Gradient-Based Recovery of Linear Projections in Multi-Index Models

Mauro Maggioni[1], M. Patrick Martin[1,2]

[1] Department of Mathematics, Johns Hopkins University, [2]mpmartin@jhu.edu

Summary

- Consider the regression problem
$$y_i = f(x_i) + \varepsilon_i$$
- Under mild assumptions on ε and x, the well-known "curse of dimensionality" implies that if f is α-Hölder regular and x lives in D-dimensional space, then the worst case L^2 approximation error decays like $n^{\frac{-\alpha}{2\alpha+D}}$
- However, if f had low dimensional structure, i.e. if there existed a $d \times D$ matrix A with orthonormal rows and function g such that
$$f(x) = g(Ax)$$
Then, if one could recover A, one should hope to be able to attain a faster convergence rate without exponential dependence on D.

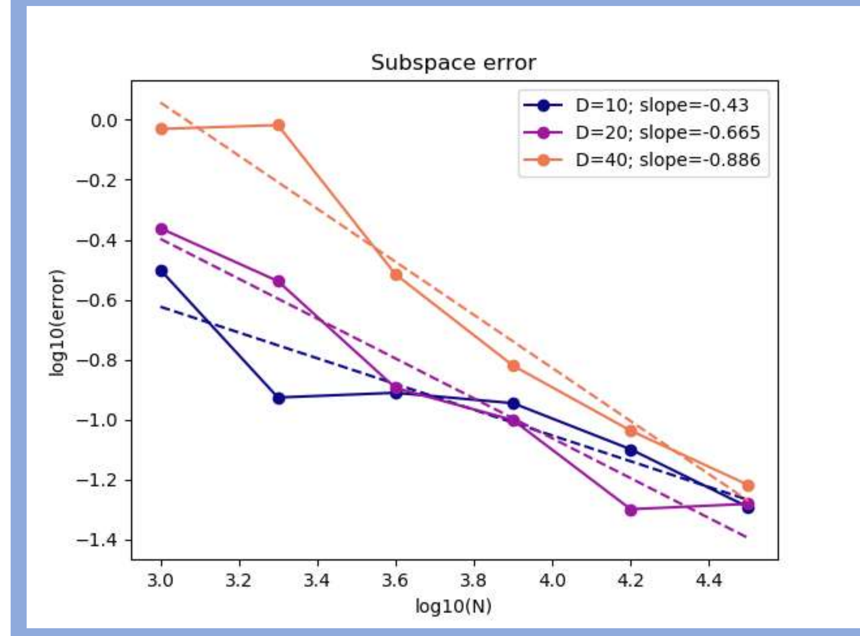
The Algorithm

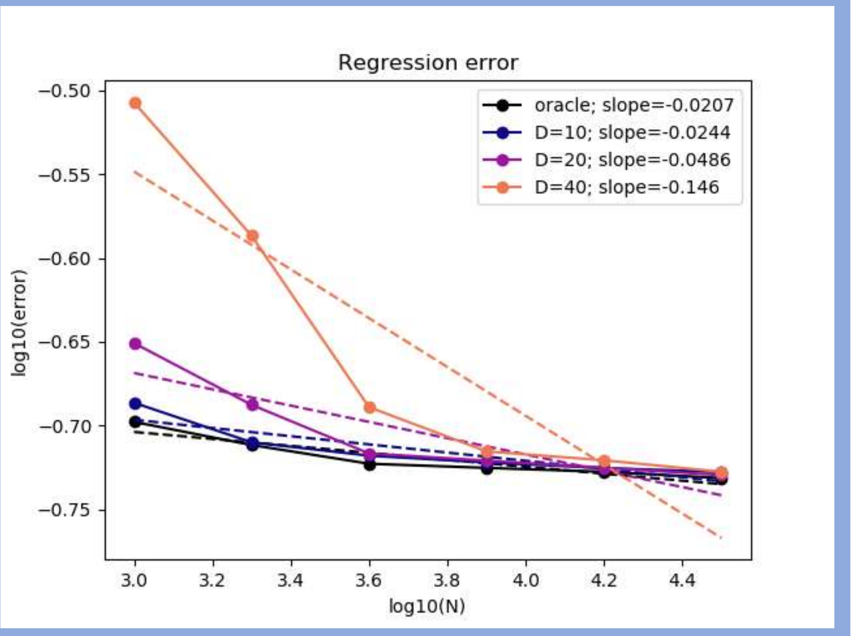
- The gradients of f must all lie in the image of A, as
$$\nabla f(x) = \nabla g(Ax) A$$
- Thus, an $N \times D$ matrix B with the estimated gradients of f as rows should concentrate around A.
- Taking the top d right singular vectors of B should result in an $\hat{A}$ moderately close to A
- This process can be repeated, with an improved estimate of A aiding the estimation of gradients, and thus improving the accuracy of $\hat{A}$
- Finally, use the learned projection in your regression algorithm of choice

