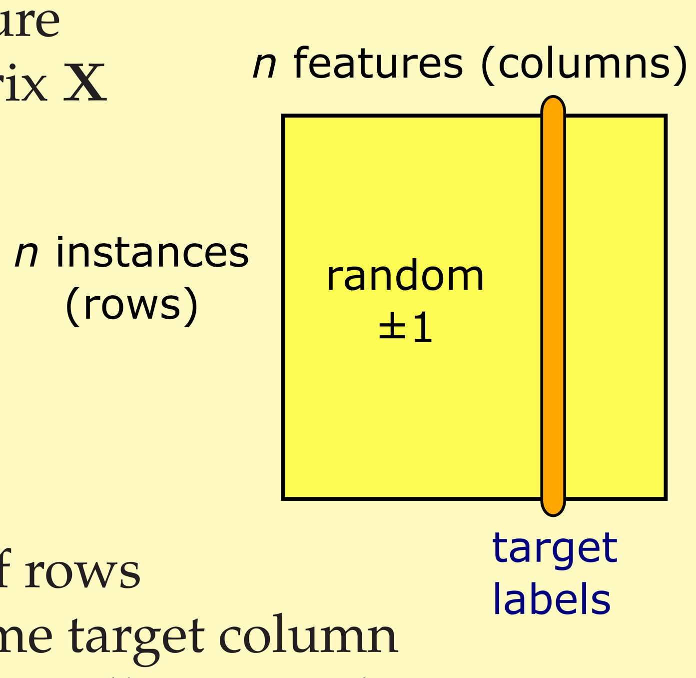
The limits of squared Euclidean distance regularization

UNIVERSITY OF CALIFORNIA SANTA CRUZ

Michał Dereziński   Manfred K. Warmuth

A simple problem

Learn single feature
of a random matrix $\mathbf{X}$

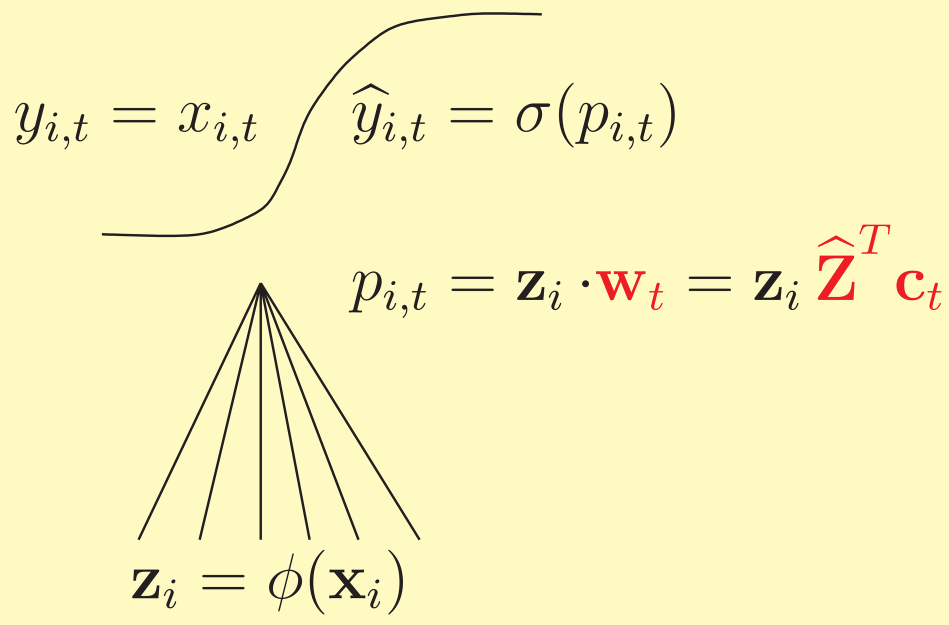


Train on subset of rows
- labeled with some target column
- loss averaged over all n examples

Sparse & linear:
- unit vector $\mathbf{e}_i$ picks out ith feature

Hard for any **kernelizable algorithm**

Prediction matrix



$\mathbf{Z} \in \mathbb{R}^{n\times q}$ is the embedded instances
$\widehat{\mathbf{Z}} \in \mathbb{R}^{k\times q}$ is the training subset of size k
$\mathbf{C} \in \mathbb{R}^{k\times n}$ is linear combination coefficients

Prediction matrix $\mathbf{P} = \mathbf{Z}\widehat{\mathbf{Z}}^T\mathbf{C}$ has rank at most k.

