Lower bounds on designs in symmetric spaces

Alex Samorodnitsky

Hebrew University

August 31, 2010

What is a design?

Definition

A design of strength t is a subset $D \subseteq X$ such that for any simple function f on X holds

$$\frac{1}{|D|} \sum_{x \in D} f(x) = \int_X f(x) d\mu(x)$$

What is a design?

Definition

A design of strength t is a subset $D \subseteq X$ such that for any simple function f on X holds

$$\frac{1}{|D|} \sum_{x \in D} f(x) = \int_X f(x) d\mu(x)$$

- Construct small designs.

What is a design?

Definition

A design of strength t is a subset $D \subseteq X$ such that for any simple function f on X holds

$$\frac{1}{|D|}\sum_{x\in D} f(x) = \int_X f(x)d\mu(x)$$

- Construct small designs.
- Or... show no small designs exist.

What is a design?

Definition

A design of strength t is a subset $D \subseteq X$ such that for any simple function f on X holds

$$\frac{1}{|D|} \sum_{x \in D} f(x) = \int_X f(x) d\mu(x)$$

- Construct small designs.
- Show no small designs exist.
- Here: the second goal.

Why?

- Interesting for Numerical Analysis : mechanical quadrature.

Why?

- Interesting for Numerical Analysis : mechanical quadrature.
- Theoretical CS: a pseudo-random set

$$\left|\mathbb{E}_D f - \mathbb{E}_X f\right| < \epsilon$$

for any simple f

Why?

- Interesting for Numerical Analysis : mechanical quadrature.
- Theoretical CS: a pseudo-random set

$$\left|\mathbb{E}_D f - \mathbb{E}_X f\right| < \epsilon$$

for any simple f

- Combinatorics: Some designs turn out to be very nice combinatorial objects, such as combinatorial designs.

Why?

- Interesting for Numerical Analysis : mechanical quadrature.
- Theoretical CS: a pseudo-random set

$$\left|\mathbb{E}_D f - \mathbb{E}_X f\right| < \epsilon$$

for any simple f

- Combinatorics: Some designs turn out to be very nice combinatorial objects, such as combinatorial designs.
- Somewhere in between: The (main) goal of this line of research is to obtain a better understanding of the Delsarte theory of linear programming bounds for codes and designs.

Some spaces

- We are interested in symmetric spaces

Some spaces

- We are interested in symmetric spaces
- That is, mostly spaces with a strong group of symmetries.

Some spaces

- We are interested in symmetric spaces
- That is, mostly spaces with a strong group of symmetries.
- Examples

1. X is the Hamming cube $\{0, 1\}^n$ or $\{-1, 1\}^n$. The group of symmetries **S** is generated by shifts $T_a: \ x \to x + a$ and permutations of coordinates.

Some spaces

- We are interested in symmetric spaces
- That is, mostly spaces with a strong group of symmetries.
- Examples

1. X is the Hamming cube $\{0, 1\}^n$ or $\{-1, 1\}^n$. The group of symmetries **S** is generated by shifts $T_a : \ x \to x + a$ and permutations of coordinates.
2. $$X = \mathbb{S}^{n-1}; \quad \mathbf{S} = \text{Rotations by the orthogonal group}$$

Some spaces

- We are interested in symmetric spaces
- That is, mostly spaces with a strong group of symmetries.
- Examples

1. X is the Hamming cube $\{0,1\}^n$ or $\{-1,1\}^n$. The group of symmetries $\mathbf{S}$ is generated by shifts $T_a: \ x \to x + a$ and permutations of coordinates.
2. $$X = \mathbb{S}^{n-1}; \quad \mathbf{S} = \text{Rotations by the orthogonal group}$$
3. X is the Johnson space, $\binom{[n]}{k}$, also known as the Hamming sphere. The group of symmetries $\mathbf{S}$ are the permutations.

Some spaces

- We are interested in symmetric spaces
- That is, mostly spaces with a strong group of symmetries.
- Examples
  1. X is the Hamming cube $\{0,1\}^n$ or $\{-1,1\}^n$. The group of symmetries $\mathbf{S}$ is generated by shifts $T_a : \ x \to x + a$ and permutations of coordinates.
  2. $$X = \mathbb{S}^{n-1}; \quad \mathbf{S} = \text{Rotations by the orthogonal group}$$
  3. X is the Johnson space, $\binom{[n]}{k}$, also known as the Hamming sphere. The group of symmetries $\mathbf{S}$ are the permutations.
- In all these examples $\mathbf{S}$ is 2-transitive.

Notions of simplicity

- For the Hamming cube $X = \{-1, 1\}^n$, simple functions of strength t are multilinear polynomials of degree at most t, that is the span of $\prod_{i \in T} x_i$ for $T \subseteq [n], \ |T| \leq t$.

Notions of simplicity

- For the Hamming cube $X = \{-1, 1\}^n$, simple functions of strength t are multilinear polynomials of degree at most t, that is the span of $\prod_{i \in T} x_i$ for $T \subseteq [n], \ |T| \leq t$.
- For the Euclidean sphere $\mathbb{S}^{n-1}$, simple functions of strength t are multivariate real polynomials of degree at most t restricted to the sphere.

