Fourier analysis of Boolean functions: Some beautiful examples

Ronald de Wolf

Centrum voor Wiskunde en Informatica

Amsterdam

Fourier analysis

Fourier analysis

- Many applications in math, physics, engineering,

Fourier analysis

- Many applications in math, physics, engineering,

  ...and in computer science

Fourier analysis

- Many applications in math, physics, engineering,

  . . . and in computer science:

  - Signal processing

Fourier analysis

- Many applications in math, physics, engineering,

  . . . and in computer science:

  - Signal processing
  - Data compression

Fourier analysis

- Many applications in math, physics, engineering,

  . . . and in computer science:

  - Signal processing
  - Data compression
  - Multiplying two polynomials

Fourier analysis

- Many applications in math, physics, engineering,

  ...and in computer science:

  - Signal processing
  - Data compression
  - Multiplying two polynomials

- These examples use Fourier analysis over cyclic groups

Fourier analysis

- Many applications in math, physics, engineering,

  . . . and in computer science:

  - Signal processing
  - Data compression
  - Multiplying two polynomials

- These examples use Fourier analysis over cyclic groups
- We will focus on Fourier analysis over the Boolean cube

Fourier analysis

- Many applications in math, physics, engineering,

  . . . and in computer science:

  - Signal processing
  - Data compression
  - Multiplying two polynomials

- These examples use Fourier analysis over cyclic groups

- We will focus on Fourier analysis over the Boolean cube $= \{0,1\}^n$, set of all n-bit strings

This has been very useful in CS

This has been very useful in CS

- Fourier coefficients measure correlations with parities

This has been very useful in CS

- Fourier coefficients measure correlations with parities
- Analysis of error-correcting codes

This has been very useful in CS

- Fourier coefficients measure correlations with parities
- Analysis of error-correcting codes
- Learning a function from examples

This has been very useful in CS

- Fourier coefficients measure correlations with parities
- Analysis of error-correcting codes
- Learning a function from examples
- The influence of variables on a function

This has been very useful in CS

- Fourier coefficients measure correlations with parities
- Analysis of error-correcting codes
- Learning a function from examples
- The influence of variables on a function
- Sensitivity of a function to noise on the inputs

This has been very useful in CS

- Fourier coefficients measure correlations with parities
- Analysis of error-correcting codes
- Learning a function from examples
- The influence of variables on a function
- Sensitivity of a function to noise on the inputs
- PCPs, NP-hardness of approximation

This has been very useful in CS

- Fourier coefficients measure correlations with parities
- Analysis of error-correcting codes
- Learning a function from examples
- The influence of variables on a function
- Sensitivity of a function to noise on the inputs
- PCPs, NP-hardness of approximation
- Cryptography

This has been very useful in CS

- Fourier coefficients measure correlations with parities
- Analysis of error-correcting codes
- Learning a function from examples
- The influence of variables on a function
- Sensitivity of a function to noise on the inputs
- PCPs, NP-hardness of approximation
- Cryptography
- Lower bounds on communication complexity

This has been very useful in CS

- Fourier coefficients measure correlations with parities
- Analysis of error-correcting codes
- Learning a function from examples
- The influence of variables on a function
- Sensitivity of a function to noise on the inputs
- PCPs, NP-hardness of approximation
- Cryptography
- Lower bounds on communication complexity
- Threshold phenomena in random graphs

This has been very useful in CS

- Fourier coefficients measure correlations with parities
- Analysis of error-correcting codes
- Learning a function from examples
- The influence of variables on a function
- Sensitivity of a function to noise on the inputs
- PCPs, NP-hardness of approximation
- Cryptography
- Lower bounds on communication complexity
- Threshold phenomena in random graphs
- Quantum computing

This has been very useful in CS

- Fourier coefficients measure correlations with parities
- Analysis of error-correcting codes
- Learning a function from examples
- The influence of variables on a function
- Sensitivity of a function to noise on the inputs
- PCPs, NP-hardness of approximation
- Cryptography
- Lower bounds on communication complexity
- Threshold phenomena in random graphs
- Quantum computing

...