Loss function

Loss averaged over all instances and all targets:

$$\frac{1}{n^2}\sum_{i,t} L(p_{i,t}, x_{i,t}),$$

Previous work. If L is the square loss, use SVD spectrum $s_1, ..., s_n$ of $\mathbf{X}$ and the rank k of $\mathbf{P}$:

$$\frac{1}{n^2}\|\mathbf{P}-\mathbf{X}\|_F^2 \geq \frac{1}{n^2}\sum_{i=k+1}^{n} s_i^2.$$

Our contribution. More general family of **C-regular** loss functions:

There is a constant $C > 0$ such that given $y \in \{1,-1\}$ and $p \in \mathbb{R}$, if $py < 0$, then $L(p, y) \geq C$.

Characterization of algorithms

Examples $(\mathbf{x}_t, y_t)$

Prediction is $\widehat{y} = \sigma(\mathbf{w}\cdot\mathbf{x})$ and
$\mathbf{w}$ = [*linear combination of training instances*]
(i.e. kernelizable)

Such as:

Gradient Descent with $\|\mathbf{w}\|_2^2$ regularization on

- Square loss (linear regression),
- Logistic loss (logistic regression),
- Hinge loss (SVM).

Our approach

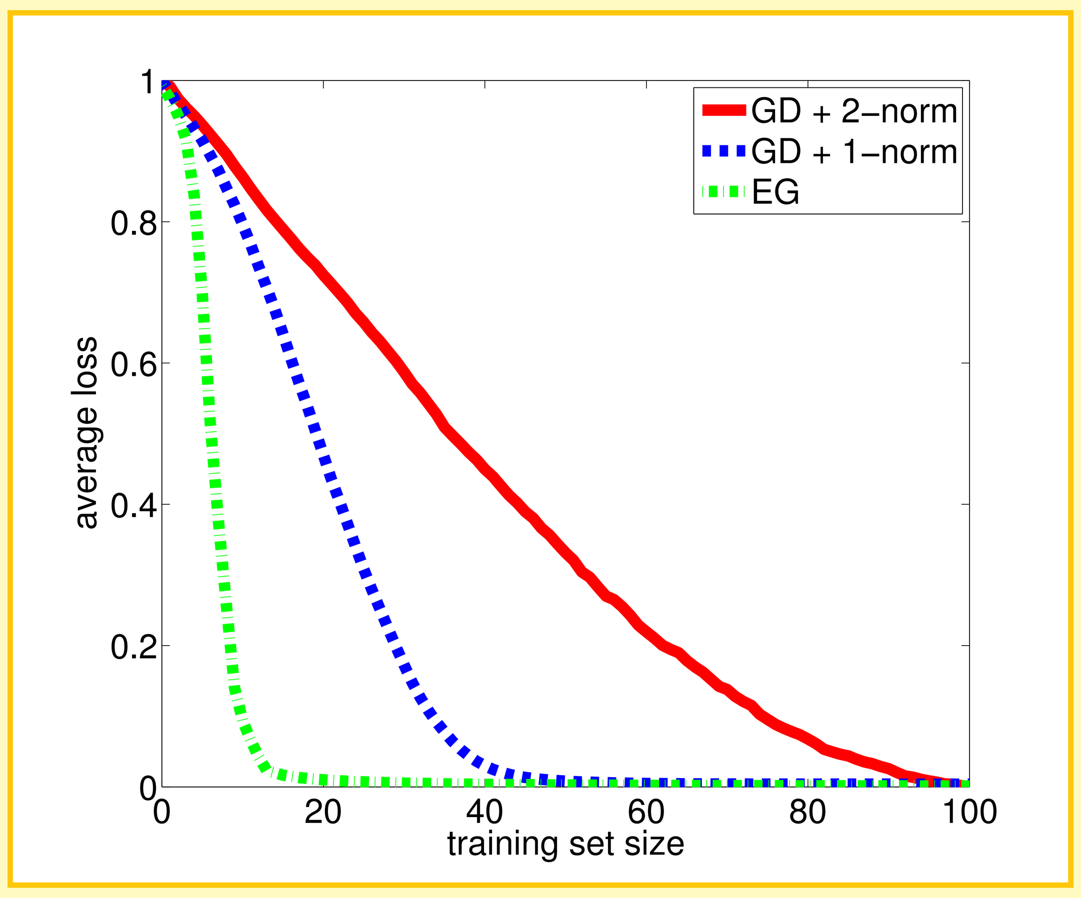
We analyze the number of **sign errors** in the linear prediction. Each one incurs loss at least C.

