How rotation invariant algorithms are fooled by noise on sparse targets

Manfred K. Warmuth — *Google*
Wojciech Kotłowski — *Poznań Univ. of Tech., Poland*
Matt Jones — *Google & Univ. of Colorado Boulder*
Ehsan Amid — *Google*

Google Research · Google DeepMind · University of Colorado Boulder

ALT 2025, Milan.

Summary

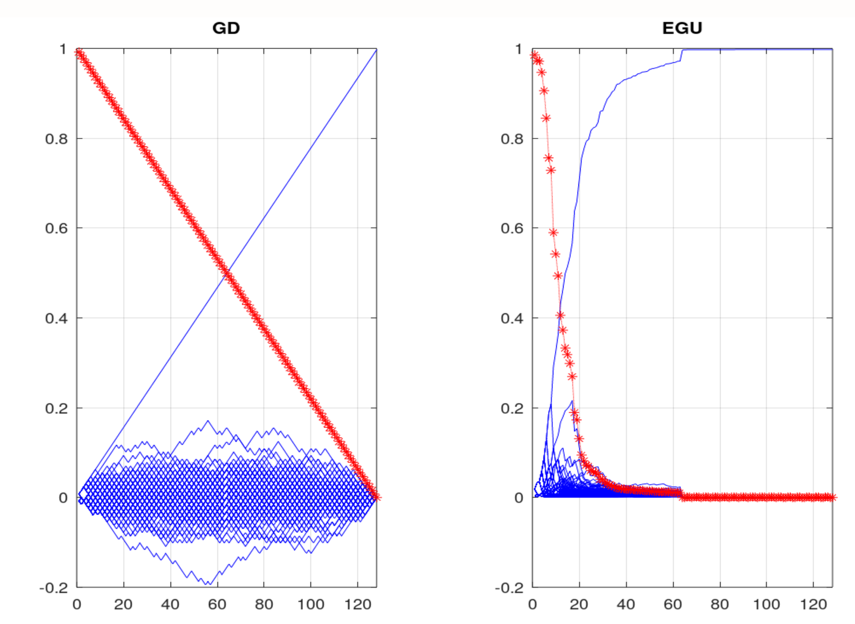
- It is known that rotation invariant algorithms are sub-optimal for sparse linear problems, when # examples $n <$ input dim. d
- We show that when noise is added to this sparse problem, rot.-inv. algorithms still sub-optimal after seeing $n > d$ examples
- We prove much better upper bounds on the same problem for a large variety of algorithms that are non-invariant by rotations.
- We analyze the gradient flow trajectories of learning algorithms

Underconstrained case ($d > n$)

$$\begin{array}{c} \boldsymbol{x}_1 \rightarrow \\ \boldsymbol{x}_2 \rightarrow \\ \boldsymbol{x}_3 \rightarrow \\ \boldsymbol{x}_4 \rightarrow \end{array} \underbrace{\begin{pmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{pmatrix}}_{d \times d \text{ Hadamard matrix } \boldsymbol{H}} \underbrace{\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}}_{\text{sparse target } \boldsymbol{e}_i} = \underbrace{\begin{pmatrix} +1 \\ +1 \\ -1 \\ -1 \end{pmatrix}}_{\text{labels } \boldsymbol{y}} \begin{array}{c} \leftarrow y_1 \\ \leftarrow y_2 \\ \leftarrow y_3 \\ \leftarrow y_4 \end{array}$$

Algorithm receives $n < d$ examples and predicts labels for the remaining examples
Evaluated by the average squared error loss on all d examples

GD fooled by sparse Hadamard problem ($d = 128$)