Notions of simplicity

- For the Hamming cube $X = \{-1, 1\}^n$, simple functions of strength t are multilinear polynomials of degree at most t, that is the span of $\prod_{i \in T} x_i$ for $T \subseteq [n],\ |T| \leq t$.
- For the Euclidean sphere $\mathbb{S}^{n-1}$, simple functions of strength t are multivariate real polynomials of degree at most t restricted to the sphere.
- For the Hamming sphere...

Notions of simplicity

- For the Hamming cube $X = \{-1, 1\}^n$, simple functions of strength t are multilinear polynomials of degree at most t, that is the span of $\prod_{i \in T} x_i$ for $T \subseteq [n], \ |T| \leq t$.
- For the Euclidean sphere $\mathbb{S}^{n-1}$, simple functions of strength t are multivariate real polynomials of degree at most t restricted to the sphere.
- For the Hamming sphere... There is a notion of simplicity leading to nice designs

Example: One-dimensional sphere, aka the circle

- $$\mathbb{S}^1 \;=\; (cos(\phi), sin(\phi)), \;\; 0 \leq \phi < 2\pi$$

Example: One-dimensional sphere, aka the circle

- $$\mathbb{S}^1 = (cos(\phi), sin(\phi)), \;\; 0 \leq \phi < 2\pi$$
- Simple functions are spanned by $cos(k\phi)$, $sin(k\phi)$, $k = 0, ..., t$ on $[0, 2\pi)$.

Example: One-dimensional sphere, aka the circle

- $$\mathbb{S}^1 = (cos(\phi), sin(\phi)), \;\; 0 \leq \phi < 2\pi$$
- Simple functions are spanned by $cos(k\phi)$, $sin(k\phi)$, $k = 0, ..., t$ on $[0, 2\pi)$.
- We need all of these basis functions f to satisfy the design condition.

Example: One-dimensional sphere, aka the circle

- $$\mathbb{S}^1 \;=\; (cos(\phi), sin(\phi)), \;\; 0 \leq \phi < 2\pi$$

- Simple functions are spanned by $cos(k\phi)$, $sin(k\phi)$, $k = 0, ..., t$ on $[0, 2\pi)$.

- We need all of these basis functions f to satisfy the design condition. That is, for $0 < k \leq t$,

$$\frac{1}{|D|}\sum_{\phi \in D} cos(k\phi) = \frac{1}{|D|}\sum_{\phi \in D} sin(k\phi) = 0$$

Circle continued

- For $t = 0$, take $D = \{0\}$.

Circle continued

- For $t = 0$, take $D = \{0\}$.
- For $t = 1$, take $D = \{0, \pi\}$.

$$cos(0) + cos(\pi) = sin(0) + sin(\pi) = 0$$

Circle continued

- For $t = 0$, take $D = \{0\}$.
- For $t = 1$, take $D = \{0, \pi\}$.

$$cos(0) + cos(\pi) = sin(0) + sin(\pi) = 0$$

- For $t = 2$, take $D = \{0, \frac{2\pi}{3}, \frac{4\pi}{3}\}$.

$$cos(0) + cos\left(\frac{2\pi}{3}\right) + cos\left(\frac{4\pi}{3}\right) = sin(0) + sin\left(\frac{2\pi}{3}\right) + sin\left(\frac{4\pi}{3}\right) = 0$$

Circle continued

- For $t = 0$, take $D = \{0\}$.
- For $t = 1$, take $D = \{0, \pi\}$.

$$cos(0) + cos(\pi) = sin(0) + sin(\pi) = 0$$

- For $t = 2$, take $D = \{0, \frac{2\pi}{3}, \frac{4\pi}{3}\}$.

$$cos(0)+cos\left(\frac{2\pi}{3}\right)+cos\left(\frac{4\pi}{3}\right) = sin(0)+sin\left(\frac{2\pi}{3}\right)+sin\left(\frac{4\pi}{3}\right) = 0$$

- For any t, an equipartition of the circle into $t + 1$ points is a minimal design.

Circle continued

- For $t = 0$, take $D = \{0\}$.
- For $t = 1$, take $D = \{0, \pi\}$.

$$cos(0) + cos(\pi) = sin(0) + sin(\pi) = 0$$

- For $t = 2$, take $D = \left\{0, \frac{2\pi}{3}, \frac{4\pi}{3}\right\}$.

$$cos(0)+cos\left(\frac{2\pi}{3}\right)+cos\left(\frac{4\pi}{3}\right) = sin(0)+sin\left(\frac{2\pi}{3}\right)+sin\left(\frac{4\pi}{3}\right) = 0$$

- For any t, an equipartition of the circle into $t + 1$ points is a minimal design.
- Nice question, nice answer.

Second example: The Hamming cube

- Simple functions are spanned by $\prod_{i \in T} x_i$, for $0 \leq |T| \leq t$.

Second example: The Hamming cube

- Simple functions are spanned by $\prod_{i \in T} x_i$, for $0 \leq |T| \leq t$.
- We need all of these basis functions f to satisfy the design condition.

