Learning Strikes Again: the Case of the DRS Signature Scheme

Yang Yu[1] Léo Ducas[2]

[1]Tsinghua University

[2]Centrum Wiskunde & Informatica

Asiacrypt 2018
Brisbane, Australia

This is a cryptanalysis work...

- Target: DRS — a NIST lattice-based signature proposal

This is a cryptanalysis work...

- Target: DRS — a NIST lattice-based signature proposal
- Techniques: learning & lattice

This is a cryptanalysis work...

- Target: DRS — a NIST lattice-based signature proposal
- Techniques: learning & lattice
  - Statistical learning $\Rightarrow$ secret key information leaks

This is a cryptanalysis work...

- Target: DRS — a NIST lattice-based signature proposal
- Techniques: learning & lattice
  - Statistical learning $\Rightarrow$ secret key information leaks
  - Lattice techniques $\Rightarrow$ better use of leaks

This is a cryptanalysis work...

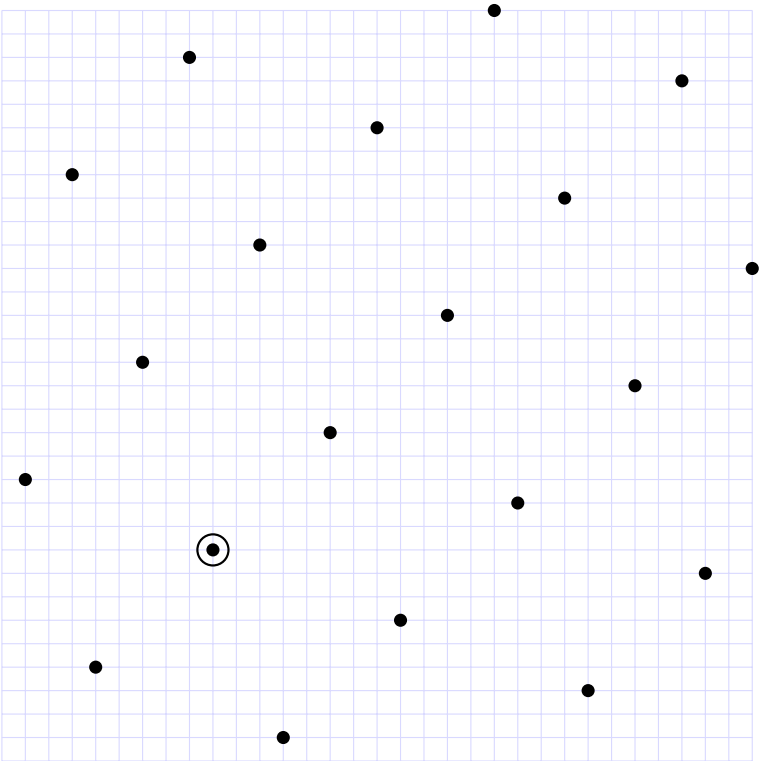
- Target: DRS — a NIST lattice-based signature proposal
- Techniques: learning & lattice
  - Statistical learning $\Rightarrow$ secret key information leaks
  - Lattice techniques $\Rightarrow$ better use of leaks
- They claim that Parameter Set-I offers **at least 128-bits** of security. We show that it actually offers **at most 80-bits** of security!

Outline

1. Background
2. DRS signature
3. Learning secret key coefficients
4. Exploiting the leaks

Outline

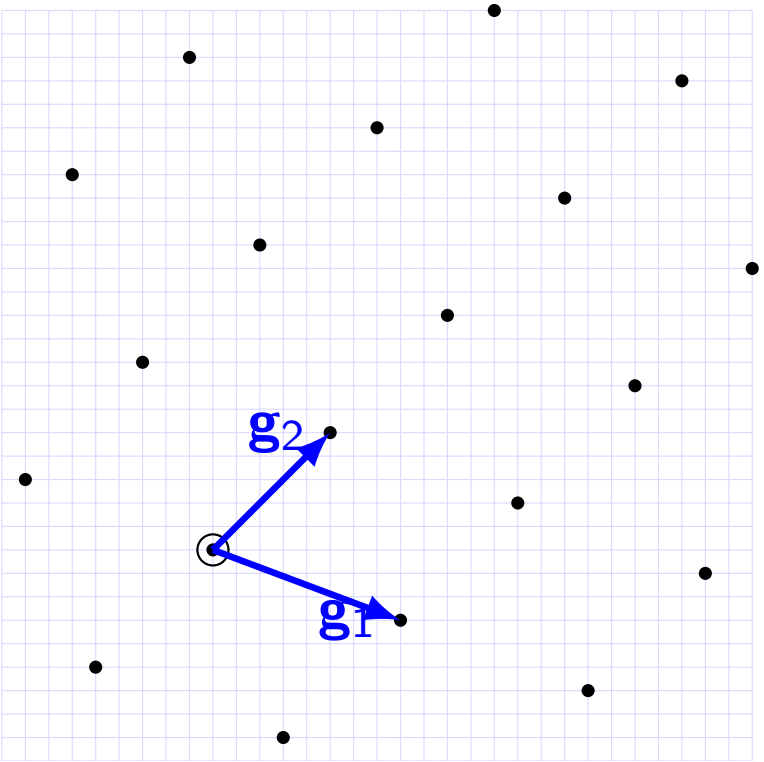
Lattice



Definition

A lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^m$.

Lattice

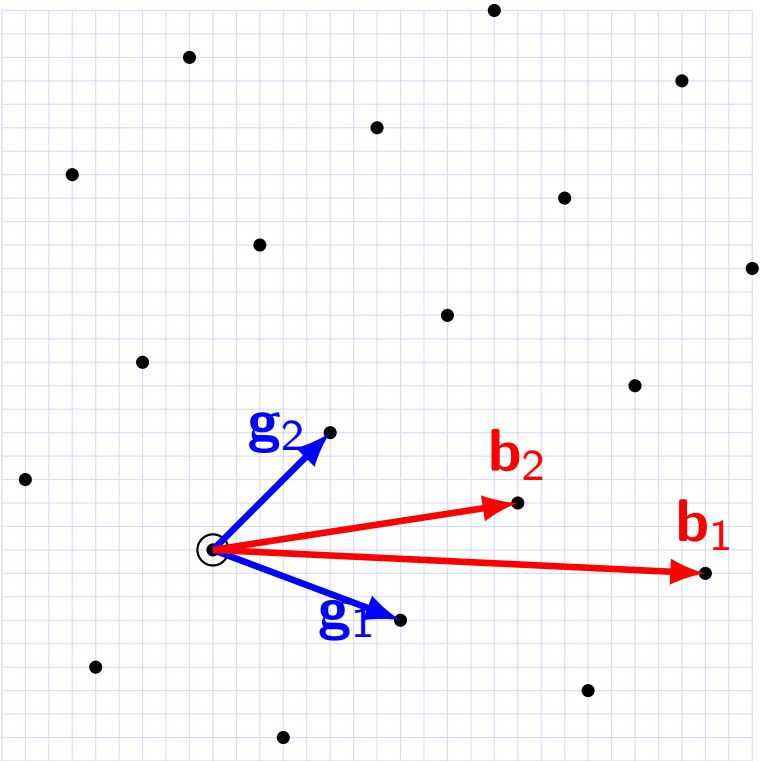


Definition

A lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^m$.

A lattice is generated by its basis $\mathbf{G} = (\mathbf{g}_1, \cdots, \mathbf{g}_n) \in \mathbb{R}^{n \times m}$, e.g. $\mathcal{L} = \{\mathbf{x}\mathbf{G} \mid \mathbf{x} \in \mathbb{Z}^n\}$.

Lattice



Definition

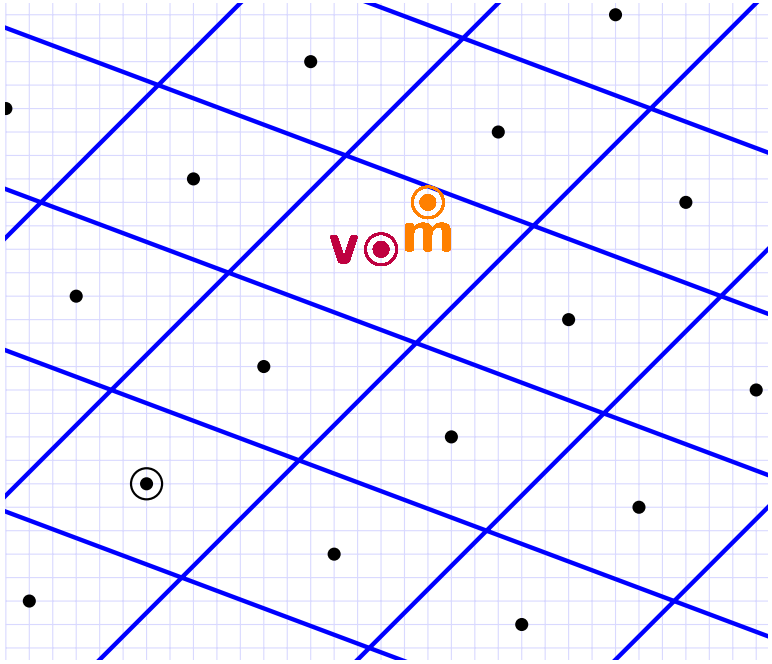
A lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^m$.

