Lower Bounds for Tolerant Junta and Unateness Testing via Rejection Sampling of Graphs

Amit Levi
Erik Waingarten
The Model
The Model

\[ G = ([n], E) \]
The Model

\[ G = ([n], E) \]

An oracle \( \mathcal{O} \) sampling edges u.a.r
The Model

\[ G = ([n], E) \]

An oracle \( \mathcal{O} \) sampling edges u.a.r

Queries: \( L_1, \ldots, L_t \subseteq [n] \)
The Model

\[ G = ([n], E) \]

An oracle \( \mathcal{O} \) sampling edges u.a.r

Queries: \( L_1, \ldots, L_t \subseteq [n] \)

For each query \( L_j \), the oracle samples a random edge \((u_j, v_j)\)
**The Model**

\[ G = ([n], E) \]

An oracle \( \mathcal{O} \) sampling edges u.a.r

Queries: \( L_1, \ldots, L_t \subseteq [n] \)

For each query \( L_j \), the oracle samples a random edge \((u_j, v_j)\)
and return: \( L_j \cap \{u_j, v_j\} \)
The Model

\[ G = ([n], E) \]

An oracle \( \mathcal{O} \) sampling edges u.a.r

Queries: \( L_1, \ldots, L_t \subseteq [n] \)

For each query \( L_j \), the oracle samples a random edge \( (u_j, v_j) \) and return: \( L_j \cap \{u_j, v_j\} \)

The complexity of the algorithm is defined as:

\[ \sum_{j=1}^{t} |L_j| \]
The Model

An oracle $\mathcal{O}$ sampling edges u.a.r

Queries: $L_1, \ldots, L_t \subseteq [n]$

For each query $L_j$, the oracle samples a random edge $(u_j, v_j)$ and return: $L_j \cap \{u_j, v_j\}$

The complexity of the algorithm is defined as:

$$\sum_{j=1}^{t} |L_j|$$
The Model

\[ G = ([n], E) \]

An oracle \( O \) sampling edges u.a.r

Queries: \( L_1, \ldots, L_t \subseteq [n] \)

For each query \( L_j \), the oracle samples a random edge \( (u_j, v_j) \) and return: \( L_j \cap \{u_j, v_j\} \)

The complexity of the algorithm is defined as:

\[
\sum_{j=1}^{t} |L_j|
\]
The Model

\[ G = ([n], E) \]

An oracle \( \mathcal{O} \) sampling edges u.a.r

Queries: \( L_1, \ldots, L_t \subseteq [n] \)

For each query \( L_j \), the oracle samples a random edge \( (u_j, v_j) \) and return: \( L_j \cap \{u_j, v_j\} \)

The complexity of the algorithm is defined as:

\[ \sum_{j=1}^{t} |L_j| \]
The Model

\[ G = ([n], E) \]

An oracle \( \mathcal{O} \) sampling edges u.a.r

Queries: \( L_1, \ldots, L_t \subseteq [n] \)

For each query \( L_j \), the oracle samples a random edge \( (u_j, v_j) \) and return: \( L_j \cap \{u_j, v_j\} \)

The complexity of the algorithm is defined as:

\[ \sum_{j=1}^{t} |L_j| \]
The Model

\[ G = ([n], E) \]

An oracle \( O \) sampling edges u.a.r

Queries: \( L_1, \ldots, L_t \subseteq [n] \)

For each query \( L_j \), the oracle samples a random edge \((u_j, v_j)\) and return: \( L_j \cap \{u_j, v_j\} \)

The complexity of the algorithm is defined as:

\[ \sum_{j=1}^{t} |L_j| \]
The Model

\[ G = ([n], E) \]

An oracle \( O \) sampling edges u.a.r

Queries: \( L_1, \ldots, L_t \subseteq [n] \)

For each query \( L_j \), the oracle samples a random edge \( (u_j, v_j) \) and return: \( L_j \cap \{u_j, v_j\} \)

The complexity of the algorithm is defined as:

\[ \sum_{j=1}^{t} |L_j| \]
The Model

\[ G = ([n], E) \]

An oracle \( \mathcal{O} \) sampling edges u.a.r

Queries: \( L_1, \ldots, L_t \subseteq [n] \)

For each query \( L_j \), the oracle samples a random edge \( (u_j, v_j) \)
and return: \( L_j \cap \{u_j, v_j\} \)

The complexity of the algorithm is defined as:

\[ \sum_{j=1}^{t} |L_j| \]
The Model

$$G = ([n], E)$$

An oracle $\mathcal{O}$ sampling edges u.a.r

Queries: $L_1, \ldots, L_t \subseteq [n]$

For each query $L_j$, the oracle samples a random edge $(u_j, v_j)$ and return: $L_j \cap \{u_j, v_j\}$

The complexity of the algorithm is defined as:

$$\sum_{j=1}^{t} |L_j|$$
The Model

\[ G = ([n], E) \]

An oracle \( \mathcal{O} \) sampling edges u.a.r

Queries: \( L_1, \ldots, L_t \subseteq [n] \)