Second example: The Hamming cube

- Simple functions are spanned by $\prod_{i \in T} x_i$, for $0 \leq |T| \leq t$.
- We need all of these basis functions f to satisfy the design condition. That is, for $0 < k \leq t$, and for any subset $T \subseteq [n]$ with $|T| = k$

$$\sum_{x \in D} \prod_{i \in T} x_i = 0$$

Second example: The Hamming cube

- Simple functions are spanned by $\prod_{i \in T} x_i$, for $0 \leq |T| \leq t$.
- We need all of these basis functions f to satisfy the design condition. That is, for $0 < k \leq t$, and for any subset $T \subseteq [n]$ with $|T| = k$

$$\sum_{x \in D} \prod_{i \in T} x_i = 0$$

- For $t = 1$, want for all $i = 1 \ldots n$

$$\sum_{x \in D} x_i = 0$$

Second example: The Hamming cube

- Simple functions are spanned by $\prod_{i \in T} x_i$, for $0 \leq |T| \leq t$.
- We need all of these basis functions f to satisfy the design condition. That is, for $0 < k \leq t$, and for any subset $T \subseteq [n]$ with $|T| = k$

$$\sum_{x \in D} \prod_{i \in T} x_i = 0$$

- For $t = 1$, want for all $i = 1 \ldots n$

$$\sum_{x \in D} x_i = 0$$

- That is, any coordinate to be 1 and -1 same number of times on D.

Second example: The Hamming cube

- Simple functions are spanned by $\prod_{i \in T} x_i$, for $0 \leq |T| \leq t$.
- We need all of these basis functions f to satisfy the design condition. That is, for $0 < k \leq t$, and for any subset $T \subseteq [n]$ with $|T| = k$

$$\sum_{x \in D} \prod_{i \in T} x_i = 0$$

- For $t = 1$, want for all $i = 1 \ldots n$

$$\sum_{x \in D} x_i = 0$$

- That is, any coordinate to be 1 and -1 same number of times on D.
- Take $D = \{(1, 1, \ldots, 1), (-1, -1, \ldots, -1)\}$.

Hamming cube - continued

- $t = 2$. Want for all $i < j$

$$\sum_{x \in D} x_i x_j = 0$$

Hamming cube - continued

- $t = 2$. Want for all $i < j$

$$\sum_{x \in D} x_i x_j = 0$$

- Easy to check: any pair (i, j) of coordinates has to contain all 4 choices $(1, 1), (-1, 1), (1, -1), (-1, -1)$ same number of times.

Hamming cube - continued

- $t = 2$. Want for all $i < j$

$$\sum_{x \in D} x_i x_j = 0$$

- Easy to check: any pair (i, j) of coordinates has to contain all 4 choices $(1, 1), (-1, 1), (1, -1), (-1, -1)$ same number of times.
- Can take D to be a Hadamard code or any binary linear code with dual distance 3. $|D| \approx \log n$ and this is optimal.

Hamming cube - continued

- In general, for any t, need a t-wise independent set.

Hamming cube - continued

- In general, for any t, need a t-wise independent set.
- Can do reasonably well for small t, by taking linear codes with dual distance $t + 1$. Worse for large t.

Hamming cube - continued

- In general, for any t, need a t-wise independent set.
- Can do reasonably well for small t, by taking linear codes with dual distance $t+1$. Worse for large t.
- Nice question, mostly nice answers...

Hamming cube - continued

- In general, for any t, need a t-wise independent set.
- Can do reasonably well for small t, by taking linear codes with dual distance $t+1$. Worse for large t.
- Nice question, mostly nice answers... Downhill from now on...

Third example: The Hamming sphere $\binom{[n]}{k}$

- There is a notion of simple functions of strength t, $t = 0, ..., k$.

Third example: The Hamming sphere $\binom{[n]}{k}$

- There is a notion of simple functions of strength t, $t = 0, ..., k$.
- For this notion a design of strength t is a combinatorial design $S_\lambda(t, k, v)$.

Third example: The Hamming sphere $\binom{[n]}{k}$

- There is a notion of simple functions of strength t, $t = 0, ..., k$.
- For this notion a design of strength t is a combinatorial design $S_\lambda(t, k, v)$.
- a not-so-nice question, nice answer.

Third example: The Hamming sphere $\binom{[n]}{k}$

- There is a notion of simple functions of strength t, $t = 0, ..., k$.
- For this notion a design of strength t is a combinatorial design $S_\lambda(t, k, v)$.
- a not-so-nice question, nice answer.
- What about lower bounds on the size of designs?

- We have considered spaces with many symmetries, that is isometries - distance preserving transformations.

- We have considered spaces with many symmetries, that is isometries - distance preserving transformations.