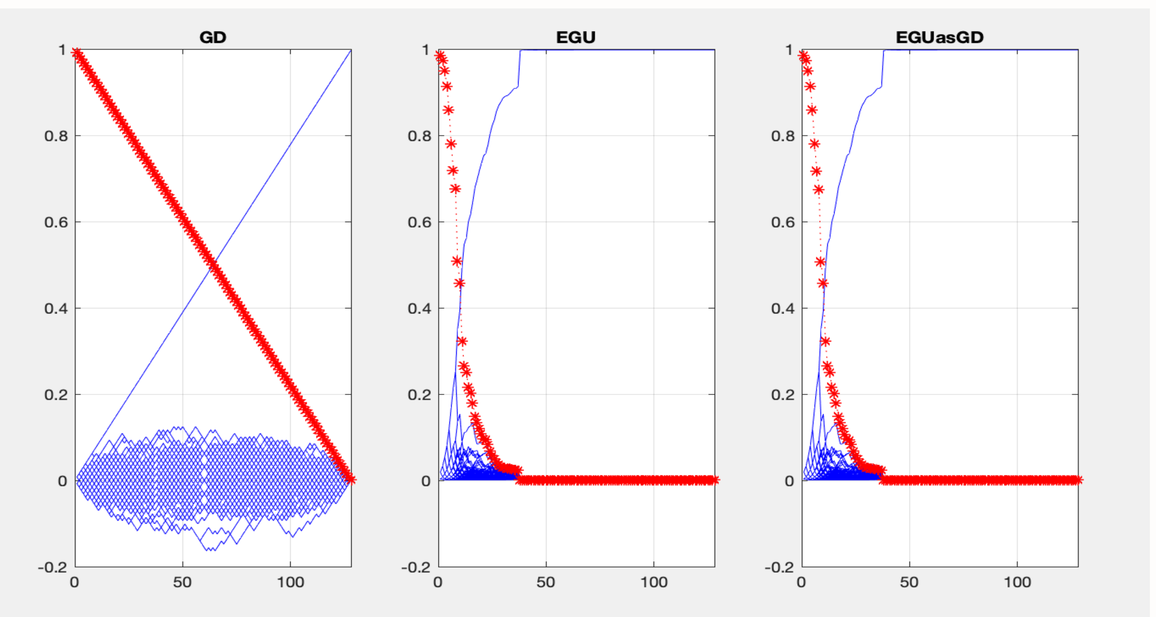
Average loss of Gradient Descent (GD) $1 - \frac{n}{d}$ after n examples
(GD predicts 0 on unseen)

Average loss of Exponentiated Gradient alg. (EGU) $O(\frac{\log d}{n})$

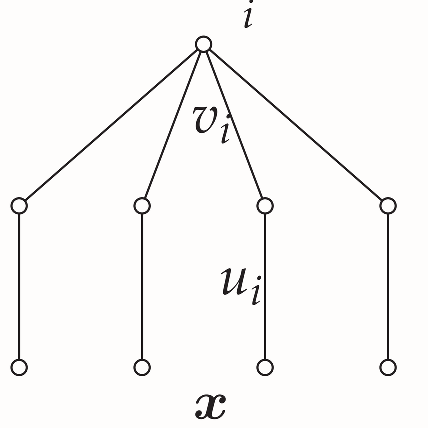
Essentially same on random $\pm$ matrices

To handle sparsity you can stick with GD

Surprise: GD on simple two-layer linear net (called "spindly") simulates EGU and cracks Hadamard problem [A. & W., 2020]



$$f(\boldsymbol{x}) = \sum_i v_i u_i x_i$$



Fooling goes hand in hand with rotation invariance

Algorithm is *rotation-invariant*, if predictions unchanged after rotating

$$\hat{y}(\underbrace{\boldsymbol{U}\boldsymbol{x}}_{\text{test}} \,|\, \underbrace{(\boldsymbol{X}\boldsymbol{U}^\top, \boldsymbol{y})}_{\text{train}}) = \hat{y}(\underbrace{\boldsymbol{x}}_{\text{test}} \,|\, \underbrace{(\boldsymbol{X}, \boldsymbol{y})}_{\text{train}})$$

Examples: linear, logistic regression, any neural network with fully connected bottom layer trained by GD

Theorem [A. et al, ALT 2021]
Any rotation invariant algorithms has average square loss $1 - \frac{n}{d}$ after n examples on Hamadard problem*

*after flipping the rows by $\pm$ random signs, or choosing the target column at random

So what – who cares about underconstrained case

In most applications, # of examples > input dimension!
All previous work becomes vacuous when $n > d$

Main contribution: **In overconstrained case, all rotation invariant algorithms still fooled when noise is added to the sparse targets** (by factor of d suboptimal)

$$\underset{n}{\boldsymbol{y}} = \underset{n,d}{\boldsymbol{X}} \boldsymbol{e}_i + \underset{n}{\boldsymbol{\xi}}, \qquad \boldsymbol{\xi} \sim N(0, \sigma^2 \underset{n}{\boldsymbol{I}})$$

