Reducing the Share Size of Weighted Threshold Secret Sharing Schemes via Chow Parameters Approximation

TCC 2024, Milan

Oriol Farràs — **Miquel Guiot**

Universitat Rovira i Virgili

6 December 2024

Financiado por la Unión Europea NextGenerationEU

GOBIERNO DE ESPAÑA — MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA — SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

Plan de Recuperación, Transformación y Resiliencia

incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

Introduction: Secret Sharing Schemes

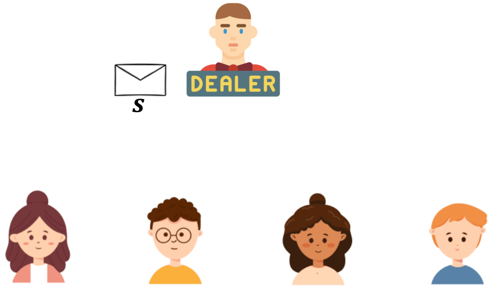
Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Introduction: Secret Sharing Schemes

Secret Sharing Scheme

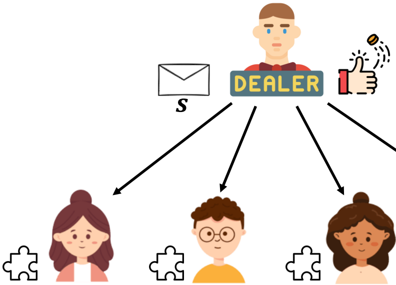
Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.



Introduction: Secret Sharing Schemes

Secret Sharing Scheme

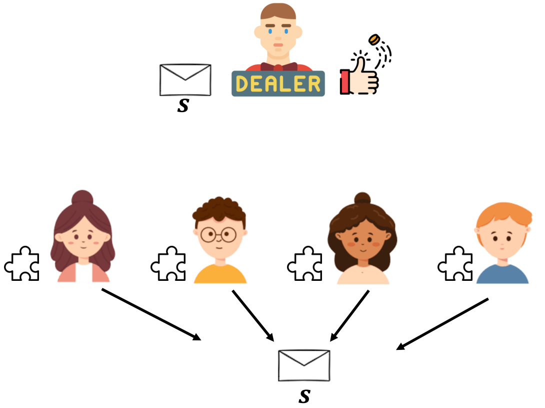
Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.



Introduction: Secret Sharing Schemes

Secret Sharing Scheme

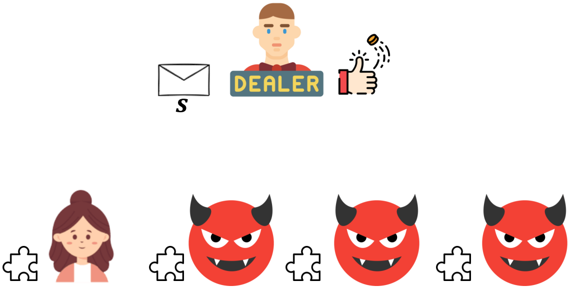
Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.



Introduction: Secret Sharing Schemes

Secret Sharing Scheme

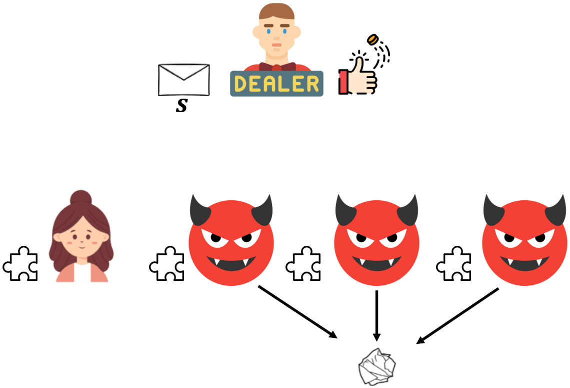
Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.



Introduction: Secret Sharing Schemes

Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.



Introduction: Secret Sharing Schemes

Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Introduction: Secret Sharing Schemes

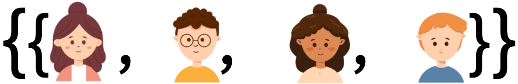
Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

$\Gamma =$



Introduction: Secret Sharing Schemes

Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Remark: Access structures are expressed as monotone Boolean functions $f : \{0,1\}^n \to \{0,1\}$.

Introduction: Secret Sharing Schemes

Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Remark: Access structures are expressed as monotone Boolean functions $f : \{0,1\}^n \to \{0,1\}$.

$\Gamma = \{\{$  ,  ,  ,  $\}\}$

$f($  ,  ,  ,  $) = 1$

Introduction: Secret Sharing Schemes

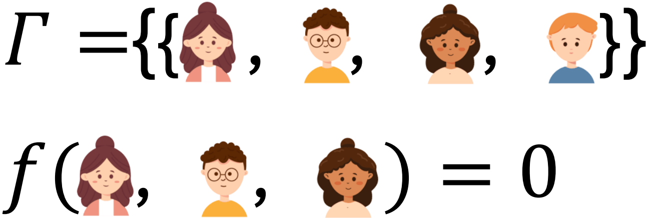
Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Remark: Access structures are expressed as monotone Boolean functions $f : \{0,1\}^n \to \{0,1\}$.



Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function: (WTF)**.

Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function: (WTF)**.

$$\Gamma = \{X \subseteq [n] : \sum_{i \in X} w_i \geq \sigma\}.$$

Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function: (WTF)**.

$$\Gamma = \{X \subseteq [n] : \sum_{i \in X} w_i \geq \sigma\}.$$

Shamir Secret Sharing Scheme

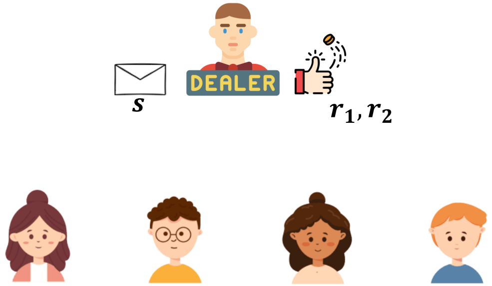
Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function: (WTF)**.

$$\Gamma = \{X \subseteq [n] : \sum_{i \in X} w_i \geq \sigma\}.$$

Shamir Secret Sharing Scheme



Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function: (WTF)**.

$$\Gamma = \{X \subseteq [n] : \sum_{i \in X} w_i \geq \sigma\}.$$

Shamir Secret Sharing Scheme

s

$$p(x) = s + r_1 x + r_2 x^2$$

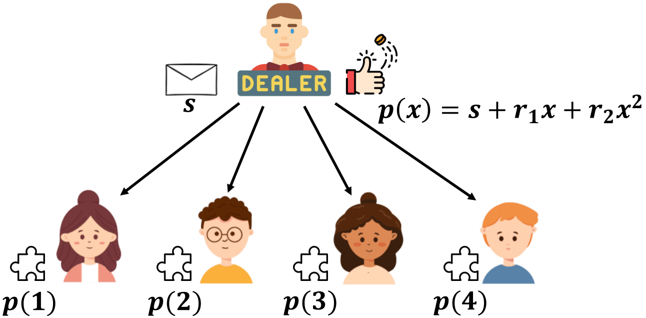
Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function: (WTF)**.

$$\Gamma = \{X \subseteq [n] : \sum_{i \in X} w_i \geq \sigma\}.$$

Shamir Secret Sharing Scheme



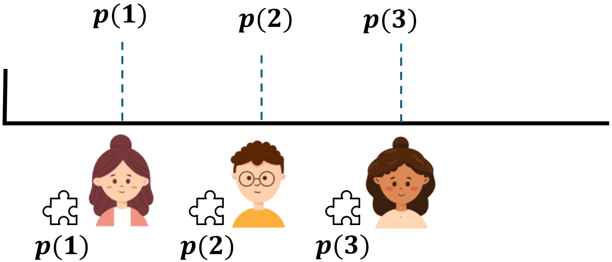
Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function: (WTF)**.

$$\Gamma = \{X \subseteq [n] : \sum_{i \in X} w_i \geq \sigma\}.$$

Shamir Secret Sharing Scheme



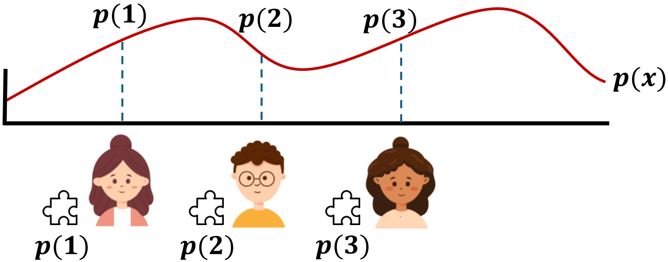
Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function: (WTF)**.

$$\Gamma = \{X \subseteq [n] : \sum_{i \in X} w_i \geq \sigma\}.$$

Shamir Secret Sharing Scheme



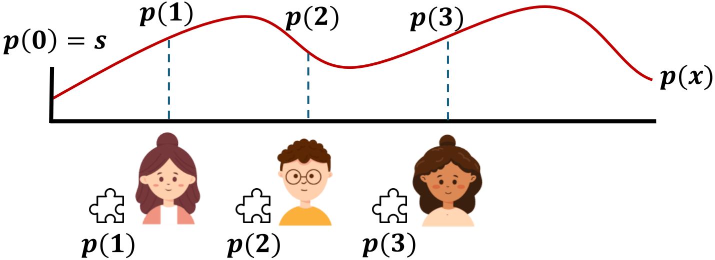
Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function: (WTF)**.

$$\Gamma = \{X \subseteq [n] : \sum_{i \in X} w_i \geq \sigma\}.$$

Shamir Secret Sharing Scheme



Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function: (WTF)**.

$$\Gamma = \{X \subseteq [n] : \sum_{i \in X} w_i \geq \sigma\}.$$

Theorem [Shamir'79]

Weighted threshold access structures admit schemes with share size $\boldsymbol{O(w_i \log W)}$.

Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function: (WTF)**.

$$\Gamma = \{X \subseteq [n] : \sum_{i \in X} w_i \geq \sigma\}.$$

Theorem [Shamir'79]

Weighted threshold access structures admit schemes with share size $\boldsymbol{O(w_i \log W)}$.

Problem [Håstadt'94]

There exists a weighted threshold access structure requiring weights of exponential size.

Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function: (WTF)**.

$$\Gamma = \{X \subseteq [n] : \sum_{i \in X} w_i \geq \sigma\}.$$

Theorem [Shamir'79]

Weighted threshold access structures admit schemes with share size $\boldsymbol{O(w_i \log W)}$.

Problem [Håstadt'94]

There exists a weighted threshold access structure requiring weights of exponential size.

Theorem [Beimel, Weinreb'06]

Weighted threshold access structures admit schemes with share size $\boldsymbol{n^{O(\log n)}}$.

Introduction: Weighted Threshold State-of-the-Art

Problem

Multiparty computation protocols and consensus mechanisms require efficient schemes for weighted threshold access structures.

Introduction: Weighted Threshold State-of-the-Art

Problem

Multiparty computation protocols and consensus mechanisms require efficient schemes for weighted threshold access structures.

Idea

To **relax** the restrictions on the access structure to obtain more **efficient** schemes.

Introduction: Weighted Threshold State-of-the-Art

Problem

Multiparty computation protocols and consensus mechanisms require efficient schemes for weighted threshold access structures.

Idea

To **relax** the restrictions on the access structure to obtain more **efficient** schemes.

$(t, T)-$Ramp Weighted Threshold Access Structure

Generalization in which subsets **below** $\boldsymbol{t}$ are forbidden and subsets **above** $\boldsymbol{T}$ are authorized.

Introduction: Weighted Threshold State-of-the-Art

Problem

Multiparty computation protocols and consensus mechanisms require efficient schemes for weighted threshold access structures.

Idea

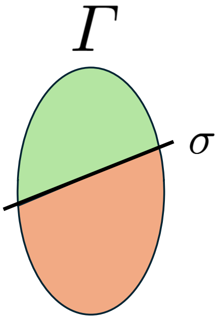
To **relax** the restrictions on the access structure to obtain more **efficient** schemes.

(t, T)−Ramp Weighted Threshold Access Structure

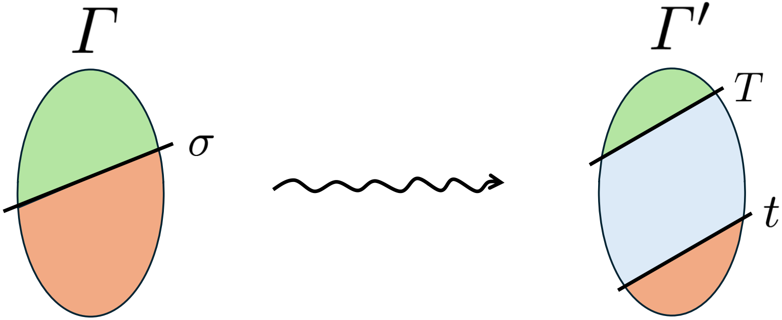
Generalization in which subsets **below t** are forbidden and subsets **above T** are authorized.

Work	Share Size	Access structure
GJMSWZ'23	$w_i = 2^{O(n \log n)}$	$(t, t + \Omega(\lambda))$-ramp WTAS
BHS'23 Rounding, TF'24	$O\left(\frac{n}{\beta-\alpha}\right)$	$(\alpha W, \beta W)$−ramp WTAS
BHS'23 BS Channels	$\max\left\{\lambda^2, \text{poly}\left(\frac{1}{\beta-\alpha}\right)\right\}$	$(\alpha W, \beta W)$−ramp WTAS

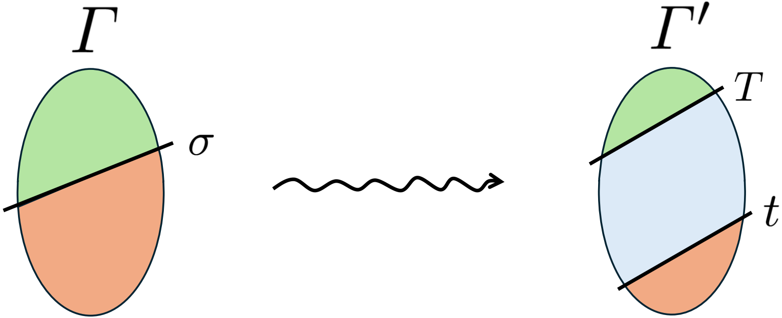
Introduction: Accuracy of the Approximation

