Secure Two-Party Distribution Testing

Alexandr Andoni Tal Malkin **Negev Shekel Nosatzki**

Department of Computer Science
Columbia University

Privacy Preserving Machine Learning 2018

December 2018

Discrete Distribution Testing

Test distributions for statistical properties using sample access.

Closeness Testing

- 2 distributions: a, b.
- Alphabet: $[n]$.
- Inputs: t samples from each of a and b.

$$\alpha_1 \ldots \alpha_t \sim a$$

$$\beta_1 \ldots \beta_t \sim b$$

Does $a = b$ or $\|a - b\|_1 > \epsilon$?

Typical Question: What is t? (sample complexity)

$t = \Theta_\epsilon(n^{2/3})$ [BFR+ 00, Val11, BFR+ 13, CDVV14, DK16, DGPP16]

Many variants:

- Instance-Optimal [ADJ+ 11, ADJ+ 12, DK16].
- Unequal sample sizes [AJOS14, BV15, DK16].
- Quantum [BHH11].

Discrete Distribution Testing

Test distributions for statistical properties using sample access.

Closeness Testing

- 2 distributions: a, b.
- Alphabet: $[n]$.
- Inputs: t samples from each of a and b.

$$\alpha_1 \ldots \alpha_t \sim a$$
$$\beta_1 \ldots \beta_t \sim b$$

Does $a = b$ or $\|a - b\|_1 > \epsilon$?

Typical Question: What is t? (sample complexity)

$t = \Theta_\epsilon(n^{2/3})$ [BFR+ 00, Val11, BFR+ 13, CDVV14, DK16, DGPP16]

Many variants:

- Instance-Optimal [ADJ+ 11, ADJ+ 12, DK16].
- Unequal sample sizes [AJOS14, BV15, DK16].
- Quantum [BHH11].

Discrete Distribution Testing

Test distributions for statistical properties using sample access.

Closeness Testing

- 2 distributions: a, b.
- Alphabet: $[n]$.
- Inputs: t samples from each of a and b.

$$\alpha_1 \dots \alpha_t \sim a$$

$$\beta_1 \dots \beta_t \sim b$$

Does $a = b$ or $\|a - b\|_1 > \epsilon$?

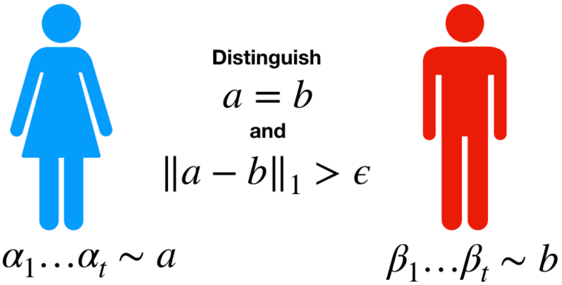
Typical Question: What is t? (sample complexity)

$t = \Theta_\epsilon(n^{2/3})$ [BFR+ 00, Val11, BFR+ 13, CDVV14, DK16, DGPP16]

Many variants:

- Instance-Optimal [ADJ+ 11, ADJ+ 12, DK16].
- Unequal sample sizes [AJOS14, BV15, DK16].
- Quantum [BHH11].

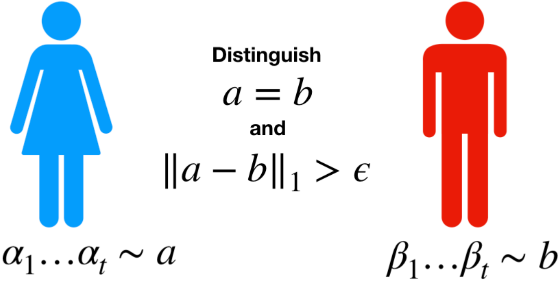
This Talk: Two Party Closeness Testing



Main Questions:

- Communication Complexity
- Security.