Fourier analysis over the Boolean cube

Fourier analysis over the Boolean cube

- Consider the space of functions $f : \{0, 1\}^n \to \mathbb{R}$

Fourier analysis over the Boolean cube

- Consider the space of functions $f : \{0,1\}^n \to \mathbb{R}$, with normalized inner product $\langle f, g\rangle = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} f(x)g(x)$

Fourier analysis over the Boolean cube

- Consider the space of functions $f : \{0,1\}^n \to \mathbb{R}$, with normalized inner product $\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)g(x)$
- The parity-functions $\chi_s(x) = (-1)^{x \cdot s} = \prod_{i:s_i=1} (-1)^{x_i}$ form an orthonormal basis of this space

Fourier analysis over the Boolean cube

- Consider the space of functions $f : \{0,1\}^n \to \mathbb{R}$, with normalized inner product $\langle f, g\rangle = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} f(x)g(x)$
- The parity-functions $\chi_s(x) = (-1)^{x\cdot s} = \prod_{i:s_i=1}(-1)^{x_i}$ form an orthonormal basis of this space
- Hence we can write $f = \sum_{s\in\{0,1\}^n} \widehat{f}(s)\chi_s$

Fourier analysis over the Boolean cube

- Consider the space of functions $f : \{0,1\}^n \to \mathbb{R}$, with normalized inner product $\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)g(x)$
- The parity-functions $\chi_s(x) = (-1)^{x \cdot s} = \prod_{i:s_i=1} (-1)^{x_i}$ form an orthonormal basis of this space
- Hence we can write $f = \sum_{s \in \{0,1\}^n} \widehat{f}(s)\chi_s$

  with $\widehat{f}(s) = \langle f, \chi_s \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)\chi_s(x)$

Fourier analysis over the Boolean cube

- Consider the space of functions $f : \{0,1\}^n \to \mathbb{R}$, with normalized inner product $\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)g(x)$
- The parity-functions $\chi_s(x) = (-1)^{x \cdot s} = \prod_{i:s_i=1}(-1)^{x_i}$ form an orthonormal basis of this space
- Hence we can write $f = \sum_{s \in \{0,1\}^n} \widehat{f}(s)\chi_s$

  with $\widehat{f}(s) = \langle f, \chi_s \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)\chi_s(x)$
- Map $f \mapsto \widehat{f}$ is proportional to unitary (length-preserving)

Fourier analysis over the Boolean cube

- Consider the space of functions $f : \{0,1\}^n \to \mathbb{R}$, with normalized inner product $\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)g(x)$
- The parity-functions $\chi_s(x) = (-1)^{x \cdot s} = \prod_{i:s_i=1} (-1)^{x_i}$ form an orthonormal basis of this space
- Hence we can write $f = \sum_{s \in \{0,1\}^n} \widehat{f}(s)\chi_s$

  with $\widehat{f}(s) = \langle f, \chi_s \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)\chi_s(x)$
- Map $f \mapsto \widehat{f}$ is proportional to unitary (length-preserving) $\Rightarrow \frac{1}{2^n} \sum_x f(x)^2 = \sum_s \widehat{f}(s)^2$ (Parseval's identity)

Examples

Examples

OR on 2 bits:

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00)$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \mathsf{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4} \sum_{x \in \{0,1\}^n} f(x) \chi_{00}(x)$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4} \sum_{x \in \{0,1\}^n} f(x) \chi_{00}(x) = \frac{1}{4}(0 + 1 + 1 + 1)$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \mathsf{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4}\sum_{x\in\{0,1\}^n} f(x)\chi_{00}(x) = \frac{1}{4}(0+1+1+1) = \frac{3}{4}$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \mathsf{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4} \sum_{x \in \{0,1\}^n} f(x) \chi_{00}(x) = \frac{1}{4}(0 + 1 + 1 + 1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4}$,