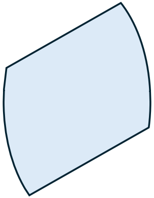


Introduction: Accuracy of the Approximation

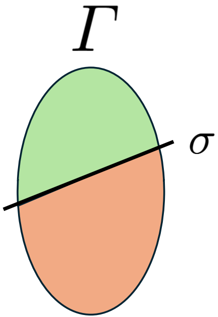


Introduction: Accuracy of the Approximation

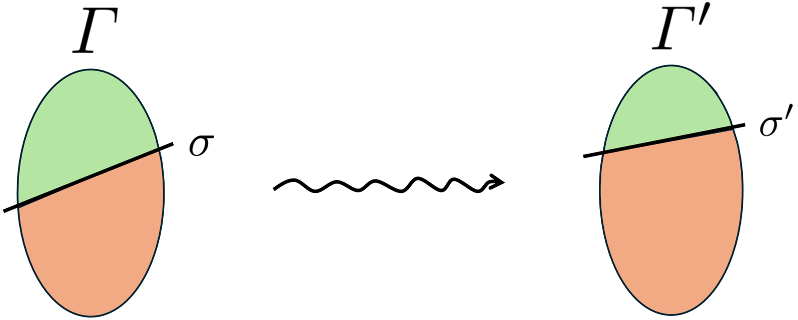


Question: How many subsets fall into  ?

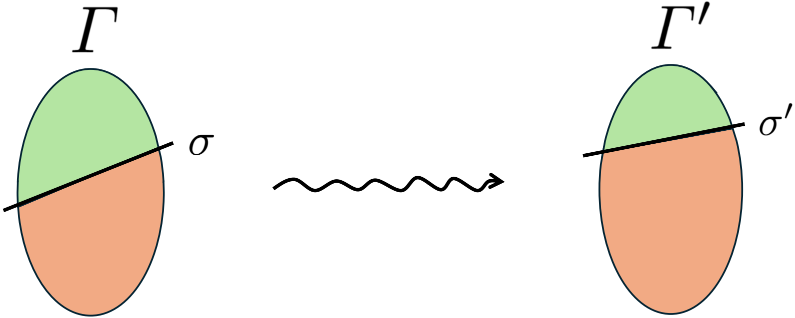
Introduction: Accuracy of the Approximation



Introduction: Accuracy of the Approximation



Introduction: Accuracy of the Approximation



Question: How many subsets change their condition from Γ to Γ'?

Introduction: Our Results

Goal: To find a good tradeoff between the efficiency and the accuracy of approximation.

Introduction: Our Results

Goal: To find a good tradeoff between the efficiency and the accuracy of approximation.

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n^{1+o(1)}}$ whose weighted threshold access structure Γ' is $\boldsymbol{o(1)}$-close to Γ.

Introduction: Our Results

Goal: To find a good tradeoff between the efficiency and the accuracy of approximation.

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n^{1+o(1)}}$ whose weighted threshold access structure Γ' is $\boldsymbol{o(1)}$-close to Γ.

Tools: Complexity theory, analysis and approximation of Boolean functions

Introduction: Our Results

Goal: To find a good tradeoff between the efficiency and the accuracy of approximation.

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $n^{1+o(1)}$ whose weighted threshold access structure Γ' is $o(1)$-close to Γ.

Tools: Complexity theory, analysis and approximation of Boolean functions

Idea

Introduction: Our Results

Goal: To find a good tradeoff between the efficiency and the accuracy of approximation.

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $n^{1+o(1)}$ whose weighted threshold access structure Γ' is $o(1)$-close to Γ.

Tools: Complexity theory, analysis and approximation of Boolean functions

Idea

$$\Gamma \xrightarrow{\varepsilon\text{-close}} \Gamma'$$

Introduction: Our Results

Goal: To find a good tradeoff between the efficiency and the accuracy of approximation.

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n^{1+o(1)}}$ whose weighted threshold access structure Γ' is $\boldsymbol{o(1)}$-close to Γ.

Tools: Complexity theory, analysis and approximation of Boolean functions

Idea

$$\begin{tikzcd}
\Gamma \arrow[rrr, dashed, "\varepsilon\text{-close}"] \arrow[d, "(1)"] & & & \Gamma' \\
f & & &
\end{tikzcd}$$

Introduction: Our Results

Goal: To find a good tradeoff between the efficiency and the accuracy of approximation.

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n^{1+o(1)}}$ whose weighted threshold access structure Γ' is $\boldsymbol{o(1)}$-close to Γ.

Tools: Complexity theory, analysis and approximation of Boolean functions

Idea

$$
\begin{tikzcd}
\Gamma \arrow[rr, dashed, "\varepsilon\text{-close}"] \arrow[d, "(1)"] & & \Gamma' \\
f \arrow[r, "(2)"] & \chi_f &
\end{tikzcd}
$$

Introduction: Our Results

Goal: To find a good tradeoff between the efficiency and the accuracy of approximation.

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n}^{\mathbf{1}+\boldsymbol{o}(\mathbf{1})}$ whose weighted threshold access structure Γ' is $\boldsymbol{o}(\mathbf{1})$-close to Γ.

Tools: Complexity theory, analysis and approximation of Boolean functions

Idea

$$\begin{tikzcd}
\Gamma \arrow[rrrr, dashed, "\varepsilon\text{-close}"] \arrow[d, "(1)"] & & & & \Gamma' \\
f \arrow[rr, "(2)"] & & \chi_f \arrow[r, "(3)"] & \chi_g &
\end{tikzcd}$$

Introduction: Our Results

Goal: To find a good tradeoff between the efficiency and the accuracy of approximation.

