A Rate-Distortion Theory for Permutation Spaces

Da Wang EECS, MIT

Arya Mazumdar ECE, Univ. Minnesota

Gregory W. Wornell EECS, MIT

ISIT 2013, Istanbul, Turkey

July 11, 2013

Codes in permutations

Error correction in permutations

- Codes with hamming distance: [I. Blake *et al.*, 1979]
- Codes with Chebyshev distance: [T. Klve *et al.*, 2010], [A. Barg and A. Mazumdar, 2010]
  - Application: rank modulation for flash memory

Lossy compression of permutations

- Largely left unattended
- Lossless compression of permutations for efficient rank query and selection: [J. Barbay and G. Navarro, 2009 & 2011], [J. Barbay *et al.*, 2012]

Given a group of elements with distinct values:

Comparison-based sorting: search for the true permutation by pairwise comparisons.

Algorithm 101: exact sorting

- To specify a permutation, need $\log_2 n! \approx \Theta(n \log n)$ bits.
- Each comparison: at most 1 bit of information
  $\Rightarrow$ need $\Omega(n \log n)$ comparisons

Given a group of elements with distinct values:

Comparison-based approximate sorting: search for the true permutation subject to certain distortion by pairwise comparisons.

Algorithm 101: exact sorting

- To specify a permutation, need $\log_2 n! \approx \Theta(n \log n)$ bits.
- Each comparison: at most 1 bit of information
  $\Rightarrow$ need $\Omega(n \log n)$ comparisons

Approximate sorting

- How many comparisons do we need for sorting with distortion D?
- How many bits do we need for specifying a permutation with distortion D?

Permutation and (approximate) sorting

Given a group of elements with distinct values:

Comparison-based approximate sorting: search for the true permutation subject to certain distortion by pairwise comparisons.

Algorithm 101: exact sorting

- To specify a permutation, need $\log_2 n! \approx \Theta(n \log n)$ bits.
- Each comparison: at most 1 bit of information
  $\Rightarrow$ need $\Omega(n \log n)$ comparisons

Approximate sorting

- How many comparisons do we need for sorting with distortion D?
- How many bits do we need for specifying a permutation with distortion D?

Rate-distortion theory!

Permutation space

- $\mathcal{S}_n$: the set of $n!$ permutations
- d: distance measure

(n, D_n) source code

- $\mathcal{C}_n \subset \mathcal{S}_n$
- for any $\sigma \in \mathcal{S}_n$, there exists $\pi \in \mathcal{C}_n$ that
$$d(\pi, \sigma) \leq D_n.$$

Rate-distortion function

Let $A(n, D_n)$ be the minimum size of the (n, D_n) source codes with distortion D_n. The minimal rate for distortion D_n is

$$R(D_n) \triangleq \frac{\log A(n, D_n)}{\log n!},$$

and the rate-distortion function is

$$R(D) \triangleq \lim_{n \to \infty} R(D_n).$$

Vector representations

- the permutation vector σ
- the inverse permutation vector σ^{-1}: $\sigma \circ \sigma^{-1} = e = [1, 2, \dots, n]$
- the inversion vector of the permutation $\mathbf{x}_\sigma$

Distances between vectors

- Kendall tau distance
- ℓ_p distances, $p = 1, 2, \dots, \infty$

In this work

Two specific permutation spaces:

- Kendall tau distance of the permutation vectors
- ℓ_1 distance of the inversion vectors

- The *Kendall tau distance* $d_\tau(\sigma_1, \sigma_2)$:
  the minimum number of swaps of adjacent elements required to change σ_1 into σ_2.

Example

- $\sigma_1 = [1, 5, 4, 2, 3]$ and $\sigma_2 = [3, 4, 5, 1, 2]$
- $d_\tau\left(\sigma_1, \sigma_2\right) = ?$

$$\sigma_1 = [1, 5, 4, 2, 3]$$

- The *Kendall tau distance* $d_\tau(\sigma_1, \sigma_2)$:
  the minimum number of swaps of adjacent elements required to change σ_1 into σ_2.

Example

- $\sigma_1 = [1, 5, 4, 2, 3]$ and $\sigma_2 = [3, 4, 5, 1, 2]$
- $d_\tau(\sigma_1, \sigma_2) = ?$

$$\sigma_1 = [1, 5, 4, \underline{2, 3}] \to [1, \underline{5, 4}, 3, 2]$$

Distance measure of permutations

Kendall tau distance

- The *Kendall tau distance* $d_\tau(\sigma_1, \sigma_2)$:
  the minimum number of swaps of adjacent elements required to change σ_1 into σ_2.

