yuwei Geng

U of SC Summer REU: Summer School on Mathematical Foundation of Data Science

July 15, 2022

Introduction: The Curse of Dimensionality

- **Problem**: when the number of parameters increases, the costs of computing the function increase exponentially
- **Goal**: reduce costs of performing parameter studies for complex, high-dimensional functions
- **Idea**: find important directions in the parameter space
  - directions along which most of the function's variability can be observed

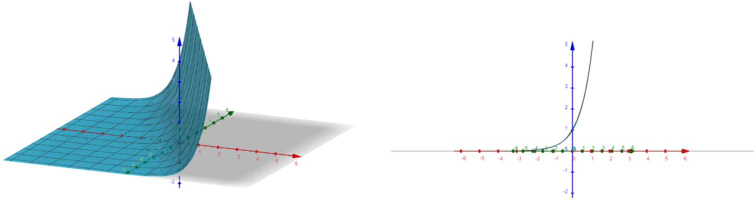


Figure: A 2D exponential function that collapses into 1 dimension when rotated.

Two Approaches

- **Active Subspace (AS) Method:** a linear approach to finding important directions, based on eigendecomposition of the gradient covariance matrix
- **Nonlinear Level Set Learning (NLL) Method:** a nonlinear approach to parameterizing level sets in a lower dimension, based on reversible neural networks called RevNets

The Active Subspace Method

Finding the Active Subspace

- Suppose f is a d-dimensional function with domain $\Omega \subset \mathbb{R}^d$
- **Goal:** find a subspace of Ω where changes in the parameters explain most of f's variability
- **Gradient Covariance Matrix:** $\boldsymbol{C} = \mathbb{E}\left[(\nabla_{\boldsymbol{x}} f)(\nabla_{\boldsymbol{x}} f)^\top\right]$

Procedure:

- Approximate $\boldsymbol{C}$ using $\hat{\boldsymbol{C}} = \frac{1}{M}\sum_{i=1}^{M}(\nabla_{\boldsymbol{x}} f_i)(\nabla_{\boldsymbol{x}} f_i)^\top$
- Perform eigendecomposition: $\hat{\boldsymbol{C}} = \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^\top$
- Search for a large gap between eigenvalues, and partition at the gap:

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_1 & \boldsymbol{W}_2 \end{bmatrix} \qquad \boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_1 & \\ & \boldsymbol{\Lambda}_2 \end{bmatrix}$$

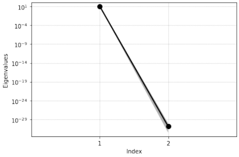
- $\boldsymbol{W}_1$ holds the eigenvectors that make up the active subspace, $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_n$
- $\boldsymbol{\Lambda}_1$ holds the eigenvalues corresponding to those eigenvectors, $\lambda_1, \ldots, \lambda_n$

Visualizing the Active Subspace

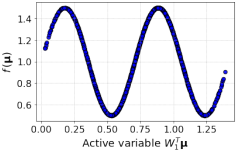
- **Eigenvalue plot:** plot all eigenvalues in decreasing order - visualize the gaps
- **Sufficient summary plot:** plot the function value against the linear combinations $\boldsymbol{\eta}_1^\top \boldsymbol{x}, \ldots, \boldsymbol{\eta}_n^\top \boldsymbol{x}$
- **Sensitivity chart:** compare the function's sensitivity to its original parameters with its sensitivity to the new active and inactive dimensions
- **Activity score:** analyze the activity of the parameters in the eigenvectors making up the active subspace
- **Grid transformation:** apply the active subspace transformation to the original parameter space and visualize the transformed space

A Simple Example

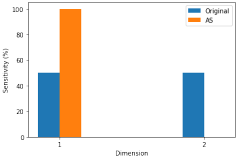
$$f(\boldsymbol{x}) = \frac{1}{2}\sin(2\pi(x_1 + x_2)) + 1$$