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $n^{1+o(1)}$ whose weighted threshold access structure Γ' is $o(1)$-close to Γ.

Tools: Complexity theory, analysis and approximation of Boolean functions

Idea

$$
\begin{tikzcd}
\Gamma \arrow[rrr, dashed, "\varepsilon\text{-close}"] \arrow[d, "(1)"] & & & \Gamma' \\
f \arrow[r, "(2)"] & \chi_f \arrow[r, "(3)"] & \chi_g \arrow[r, "(4)"] & g
\end{tikzcd}
$$

Introduction: Our Results

Goal: To find a good tradeoff between the efficiency and the accuracy of approximation.

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n}^{\mathbf{1}+\boldsymbol{o}(\mathbf{1})}$ whose weighted threshold access structure Γ' is $\boldsymbol{o}(\mathbf{1})$-close to Γ.

Tools: Complexity theory, analysis and approximation of Boolean functions

Idea

$$
\begin{tikzcd}
\Gamma \arrow[rrr, dashed, "\varepsilon\text{-close}"] \arrow[d, "(1)"] & & & \Gamma' \\
f \arrow[r, "(2)"] & \chi_f \arrow[r, "(3)"] & \chi_g \arrow[r, "(4)"] & g \arrow[u, "(5)"']
\end{tikzcd}
$$

Analysis of Boolean Functions: Chow Parameters & Distance

Chow Parameters

The $n+1$ values $\hat{f}(0) = \mathbf{E}[f(x)]$ and $\hat{f}(i) = \mathbf{E}[f(x)x_i]$ taking uniform distribution on its domain. The Chow vector is given by $\chi_f = (\hat{f}(0), \ldots, \hat{f}(n))$.

Analysis of Boolean Functions: Chow Parameters & Distance

Chow Parameters

The $n+1$ values $\hat{f}(0) = \mathbf{E}[f(x)]$ and $\hat{f}(i) = \mathbf{E}[f(x)x_i]$ taking uniform distribution on its domain. The Chow vector is given by $\chi_f = (\hat{f}(0), \ldots, \hat{f}(n))$.

Theorem [Chow'61]

Any WTF is **uniquely determined** in the space of Boolean functions by its Chow parameters.

Analysis of Boolean Functions: Chow Parameters & Distance

Chow Parameters

The $n + 1$ values $\hat{f}(0) = \mathbf{E}[f(x)]$ and $\hat{f}(i) = \mathbf{E}[f(x)x_i]$ taking uniform distribution on its domain. The Chow vector is given by $\chi_f = (\hat{f}(0), \ldots, \hat{f}(n))$.

Theorem [Chow'61]

Any WTF is **uniquely determined** in the space of Boolean functions by its Chow parameters.

Function & Chow Distance

- $\mathrm{dist}(f, g) = \mathbf{E}[|f(x) - g(x)|]$

Analysis of Boolean Functions: Chow Parameters & Distance

Chow Parameters

The $n+1$ values $\hat{f}(0) = \mathbf{E}[f(x)]$ and $\hat{f}(i) = \mathbf{E}[f(x)x_i]$ taking uniform distribution on its domain. The Chow vector is given by $\chi_f = (\hat{f}(0), \ldots, \hat{f}(n))$.

Theorem [Chow'61]

Any WTF is **uniquely determined** in the space of Boolean functions by its Chow parameters.

Function & Chow Distance

- $\mathrm{dist}(f,g) = \mathbf{E}[|f(x) - g(x)|]$
- $\mathrm{dist}_{\mathrm{Chow}}(f,g) = \|\chi_f - \chi_g\|$

Analysis of Boolean Functions: Chow Parameters & Distance

Chow Parameters

The $n+1$ values $\hat{f}(0) = \mathbf{E}[f(x)]$ and $\hat{f}(i) = \mathbf{E}[f(x)x_i]$ taking uniform distribution on its domain. The Chow vector is given by $\chi_f = (\hat{f}(0), \ldots, \hat{f}(n))$.

Theorem [Chow'61]

Any WTF is **uniquely determined** in the space of Boolean functions by its Chow parameters.

Function & Chow Distance

- $\mathrm{dist}(f,g) = \mathbf{E}[|f(x) - g(x)|]$
- $\mathrm{dist}_{\mathrm{Chow}}(f,g) = \|\chi_f - \chi_g\|$

Theorem [O'Donnell, Servedio'11 + De, Diakonikolas, Feldman, Servedio'14]

For any two Boolean functions f and g, $\mathrm{dist}(f,g) = o(1) \iff \mathrm{dist}_{\mathrm{Chow}}(f,g) = o(1)$.

Analysis of Boolean Functions: Chow Parameters & Distance

Chow Parameters

The $n+1$ values $\hat{f}(0) = \mathbf{E}[f(x)]$ and $\hat{f}(i) = \mathbf{E}[f(x)x_i]$ taking uniform distribution on its domain. The Chow vector is given by $\chi_f = (\hat{f}(0), \ldots, \hat{f}(n))$.

Theorem [Chow'61]

Any WTF is **uniquely determined** in the space of Boolean functions by its Chow parameters.