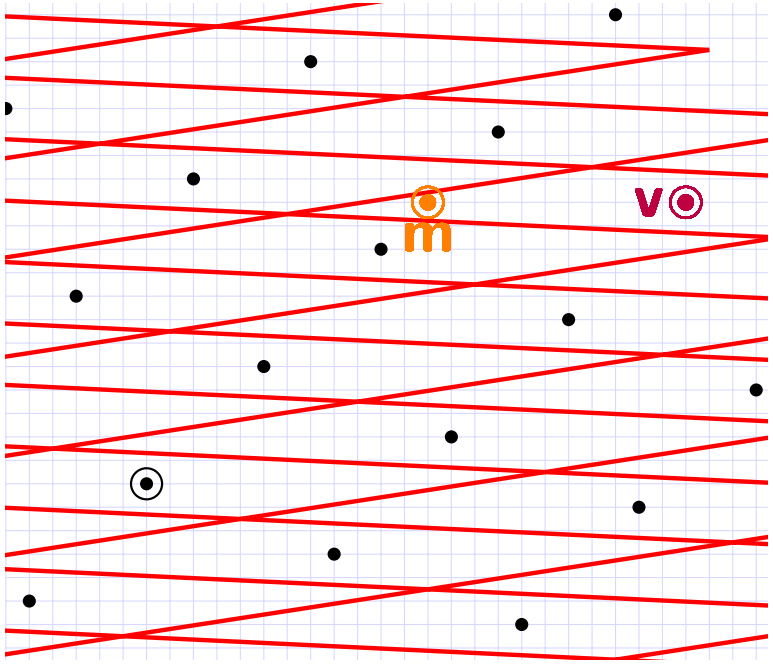
A lattice is generated by its basis $\mathbf{G} = (\mathbf{g}_1, \cdots, \mathbf{g}_n) \in \mathbb{R}^{n \times m}$, e.g. $\mathcal{L} = \{\mathbf{x}\mathbf{G} \mid \mathbf{x} \in \mathbb{Z}^n\}$.

$\mathcal{L}$ has infinitely many bases
$\mathbf{G}$ is good, $\mathbf{B}$ is bad.

Finding Close Vectors

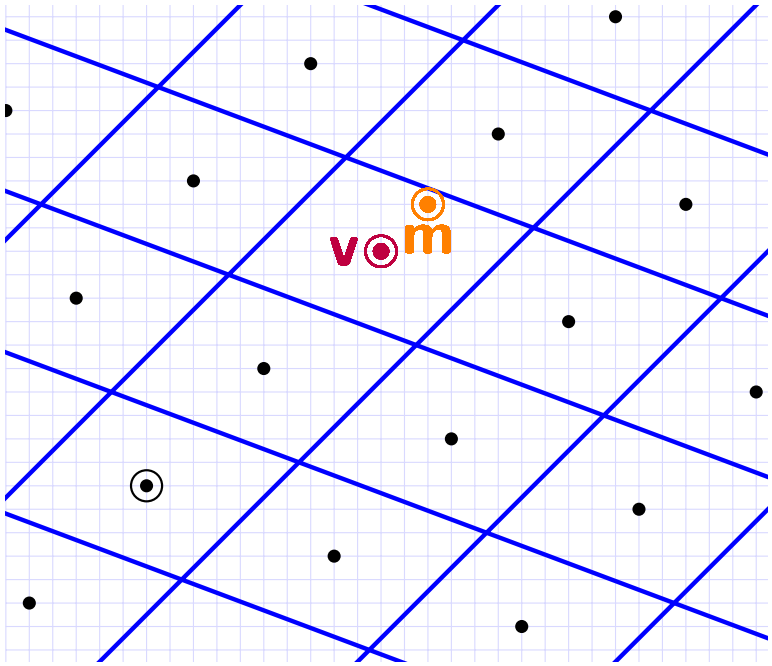
Each basis defines a parallelepiped $\mathcal{P}$.

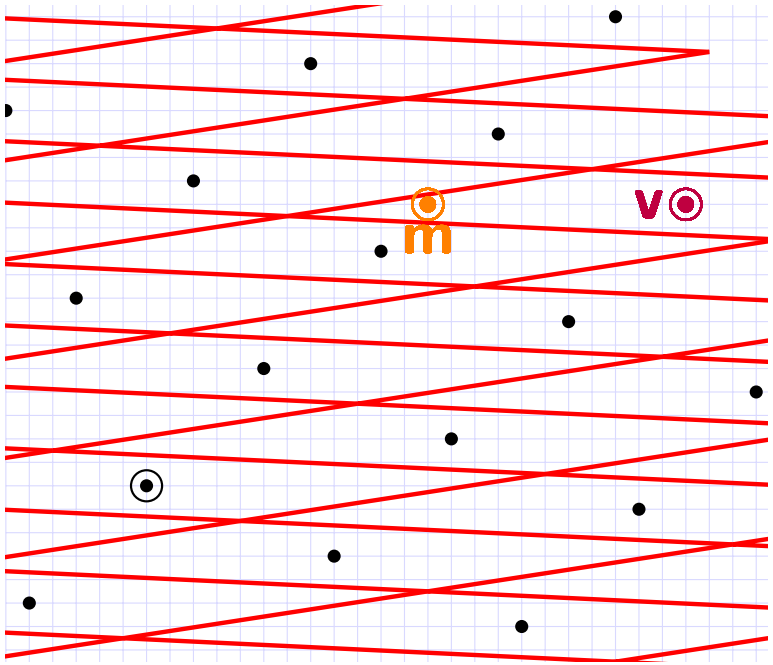




Finding Close Vectors

Each basis defines a parallelepiped $\mathcal{P}$.





Babai's round-off algorithm outputs $\mathbf{v} \in \mathcal{L}$ such that $\mathbf{v} - \mathbf{m} \in \mathcal{P}$.

GGH & NTRUSign Schemes

Public key: $\mathbf{P}$, secret key: $\mathbf{S}$

Sign

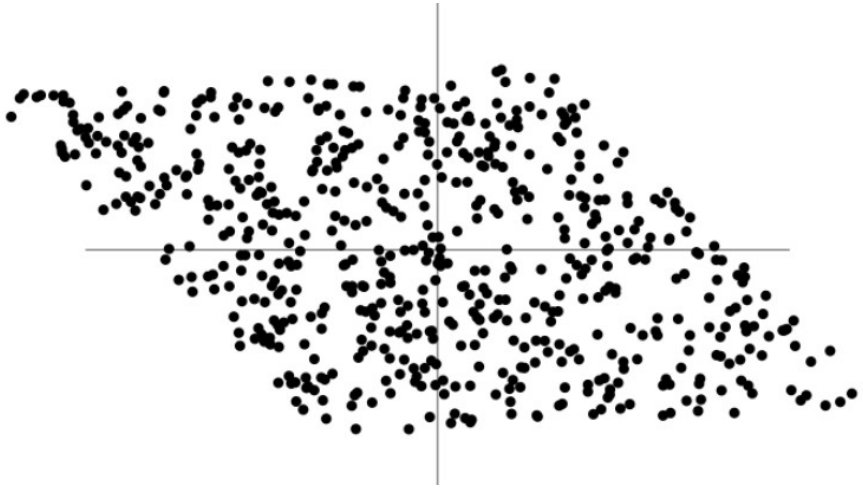
1. Hash the message to a random vector $\mathbf{m}$
2. Round $\mathbf{m}$ (using $\mathbf{S}$) to $\mathbf{v} \in \mathcal{L}$