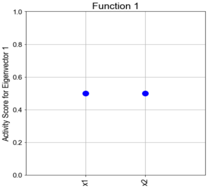
(a) Eigenvalue plot



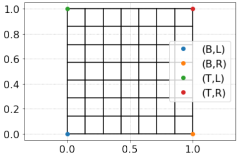
(b) Sufficient summary plot



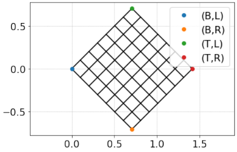
(c) Sensitivity chart



(d) Activity score



(e) Original parameter space



(f) Transformed parameter space

The Nonlinear Level Set Learning Method

Development

- Suppose f is a d-dimensional function with domain $\Omega \subset \mathbb{R}^d$
- **Goal:** create a bijective nonlinear transformation $\boldsymbol{g} : \Omega \to \mathbb{R}^d$, where for any $\boldsymbol{x} \in \Omega$, $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{z}$ so that $\boldsymbol{z}$ has a small number of "active" inputs
- We want to write $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2)$ so that $f \circ \boldsymbol{g}^{-1}$ is sensitive only to changes in $\boldsymbol{z}_1$, where $\boldsymbol{z}_1$'s dimension is much smaller than d.
- $f \circ \boldsymbol{g}^{-1}(\boldsymbol{z}) = f(\boldsymbol{x})$, so approximating $f \circ \boldsymbol{g}^{-1}$ using only the active components in $\boldsymbol{z}_1$ will let us approximate f in lower dimensions

The Neural Network

- The transformation $\boldsymbol{g}$ is trained using a RevNet with the following structure:

$$\begin{cases} \boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h\boldsymbol{K}_{n,1}^{\top}\boldsymbol{\sigma}(\boldsymbol{K}_{n,1}\boldsymbol{v}_n + \boldsymbol{b}_{n,1}) \\ \boldsymbol{v}_{n+1} = \boldsymbol{v}_n - h\boldsymbol{K}_{n,2}^{\top}\boldsymbol{\sigma}(\boldsymbol{K}_{n,2}\boldsymbol{u}_{n+1} + \boldsymbol{b}_{n,2}) \end{cases},$$

  for $n = 0, 1, \ldots, N-1$.
  - $h \in \mathbb{R}$ is the time step
  - $\boldsymbol{\sigma}$ is the activation function (usually chosen as $tanh$)
  - $\boldsymbol{K}_{n,1}, \boldsymbol{K}_{n,2}$ are weight matrices, and $\boldsymbol{b}_{n,1}, \boldsymbol{b}_{n,2}$ are bias vectors, which are all updated according to the loss function
- Letting $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{v}_0 \end{bmatrix}$ and $\boldsymbol{z} = \begin{bmatrix} \boldsymbol{u}_N \\ \boldsymbol{v}_N \end{bmatrix}$, the network yields the transformation $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{z}$
- When using the RevNet, it is also necessary to specify the number of layers, the active dimension, the learning rate, and the number of epochs, along with the time step and the activation function.

The Loss Function

- The Loss Function is used to train the RevNet with the structure outlined above
- There are two parts to the loss function, L_1 and L_2, both of which depend on the Jacobian matrix of the inverse transformation $\boldsymbol{g}^{-1}$
  - Jacobian matrix:

$$J(z) = [J_1(z), J_2(z), \dots J_d(z)] \text{ where each } J_i(z) = \left(\frac{\partial x_1}{\partial z_i}(z), \dots \frac{\partial x_d}{\partial z_i}(z)\right)^\top$$

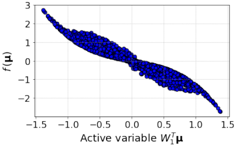
  - First component:

$$L_1 := \sum_{s=1}^{M} \sum_{i=1}^{d} \left[\omega_i \left\langle \frac{J_i(z^{(s)})}{||J_i(z^{(s)})||_2}, \nabla f(x^{(s)}) \right\rangle\right]^2,$$

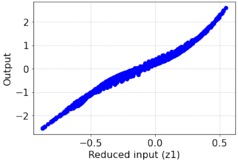
    where $\omega_1, \dots, \omega_d$ are weights (0 or 1) on each dimension.
  - Second component: $L_2 := (\det(J) - 1)^2$
- **Final loss function:** $L := L_1 + \lambda L_2$, where λ is a constant to be chosen by the user that balances the two terms