Function & Chow Distance

- $\mathrm{dist}(f, g) = \mathbf{E}[|f(x) - g(x)|]$
- $\mathrm{dist}_{\mathrm{Chow}}(f, g) = \|\chi_f - \chi_g\|$

Theorem [O'Donnell, Servedio'11 + De, Diakonikolas, Feldman, Servedio'14]

For any two Boolean functions f and g, $\mathrm{dist}(f, g) = o(1) \iff \mathrm{dist}_{\mathrm{Chow}}(f, g) = o(1)$.

Idea

To use the Chow parameters as the **building block** for approximating WTFs.

Approximation of Weighted Threshold Functions: Statement

Theorem

For any $\varepsilon \in (0, 1)$, there exists a **randomized algorithm** ApproximateWTF that given any monotone WTF over n variables f, outputs a monotone WTF g with the following properties:

Approximation of Weighted Threshold Functions: Statement

Theorem

For any $\varepsilon \in (0, 1)$, there exists a **randomized algorithm** ApproximateWTF that given any monotone WTF over n variables f, outputs a monotone WTF g with the following properties:

- g is ε-**close** to f,

Approximation of Weighted Threshold Functions: Statement

Theorem

For any $\varepsilon \in (0, 1)$, there exists a **randomized algorithm** ApproximateWTF that given any monotone WTF over n variables f, outputs a monotone WTF g with the following properties:

- g is ε-**close** to f,
- g has weights of size $\boldsymbol{O}\left(\sqrt{\boldsymbol{n}}\left(\frac{1}{\varepsilon}\right)^{\boldsymbol{O}\left(\log^2\left(\frac{1}{\varepsilon}\right)\right)}\right)$.

Approximation of Weighted Threshold Functions: Statement

Theorem

For any $\varepsilon \in (0,1)$, there exists a **randomized algorithm** ApproximateWTF that given any monotone WTF over n variables f, outputs a monotone WTF g with the following properties:

- g is ε-**close** to f,
- g has weights of size $\boldsymbol{O\left(\sqrt{n}\left(\frac{1}{\varepsilon}\right)^{O\left(\log^2\left(\frac{1}{\varepsilon}\right)\right)}\right)}$.

Remarks

Approximation of Weighted Threshold Functions: Statement

Theorem

For any $\varepsilon \in (0,1)$, there exists a **randomized algorithm** ApproximateWTF that given any monotone WTF over n variables f, outputs a monotone WTF g with the following properties:

- g is ε-**close** to f,
- g has weights of size $\boldsymbol{O}\left(\sqrt{\boldsymbol{n}}\left(\frac{1}{\varepsilon}\right)^{\boldsymbol{O}\left(\log^2\left(\frac{1}{\varepsilon}\right)\right)}\right)$.

Remarks

- It is an adaptation of the work of De, Diakonikolas, Feldman, Servedio'14 to the monotone setting.

Approximation of Weighted Threshold Functions: Statement

Theorem

For any $\varepsilon \in (0,1)$, there exists a **randomized algorithm** ApproximateWTF that given any monotone WTF over n variables f, outputs a monotone WTF g with the following properties:

- g is ε-**close** to f,
- g has weights of size $\boldsymbol{O}\left(\sqrt{\boldsymbol{n}}\left(\frac{1}{\varepsilon}\right)^{\boldsymbol{O}\left(\log^2\left(\frac{1}{\varepsilon}\right)\right)}\right)$.

Remarks

- It is an adaptation of the work of De, Diakonikolas, Feldman, Servedio'14 to the monotone setting.
- The approximation preserves the influence of the coordinates and the weight hierarchy.

Approximation of Weighted Threshold Functions: The Algorithm

$$f(x) = \text{sign}(w_1x_1 + ... + w_nx_n - t)$$

Approximation of Weighted Threshold Functions: The Algorithm

$$f(x) = \mathrm{sign}(w_1 x_1 + ... + w_n x_n - t)$$

$$\downarrow$$

$$\chi_f$$

Approximation of Weighted Threshold Functions: The Algorithm

$$f(x) = \operatorname{sign}(w_1 x_1 + \ldots + w_n x_n - t)$$

$$\Big\downarrow$$

$$\boldsymbol{\chi}_f$$

$$\Big\downarrow$$

$$g(x) = \operatorname{sign}(\hat{f}(1) x_1 + \ldots + \hat{f}(n) x_n - \hat{f}(0))$$

Approximation of Weighted Threshold Functions: The Algorithm

$$f(x) = \text{sign}(w_1 x_1 + \ldots + w_n x_n - t)$$

$$\downarrow$$

$$\boldsymbol{\chi}_f$$

$$\downarrow$$

$$g(x) = \text{sign}(\hat{f}(1)x_1 + \ldots + \hat{f}(n)x_n - \hat{f}(0))$$

$$\downarrow$$

$$\boldsymbol{\chi}_g$$

Approximation of Weighted Threshold Functions: The Algorithm

$$
\begin{array}{c}
f(x) = \text{sign}(w_1 x_1 + \ldots + w_n x_n - t) \\
\downarrow \\
\boldsymbol{\chi}_f \\
\downarrow \\
g(x) = \text{sign}(\hat{f}(1) x_1 + \ldots + \hat{f}(n) x_n - \hat{f}(0)) \\
\downarrow \\
\boldsymbol{\chi}_g \longrightarrow \|\boldsymbol{\chi}_f - \boldsymbol{\chi}_g\| \leq 2\varepsilon?
\end{array}
$$