Example

- $\sigma_1 = [1, 5, 4, 2, 3]$ and $\sigma_2 = [3, 4, 5, 1, 2]$
- $d_\tau(\sigma_1, \sigma_2) = ?$

$$\sigma_1 = [1, 5, 4, \underline{2, 3}] \rightarrow [1, \underline{5, 4}, 3, 2] \rightarrow [\underline{1, 4}, 5, 3, 2]$$

- The *Kendall tau distance* $d_\tau(\sigma_1, \sigma_2)$:
  the minimum number of swaps of adjacent elements required to change σ_1 into σ_2.

Example

- $\sigma_1 = [1, 5, 4, 2, 3]$ and $\sigma_2 = [3, 4, 5, 1, 2]$
- $d_\tau(\sigma_1, \sigma_2) = ?$

$$\sigma_1 = [1, 5, 4, \underline{2, 3}] \rightarrow [1, \underline{5, 4}, 3, 2] \rightarrow [\underline{1, 4}, 5, 3, 2] \rightarrow [4, \underline{1, 5}, 3, 2]$$

- The *Kendall tau distance* $d_\tau(\sigma_1, \sigma_2)$:
  the minimum number of swaps of adjacent elements required to change σ_1 into σ_2.

Example

- $\sigma_1 = [1, 5, 4, 2, 3]$ and $\sigma_2 = [3, 4, 5, 1, 2]$
- $d_\tau\left(\sigma_1, \sigma_2\right) =?$

$$\sigma_1 = [1, 5, 4, \underline{2, 3}] \rightarrow [1, \underline{5, 4}, 3, 2] \rightarrow [\underline{1, 4}, 5, 3, 2] \rightarrow [4, \underline{1, 5}, 3, 2]$$
$$\rightarrow [4, 5, \underline{1, 3}, 2]$$

- The *Kendall tau distance* $d_\tau(\sigma_1, \sigma_2)$:
  the minimum number of swaps of adjacent elements required to change σ_1 into σ_2.

Example

- $\sigma_1 = [1, 5, 4, 2, 3]$ and $\sigma_2 = [3, 4, 5, 1, 2]$
- $d_\tau(\sigma_1, \sigma_2) = ?$

$$\sigma_1 = [1, 5, 4, \underline{2, 3}] \rightarrow [1, \underline{5, 4}, 3, 2] \rightarrow [\underline{1, 4}, 5, 3, 2] \rightarrow [4, \underline{1, 5}, 3, 2]$$
$$\rightarrow [4, 5, \underline{1, 3}, 2] \rightarrow [4, \underline{5, 3}, 1, 2]$$

- The *Kendall tau distance* $d_\tau(\sigma_1, \sigma_2)$:
  the minimum number of swaps of adjacent elements required to change σ_1 into σ_2.

Example

- $\sigma_1 = [1, 5, 4, 2, 3]$ and $\sigma_2 = [3, 4, 5, 1, 2]$
- $d_\tau(\sigma_1, \sigma_2) = ?$

$$\sigma_1 = [1, 5, 4, \underline{2, 3}] \rightarrow [1, \underline{5, 4}, 3, 2] \rightarrow [\underline{1, 4}, 5, 3, 2] \rightarrow [4, \underline{1, 5}, 3, 2]$$
$$\rightarrow [4, 5, \underline{1, 3}, 2] \rightarrow [4, \underline{5, 3}, 1, 2] \rightarrow [\underline{4, 3}, 5, 1, 2]$$

Distance measure of permutations

Kendall tau distance

- The *Kendall tau distance* $d_\tau(\sigma_1, \sigma_2)$:
  the minimum number of swaps of adjacent elements required to change σ_1 into σ_2.

Example

- $\sigma_1 = [1, 5, 4, 2, 3]$ and $\sigma_2 = [3, 4, 5, 1, 2]$
- $d_\tau(\sigma_1, \sigma_2) = ?$

$$\begin{aligned}\sigma_1 = [1, 5, 4, \underline{2, 3}] &\rightarrow [1, \underline{5, 4}, 3, 2] \rightarrow [\underline{1, 4}, 5, 3, 2] \rightarrow [4, \underline{1, 5}, 3, 2] \\ &\rightarrow [4, 5, \underline{1, 3}, 2] \rightarrow [4, \underline{5, 3}, 1, 2] \rightarrow [\underline{4, 3}, 5, 1, 2] \\ &\rightarrow [3, 4, 5, 1, 2]\end{aligned}$$

- The *Kendall tau distance* $d_\tau(\sigma_1, \sigma_2)$:
  the minimum number of swaps of adjacent elements required to change σ_1 into σ_2.