A Simple Example

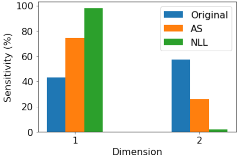
$$f(\boldsymbol{x}) = x_1^3 + x_2^3 + 0.2x_1 + 0.6x_2$$



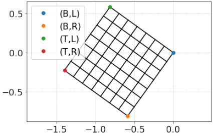
(a) AS plot



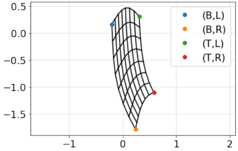
(b) NLL plot



(c) Sensitivity chart



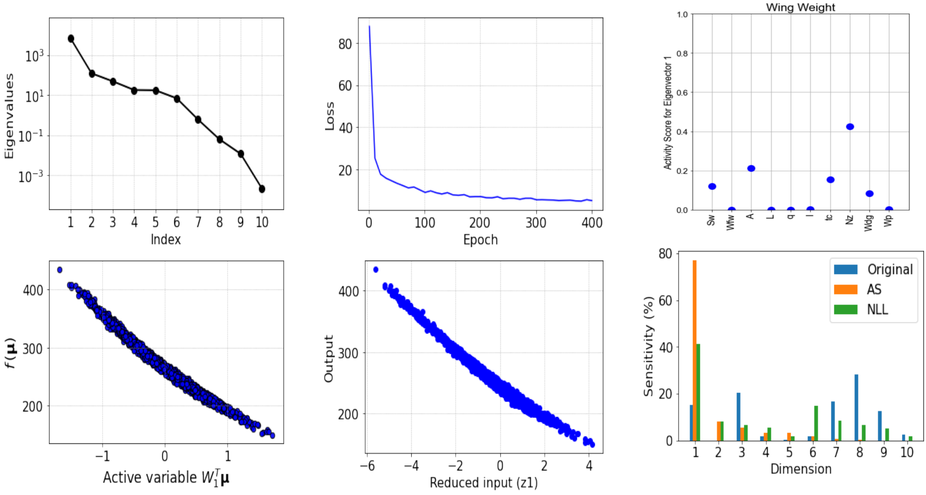
(d) AS transformation



(e) NLL transformation

Real-World Applications

Wing Weight Function

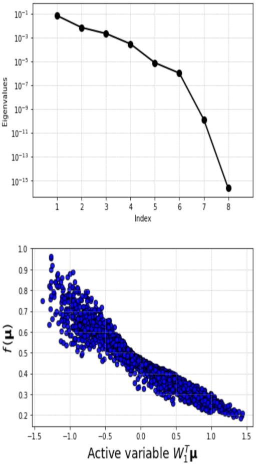


(a) AS method

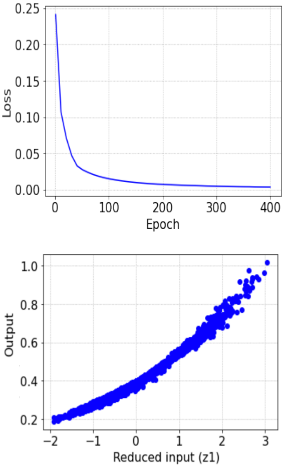
(b) NLL method

(c) Activity score and sensitivity comparison

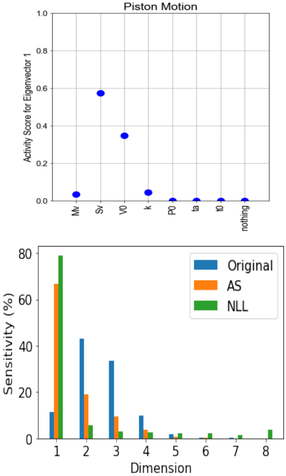
Piston Simulation Function



(a) AS method

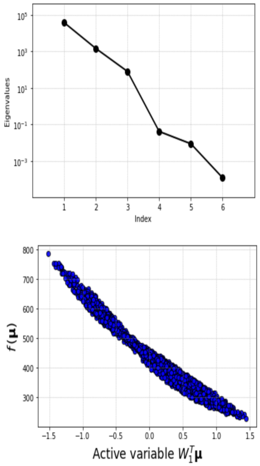


(b) NLL method

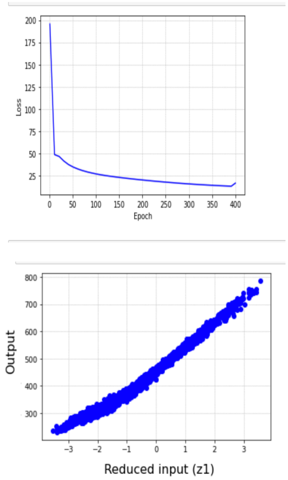


(c) Activity score and sensitivity comparison

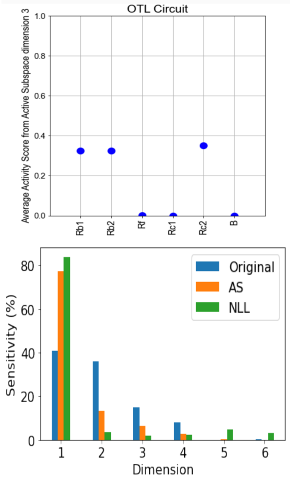
OTL Circuit Function



(a) AS method



(b) NLL method



(c) Activity score and sensitivity comparison

Conclusion

Outcomes

- Understood the theoretical development of the AS and NLL methods for dimension reduction
- Learned to visualize the transformations provided by these methods, evaluate their performance, and determine the benefits and drawbacks of each method in various situations
- Used these methods for real-world applications in engineering

Future Directions

- Writing code to automate the process of finding optimal hyperparameters
  - Could use sensitivity measures to evaluate the performance of the transformation created by certain hyperparameters
- Further reducing computational costs by more efficiently approximating gradients and Jacobians
- Applying AS and NLL methods to vector-valued functions
- Using AS and NLL response surfaces for optimizing a function or estimating its average value