Counting arguments show that a low-rank matrix $\mathbf{P}$ will have a large number of sign-errors.

A **linear lower bound** is obtained.

Main Theorem
Let L be a C-regular loss function. A random $n\times n$ data matrix $\mathbf{X}$ almost certainly has the property that for any kernelized algorithm, the average loss L after observing k instances is at least $4C\left(\frac{1}{20}-\frac{k}{n}\right)$.

Hardness



The problem is easy - **VC dimension** is $\log n$

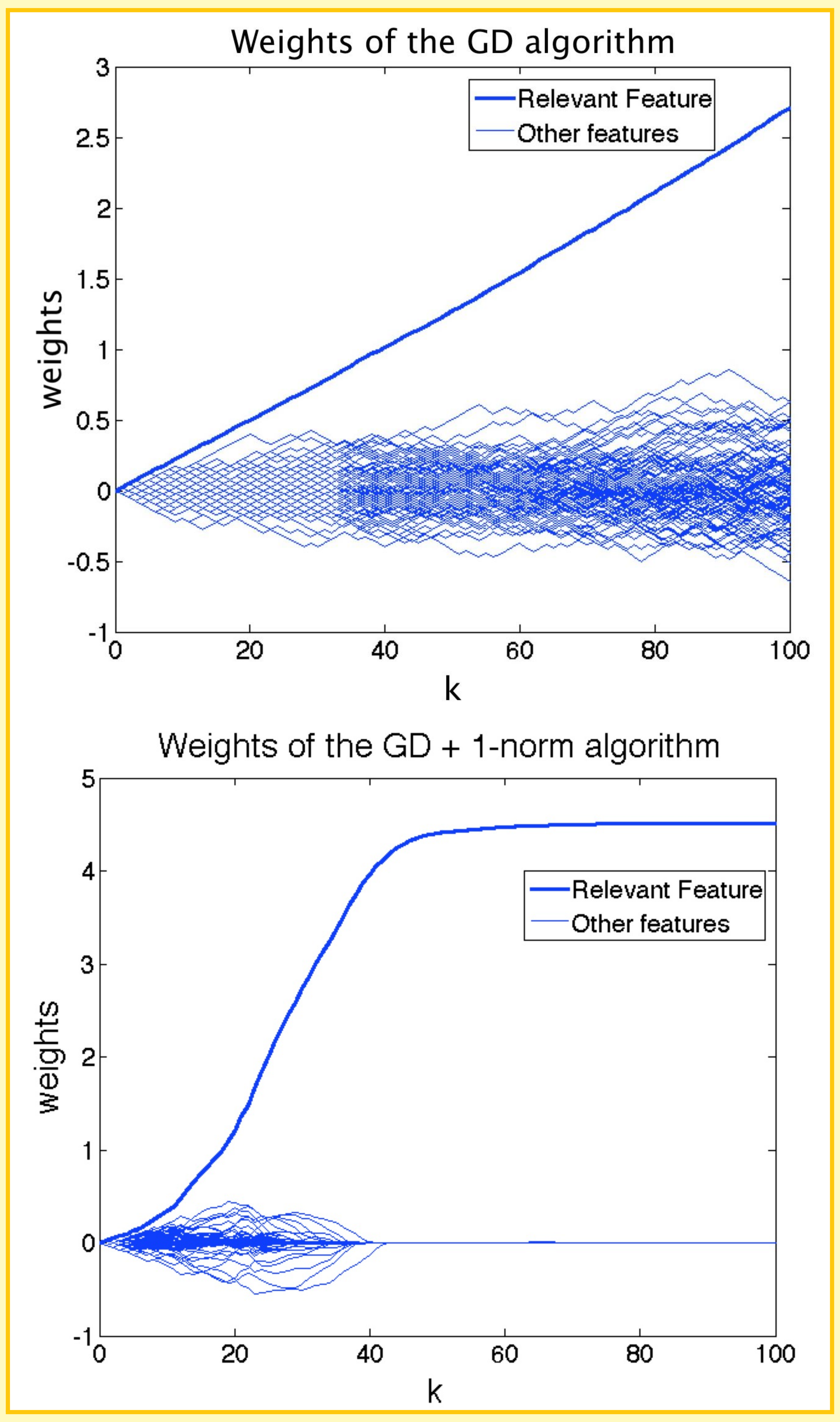
Good algorithms:

1. GD with **1-norm regularization**,
2. **Exponentiated Gradient** algorithm

We show that any **kernelizable algorithm** requires $\Omega(n)$ instances

Kernelization does not help
- hard for **any embedding**
- when averaged over targets as well

Weights plotted



Proof sketch

Data matrix $\mathbf{X}$ is (k, r)-learnable if there exists a prediction matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ of rank $\leq k$ with at most r sign errors

$\mathbf{P}$			sign($\mathbf{P}$)			$\mathbf{X}$		
2	−3	1	+	−	+	+	+	+
5	1	−2	+	+	−	−	+	−
−3	−2	−8	−	−	−	−	+	+

$r = 4$
loss $\geq \frac{4C}{n^2}$

Define:	Number of:
$\text{signs}_n(k)$	$\{\text{sign}(\mathbf{P}) \mid \text{rank}(\mathbf{P}) \leq k\}$
$\text{changes}_n(r)$	patterns from $\leq r$ sign-changes
$\text{easy}_n(k, r)$	(k, r)-learnable data matrices $\mathbf{X}$
$\text{all}_n = 2^{n^2}$	all possible data matrices $\mathbf{X}$

$$\text{easy}_n(k, r) \leq \text{signs}_n(k)\cdot\text{changes}_n(r)$$

$$\text{signs}_n(k) \leq 2^{(3+\log\frac{k}{n})(4kn+n)}$$

$$\text{changes}_n(r) = \sum_{i=0}^{r}\binom{n^2}{i} \leq 2^{H(\frac{r}{n^2})n^2}$$

$$\Downarrow\Downarrow\Downarrow$$

$$\text{easy}_n\left(k, 4n^2\left(1/20 - k/n\right)\right) \ll 2^{n^2} = \text{all}_n$$

$$\text{loss} \geq 4C\left(\frac{1}{20}-\frac{k}{n}\right) \quad \text{for almost all } \mathbf{X}$$

Conjecture: bound holds for deep neural nets

Remains **hard** for any **deep neural net** trained with Gradient Descent + 2-norm regularization

1-norm regularization works fine

Adding **hidden layers** does not help
Changing **transfer function** does not help
Dropout does not help

Only experimental evidence