Verify

1. Check $\mathbf{v} \in \mathcal{L}$ (using $\mathbf{P}$)
2. Check $\mathbf{v}$ is close to $\mathbf{m}$

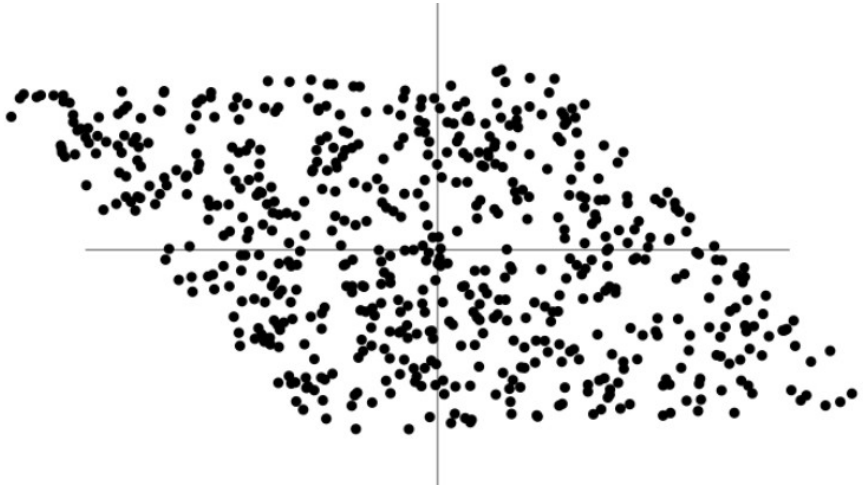
GGH & NTRUSign are insecure!

$\mathbf{v} - \mathbf{m} \in \mathcal{P}(\mathbf{S}) \Rightarrow (\mathbf{v}, \mathbf{m})$ leaks some information of $\mathbf{S}$.



GGH & NTRUSign are insecure!

$\mathbf{v} - \mathbf{m} \in \mathcal{P}(\mathbf{S}) \Rightarrow (\mathbf{v}, \mathbf{m})$ leaks some information of $\mathbf{S}$.



GGH and NTRUSign were broken by "learning the parallelepiped" [NR06].

Some countermeasures were also broken by a similar attack [DN12].

Countermeasures

Let us focus on Hash-then-Sign approach!

Provably secure method [GPV08]

- rounding based on Gaussian sampling
- $\mathbf{v} - \mathbf{m}$ is independent of $\mathbf{S}$

Countermeasures

Let us focus on Hash-then-Sign approach!

Provably secure method [GPV08]

- rounding based on Gaussian sampling
- $\mathbf{v} - \mathbf{m}$ is independent of $\mathbf{S}$

Heuristic method [PSW08]

- rounding based on CVP w.r.t ℓ_∞-norm
- the support of $\mathbf{v} - \mathbf{m}$ is independent of $\mathbf{S}$
- DRS [PSDS17] is an instantiation, submitted to the NIST.

Outline

DRS

DRS = **D**iagonal-dominant **R**eduction **S**ignature

DRS

DRS = **D**iagonal-dominant **R**eduction **S**ignature

Parameters: (n, D, b, N_b, N_1)

- n : the dimension
- D : the diagonal coefficient
- b : the magnitude of the large coefficients (*i.e.* $\{\pm b\}$)
- N_b : the number of large coefficients per row vector
- N_1 : the number of small coefficients (*i.e.* $\{\pm 1\}$) per row vector

$$\mathbf{S} = \begin{pmatrix} D & & & \\ & D & & \\ & & \ddots & \\ & & & D \end{pmatrix} +$$

DRS

DRS = **D**iagonal-dominant **R**eduction **S**ignature

Parameters: (n, D, b, N_b, N_1)

- n : the dimension
- D : the diagonal coefficient
- b : the magnitude of the large coefficients (*i.e.* $\{\pm b\}$)
- N_b : the number of large coefficients per row vector
- N_1 : the number of small coefficients (*i.e.* $\{\pm 1\}$) per row vector

$$\mathbf{S} = \begin{pmatrix} D & & & \\ & D & & \\ & & \ddots & \\ & & & D \end{pmatrix} +$$

$\leftarrow$ "absolute circulant"

Message reduction algorithm

Input: a message $\mathbf{m} \in \mathbb{Z}^n$, the secret matrix $\mathbf{S}$

Output: a reduced message $\mathbf{w}$ such that $\mathbf{w} - \mathbf{m} \in \mathcal{L}$ and $\|\mathbf{w}\|_\infty < D$

1: $\mathbf{w} \leftarrow \mathbf{m}, i \leftarrow 0$
2: **repeat**
3: $\quad \mathbf{w} \leftarrow \mathbf{w} - \lfloor \frac{w_i}{D} \rfloor \cdot \mathbf{s}_i$
4: $\quad i \leftarrow (i+1) \bmod n$
5: **until** $\|\mathbf{w}\|_\infty < D$
6: return $\mathbf{w}$

Message reduction algorithm

Input: a message $\mathbf{m} \in \mathbb{Z}^n$, the secret matrix $\mathbf{S}$
Output: a reduced message $\mathbf{w}$ such that $\mathbf{w} - \mathbf{m} \in \mathcal{L}$ and $\|\mathbf{w}\|_\infty < D$

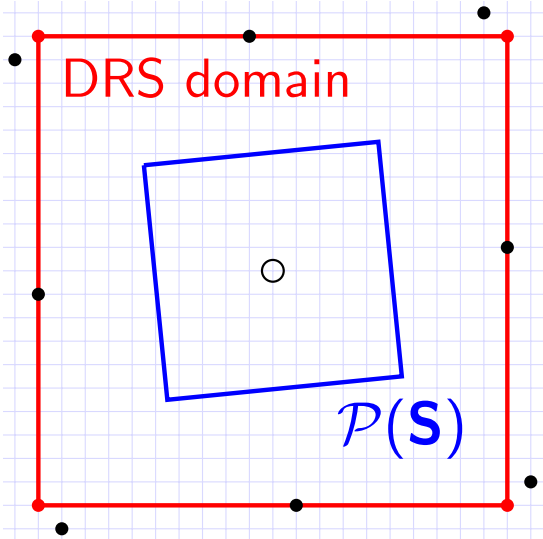
1: $\mathbf{w} \leftarrow \mathbf{m}, i \leftarrow 0$
2: **repeat**
3: $\quad \mathbf{w} \leftarrow \mathbf{w} - \lfloor \frac{w_i}{D} \rfloor \cdot \mathbf{s}_i$
4: $\quad i \leftarrow (i+1) \bmod n$
5: **until** $\|\mathbf{w}\|_\infty < D$
6: return $\mathbf{w}$

Intuition: use $\mathbf{s}_i$ to reduce

- w_i decreases a lot
- for $j \neq i$, w_j increases a bit
- $\|\mathbf{w}\|_1$ is reduced $\Rightarrow$ reduction always terminates!

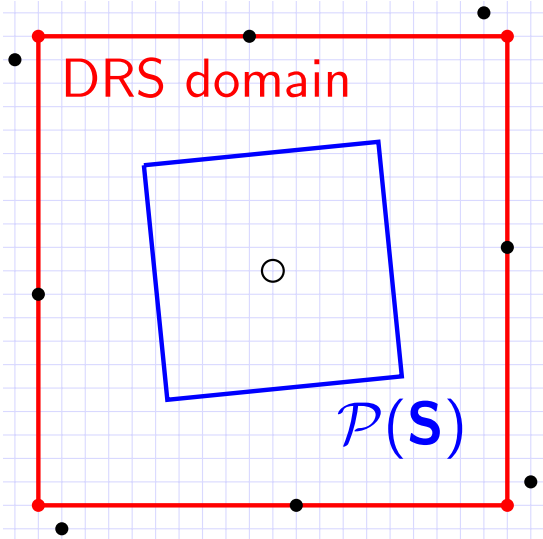
Resistance to NR attack

The support of $\mathbf{w}$: $(-D, D)^n$



Resistance to NR attack

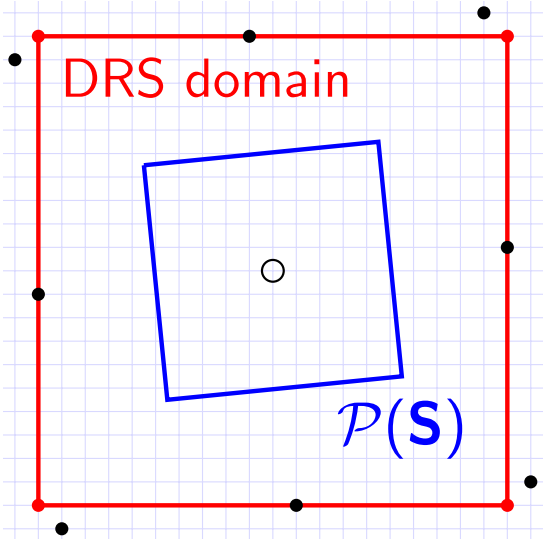
The support of $\mathbf{w}$: $(-D, D)^n$