- We have considered spaces with many symmetries, that is isometries - distance preserving transformations.
- Now, more attention to the distance. In our example spaces:

- We have considered spaces with many symmetries, that is isometries - distance preserving transformations.
- Now, more attention to the distance. In our example spaces:

1. The Hamming cube: the Hamming distance (number of coordinates two strings differ in).

- We have considered spaces with many symmetries, that is isometries - distance preserving transformations.
- Now, more attention to the distance. In our example spaces:
  1. The Hamming cube: the Hamming distance (number of coordinates two strings differ in).
  2. The Euclidean sphere: The Euclidean distance.

- We have considered spaces with many symmetries, that is isometries - distance preserving transformations.
- Now, more attention to the distance. In our example spaces:
  1. The Hamming cube: the Hamming distance (number of coordinates two strings differ in).
  2. The Euclidean sphere: The Euclidean distance.
  3. The Hamming sphere: The Hamming distance divided by 2.

Codes

- An (Error-correcting) code of minimal distance d is a subset $C \subseteq X$ such that the distance between any two distinct points in X is at least d.

Codes

- An (Error-correcting) code of minimal distance d is a subset $C \subseteq X$ such that the distance between any two distinct points in X is at least d.
- It is a disjoint packing of metric balls of radius $d/2$.

Codes

- An (Error-correcting) code of minimal distance d is a subset $C \subseteq X$ such that the distance between any two distinct points in X is at least d.
- It is a disjoint packing of metric balls of radius $d/2$.
- Want to construct large codes, or prove that large codes do not exist.

Codes

- An (Error-correcting) code of minimal distance d is a subset $C \subseteq X$ such that the distance between any two distinct points in X is at least d.
- It is a disjoint packing of metric balls of radius $d/2$.
- Want to construct large codes, or prove that large codes do not exist.
- Here: The second goal. What is the largest possible cardinality $A(X, d)$ of a code with minimal distance d?

Codes

- An (Error-correcting) code of minimal distance d is a subset $C \subseteq X$ such that the distance between any two distinct points in X is at least d.
- It is a disjoint packing of metric balls of radius $d/2$.
- Want to construct large codes, or prove that large codes do not exist.
- Here: The second goal. What is the largest possible cardinality $A(X,d)$ of a code with minimal distance d?
- Classical bounds:

$$\frac{|X|}{|B(d)|} \leq A(X,d) \leq \frac{|X|}{|B\,(d/2)\,|}$$

Linear programming bounds

Delsarte '73

- Hamming's bound

$$A(X, d) \leq \frac{|X|}{|B(d/2)|}$$

uses the fact that X has a 1-transitive group of symmetries.

Linear programming bounds

Delsarte '73

- Hamming's bound

$$A(X,d) \leq \frac{|X|}{|B(d/2)|}$$

uses the fact that X has a 1-transitive group of symmetries.

- LP-approach uses more of geometry of the space X to get stronger upper bounds on $A(X,d)$.

Linear programming bounds

Delsarte '73

- Hamming's bound

$$A(X, d) \leq \frac{|X|}{|B(d/2)|}$$

uses the fact that X has a 1-transitive group of symmetries.

- LP-approach uses more of geometry of the space X to get stronger upper bounds on $A(X, d)$.

- In this talk we use 2-transitivity of the symmetry group. Delsarte's approach holds in higher generality.

Delsarte's linear programming approach for the Hamming cube

- Shifts and permutations form a 2-transitive group of isometries for the Hamming cube.

Delsarte's linear programming approach for the Hamming cube

- Shifts and permutations form a 2-transitive group of isometries for the Hamming cube.
- For any two pairs of points (x, y), (x', y') with $d(x, y) = d(x', y')$ there is an isometry I such that

$$I: \; x \mapsto x' \quad \text{and} \quad y \mapsto y'$$

Delsarte's linear programming approach for the Hamming cube

- Shifts and permutations form a 2-transitive group of isometries for the Hamming cube.
- Pairwise distances - in the space. Let

$$A_i(x, y) = 1 \;\text{ if } d(x, y) = i \quad \text{and} \quad A_i(x, y) = 0 \;\text{ if } d(x, y) \neq i$$

Delsarte's linear programming approach for the Hamming cube

- Shifts and permutations form a 2-transitive group of isometries for the Hamming cube.
- Pairwise distances - in the space. Let

$$A_i(x,y) = 1 \ \text{ if } d(x,y) = i \quad \text{and} \quad A_i(x,y) = 0 \ \text{ if } d(x,y) \neq i$$

If the metric space is 2-transitive, the matrices $\{A_i\}$ span a commutative matrix algebra over the reals.

Delsarte's linear programming approach for the Hamming cube

- Shifts and permutations form a 2-transitive group of isometries for the Hamming cube.
- Pairwise distances - in the space. Let

$$A_i(x, y) = 1 \text{ if } d(x, y) = i \quad \text{and} \quad A_i(x, y) = 0 \text{ if } d(x, y) \neq i$$

If the metric space is 2-transitive, the matrices $\{A_i\}$ span a commutative matrix algebra over the reals.

- This is a Bose-Mesner algebra of the Hamming association scheme. Association schemes were introduced by statisticians to deal with difficulties in constructing combinatorial designs.