For each query \( L_j \), the oracle samples a random edge \( (u_j, v_j) \)
and return: \( L_j \cap \{u_j, v_j\} \)

The complexity of the algorithm is defined as:

\[ \sum_{j=1}^{t} |L_j| \]
Results
Results

Thm 1: Testing Bipartiteness (non adaptively) in the rejection sampling model requires complexity of $\tilde{\Omega}(n^2)$.
Results

**Thm 1**: Testing Bipartiteness (non adaptively) in the rejection sampling model requires complexity of $\tilde{\Omega}(n^2)$. 
**Results**

**Thm 1:** Testing Bipartiteness (non adaptively) in the rejection sampling model requires complexity of $\tilde{O}(n^2)$.

**Def:** a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is a $k$- junta if it depends on at most $k$ of its variables.
Results

**Thm 1:** Testing Bipartiteness (non adaptively) in the rejection sampling model requires complexity of $\tilde{\Omega}(n^2)$.

**Def:** a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is a $k$- junta if it depends on at most $k$ of its variables.

**Thm 2:** There exist $0 < \epsilon_0 < \epsilon_1 < 1$ such that distinguishing a function $\epsilon_0$-close to a $k$-junta and a function $\epsilon_1$-far from any $k$-junta requires $\tilde{\Omega}(k^2)$ non-adaptive queries.
Results

**Thm 1**: Testing Bipartiteness (non adaptively) in the rejection sampling model requires complexity of $\widetilde{\Omega}(n^2)$.

**Def**: a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is a $k$-junta if it depends on at most $k$ of its variables.

**Thm 2**: There exist $0 < \epsilon_0 < \epsilon_1 < 1$ such that distinguishing a function $\epsilon_0$-close to a $k$-junta and a function $\epsilon_1$-far from any $k$-junta requires $\widetilde{\Omega}(k^2)$ non-adaptive queries.

Given the non-adaptive $\widetilde{O}(k^{3/2})$ tester of Blais [Bla08] (where $\epsilon_0 = 0$), we conclude that non-adaptive tolerant junta testing requires more queries than non-tolerant testing.
Results

**Thm 1:** Testing Bipartiteness (non adaptively) in the rejection sampling model requires complexity of $\tilde{\Omega}(n^2)$. 
Results

**Thm 1:** Testing Bipartiteness (non adaptively) in the rejection sampling model requires complexity of $\Omega(n^2)$.

**Def:** a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is unate if it is either non-increasing or non-decreasing in every variable.
Results

**Thm 1**: Testing Bipartiteness (non adaptively) in the rejection sampling model requires complexity of $\tilde{\Omega}(n^2)$.

**Def**: a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is unate if it is either non-increasing or non-decreasing in every variable.

**Thm 2** : There exist $0 < \epsilon_0 < \epsilon_1 < 1$ such that distinguishing a function $\epsilon_0$-close to a unate and a function $\epsilon_1$-far from any unate function requires $\tilde{\Omega}(n^{3/2})$ non-adaptive queries.
Results

**Thm 1**: Testing Bipartiteness (non adaptively) in the rejection sampling model requires complexity of $\tilde{\Omega}(n^2)$.

**Def**: a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is unate if it is either non-increasing or non-decreasing in every variable.

**Thm 2**: There exist $0 < \epsilon_0 < \epsilon_1 < 1$ such that distinguishing a function $\epsilon_0$-close to a unate and a function $\epsilon_1$-far from any unate function requires $\tilde{\Omega}(n^{3/2})$ non-adaptive queries.

**Thm 3**: There exist $0 < \epsilon_0 < \epsilon_1 < 1$ such that distinguishing a function $\epsilon_0$-close to a unate and a function $\epsilon_1$-far from any unate function requires $\tilde{\Omega}(n)$ queries.
Results

**Thm 1**: Testing Bipartiteness (non adaptively) in the rejection sampling model requires complexity of $\tilde{\Omega}(n^2)$.

**Def**: a function $f : \{0, 1\}^n \to \{0, 1\}$ is unate if it is either non-increasing or non-decreasing in every variable.

**Thm 2**: There exist $0 < \epsilon_0 < \epsilon_1 < 1$ such that distinguishing a function $\epsilon_0$-close to a unate and a function $\epsilon_1$-far from any unate function requires $\tilde{\Omega}(n^{3/2})$ non-adaptive queries.

**Thm 3**: There exist $0 < \epsilon_0 < \epsilon_1 < 1$ such that distinguishing a function $\epsilon_0$-close to a unate and a function $\epsilon_1$-far from any unate function requires $\tilde{\Omega}(n)$ queries.

Given the non-adaptive $\tilde{O}(n)$ tester of Baleshzar et al. [BCP+17] (where $\epsilon_0 = 0$), and the adaptive $\tilde{O}(n^{3/4})$ tester of Chen et al. [CWX17], we conclude that tolerant unateness testing requires more queries than non-tolerant testing in both settings.