References I

- Paul Constantine, Eric Dow, and Qiqi Wang. "Active Subspace Methods in Theory and Practice: Applications to Kriging Surfaces.". In: *SIAM Journal on Scientific Computing* 36 (2014)
- Paul Constantine. *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*. SIAM Spotlights. Society for Industrial and Applied Mathematics, 2015. ISBN: 9781611973860. URL: https://books.google.com/books?id=AuJ9BwAAQBAJ
- Guannan Zhang, Jiaxin Zhang, and Jacob Hinkle. *Learning nonlinear level sets for dimensionality reduction in function approximation*. 2019. DOI: 10.48550/ARXIV.1902.10652. URL: https://arxiv.org/abs/1902.10652
- Francesco Romor, Marco Tezzele, and Gianluigi Rozza. "ATHENA: Advanced Techniques for High dimensional parameter spaces to Enhance Numerical Analysis". In: *Software Impacts* 10 (2021), p. 100133. DOI: 10.1016/j.simpa.2021.100133

References II

- Paul Constantine. "Active Subspaces". In: *Software Impacts* (2010). DOI: 10.5281/zenodo.158941
- Sonja Surjanovic and Derek Bingham. "Virtual Library of Simulation Experiments: Test Functions and Datasets". In: (2013). URL: http://www.sfu.ca/~ssurjano/wingweight.html
- Sonja Surjanovic and Derek Bingham. "Virtual Library of Simulation Experiments: Test Functions and Datasets". In: (2013). URL: https://www.sfu.ca/~ssurjano/piston.html
- Sonja Surjanovic and Derek Bingham. "Virtual Library of Simulation Experiments: Test Functions and Datasets". In: (2013). URL: https://www.sfu.ca/~ssurjano/otlcircuit.html
- Paul Diaz et al. "A Modified SEIR Model for the Spread of Ebola in Western Africa and Metrics for Resource Allocation". In: *Cornell University* 3 (2016), p. 28. DOI: 10.48550/arXiv.1603.04955