Delsarte's linear programming approach for the Hamming cube

- Shifts and permutations form a 2-transitive group of isometries for the Hamming cube.
- Pairwise distances - in the space. Let

$$A_i(x,y) = 1 \;\text{ if } d(x,y) = i \quad \text{and} \quad A_i(x,y) = 0 \;\text{ if } d(x,y) \neq i$$

If the metric space is 2-transitive, the matrices $\{A_i\}$ span a commutative matrix algebra over the reals.

- Pairwise distances - in the code C.

$$a_i = \frac{1}{|C|} \cdot \Big| \{(x,y) \in C \times C : \; d(x,y) = i\} \Big|$$

The distance distribution $(a_0 ... a_n)$ satisfies a system of linear constraints derived from the matrix algebra.

Delsarte's linear program

Maximize

$$\sum_{k=0}^{n} a_k$$

Given

- $$a_k \geq 0; \quad a_0 = 1 \quad ; a_1 = ... a_{d-1} = 0$$
- $$\sum_{k=0}^{n} Q_{i,k} a_k \geq 0, \quad i = 0 ... n$$

Delsarte's linear program

Maximize

$$\sum_{k=0}^{n} a_k$$

Given

- $$a_k \geq 0; \quad a_0 = 1 \quad ; a_1 = ...a_{d-1} = 0$$
- $$\sum_{k=0}^{n} Q_{i,k} a_k \geq 0, \quad i = 0...n$$

- A linear programming relaxation of the original problem.

Linear programming bounds

- Any feasible solution of the dual program gives an upper bound on $A(n, d)$.

Linear programming bounds

- Any feasible solution of the dual program gives an upper bound on $A(n, d)$.
- The coefficients $Q_{i,k}$ of the linear constraints on distance distribution are bad news.

Linear programming bounds

- Any feasible solution of the dual program gives an upper bound on $A(n, d)$.
- The coefficients $Q_{i,k}$ of the linear constraints on distance distribution are bad news.
- Delsarte - these coefficients are values of orthogonal polynomials.

Linear programming bounds

- Any feasible solution of the dual program gives an upper bound on $A(n, d)$.
- The coefficients $Q_{i,k}$ of the linear constraints on distance distribution are bad news.
- Delsarte - these coefficients are values of orthogonal polynomials.
- Obtaining an LP bound reduces to an extremal problem for orthogonal polynomials.

The best known bounds

- MRRW '77, using orthogonal polynomial theory: The first linear programming bound for codes

$$A(X,d) \leq 2^{H\left(\frac{1}{2}-\sqrt{\delta(1-\delta)}\right)\cdot n}$$

The best known bounds

- MRRW '77, using orthogonal polynomial theory: The first linear programming bound for codes

$$A(X,d) \leq 2^{H\left(\frac{1}{2}-\sqrt{\delta(1-\delta)}\right)\cdot n}$$

- The chain of events: Combinatorial problem –> relaxation (using algebra) –> optimization problem –> analytic problem (extremal problem in orthogonal polynomials).

The best known bounds

- MRRW '77, using orthogonal polynomial theory: The first linear programming bound for codes

$$A(X, d) \leq 2^{H\left(\frac{1}{2}-\sqrt{\delta(1-\delta)}\right)\cdot n}$$

- The chain of events: Combinatorial problem –> relaxation (using algebra) –> optimization problem –> analytic problem (extremal problem in orthogonal polynomials).
- What about designs?

Designs in association schemes

- Delsarte: in nice familiar spaces the distance distribution of a design satisfies linear constraints:

Designs in symmetric spaces

- Delsarte: in nice familiar spaces the distance distribution of a design satisfies linear constraints:

$$\sum_{k=0}^{n} Q_{i,k} a_k = 0, \quad i = 1 \ldots t$$

Designs in symmetric spaces

- Delsarte: in nice familiar spaces the distance distribution of a design satisfies linear constraints:

$$\sum_{k=0}^{n} Q_{i,k} a_k = 0, \quad i = 1 \ldots t$$

- This immediately leads to Delsarte's linear program for designs.

Delsarte's linear program for designs

Minimize

$$\sum_{k=0}^{n} a_k$$

Given

- $$a_k \geq 0; \quad a_0 = 1;$$
- $$\sum_{k=0}^{n} Q_{i,k} a_k \geq 0, \quad \sum_{k=0}^{n} Q_{i,k} a_k = 0, \; i = 1, ..., t$$

Delsarte's linear program for designs

Minimize

$$\sum_{k=0}^{n} a_k$$

Given

- $$a_k \geq 0; \quad a_0 = 1;$$
- $$\sum_{k=0}^{n} Q_{i,k} a_k \geq 0, \quad \sum_{k=0}^{n} Q_{i,k} a_k = 0, \; i = 1, ..., t$$

- A formal dual of the coding problem.

Delsarte's linear program for designs

Minimize

$$\sum_{k=0}^{n} a_k$$

Given

- $$a_k \geq 0; \quad a_0 = 1;$$
- $$\sum_{k=0}^{n} Q_{i,k} a_k \geq 0, \quad \sum_{k=0}^{n} Q_{i,k} a_k = 0, \; i = 1, ..., t$$

- A formal dual of the coding problem.
- The MRRW solution leads to the first linear programming bound for designs.