Example

- $\sigma_1 = [1, 5, 4, 2, 3]$ and $\sigma_2 = [3, 4, 5, 1, 2]$
- $d_\tau(\sigma_1, \sigma_2) = 7$

$$\sigma_1 = [1, 5, 4, \underline{2, 3}] \to [1, \underline{5, 4}, 3, 2] \to [\underline{1, 4}, 5, 3, 2] \to [4, \underline{1, 5}, 3, 2]$$
$$\to [4, 5, \underline{1, 3}, 2] \to [4, \underline{5, 3}, 1, 2] \to [\underline{4, 3}, 5, 1, 2]$$
$$\to [3, 4, 5, 1, 2] = \sigma_2$$

- The *Kendall tau distance* $d_\tau(\sigma_1, \sigma_2)$:
  the minimum number of swaps of adjacent elements required to change σ_1 into σ_2.

Example

- $\sigma_1 = [1, 5, 4, 2, 3]$ and $\sigma_2 = [3, 4, 5, 1, 2]$
- $d_\tau(\sigma_1, \sigma_2) = 7$

$$\begin{aligned}\sigma_1 = [1, 5, 4, \underline{2, 3}] &\rightarrow [1, \underline{5, 4}, 3, 2] \rightarrow [\underline{1, 4}, 5, 3, 2] \rightarrow [4, \underline{1, 5}, 3, 2] \\ &\rightarrow [4, 5, \underline{1, 3}, 2] \rightarrow [4, \underline{5, 3}, 1, 2] \rightarrow [\underline{4, 3}, 5, 1, 2] \\ &\rightarrow [3, 4, 5, 1, 2] = \sigma_2\end{aligned}$$

Properties

- upper bounded by $\binom{n}{2}$
- $d_\tau(\sigma, e)$ = number of swaps in bubble sort

Inversion

- An *inversion* in a permutation σ: a pair $(\sigma(i), \sigma(j))$ such that $i < j$ and $\sigma(i) > \sigma(j)$.
  - Inversions in $\sigma_1 = [1, 5, 4, 2, 3]$: $(5, 4), (5, 2), (5, 3), (4, 2), (4, 3)$
  - Inversions in $\sigma_2 = [3, 4, 5, 1, 2]$: $(3, 1), (3, 2), (4, 1), (4, 2), (5, 1), (5, 2)$

Inversion vector $\mathbf{x}_\sigma \in [0 : 1] \times [0 : 2] \times \cdots \times [0 : n-1]$

$$\mathbf{x}_\sigma(i) = \text{the number of inversions in } \sigma \text{ in which } i+1 \text{ is the first element}$$
$$i = 1, 2, \ldots, n-1.$$

Examples

$$\sigma_1 = [1, 5, 4, 2, 3] \Rightarrow \mathbf{x}_{\sigma_1} = [0, 0, 2, 3]$$
$$\sigma_2 = [3, 4, 5, 1, 2] \Rightarrow \mathbf{x}_{\sigma_2} = [0, 2, 2, 2]$$
$$d_{\mathbf{x},\ell_1}(\sigma_1, \sigma_2) = d_{\ell_1}([0, 0, 2, 3], [0, 2, 2, 2]) = 3$$

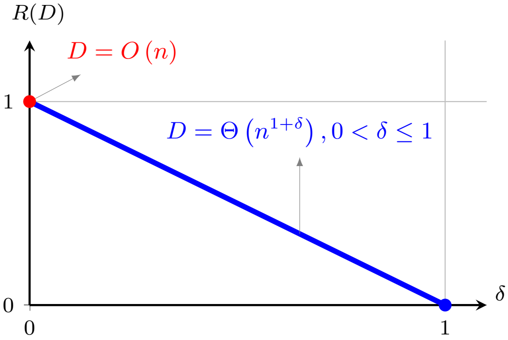
- Inversion vector: a common measure of sortedness
- $d_{\mathbf{x},\ell_1}(e, \sigma)$: evaluation metric for ranking system

Main result

Theorem (Rate distortion function)

In both permutation spaces $\mathcal{X}(\mathcal{S}_n, d_\tau)$ *and* $\mathcal{X}(\mathcal{S}_n, d_{\mathbf{x},\ell_1})$,

$$R(D) = \begin{cases} 1 & \text{if } D = O(n) \\ 1-\delta & \text{if } D = \Theta\left(n^{1+\delta}\right), \quad 0 < \delta \leq 1 \end{cases}.$$