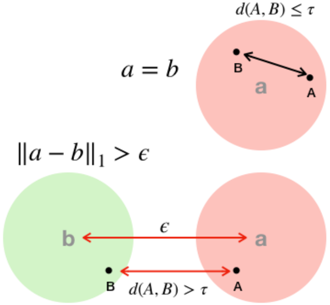
This Talk: Two Party Closeness Testing



Main Questions:

- **Communication Complexity**
- **Security.**

Testing Closeness - Known Reductions [CDVV14,DK16]

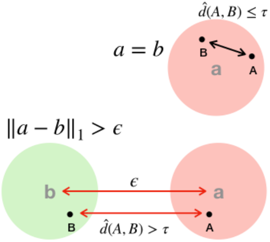


$$d(A,B) = \frac{1}{t}\sqrt{\sum_{i\in[n]} (A_i - B_i)^2 - 2t}$$

(A_i, B_i are the no. of occurrences of the i^{th} letter in each set.)

- Tool: $\ell 1$ to $\ell 2$ reduction.
- Compute *count-distance* for 2 sets of t samples $A \sim a, B \sim b$.
- Compare to some threshold τ to estimate if they originated from SAME or ϵ-FAR distributions.
- Reductions use "splitting" / "flattening" techniques.
- This results in adjusted alphabet, that **depends on Bob's inputs.**

Improving communication (still insecurely)



$$d(A,B) = \frac{1}{t}\sqrt{\sum_{i\in[n]}(A_i - B_i)^2 - 2t}$$

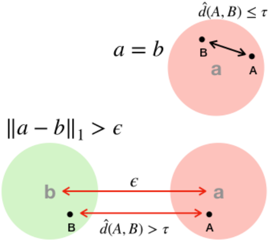
- Alice and Bob estimate $\hat{d}(A,B)$ by sketching $\|A-B\|_2^2$ approximation and comparing to threshold τ.
- With **more samples**, can tolerate **cruder approximation**, gaining **communication efficiency**.

Communication Complexity: $\tilde{\Theta}_\epsilon(n^2/t^2)$

Examples:

- With $t = \Theta_\epsilon(n^{2/3})$, need to communicate **near-all** of them.
- With **linear** sample size, we allow $\tilde{O}_\epsilon(1)$ **communication**.

Improving communication (still insecurely)



$$d(A,B) = \frac{1}{t}\sqrt{\sum_{i\in[n]} (A_i - B_i)^2 - 2t}$$

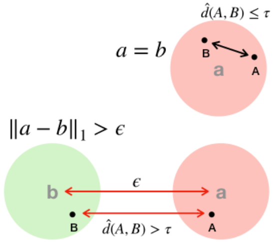
- Alice and Bob estimate $\hat{d}(A,B)$ by sketching $\|A - B\|_2^2$ approximation and comparing to threshold τ.
- With **more samples**, can tolerate **cruder approximation**, gaining **communication efficiency**.

Communication Complexity: $\tilde{\Theta}_\epsilon(n^2/t^2)$

Examples:

- With $t = \Theta_\epsilon(n^{2/3})$, need to communicate **near-all** of them.
- With **linear** sample size, we allow $\tilde{O}_\epsilon(1)$ **communication**.

Improving communication (still insecurely)



$$d(A,B) = \frac{1}{t}\sqrt{\sum_{i\in[n]} (A_i - B_i)^2 - 2t}$$

- Alice and Bob estimate $\hat{d}(A,B)$ by sketching $\|A-B\|_2^2$ approximation and comparing to threshold τ.
- With **more samples**, can tolerate **cruder approximation**, gaining **communication efficiency**.

Communication Complexity: $\tilde{\Theta}_\epsilon(n^2/t^2)$

Examples:

- With $t = \Theta_\epsilon(n^{2/3})$, need to communicate **near-all** of them.
- With **linear** sample size, we allow $\tilde{O}_\epsilon(1)$ **communication**.