Approximation of Weighted Threshold Functions: The Algorithm

$$f(x) = \mathrm{sign}(w_1 x_1 + \ldots + w_n x_n - t)$$

$$\downarrow$$

$$\boldsymbol{\chi}_f$$

$$\downarrow$$

$$g(x) = \mathrm{sign}(\hat{f}(1)x_1 + \ldots + \hat{f}(n)x_n - \hat{f}(0))$$

$$\downarrow$$

$$\boldsymbol{\chi}_g \longrightarrow \|\boldsymbol{\chi}_f - \boldsymbol{\chi}_g\| \leq 2\varepsilon?$$

$$h(x) = \hat{g}(0) - \hat{f}(0) + \sum_{i=1}^{n} (\hat{f}(i) - \hat{g}(i))x_i \xleftarrow{\text{No}} \Diamond$$

Approximation of Weighted Threshold Functions: The Algorithm

$$f(x) = \mathrm{sign}(w_1 x_1 + \ldots + w_n x_n - t)$$

$$\downarrow$$

$$\boldsymbol{\chi}_f$$

$$\downarrow$$

$$g(x) = \mathrm{sign}(\hat{f}(1) x_1 + \ldots + \hat{f}(n) x_n - \hat{f}(0))$$

$$\downarrow$$

$$\boldsymbol{\chi}_g \longrightarrow \|\boldsymbol{\chi}_f - \boldsymbol{\chi}_g\| \leq 2\varepsilon?$$

No

$$h(x) = \hat{g}(0) - \hat{f}(0) + \sum_{i=1}^{n} (\hat{f}(i) - \hat{g}(i)) x_i$$

$$\downarrow$$

$$g(x) = \mathrm{sign}(g(x) + h(x))$$

Approximation of Weighted Threshold Functions: The Algorithm

$$f(x) = \mathrm{sign}(w_1 x_1 + \dots + w_n x_n - t)$$

$$\downarrow$$

$$\boldsymbol{\chi}_f$$

$$\downarrow$$

$$g(x) = \mathrm{sign}(\hat{f}(1)x_1 + \dots + \hat{f}(n)x_n - \hat{f}(0))$$

$$\downarrow$$

$$\boldsymbol{\chi}_g \longrightarrow \|\boldsymbol{\chi}_f - \boldsymbol{\chi}_g\| \leq 2\varepsilon?$$

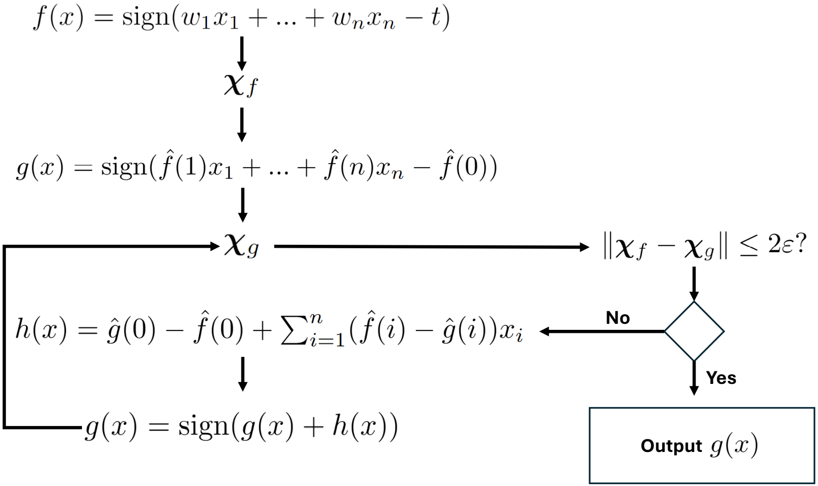
No

$$h(x) = \hat{g}(0) - \hat{f}(0) + \sum_{i=1}^{n}(\hat{f}(i) - \hat{g}(i))x_i$$

$$\downarrow$$

$$g(x) = \mathrm{sign}(g(x) + h(x))$$

Approximation of Weighted Threshold Functions: The Algorithm



Secret Sharing Schemes Construction

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n^{1+o(1)}}$ whose access structure Γ' is **$\boldsymbol{o(1)}$-close** to Γ.

Secret Sharing Schemes Construction

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n^{1+o(1)}}$ whose access structure Γ' is $\boldsymbol{o(1)}$**-close** to Γ.

Idea

Secret Sharing Schemes Construction

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n^{1+o(1)}}$ whose access structure Γ' is $\boldsymbol{o(1)}$**-close** to Γ.

Idea

1. Define a new access structure f' from f by discarding all parties with $\hat{f}(i) < \frac{1}{2n\log^k(n)}$.

Secret Sharing Schemes Construction

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n^{1+o(1)}}$ whose access structure Γ' is $\boldsymbol{o(1)}$**-close** to Γ.