Remarks

- Given two permutations σ_1 and σ_2, [A. Mazumdar *et al.*, 2013] shows

$$d_{\mathbf{x},\ell_1}(\sigma_1,\sigma_2) \leq d_\tau(\sigma_1,\sigma_2)$$

  - (n,D) code for $d_\tau(\cdot,\cdot) \Rightarrow (n,D)$ code for $d_{\mathbf{x},\ell_1}(\cdot,\cdot)$
  - No known non-trivial lower bound for $d_{\mathbf{x},\ell_1}(\sigma_1,\sigma_2)$ in terms of $d_\tau(\sigma_1,\sigma_2)$.

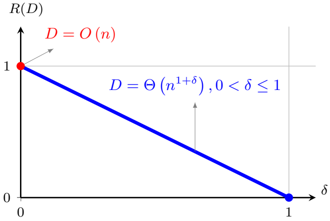
- In the moderate distortion regime, where

$$D = \Theta\left(n^{1+\delta}\right), \quad 0 \leq \delta < 1,$$

the rate for worst-case distortion is the same for average-case distortion with uniform distribution on $\mathcal{S}_n$.

Two end-points of the rate distortion function

$$D_n = O(n) \Rightarrow \quad R(D) = 1 \quad \text{(small distortion)}$$
$$D_n = \Omega(n^2) \Rightarrow \quad R(D) = 0 \quad \text{(large distortion).}$$



Higher order rates in the codebook size

$$r(D_n) \triangleq \log A(n, D_n) - R(D) \cdot \log n!$$

- $r(D_n) \leq 0$ when $R(D) = 1$
- $r(D_n) \geq 0$ when $R(D) = 0$
- Exact characterization is open: present upper and lower bounds.
  - upper bound: achievability
  - lower bound: converse

Kendall tau distance

In the permutation space $\mathcal{X}(\mathcal{S}_n, d_\tau)$, when $D = an^\delta, 0 < \delta \leq 1$,

$$\underline{r^{\mathrm{s}}_\tau(D)} \leq r(D) \leq \overline{r^{\mathrm{s}}_\tau(D)},$$

where

$$\underline{r^{\mathrm{s}}_\tau(D)} = \begin{cases} -a(1-\delta)n^\delta \log n + O\left(n^\delta\right), & 0 < \delta < 1 \\ -n\left[(1+a)\log(1+a) - a\log a\right] + o(n), & \delta = 1 \end{cases},$$

$$\overline{r^{\mathrm{s}}_\tau(D)} = \begin{cases} -n^\delta \frac{a\log 2}{2} + O(1), & 0 < a < 1 \\ -n^\delta \frac{\log \lfloor 2a \rfloor !}{\lfloor 2a \rfloor} + O(1), & a \geq 1 \end{cases}.$$

When $D = bn^2, 0 < b \leq 1/2$, $\underline{r^{\mathrm{l}}_\tau(D)} \leq r(D) \leq \overline{r^{\mathrm{l}}_\tau(D)}$, where

$$\underline{r^{\mathrm{l}}_\tau(D)} = \max\left\{0, n\log 1/\left(2be^2\right)\right\},$$

$$\overline{r^{\mathrm{l}}_\tau(D)} = n\log\lceil 1/(2b)\rceil + O(\log n).$$

ℓ_1 distance of inversion vectors

In the permutation space $\mathcal{X}\left(\mathcal{S}_n, d_{\mathbf{x},\ell_1}\right)$, when $D = an^\delta, 0 < \delta \leq 1$,

$$\underline{r^{\mathrm{s}}_{\mathbf{x},\ell_1}(D)} \leq r(D) \leq \overline{r^{\mathrm{s}}_{\mathbf{x},\ell_1}(D)},$$

where $\underline{r^{\mathrm{s}}_{\mathbf{x},\ell_1}(D)} = \underline{r^{\mathrm{s}}_{\tau}(D)} - n^\delta \log 2$ and

$$\overline{r_{\mathbf{x},\ell_1}(D)} = \begin{cases} -\left\lfloor n^\delta \right\rfloor \log(2a-1) & a > 1 \\ -\left\lceil an^\delta \right\rceil \log 3 & 0 < a \leq 1 \end{cases}.$$