Adding Security

- Applying **generic techniques** for secure computation is **prohibitive** in our context, as we care for **sublinear communication**.
- $\|A - B\|_2^2$ can be estimated securely and efficiently using a secure (garbled) circuit with **external memory** [IW06].
- But reductions estimators use an adjusted alphabet that "depend on Bob's samples".

Goal: Securely estimating $\|A_S - B_S\|_2^2$

(where A_S, B_S represent samples over the adjusted alphabet)

- We need a secure way for Alice and Bob to agree on an alphabet.

Observation: *Most letters multiplicity is not affected by alphabet change.*

Adding Security

- Applying **generic techniques** for secure computation is **prohibitive** in our context, as we care for **sublinear communication**.
- $\|A - B\|_2^2$ can be estimated securely and efficiently using a secure (garbled) circuit with **external memory** [IW06].
- But reductions estimators use an adjusted alphabet that "depend on Bob's samples".

Goal: Securely estimating $\|A_S - B_S\|_2^2$

(where A_S, B_S represent samples over the adjusted alphabet)

- We need a secure way for Alice and Bob to agree on an alphabet.

Observation: *Most letters multiplicity is not affected by alphabet change.*

Adding Security

- Applying **generic techniques** for secure computation is **prohibitive** in our context, as we care for **sublinear communication**.
- $\|A - B\|_2^2$ can be estimated securely and efficiently using a secure (garbled) circuit with **external memory** [IW06].
- But reductions estimators use an adjusted alphabet that "depend on Bob's samples".

Goal: Securely estimating $\|A_S - B_S\|_2^2$

(where A_S, B_S represent samples over the adjusted alphabet)

- We need a secure way for Alice and Bob to agree on an alphabet.

Observation: *Most letters multiplicity is not affected by alphabet change.*

Adding Security

- Applying **generic techniques** for secure computation is **prohibitive** in our context, as we care for **sublinear communication**.
- $\|A - B\|_2^2$ can be estimated securely and efficiently using a secure (garbled) circuit with **external memory** [IW06].
- But reductions estimators use an adjusted alphabet that "depend on Bob's samples".

Goal: Securely estimating $\|A_S - B_S\|_2^2$

(where A_S, B_S represent samples over the adjusted alphabet)

- We need a secure way for Alice and Bob to agree on an alphabet.

Observation: *Most letters multiplicity is not affected by alphabet change.*

Adding Security

- Applying **generic techniques** for secure computation is **prohibitive** in our context, as we care for **sublinear communication**.
- $\|A - B\|_2^2$ can be estimated securely and efficiently using a secure (garbled) circuit with **external memory** [IW06].
- But reductions estimators use an adjusted alphabet that "depend on Bob's samples".

Goal: Securely estimating $\|A_S - B_S\|_2^2$

(where A_S, B_S represent samples over the adjusted alphabet)

- We need a secure way for Alice and Bob to agree on an alphabet.

Observation: *Most letters multiplicity is not affected by alphabet change.*

Solution Overview

Goal: Securely estimating $\|A_S - B_S\|_2^2$

(where A_S, B_S represent samples over the adjusted alphabet)

- Secure circuit estimates some distance of the original alphabet.
- Such estimation is then adjusted by the circuit to account for the adjusted alphabet and "heavy" letters.
- Offline preparation of (polynomial) external memory enable efficiency and correctness.

Solution Overview

Goal: Securely estimating $\|A_S - B_S\|_2^2$

(where A_S, B_S represent samples over the adjusted alphabet)

- Secure circuit estimates some distance of the original alphabet.
- Such estimation is then adjusted by the circuit to account for the adjusted alphabet and "heavy" letters.
- Offline preparation of (polynomial) external memory enable efficiency and correctness.