Idea

1. Define a new access structure f' from f by discarding all parties with $\hat{f}(i) < \frac{1}{2n\log^k(n)}$.
2. Approximate f' using the algorithm ApproximateWTF to obtain g.

Secret Sharing Schemes Construction

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n^{1+o(1)}}$ whose access structure Γ' is $\boldsymbol{o(1)}$**-close** to Γ.

Idea

1. Define a new access structure f' from f by discarding all parties with $\hat{f}(i) < \frac{1}{2n\log^k(n)}$.
2. Approximate f' using the algorithm ApproximateWTF to obtain g.
3. The weighted threshold access structure given by g has weights of size $n^{1+o(1)}$.

Secret Sharing Schemes Construction

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n^{1+o(1)}}$ whose access structure Γ' is $\boldsymbol{o(1)}$**-close** to Γ.

Idea

1. Define a new access structure f' from f by discarding all parties with $\hat{f}(i) < \frac{1}{2n\log^k(n)}$.
2. Approximate f' using the algorithm ApproximateWTF to obtain g.
3. The weighted threshold access structure given by g has weights of size $n^{1+o(1)}$.
4. Apply Shamir secret sharing scheme to g.

Secret Sharing Schemes Construction

Theorem

For any weighted threshold access structure Γ, there exists a secret sharing scheme with share size $\boldsymbol{n^{1+o(1)}}$ whose access structure Γ' is $\boldsymbol{o(1)}$**-close** to Γ.

Idea

1. Define a new access structure f' from f by discarding all parties with $\hat{f}(i) < \frac{1}{2n\log^k(n)}$.
2. Approximate f' using the algorithm ApproximateWTF to obtain g.
3. The weighted threshold access structure given by g has weights of size $n^{1+o(1)}$.
4. Apply Shamir secret sharing scheme to g.

Side Note

Efficient schemes for **any** weighted threshold access structure can be build from **computational assumptions**.

Closing Remarks & Open Questions

Closing Remarks

Closing Remarks & Open Questions

Closing Remarks

- The share size of existing schemes for weighted threshold access structures have a strong dependency on the size of the weights.

Closing Remarks & Open Questions

Closing Remarks

- The share size of existing schemes for weighted threshold access structures have a strong dependency on the size of the weights.
- The state-of-the-art solution is to find a good tradeoff between efficiency and accuracy.

Closing Remarks & Open Questions

Closing Remarks

- The share size of existing schemes for weighted threshold access structures have a strong dependency on the size of the weights.
- The state-of-the-art solution is to find a good tradeoff between efficiency and accuracy.
- Our proposal ensures almost linear share size at the cost of $o(1)$ error.

Closing Remarks & Open Questions

Closing Remarks

- The share size of existing schemes for weighted threshold access structures have a strong dependency on the size of the weights.
- The state-of-the-art solution is to find a good tradeoff between efficiency and accuracy.
- Our proposal ensures almost linear share size at the cost of $o(1)$ error.
- Moving towards the computational setting allows more efficient schemes.

Closing Remarks & Open Questions

Closing Remarks

- The share size of existing schemes for weighted threshold access structures have a strong dependency on the size of the weights.
- The state-of-the-art solution is to find a good tradeoff between efficiency and accuracy.
- Our proposal ensures almost linear share size at the cost of $o(1)$ error.
- Moving towards the computational setting allows more efficient schemes.

Open Questions

Closing Remarks & Open Questions

Closing Remarks

- The share size of existing schemes for weighted threshold access structures have a strong dependency on the size of the weights.
- The state-of-the-art solution is to find a good tradeoff between efficiency and accuracy.
- Our proposal ensures almost linear share size at the cost of $o(1)$ error.
- Moving towards the computational setting allows more efficient schemes.

Open Questions

- Is there any scheme with polynomial share size for weighted threshold access structures?

Closing Remarks & Open Questions

Closing Remarks

- The share size of existing schemes for weighted threshold access structures have a strong dependency on the size of the weights.
- The state-of-the-art solution is to find a good tradeoff between efficiency and accuracy.
- Our proposal ensures almost linear share size at the cost of $o(1)$ error.
- Moving towards the computational setting allows more efficient schemes.

Open Questions

- Is there any scheme with polynomial share size for weighted threshold access structures?
- Can we improve the $\Omega(\sqrt{n})$ lower bound for these access structures?

Closing Remarks & Open Questions

Closing Remarks

- The share size of existing schemes for weighted threshold access structures have a strong dependency on the size of the weights.
- The state-of-the-art solution is to find a good tradeoff between efficiency and accuracy.
- Our proposal ensures almost linear share size at the cost of $o(1)$ error.
- Moving towards the computational setting allows more efficient schemes.

Open Questions

- Is there any scheme with polynomial share size for weighted threshold access structures?
- Can we improve the $\Omega(\sqrt{n})$ lower bound for these access structures?
- Can we relate the notion of distance with the ramp criteria?

Thank You!

Check the Full Version of our Paper!