$\boldsymbol{X}$ – matrix with orthogonal rows or drawn from rotationally symmetric distribution

Algorithms evaluated by their excess risk relative to $\boldsymbol{e}_i$

$$\mathbb{E}\left[(\hat{y} - \boldsymbol{x}_{te}^\top \boldsymbol{e}_i)^2\right], \text{ where } \boldsymbol{x}_{te} \text{ random row/sample and random noise}$$

Lower bound

Theorem: The expected error of any rotation-invariant learning algorithm is at least

$$\frac{d-1}{d} \frac{\sigma^2}{\sigma^2 + n/d} \qquad (\text{with fixed } \sigma, \text{ error} \sim d/n)$$

Proof essentially by a Bayesian argument:

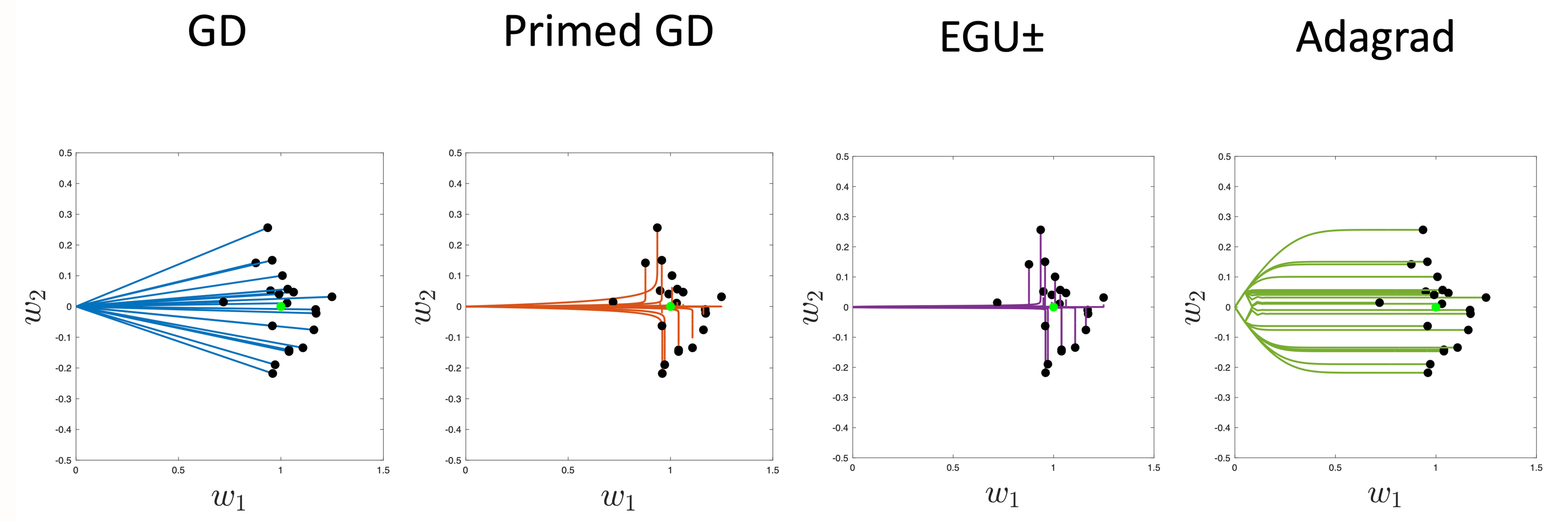
- Target vector $\boldsymbol{w}^\star$ drawn uniformly from a unit sphere
- Lower bound for any algorithm by bounding the error of the optimal (Bayesian) algorithm
- Due to rotation symmetry of the input distribution, rotation invariant algorithms have the same error for every target $\boldsymbol{w}^\star$, in particular $\boldsymbol{w}^\star = \boldsymbol{e}_1$.