Solution Overview

Goal: Securely estimating $\|A_S - B_S\|_2^2$

(where A_S, B_S represent samples over the adjusted alphabet)

- Secure circuit estimates some distance of the original alphabet.
- Such estimation is then adjusted by the circuit to account for the adjusted alphabet and “heavy” letters.
- Offline preparation of (polynomial) external memory enable efficiency and correctness.

Solution Overview

Goal: Securely estimating $\|A_S - B_S\|_2^2$

(where A_S, B_S represent samples over the adjusted alphabet)

- Secure circuit estimates some distance of the original alphabet.
- Such estimation is then adjusted by the circuit to account for the adjusted alphabet and "heavy" letters.
- Offline preparation of (polynomial) external memory enable efficiency and correctness.

Secure Closeness: Methods

1. **Adapted Reduction**: adjust alphabet using split set S sampled from both a and b. (avoiding insecure part in reduction)

2. **Capped Samples**: estimate *capped sample distance* $\|A' - B'\|_2^2$. (which is of a similar magnitude as $\|A_S - B_S\|_2^2$, over the adjusted alphabet)

Split Samples: Recasted samples randomly placed in 1-of-s bins, based on sample multiplicity in multi-set S

$$A = \begin{bmatrix} 6 \\ 0 \\ 7 \\ 1 \end{bmatrix} \quad \rightarrow \quad A_S = \begin{bmatrix} 6 & & \\ 0 & & \\ 2 & 4 & 1 \\ 0 & 1 & \end{bmatrix}$$

$$S = \{3, 3, 4\}$$

Capped Samples: Count samples up to L.

$$A = \begin{bmatrix} 6 \\ 0 \\ 7 \\ 1 \end{bmatrix} \quad \rightarrow \quad A' = \begin{bmatrix} 5 \\ 0 \\ 5 \\ 1 \end{bmatrix}$$

$$L = 5$$

Secure Closeness: Methods

1. **Adapted Reduction**: adjust alphabet using split set S sampled from both a and b. (avoiding insecure part in reduction)

2. **Capped Samples**: estimate *capped sample distance* $\|A' - B'\|_2^2$. (which is of a similar magnitude as $\|A_S - B_S\|_2^2$, over the adjusted alphabet)

Split Samples: Recasted samples randomly placed in 1-of-s bins, based on sample multiplicity in multi-set S

$$A = \begin{bmatrix} 6 \\ 0 \\ 7 \\ 1 \end{bmatrix} \quad \rightarrow \quad A_S = \begin{bmatrix} 6 & & \\ 0 & & \\ 2 & 4 & 1 \\ 0 & 1 & \end{bmatrix}$$

$$S = \{3, 3, 4\}$$

Capped Samples: Count samples up to L.

$$A = \begin{bmatrix} 6 \\ 0 \\ 7 \\ 1 \end{bmatrix} \quad \rightarrow \quad A' = \begin{bmatrix} 5 \\ 0 \\ 5 \\ 1 \end{bmatrix}$$

$$L = 5$$

Secure Closeness: Methods (cont)

3. **Adjust for "heavy letters"**: compute $\|A' - B'\|_2^2 - \|A_S - B_S\|_2^2$ exactly.
   (function of a small number of letters. can be computed over a small-sized circuit)

Split Samples: Recasted samples randomly placed in 1-of-s bins, based on sample multiplicity in multi-set S

$$A = \begin{bmatrix} 6 \\ 0 \\ 7 \\ 1 \end{bmatrix} \quad \rightarrow \quad A_S = \begin{bmatrix} 6 & & \\ 0 & & \\ 2 & 4 & 1 \\ 0 & 1 & \end{bmatrix}$$

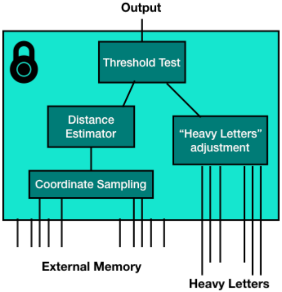
$$S = \{3, 3, 4\}$$

Capped Samples: Count samples up to L.