The support is "zero-knowledge"

Resistance to NR attack

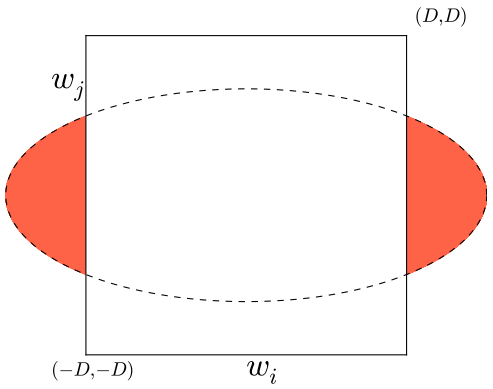
The support of $\mathbf{w}$: $(-D, D)^n$

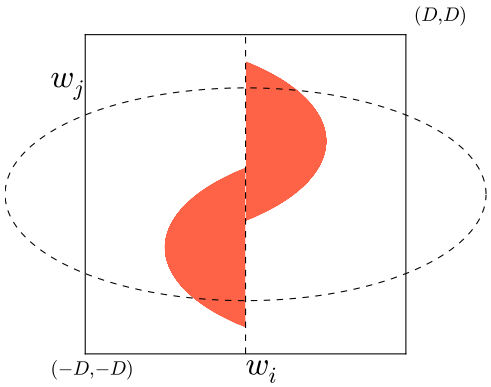


The support is "zero-knowledge", but **maybe the distribution is not!**

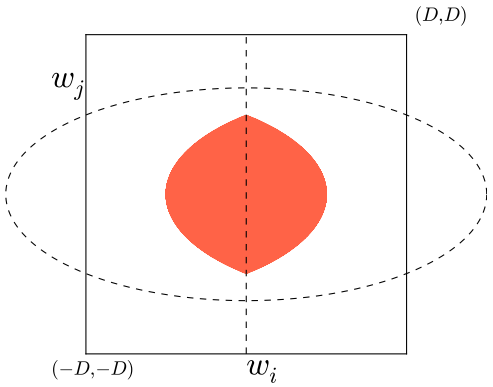
Outline

Intuition

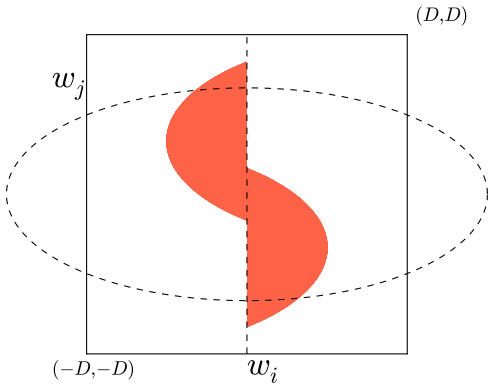




$\mathbf{S}_{i,j} = -b$



$\mathbf{S}_{i,j} = 0$



$\mathbf{S}_{i,j} = b$

Figure out the model

Can we devise a formula $\mathbf{S}_{i,j} \approx f(W_{i,j})$?

Figure out the model

Can we devise a formula $\mathbf{S}_{i,j} \approx f(W_{i,j})$? Seems complicated!

- cascading phenomenon: a reduction triggers another one.
- other parasite correlations

Figure out the model

Can we devise a formula $\mathbf{S}_{i,j} \approx f(W_{i,j})$? Seems complicated!

- cascading phenomenon: a reduction triggers another one.
- other parasite correlations

$\Rightarrow$ **Search for the best linear fit f ?**

Figure out the model

Can we devise a formula $\mathbf{S}_{i,j} \approx f(W_{i,j})$? Seems complicated!

- cascading phenomenon: a reduction triggers another one.
- other parasite correlations

$\Rightarrow$ **Search for the best linear fit f ?**

Search space for all linear f: too large!

Figure out the model

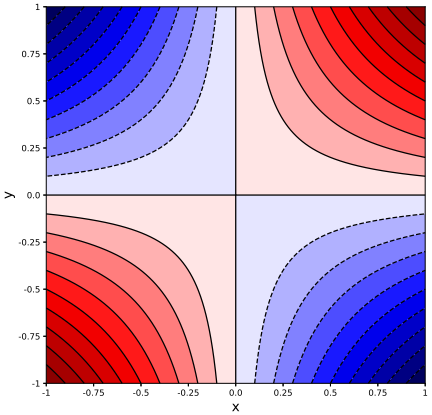
Can we devise a formula $\mathbf{S}_{i,j} \approx f(W_{i,j})$? Seems complicated!

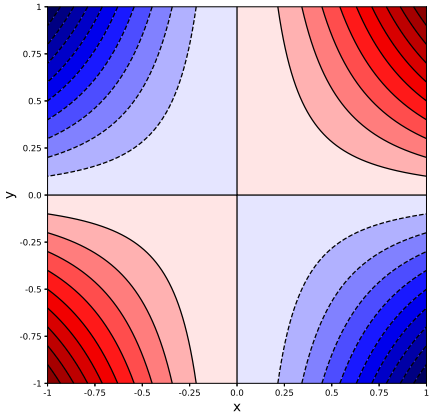
- cascading phenomenon: a reduction triggers another one.
- other parasite correlations

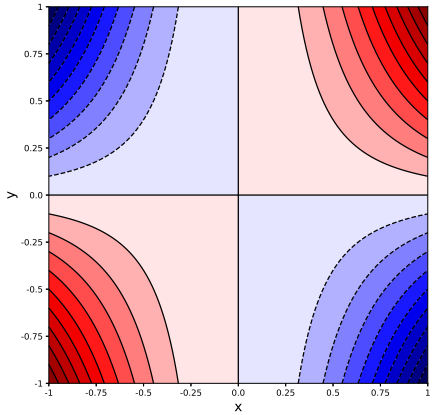
$\Rightarrow$ **Search for the best linear fit f ?**

Search space for all linear f: too large!

$\Rightarrow$ **choose some features $\{f_i\}$ and search in** $\mathrm{span}(\{f_i\})$**, i.e.** $f = \sum x_\ell f_\ell$







Training — feature selection

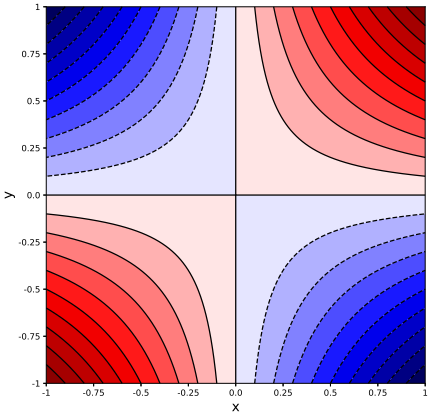
Lower degree moments:

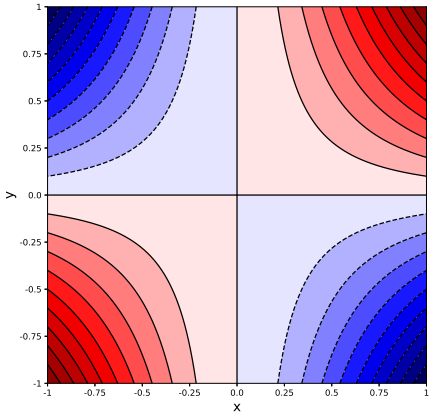
$f_1(W) = \mathbb{E}(w_i w_j)$ $\quad$ $f_2(W) = \mathbb{E}(w_i \cdot |w_i|^{1/2} \cdot w_j)$ $\quad$ $f_3(W) = \mathbb{E}(w_i \cdot |w_i| \cdot w_j)$

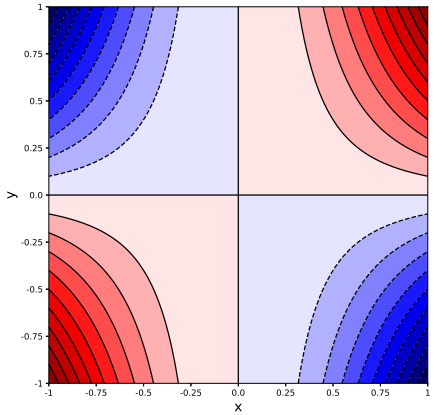
Training — feature selection

Lower degree moments:

$f_1(W) = \mathbb{E}(w_i w_j)$ $\qquad$ $f_2(W) = \mathbb{E}(w_i \cdot |w_i|^{1/2} \cdot w_j)$ $\qquad$ $f_3(W) = \mathbb{E}(w_i \cdot |w_i| \cdot w_j)$

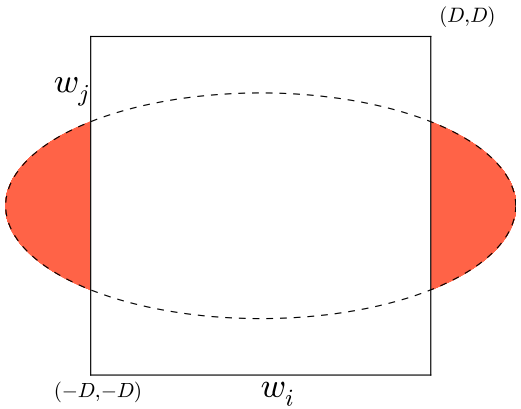


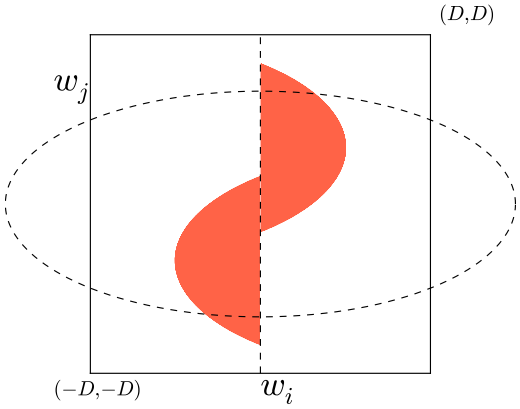




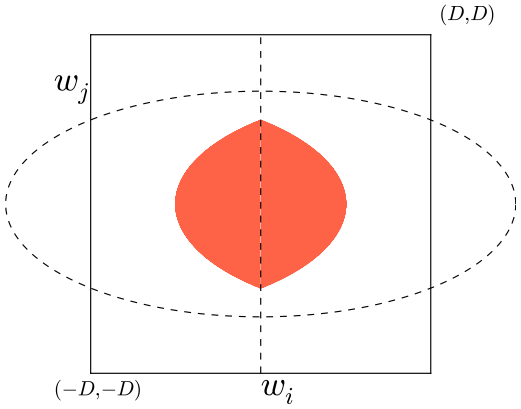
Not enough!

Training — feature selection

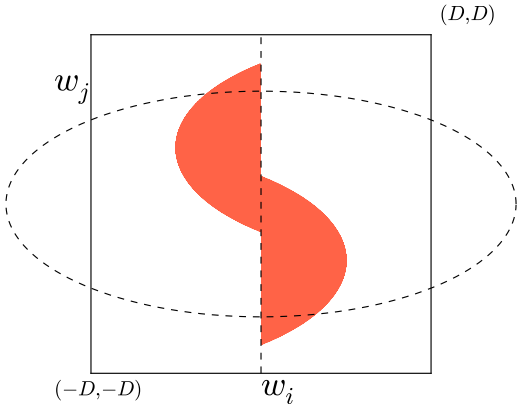




$\mathbf{S}_{i,j} = -b$



$\mathbf{S}_{i,j} = 0$

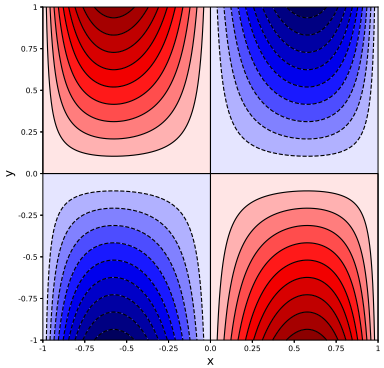


$\mathbf{S}_{i,j} = b$

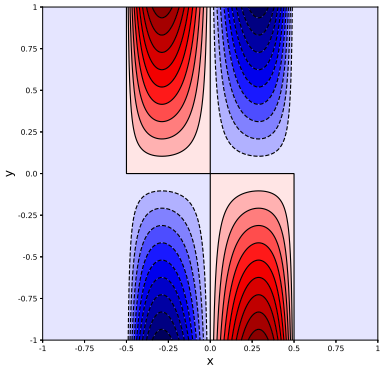
Training — feature selection

Pay more attention to the central region (i.e. $|w_i|$ small).

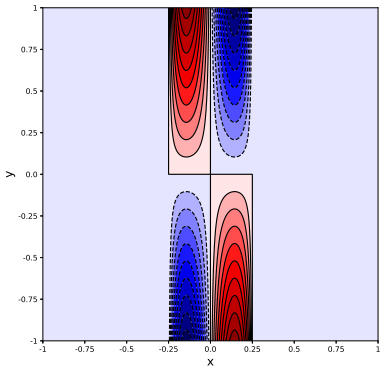
$$f_4 = \mathbb{E}(w_i(w_i - 1)(w_i + 1)w_j)$$



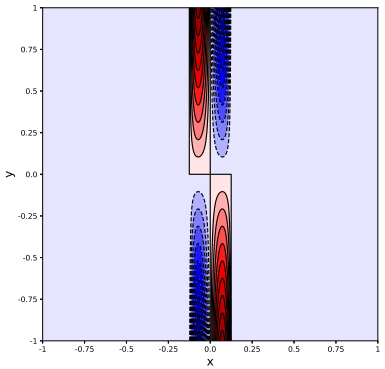
$$f_5 = \mathbb{E}(2w_i(2w_i - 1)(2w_i + 1)w_j \mid |2w_i| \leq 1)$$



$$f_6 = \mathbb{E}(4w_i(4w_i - 1)(4w_i + 1)w_j \mid |4w_i| \leq 1)$$



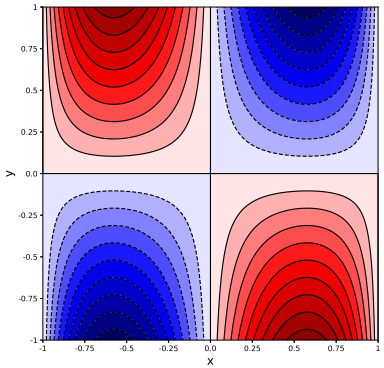
$$f_7 = \mathbb{E}(8w_i(8w_i - 1)(8w_i + 1)w_j \mid |8w_i| \leq 1)$$



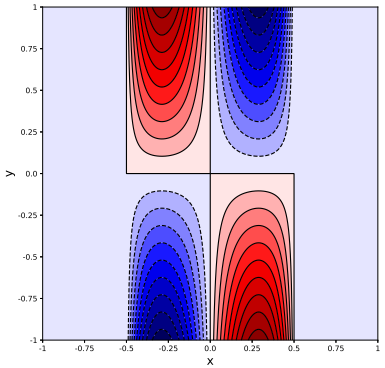
Training — feature selection

Pay more attention to the central region (i.e. $|w_i|$ small).

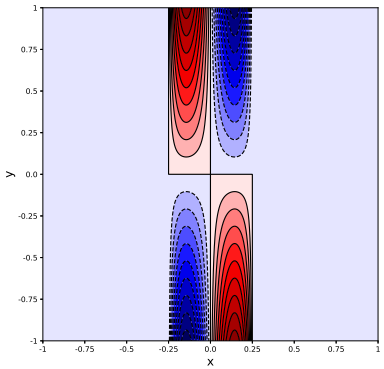
$$f_4 = \mathbb{E}(w_i(w_i - 1)(w_i + 1)w_j)$$