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4}\sum_{x\in\{0,1\}^n} f(x)\chi_{00}(x) = \frac{1}{4}(0+1+1+1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4},\ \ \widehat{f}(10) = -\frac{1}{4},$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4}\sum_{x\in\{0,1\}^n} f(x)\chi_{00}(x) = \frac{1}{4}(0+1+1+1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4},\ \widehat{f}(10) = -\frac{1}{4},\ \widehat{f}(11) = -\frac{1}{4}$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4} \sum_{x \in \{0,1\}^n} f(x)\chi_{00}(x) = \frac{1}{4}(0 + 1 + 1 + 1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4},\ \ \widehat{f}(10) = -\frac{1}{4},\ \ \widehat{f}(11) = -\frac{1}{4}$
- Note: $\widehat{f}(00) = \text{Exp}_x[f(x)]$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \mathsf{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4} \sum_{x \in \{0,1\}^n} f(x) \chi_{00}(x) = \frac{1}{4}(0 + 1 + 1 + 1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4}$, $\widehat{f}(10) = -\frac{1}{4}$, $\widehat{f}(11) = -\frac{1}{4}$
- Note: $\widehat{f}(00) = \mathsf{Exp}_x[f(x)]$, and $f(00) = \sum_s \widehat{f}(s)$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4} \sum_{x \in \{0,1\}^n} f(x)\chi_{00}(x) = \frac{1}{4}(0 + 1 + 1 + 1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4}$, $\widehat{f}(10) = -\frac{1}{4}$, $\widehat{f}(11) = -\frac{1}{4}$
- Note: $\widehat{f}(00) = \text{Exp}_x[f(x)]$, and $f(00) = \sum_s \widehat{f}(s)$
- Parseval: $\displaystyle\frac{1}{4} \sum_x f(x)^2$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4} \sum_{x \in \{0,1\}^n} f(x)\chi_{00}(x) = \frac{1}{4}(0 + 1 + 1 + 1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4}, \ \widehat{f}(10) = -\frac{1}{4}, \ \widehat{f}(11) = -\frac{1}{4}$
- Note: $\widehat{f}(00) = \text{Exp}_x[f(x)]$, and $f(00) = \sum_s \widehat{f}(s)$
- Parseval: $\dfrac{1}{4} \displaystyle\sum_x f(x)^2 = \dfrac{3}{4}$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4} \sum_{x \in \{0,1\}^n} f(x) \chi_{00}(x) = \frac{1}{4}(0 + 1 + 1 + 1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4}$, $\widehat{f}(10) = -\frac{1}{4}$, $\widehat{f}(11) = -\frac{1}{4}$
- Note: $\widehat{f}(00) = \text{Exp}_x[f(x)]$, and $f(00) = \sum_s \widehat{f}(s)$
- Parseval: $\displaystyle \frac{1}{4} \sum_x f(x)^2 = \frac{3}{4} = \sum_s \widehat{f}(s)^2$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4}\sum_{x\in\{0,1\}^n} f(x)\chi_{00}(x) = \frac{1}{4}(0+1+1+1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4},\ \widehat{f}(10) = -\frac{1}{4},\ \widehat{f}(11) = -\frac{1}{4}$
- Note: $\widehat{f}(00) = \text{Exp}_x[f(x)]$, and $f(00) = \sum_s \widehat{f}(s)$
- Parseval: $\displaystyle\frac{1}{4}\sum_x f(x)^2 = \frac{3}{4} = \sum_s \widehat{f}(s)^2$

PARITY on n bits

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4} \sum_{x \in \{0,1\}^n} f(x) \chi_{00}(x) = \frac{1}{4}(0 + 1 + 1 + 1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4}, \ \widehat{f}(10) = -\frac{1}{4}, \ \widehat{f}(11) = -\frac{1}{4}$
- Note: $\widehat{f}(00) = \text{Exp}_x[f(x)]$, and $f(00) = \sum_s \widehat{f}(s)$
- Parseval: $\displaystyle \frac{1}{4} \sum_x f(x)^2 = \frac{3}{4} = \sum_s \widehat{f}(s)^2$