Numerical Experiments

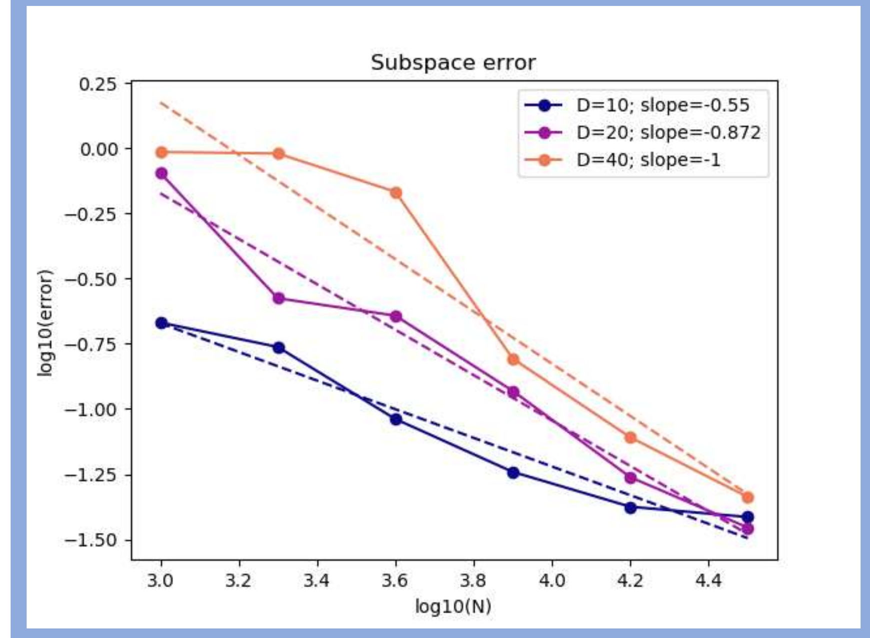
- There are two outcomes to track: the largest angle between the regressed and true subspaces, and the final regression error (using 5-nearest-neighbor regression, normalized by variance).
- For the L^2 regression error, we also plot the normalized error when regressing using the true projection ("oracle").
- In both cases, $\vec{x} \in [-1,1]^D$

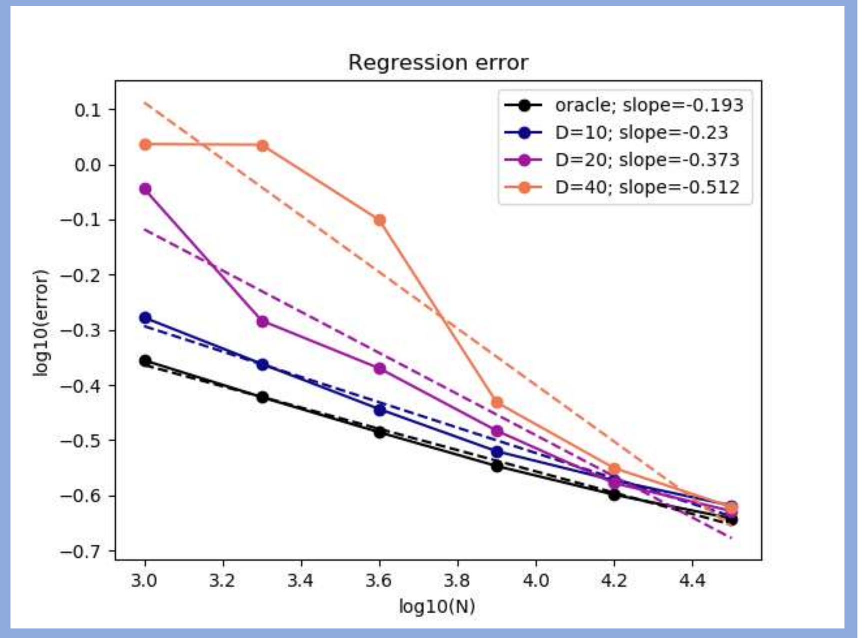
$$g(z) = \sin\left(\frac{\pi}{6} z_1 - \frac{\pi}{2}\right) + z_2$$





$$g(z) = |z_1| + |z_2| + |z_3| + |z_4|$$





Pseudocode

Input: $\{(x_i, y_i)\}_{i=1}^N$, $d \in \mathbb{Z}^+$, $\hat{A}_0 \in \mathbb{R}^{d \times D}$
while not converged:
  $B_k = \boldsymbol{gradient}(x_k, y_k, \hat{A}_t, \{(x_i, y_i)\}_i)$
  $\hat{A}_{t+1} = \boldsymbol{svd\_right}(B)[:d]$
  $t = t + 1$
Output: $\hat{A}_t$

Sketch of Theory

- Gradient computation is done by solving the weighted-least squares problem
$$\Delta y_{k,i} = \langle \Delta x_{k,i}, \hat{\nabla} f(x_k) \rangle + \xi_{k,i}$$
on the m_t datapoints that minimize
$$\left\| \hat{A}_t (x_i - x_k) \right\|_2$$
where $m_0 \approx N$ and decreases to an appropriate value as $\hat{A}_t$ becomes more accurate.
- The distribution of $\hat{\nabla} f(x_i)$ has two main contributions to its norm:
  - First, the true solution to the least-squares problem $\tilde{\nabla} f(x_i)$, which lies in A
  - Second, the variance of the estimate, which is concentrated in $\hat{A}_t$
- As $\hat{A}_t$ aligns with A, these reinforce and further improve the projection accuracy

Acknowledgements

Johns Hopkins University
Maryland Advanced Research Computing Center