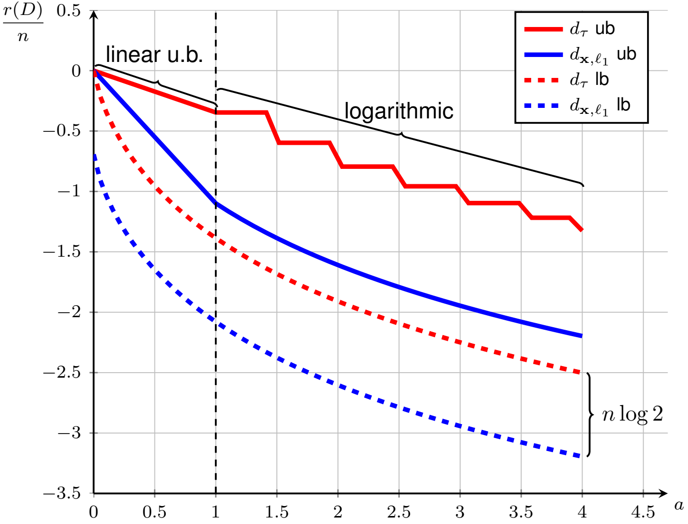
When $D = bn^2, 0 < b \leq 1/2$,

$$\underline{r^{\mathrm{l}}_{\mathbf{x},\ell_1}(D)} \leq r(D) \leq \overline{r^{\mathrm{l}}_{\mathbf{x},\ell_1}(D)},$$

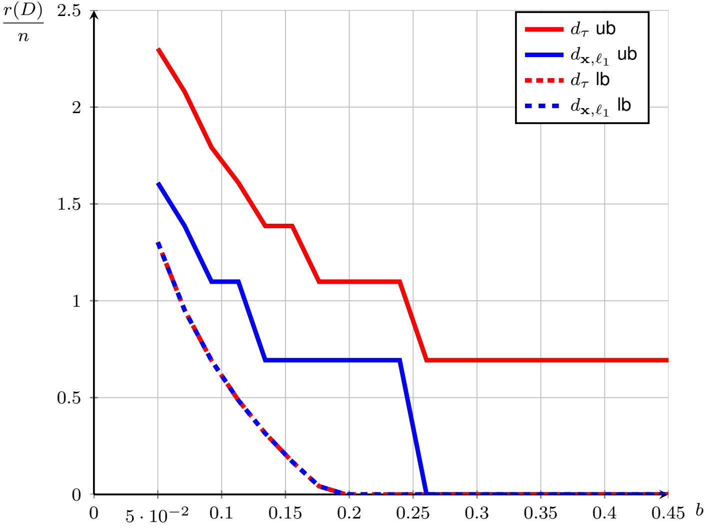
where $\underline{r^{\mathrm{l}}_{\mathbf{x},\ell_1}(D)} = \underline{r^{\mathrm{l}}_{\tau}(D)}$ and $\overline{r^{\mathrm{l}}_{\mathbf{x},\ell_1}(D)} = n \log \left\lceil 1/(4b) \right\rceil + O\left(1\right).$

Higher order rates

Small distortion region: $D = an$



Large distortion region: $D = bn^2$



Achievability

Achievability for Kendall tau distance

Sorting subsequences

Quantization by sorting subsequences

Given $\sigma \in \mathcal{S}_n$, we quantize it to π by sorting k subsequence with length m

$\sigma_1, \sigma_2, \ldots, \sigma_m$	$\sigma_{m+1}, \ldots, \sigma_{2m}$	$\cdots$	$\sigma_{(k-1)m+1}, \ldots, \sigma_{km}$	$\sigma_{km+1}, \ldots, \sigma_n$
sort	sort		sort	keep

- **Codebook size**

$$|\mathcal{C}(k, m, n)| = n!/\left(m!^k\right)$$

- **Maximal distortion**

$$D(k, m) \leq km^2/2$$

To achieve moderate distortion $D = \Theta\left(n^{1+\delta}\right), 0 \leq \delta < 1$

$$\begin{cases} m \approx 2D/n = \Theta\left(n^{\delta}\right) \\ k \approx n/m = \Theta\left(n^{1-\delta}\right) \end{cases} \Rightarrow |\mathcal{C}(k, m, n)| \approx n \log n - k \cdot m \log m = (1-\delta) n \log n$$

Component-wise scalar quantization

- Quantize the k-th coordinate uniformly by m_k points ($k = 2, 3, \ldots, n$)
- Product structure of the space: $[0:1] \times [0:2] \times \cdots \times [0:n-1]$