PARITY on n bits, with TRUE$= -1$, FALSE$= +1$:

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4} \sum_{x \in \{0,1\}^n} f(x) \chi_{00}(x) = \frac{1}{4}(0 + 1 + 1 + 1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4},\ \ \widehat{f}(10) = -\frac{1}{4},\ \ \widehat{f}(11) = -\frac{1}{4}$
- Note: $\widehat{f}(00) = \text{Exp}_x[f(x)]$, and $f(00) = \sum_s \widehat{f}(s)$
- Parseval: $\displaystyle \frac{1}{4} \sum_x f(x)^2 = \frac{3}{4} = \sum_s \widehat{f}(s)^2$

PARITY on n bits, with TRUE$= -1$, FALSE$= +1$:

- $f(x)$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4}\sum_{x\in\{0,1\}^n} f(x)\chi_{00}(x) = \frac{1}{4}(0+1+1+1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4},\ \widehat{f}(10) = -\frac{1}{4},\ \widehat{f}(11) = -\frac{1}{4}$
- Note: $\widehat{f}(00) = \text{Exp}_x[f(x)]$, and $f(00) = \sum_s \widehat{f}(s)$
- Parseval: $\displaystyle\frac{1}{4}\sum_x f(x)^2 = \frac{3}{4} = \sum_s \widehat{f}(s)^2$

PARITY on n bits, with TRUE$= -1$, FALSE$= +1$:

- $f(x) = \chi_{1^n}(x) = (-1)^{|x|}$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4}\sum_{x\in\{0,1\}^n} f(x)\chi_{00}(x) = \frac{1}{4}(0+1+1+1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4},\ \widehat{f}(10) = -\frac{1}{4},\ \widehat{f}(11) = -\frac{1}{4}$
- Note: $\widehat{f}(00) = \text{Exp}_x[f(x)]$, and $f(00) = \sum_s \widehat{f}(s)$
- Parseval: $\displaystyle \frac{1}{4}\sum_x f(x)^2 = \frac{3}{4} = \sum_s \widehat{f}(s)^2$

PARITY on n bits, with TRUE$= -1$, FALSE$= +1$:

- $f(x) = \chi_{1^n}(x) = (-1)^{|x|}$, so $\widehat{f}(1^n) = 1$

Examples

OR on 2 bits:

- $f(x_1, x_2) = \text{OR}(x_1, x_2) \in \{0, 1\}$
- $\widehat{f}(00) = \frac{1}{4} \sum_{x \in \{0,1\}^n} f(x) \chi_{00}(x) = \frac{1}{4}(0 + 1 + 1 + 1) = \frac{3}{4}$
  $\widehat{f}(01) = -\frac{1}{4}, \ \widehat{f}(10) = -\frac{1}{4}, \ \widehat{f}(11) = -\frac{1}{4}$
- Note: $\widehat{f}(00) = \text{Exp}_x[f(x)]$, and $f(00) = \sum_s \widehat{f}(s)$
- Parseval: $\displaystyle \frac{1}{4} \sum_x f(x)^2 = \frac{3}{4} = \sum_s \widehat{f}(s)^2$

PARITY on n bits, with TRUE$= -1$, FALSE$= +1$:

- $f(x) = \chi_{1^n}(x) = (-1)^{|x|}$, so $\widehat{f}(1^n) = 1$
- all other $\widehat{f}(s)$ are 0

(1) Approximating functions with parities

(1) Approximating functions with parities

- Suppose $f : \{0,1\}^n \to \{\pm 1\}$ has small Fourier degree d

(1) Approximating functions with parities

- Suppose $f : \{0,1\}^n \to \{\pm 1\}$ has small Fourier degree d:

$$f = \sum_{s:|s| \le d} \widehat{f}(s) \chi_s$$

(1) Approximating functions with parities

- Suppose $f : \{0,1\}^n \to \{\pm 1\}$ has small Fourier degree d:

$$f = \sum_{s:|s|\le d} \widehat{f}(s)\chi_s$$

- Then there exists a parity-function on at most d bits that has non-trivial correlation with f

(1) Approximating functions with parities

- Suppose $f : \{0,1\}^n \to \{\pm 1\}$ has small Fourier degree d:

$$f = \sum_{s:|s|\leq d} \widehat{f}(s)\chi_s$$

- Then there exists a parity-function on at most d bits that has non-trivial correlation with f
- Why?