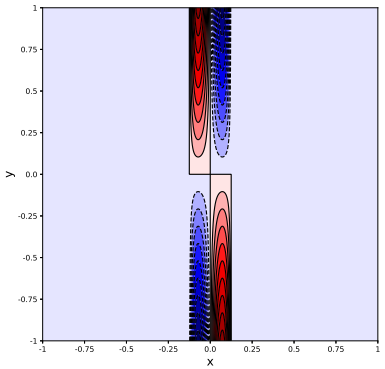
$$f_5 = \mathbb{E}(2w_i(2w_i - 1)(2w_i + 1)w_j \mid |2w_i| \leq 1)$$



$$f_6 = \mathbb{E}(4w_i(4w_i - 1)(4w_i + 1)w_j \mid |4w_i| \leq 1)$$



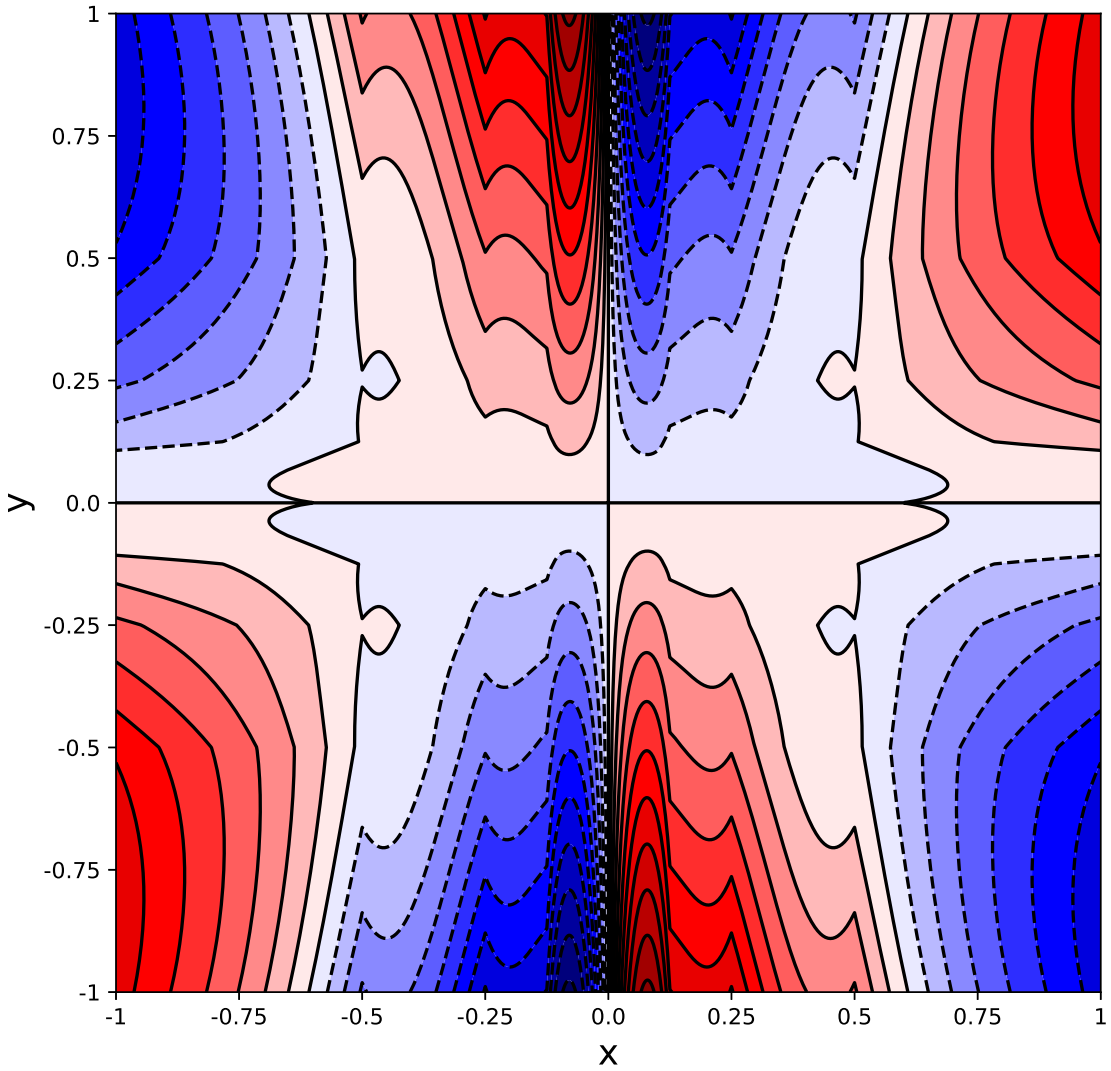
$$f_7 = \mathbb{E}(8w_i(8w_i - 1)(8w_i + 1)w_j \mid |8w_i| \leq 1)$$



Together with transposes (i.e. $f^t(w_i, w_j) = f(w_j, w_i)$), we finally selected $7 \times 2 - 1 = 13$ features in experiments.