Discussion

- Recap: Linear programming approach reduces the problem of bounds for codes and designs to a difficult analytic problem in orthogonal polynomials.

Discussion

- Linear programming approach reduces the problem of bounds for codes and designs to a difficult analytic problem in orthogonal polynomials.
- It works very well, leading to the best known bounds for codes and designs.

Discussion

- Linear programming approach reduces the problem of bounds for codes and designs to a difficult analytic problem in orthogonal polynomials.
- It works very well, leading to the best known bounds for codes and designs.
- Is MRRW's solution leading to LP bounds optimal?

Discussion

- Linear programming approach reduces the problem of bounds for codes and designs to a difficult analytic problem in orthogonal polynomials.
- It works very well, leading to the best known bounds for codes and designs.
- Is MRRW's solution leading to LP bounds optimal?
- In fact, MRRW conjecture it is not optimal.
- However, there are results (Barg-Jaffe etc) suggesting its optimality.

Discussion

- Linear programming approach reduces the problem of bounds for codes and designs to a difficult analytic problem in orthogonal polynomials.
- It works mysteriously well, leading to the best known bounds for codes and designs.
- Is MRRW's solution leading to LP bounds optimal?
- Why does the LP approach work so well?

Discussion

- Linear programming approach reduces the problem of bounds for codes and designs to a difficult analytic problem in orthogonal polynomials.
- It works mysteriously well, leading to the best known bounds for codes and designs.
- Is MRRW's solution leading to LP bounds optimal?
- Why does the LP approach work so well?
- Can we improve on LP bounds?

Discussion

- Linear programming approach reduces the problem of bounds for codes and designs to a difficult analytic problem in orthogonal polynomials.
- It works mysteriously well, leading to the best known bounds for codes and designs.
- Is MRRW's solution leading to LP bounds optimal?
- Why does the LP approach work so well?
- Can we improve on LP bounds? Failing that, can we at least understand them better?

Other proofs and improvements

Some history

- Schrijver'05, Bachoc-Valentin'07: semi-definite relaxation for the coding problem. Improves over LP bounds for small parameters.

Other proofs and improvements

Some history

- Schrijver'05, Bachoc-Valentin'07: semi-definite relaxation for the coding problem. Improves over LP bounds for small parameters. Hard to analyze analytically.

Other proofs and improvements

Some history

- Schrijver'05, Bachoc-Valentin'07: semi-definite relaxation for the coding problem. Improves over LP bounds for small parameters.
- Kalai-Linial '89 A Fourier-analytic approach to obtain Delsarte's linear program for codes and designs.

Other proofs and improvements

Some history

- Schrijver'05, Bachoc-Valentin'07: semi-definite relaxation for the coding problem. Improves over LP bounds for small parameters.
- Kalai-Linial '89 A Fourier-analytic approach to obtain Delsarte's linear program for codes and designs. Works with the whole space rather than with the distance distribution.

Other proofs and improvements

Some history

- Schrijver'05, Bachoc-Valentin'07: semi-definite relaxation for the coding problem. Improves over LP bounds for small parameters.
- Kalai-Linial '89 A Fourier-analytic approach to obtain Delsarte's linear program for codes and designs. Works with the whole space rather than with the distance distribution.
- Friedman-Tillich '02: A different proof of the first LP bound for binary linear codes, bounding the eigenvalue of a certain related Cayley graph.

Other proofs and improvements

Some history

- Schrijver'05, Bachoc-Valentin'07: semi-definite relaxation for the coding problem. Improves over LP bounds for small parameters.
- Kalai-Linial '89 A Fourier-analytic approach to obtain Delsarte's linear program for codes and designs. Works with the whole space rather than with the distance distribution.
- Friedman-Tillich '02: A different proof of the first LP bound for binary linear codes, bounding the eigenvalue of a certain related Cayley graph. Works with the whole space.

Other proofs and improvements

Some history

- Schrijver'05, Bachoc-Valentin'07: semi-definite relaxation for the coding problem.
- Kalai-Linial '89 A Fourier-analytic approach to obtain Delsarte's linear program for codes and designs.
- Friedman-Tillich '02: A different proof of the first LP bound for binary linear codes, bounding the eigenvalue of a certain related Cayley graph.
- Navon-S.'05: Combining the two approaches above. A Fourier-analytic proof of lower bounds for designs in the Hamming cube.

Other proofs and improvements

Some history

- Schrijver'05, Bachoc-Valentin'07: semi-definite relaxation for the coding problem.
- Kalai-Linial '89 A Fourier-analytic approach to obtain Delsarte's linear program for codes and designs.
- Friedman-Tillich '02: A different proof of the first LP bound for binary linear codes, bounding the eigenvalue of a certain related Cayley graph.
- Navon-S.'05: Combining the two approaches above. A Fourier-analytic proof of lower bounds for designs in the Hamming cube. A design is large because a union of small Hamming balls around its points "covers" the whole space.

This talk

Sales pitch

- We prove lower bounds on designs in symmetric spaces, generalizing the result for the Hamming cube.