- Codebook size:

$$M_n = \prod_{k=2}^{n} m_k$$

- Maximal distortion

$$D_n = \sum_{k=2}^{n} D_k$$

$$D_k = \lceil (k/m_k - 1)/2 \rceil$$

To achieve moderate distortion $D = \Theta\left(n^{1+\delta}\right), 0 \le \delta < 1$

$$m_k \approx \frac{n^2}{2D} = \Theta\left(n^{1-\delta}\right) \quad \Rightarrow \begin{cases} D_k & \approx \frac{kD}{n^2} = k\Theta\left(D^{1-\delta}\right) \\ M_n & \approx (1-\delta) n \log n \end{cases}.$$

Converse

D-balls $B_d(\sigma, D)$

- Distance measure: $d(\cdot, \cdot)$
- Center: σ
- Radius: D
- Maximum size: $N_d(D)$.

$n!$ divided by the upper bound of $N_d(D)$ provides converse results.

Key lemmas via combinatorial arguments

Kendall tau distance

For $0 \leq D \leq n$,

$$N_\tau (D) \leq \binom{n+D-1}{D}.$$

ℓ_1 distance of the inversion vectors

For $0 \leq D \leq n(n-1)/2$,

$$N_{\mathbf{x},\ell_1} (D) \leq 2^{\min\{n,D\}} \binom{n+D}{D}.$$

Recap

- Information theory provides the fundamental trade-off between complexity and accuracy in approximate sorting.
  - Can be generalized to other comparison-based algorithms.
- Achievability: sorting subsequences and quantizing coordinates
  - both support successive refinement
- Converse: geometry of the permutation spaces

Future directions

- Sharper bounds for higher order rates
- Other distance measures of interest
- Design of approximate sorting algorithms

Backup slides

Equivalent procedure in the inverse permutation domain

1. Construct a vector $a(\sigma)$ such that for $1 \leq i \leq k$,

$$a(i) = j \text{ if } \sigma^{-1}(i) \in [(j-1)m+1, jm], 1 \leq j \leq k.$$

   Then a contains exactly m values of integers j.
2. Form a permutation π' by replacing the length-m subsequence of a that corresponds to value j by vector $[(j-1)m+1, (j-1)m+2, \ldots, jm]$.

- The total number of inversions in σ is $I_n(\sigma)$.
  - $I_5(\sigma_1) = 5$, $I_5(\sigma_2) = 6$
- The number of permutations with k inversions:

$$K_n(k) \triangleq |\{\sigma \in \mathcal{S}_n : I_n(\sigma) = k\}|$$

Lemma

For $0 \leq D \leq n$,

$$N_{\tau}(D) \leq \binom{n+D-1}{D}.$$

Proof sketch.

Let the number of permutations in $\mathcal{S}_n$ with at most k inversions be $T_n(d) \triangleq \sum_{k=0}^{d} K_n(k)$. Then

$$N_{\tau}(D) = T_n(D),$$

By induction, $T_n(D) = K_{n+1}(D)$ when $D \leq n$. Then noting that for $k < n$, $K_n(k) = K_n(k-1) + K_{n-1}(k)$ ([Knuth 1998, Section 5.1.1]) and for any $n \geq 2$,

$$K_n(0) = 1, \quad K_n(1) = n-1, \quad K_n(2) = \binom{n}{2} - 1,$$

The proof can be completed by induction. □

Lemma

For $0 \leq D \leq n(n-1)/2$*,*

$$N_{\mathbf{x},\ell_1}(D) \leq 2^{\min\{n,D\}}\binom{n+D}{D}.$$

Proof sketch.

For any $\sigma \in \mathcal{S}_n$, let $\mathbf{x} = \mathbf{x}_\sigma \in \mathcal{G}_n$, then $|B_{\mathbf{x},\ell_1}(D)| = \sum_{r=0}^{D} |\{\mathbf{y} \in \mathcal{G}_n : d_{\ell_1}(\mathbf{x},\mathbf{y}) = r\}|$. Let $\mathbf{d} \triangleq |\mathbf{x}-\mathbf{y}|$, and $Q(n,r)$ be the number of integer solutions of the equation $z_1 + z_2 + \ldots + z_n = r$ with $z_i \geq 0, 0 \leq i \leq n$, then noting $|\{\mathbf{y} \in \mathcal{G}_n : d_{\ell_1}(\mathbf{x},\mathbf{y}) = r\}| \leq 2^{\min\{n,D\}}Q(n,r)$ and by upper bounding $Q(n,r)$ we complete the proof. □