$$A = \begin{bmatrix} 6 \\ 0 \\ 7 \\ 1 \end{bmatrix} \quad \rightarrow \quad A' = \begin{bmatrix} 5 \\ 0 \\ 5 \\ 1 \end{bmatrix}$$

$$L = 5$$

Secure Circuit Sketch



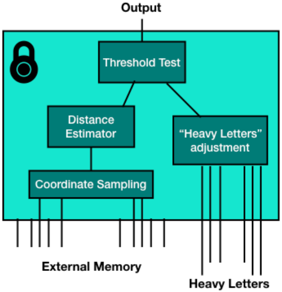
1. Sample multiset S from Alice, Bob.
2. Approximate by sampling from external memory $\|A' - B'\|_2^2$.
3. Compute $\|A_S - B_S\|_2^2 - \|A' - B'\|_2^2$
4. Output "SAME" iff $(2) + (3) \leq \tau$

Entire computation is over a secure circuit. Simulating the output provides security by composition theorems.

Circuit is of size $\tilde{O}_\epsilon(poly(k) \cdot n^2/t^2)$

Communication overhead is a function of security parameter k independent of n (assuming PRG/OT).

Secure Circuit Sketch



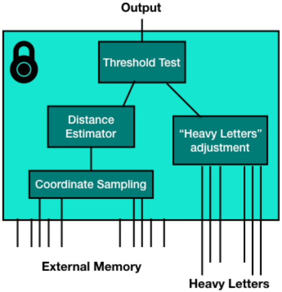
1. Sample multiset S from Alice, Bob.
2. Approximate by sampling from external memory $\|A' - B'\|_2^2$.
3. Compute $\|A_S - B_S\|_2^2 - \|A' - B'\|_2^2$
4. Output "SAME" iff $(2) + (3) \leq \tau$

Entire computation is over a secure circuit. Simulating the output provides security by composition theorems.

Circuit is of size $\tilde{O}_\epsilon(poly(k) \cdot n^2/t^2)$

Communication overhead is a function of security parameter k independent of n (assuming PRG/OT).

Secure Circuit Sketch



1. Sample multiset S from Alice, Bob.
2. Approximate by sampling from external memory $\|A' - B'\|_2^2$.
3. Compute $\|A_S - B_S\|_2^2 - \|A' - B'\|_2^2$
4. Output "SAME" iff $(2) + (3) \leq \tau$

Entire computation is over a secure circuit. Simulating the output provides security by composition theorems.

Circuit is of size $\tilde{O}_\epsilon(poly(k) \cdot n^2/t^2)$

Communication overhead is a function of security parameter k independent of n (assuming PRG/OT).

Conclusions

- **Two Party Closeness Testing** can be computed **securely** with $\tilde{\Theta}_{\epsilon,k}(n^2/t^2)$ communication under standard cryptographic assumptions.
- We also provide (secure) **Two Party Independence Testing** protocols using $\tilde{\Theta}_{\epsilon,k}(n^2m/t^2 + nm/t + \sqrt{m})$ communication.
- We show **tightness** for Closeness Testing, and for some of the parameter regimes of Independence Testing.
- **More Samples** $\Leftrightarrow$ **Less Communication**.

Thank you!

Questions?

Conclusions

- **Two Party Closeness Testing** can be computed **securely** with $\tilde{\Theta}_{\epsilon,k}(n^2/t^2)$ communication under standard cryptographic assumptions.
- We also provide (secure) **Two Party Independence Testing** protocols using $\tilde{\Theta}_{\epsilon,k}(n^2 m/t^2 + nm/t + \sqrt{m})$ communication.
- We show **tightness** for Closeness Testing, and for some of the parameter regimes of Independence Testing.
- **More Samples** $\Leftrightarrow$ **Less Communication**.

Thank you!

Questions?