This talk

Sales pitch

- We prove lower bounds on designs in symmetric spaces, generalizing the result for the Hamming cube.
A design is large because a union of small metric balls around its points "covers" the whole space.

This talk

Sales pitch

- We prove lower bounds on designs in symmetric spaces, generalizing the result for the Hamming cube.
A design is large because a union of small metric balls around its points "covers" the whole space.
- We do not add to geometric understanding of the space, compared to Delsarte's approach, but use the symmetries of the space in a (seemingly) different way.

This talk

Sales pitch

- We prove lower bounds on designs in symmetric spaces, generalizing the result for the Hamming cube.
A design is large because a union of small metric balls around its points "covers" the whole space.
- We do not add to geometric understanding of the space, compared to Delsarte's approach, but use the symmetries of the space in a (seemingly) different way.
- Our proof is easy, using simple linear algebra and the symmetries of the space directly (rather than going through Fourier analysis).

This talk

Sales pitch

- We prove lower bounds on designs in symmetric spaces, generalizing the result for the Hamming cube.
  A design is large because a union of small metric balls around its points "covers" the whole space.
- We do not add to geometric understanding of the space, compared to Delsarte's approach, but use the symmetries of the space in a (seemingly) different way.
- Our proof is easy, using simple linear algebra and the symmetries of the space directly (rather than going through Fourier analysis). However, in the general case, need the language and some claims from Fourier analysis on symmetric spaces.

Some examples

- For any t there is a radius $r = r(t)$ (depending on the space) such that

Some examples

- For any t there is a radius $r = r(t)$ (depending on the space) such that
  1. Union of Hamming balls of radius $r(t)$ around a t-wise independent set "covers" the Hamming cube.

Some examples

- For any t there is a radius $r = r(t)$ (depending on the space) such that

1. Union of Hamming balls of radius $r(t)$ around a t-wise independent set "covers" the Hamming cube.
2. Union of spherical caps of radius $r(t)$ around a spherical design of strength t "covers" the Euclidean sphere.

Some examples

- For any t there is a radius $r = r(t)$ (depending on the space) such that

1. Union of Hamming balls of radius $r(t)$ around a t-wise independent set "covers" the Hamming cube.
2. Union of spherical caps of radius $r(t)$ around a spherical design of strength t "covers" the Euclidean sphere.
3. Union of "Hamming spherical caps" of radius $r(t)$ around a combinatorial t-design "covers" the Hamming sphere.

Some examples

- For any t there is a radius $r = r(t)$ (depending on the space) such that
  1. Union of Hamming balls of radius $r(t)$ around a t-wise independent set "covers" the Hamming cube.
  2. Union of spherical caps of radius $r(t)$ around a spherical design of strength t "covers" the Euclidean sphere.
  3. Union of "Hamming spherical caps" of radius $r(t)$ around a combinatorial t-design "covers" the Hamming sphere.
- Recovers the first linear programming bound for these spaces.

Some ingredients of this approach

- A simple description of simple functions.

Some ingredients of this approach

- A simple description of simple functions.
- The Laplacian.

1. For the Euclidean space $\mathbb{R}^n$

$$Lf = -\left(\frac{\partial^2 f}{\partial x_1^2} + \ldots + \frac{\partial^2 f}{\partial x_n^2}\right)$$

Some ingredients of this approach

- A simple description of simple functions.
- The Laplacian.

1. For the Euclidean space $\mathbb{R}^n$

$$Lf = -\left(\frac{\partial^2 f}{\partial x_1^2} + \dots + \frac{\partial^2 f}{\partial x_n^2}\right)$$

2. For the Euclidean sphere $\mathbb{S}^{n-1}$ the Laplacian is the restriction of the Laplacian on $\mathbb{R}^n$.

Some ingredients of this approach

- A simple description of simple functions.
- The Laplacian.

1. For the Euclidean space $\mathbb{R}^n$

$$Lf = -\left(\frac{\partial^2 f}{\partial x_1^2} + \ldots + \frac{\partial^2 f}{\partial x_n^2}\right)$$

2. For the Euclidean sphere $\mathbb{S}^{n-1}$ the Laplacian is the restriction of the Laplacian on $\mathbb{R}^n$. (Less easy than it looks.)

Some ingredients of this approach

- A simple description of simple functions.
- The Laplacian.

1. For the Euclidean space $\mathbb{R}^n$

$$Lf = -\left(\frac{\partial^2 f}{\partial x_1^2} + \ldots + \frac{\partial^2 f}{\partial x_n^2}\right)$$

2. For the Euclidean sphere $\mathbb{S}^{n-1}$ the Laplacian is the restriction of the Laplacian on $\mathbb{R}^n$.
3. For k-regular graphs it is the usual graph Laplacian

$$Lf(x) = k \cdot f(x) - \sum_{y \sim x} f(y)$$

Some ingredients of this approach

- A simple description of simple functions.
- The Laplacian.