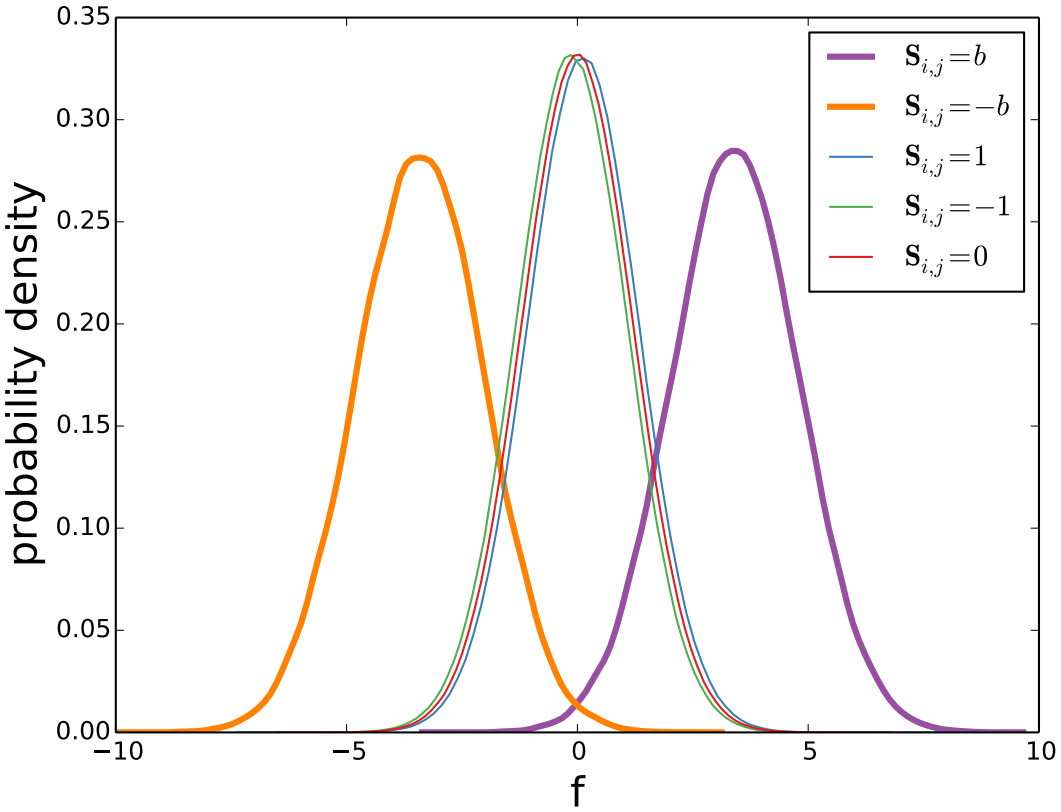
The model

$$f = \sum x_\ell f_\ell$$



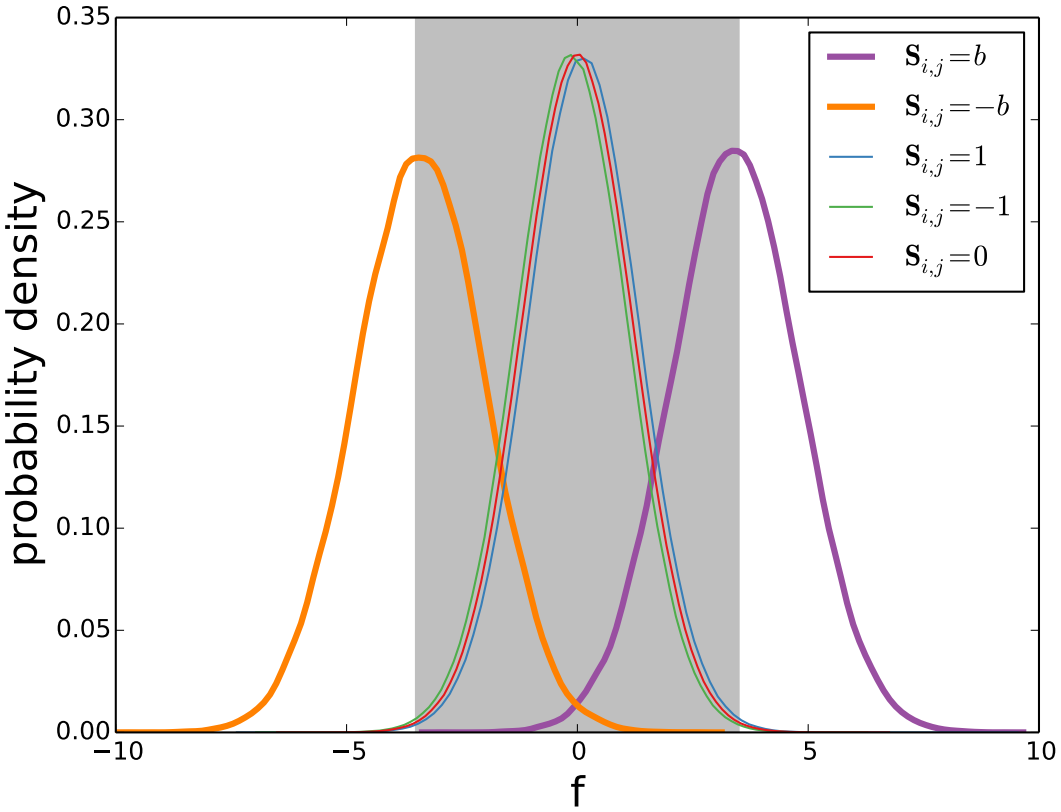
Learning

Let's learn a new **S** as $\mathbf{S}' = f(W)$!

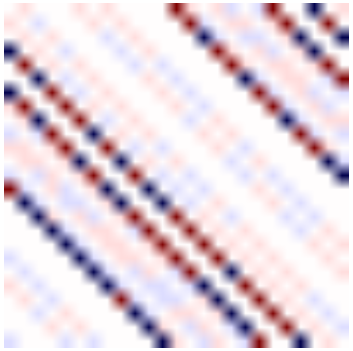


Learning

Let's learn a new **S** as $\mathbf{S}' = f(W)$!

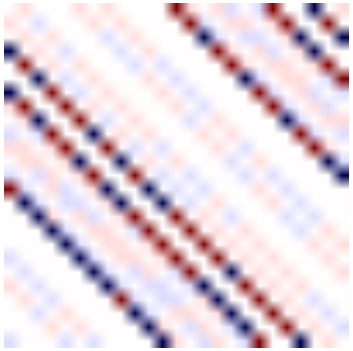


Learning

$\mathbf{S} = D \cdot I +$  is "absolute circulant"

$\Rightarrow$ more confidence via diagonal amplification

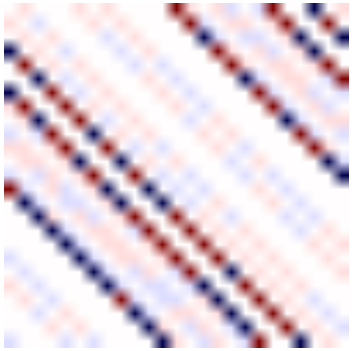
Learning

$\mathbf{S} = D \cdot I +$  is "absolute circulant"

$\Rightarrow$ more confidence via diagonal amplification

- focus on absolute values and put guesses in a same diagonal together

Learning

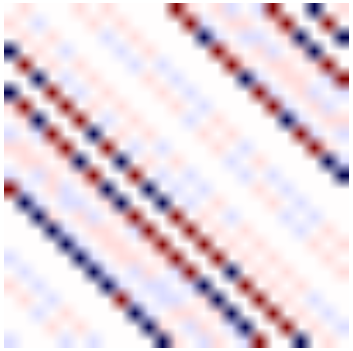
$\mathbf{S} = D \cdot I +$  is "absolute circulant"

$\Rightarrow$ more confidence via diagonal amplification

- focus on absolute values and put guesses in a same diagonal together

We locate all large coefficients successfully!

Learning

$\mathbf{S} = D \cdot I +$  is "absolute circulant"

$\Rightarrow$ more confidence via diagonal amplification

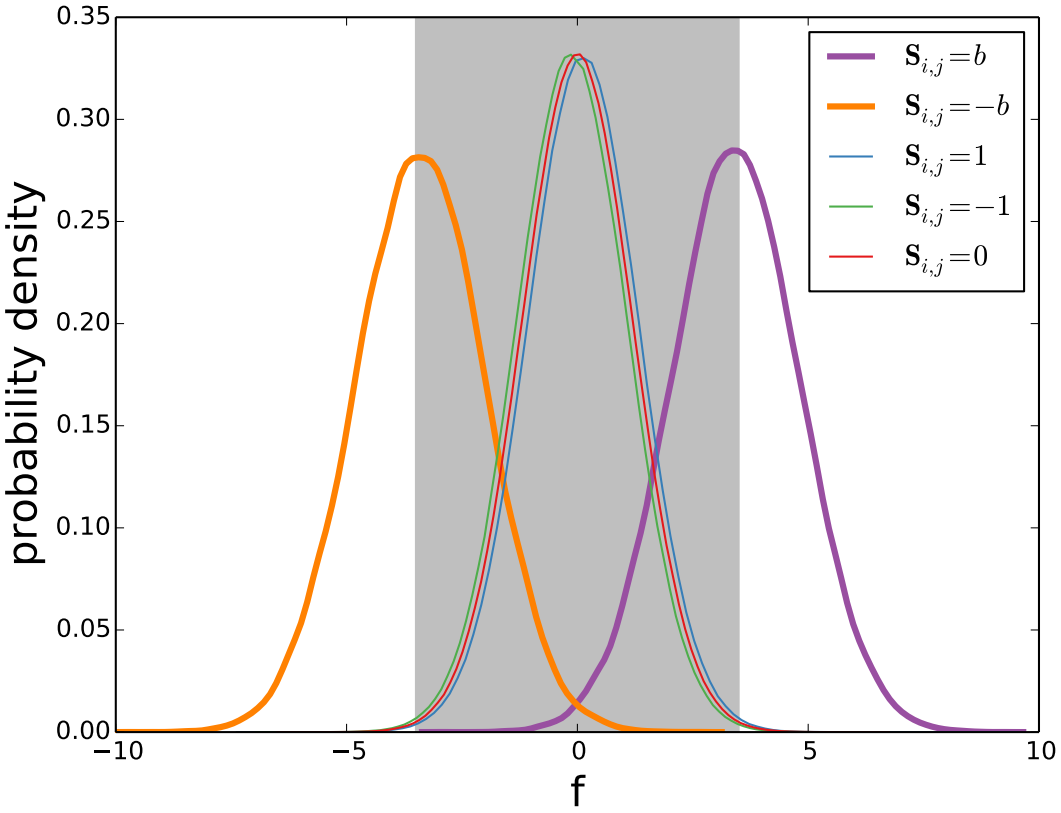
- focus on absolute values and put guesses in a same diagonal together

We locate all large coefficients successfully!

but we are still missing the signs!