Upper bounds

For versions of EGU, and spindly:
with early stopping (crucial), the error decreases as $\sim \frac{\log d}{n}$:
($\frac{d}{\log d}$ faster than rotation-invariant algorithms)

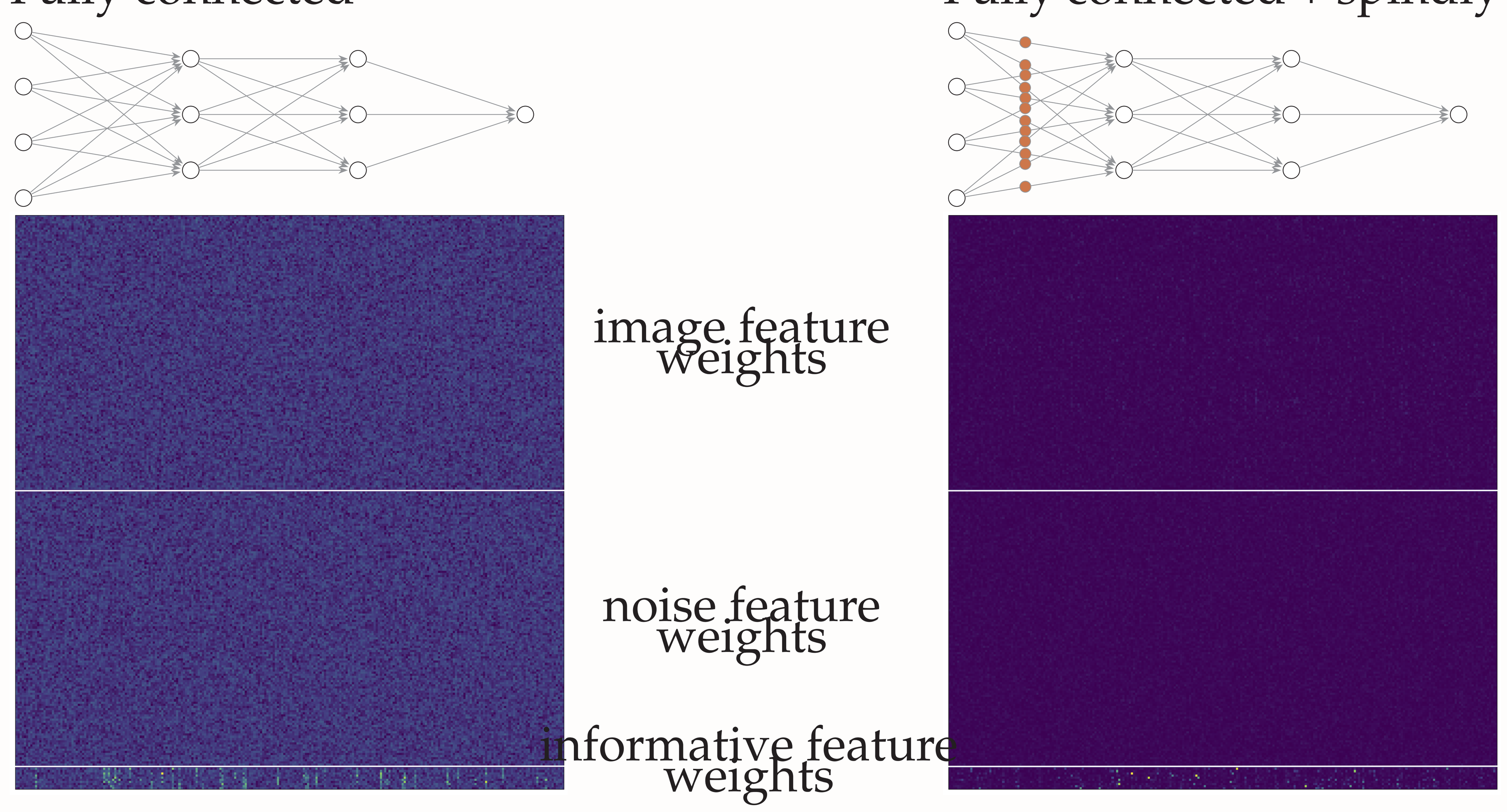
- Many technical details
- New alg. called "priming GD" does not have the $\log d$ factor
- Conjecture: they all don't have this factor
- Similar upper bound with $\log d$ factor known for Lasso

Gradient flow trajectories: $d = 2$



- Analytic solutions to ODEs for continuous algorithms
- GD and rotation invariant algorithms go straight to LS solution
- EGU and relatives biased toward sparse solutions
- Adagrad and relatives biased toward dense solutions

Fashion MNIST experiments



Test accuracy:	Fully conn.	Spindly
only image features	85%	85%
image + noise features	71%	85%
image + noise + informative	98%	100%