1. For the Euclidean space $\mathbb{R}^n$

$$Lf = -\left(\frac{\partial^2 f}{\partial x_1^2} + \ldots + \frac{\partial^2 f}{\partial x_n^2}\right)$$

2. For the Euclidean sphere $\mathbb{S}^{n-1}$ the Laplacian is the restriction of the Laplacian on $\mathbb{R}^n$.
3. For k-regular graphs it is the usual graph Laplacian (This takes care of the Hamming cube and sphere.)

$$Lf(x) = k \cdot f(x) - \sum_{y \sim x} f(y)$$

Some ingredients of this approach - continued

- Definition: The space of simple functions of strength t is spanned by the eigenfunctions of the Laplacian which belong to the first t eigenvalues.

Some ingredients of this approach - continued

- Definition: The space of simple functions of strength t is spanned by the eigenfunctions of the Laplacian which belong to the first t eigenvalues.
- Definition: The eigenvalue of a subset $\Omega \subseteq X$ is the minimal eigenvalue of the Laplacian restricted to functions supported on Ω (with appropriate boundary and smoothness conditions if needed.)

Main Theorem

Let D be a design of strength t on X. Let Ω be a subset of X with eigenvalue λ. Then, assuming X and Ω are sufficiently symmetric, and $\lambda < \lambda_t$, we have

$$|D| \geq \frac{\lambda_t - \lambda}{\lambda_t} \cdot \frac{|X|}{|\Omega|}$$

In fact, a union of isomorphic copies of Ω taken around each point of D covers X (up to a $\frac{\lambda_t - \lambda}{\lambda_t}$-factor).

Proof by hand-waiving

- D is a design of strength t. Let $|D| = d$. Let $\Omega_1, ..., \Omega_d$ be copies of Ω around the points of D.

Proof by hand-waiving

- D is a design of strength t. Let $|D| = d$. Let $\Omega_1, ..., \Omega_d$ be copies of Ω around the points of D.
- We define a function F on $\cup_i \Omega_i$ and show

$$\frac{\|F\|_2^2}{\|F\|_1^2} \leq \frac{\lambda_t}{\lambda_t - \lambda}$$

Proof by hand-waiving

- D is a design of strength t. Let $|D| = d$. Let $\Omega_1, ..., \Omega_d$ be copies of Ω around the points of D.
- We define a function F on $\cup_i \Omega_i$ and show

$$\frac{\|F\|_2^2}{\|F\|_1^2} \leq \frac{\lambda_t}{\lambda_t - \lambda}$$

- This means the support size of F is at least $\frac{\lambda_t - \lambda}{\lambda_t} \cdot |X|$. (Done).

Proof by hand-waiving: the main point

- Let f_i be a function with eigenvalue $\lambda = \lambda(\Omega)$ supported on Ω_i. Take $F = \sum_{i=1}^{d} f_i$.

Proof by hand-waiving: the main point

- Let f_i be a function with eigenvalue $\lambda = \lambda(\Omega)$ supported on Ω_i. Take $F = \sum_{i=1}^{d} f_i$.
- $$\langle f_i, Lf_i \rangle \leq \lambda \langle f_i, f_i \rangle \implies \langle F, LF \rangle \leq \lambda \langle F, F \rangle$$

Proof by hand-waiving: the main point

- Let f_i be a function with eigenvalue $\lambda = \lambda(\Omega)$ supported on Ω_i. Take $F = \sum_{i=1}^{d} f_i$.
-
$$\langle f_i, Lf_i \rangle \leq \lambda \langle f_i, f_i \rangle \quad \implies \quad \langle F, LF \rangle \leq \lambda \langle F, F \rangle$$

- F inherits the design properties:

$\langle F, \phi \rangle = 0$ for any eigenfunction ϕ with eigenvalues $\lambda_1 < ... < \lambda_t$

Proof by hand-waiving: the main point

- Let f_i be a function with eigenvalue $\lambda = \lambda(\Omega)$ supported on Ω_i. Take $F = \sum_{i=1}^{d} f_i$.
- $$\langle f_i, Lf_i \rangle \leq \lambda \langle f_i, f_i \rangle \quad \implies \quad \langle F, LF \rangle \leq \lambda \langle F, F \rangle$$
- F inherits the design properties:

$\langle F, \phi \rangle = 0$ for any eigenfunction ϕ with eigenvalues $\lambda_1 < ... < \lambda_t$

- This means F is spanned by the constant function and eigenfunctions with large eigenvalues.

Proof by hand-waiving: the main point

- Let f_i be a function with eigenvalue $\lambda = \lambda(\Omega)$ supported on Ω_i. Take $F = \sum_{i=1}^{d} f_i$.
- $$\langle f_i, Lf_i \rangle \leq \lambda \langle f_i, f_i \rangle \quad \implies \quad \langle F, LF \rangle \leq \lambda \langle F, F \rangle$$
- F inherits the design properties:

$$\langle F, \phi \rangle = 0 \quad \text{for any eigenfunction } \phi \text{ with eigenvalues } \lambda_1 < ... < \lambda_t$$

- This means F is spanned by the constant function and eigenfunctions with large eigenvalues.
- This has to mean F has a large constant component. Done