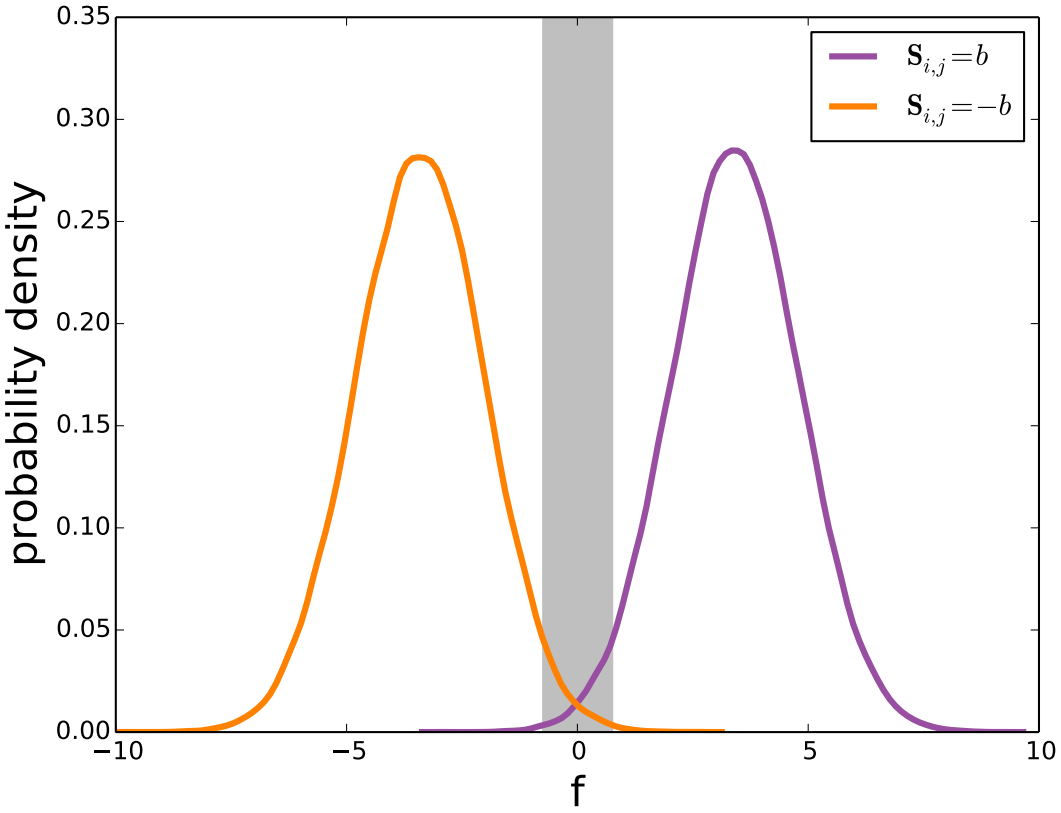
Learning

$\mathbf{S}_{i,j} \in \{\pm b, \pm 1, 0\}$



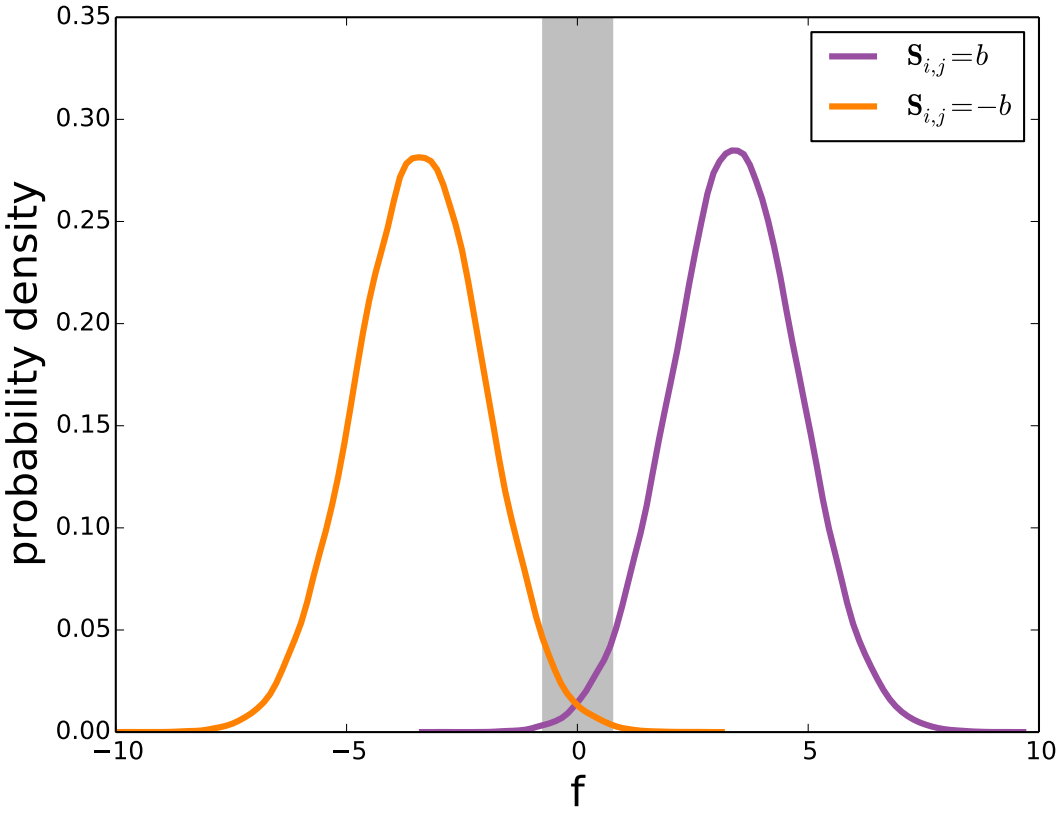
Learning

$\mathbf{S}_{i,j} \in \{\pm b\}$



Learning

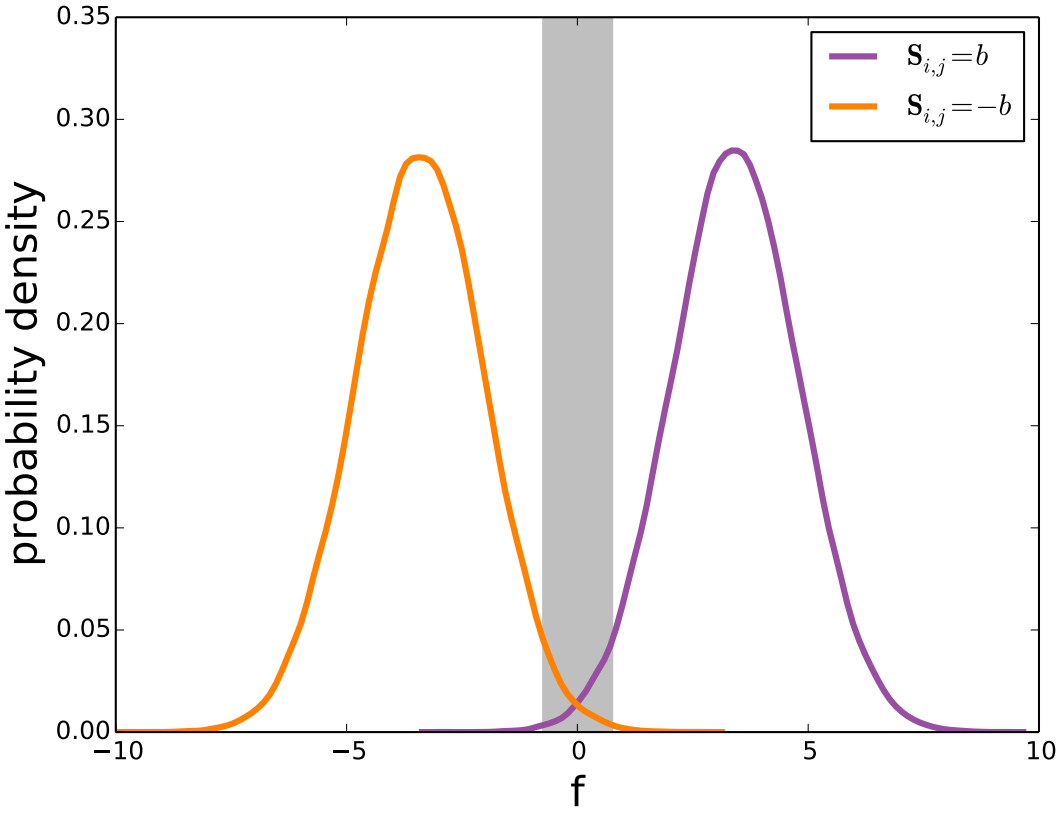
$\mathbf{S}_{i,j} \in \{\pm b\}$



We can determine all large coefficients in one row!

Learning

$\mathbf{S}_{i,j} \in \{\pm b\}$



We can determine all large coefficients in one row!

However, it is still hard to learn small coefficients...

Outline

Leaks help a lot!

Attack without leaks

- dim $= n + 1$, short vector of length $\sqrt{b^2 \cdot N_b + N_1 + 1}$
- cost: $> 2^{128}$

Leaks help a lot!

Attack without leaks

- dim $= n + 1$, short vector of length $\sqrt{\mathbf{b^2} \cdot \mathbf{N_b} + \mathbf{N_1} + \mathbf{1}}$
- cost: $> 2^{128}$

Naive attack with leaks

- dim $= n + 1$, short vector of length $\sqrt{\mathbf{N_1} + \mathbf{1}}$
- cost: 2^{78}

Leaks help a lot!

Attack without leaks

- dim $= n + 1$, short vector of length $\sqrt{b^2 \cdot N_b + N_1 + 1}$
- cost: $> 2^{128}$

Naive attack with leaks

- dim $= \mathbf{n} + \mathbf{1}$, short vector of length $\sqrt{N_1 + 1}$
- cost: 2^{78}

Improved attack with leaks

- dim $= \mathbf{n} - \mathbf{N_b}$, short vector of length $\sqrt{N_1 + 1}$
- cost: 2^{73}

Conclusion

We present a statistical attack against DRS:

- given 100 000 signatures, security is below 80-bits;
- even less with the current progress of lattice algorithms.

Thank you!