(1) Approximating functions with parities

- Suppose $f : \{0,1\}^n \to \{\pm 1\}$ has small Fourier degree d:

$$f = \sum_{s:|s|\leq d} \widehat{f}(s)\chi_s$$

- Then there exists a parity-function on at most d bits that has non-trivial correlation with f
- Why? $\sum_{s:|s|\leq d} \widehat{f}(s)^2$

(1) Approximating functions with parities

- Suppose $f : \{0,1\}^n \to \{\pm 1\}$ has small Fourier degree d:

$$f = \sum_{s:|s|\le d} \widehat{f}(s)\chi_s$$

- Then there exists a parity-function on at most d bits that has non-trivial correlation with f
- Why? $\sum_{s:|s|\le d} \widehat{f}(s)^2 = \frac{1}{2^n}\sum_{x\in\{0,1\}^n} f(x)^2 = 1$ (Parseval).

(1) Approximating functions with parities

- Suppose $f : \{0,1\}^n \to \{\pm 1\}$ has small Fourier degree d:

$$f = \sum_{s:|s|\leq d} \widehat{f}(s)\chi_s$$

- Then there exists a parity-function on at most d bits that has non-trivial correlation with f

- Why? $\sum_{s:|s|\leq d} \widehat{f}(s)^2 = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} f(x)^2 = 1$ (Parseval).

  This is a sum over $\leq n^d$ terms.

(1) Approximating functions with parities

- Suppose $f : \{0,1\}^n \to \{\pm 1\}$ has small Fourier degree d:

$$f = \sum_{s:|s|\leq d} \widehat{f}(s)\chi_s$$

- Then there exists a parity-function on at most d bits that has non-trivial correlation with f
- Why? $\sum_{s:|s|\leq d} \widehat{f}(s)^2 = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} f(x)^2 = 1$ (Parseval).

  This is a sum over $\leq n^d$ terms. Hence $\exists s$ with

(1) Approximating functions with parities

- Suppose $f : \{0,1\}^n \to \{\pm 1\}$ has small Fourier degree d:

$$f = \sum_{s:|s|\leq d} \widehat{f}(s)\chi_s$$

- Then there exists a parity-function on at most d bits that has non-trivial correlation with f

- Why? $\sum_{s:|s|\leq d} \widehat{f}(s)^2 = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)^2 = 1$ (Parseval).

  This is a sum over $\leq n^d$ terms. Hence $\exists s$ with $\frac{1}{n^d} \leq \widehat{f}(s)^2$

(1) Approximating functions with parities

- Suppose $f : \{0,1\}^n \to \{\pm 1\}$ has small Fourier degree d:

$$f = \sum_{s:|s|\le d} \widehat{f}(s)\chi_s$$

- Then there exists a parity-function on at most d bits that has non-trivial correlation with f

- Why? $\sum_{s:|s|\le d} \widehat{f}(s)^2 = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} f(x)^2 = 1$ (Parseval).

  This is a sum over $\le n^d$ terms. Hence $\exists s$ with

$$\frac{1}{n^d} \le \widehat{f}(s)^2 = |\frac{1}{2^n} \sum_{x\in\{0,1\}^n} f(x)\chi_s(x)|^2$$

(1) Approximating functions with parities

- Suppose $f : \{0,1\}^n \to \{\pm 1\}$ has small Fourier degree d:

$$f = \sum_{s:|s|\le d} \widehat{f}(s)\chi_s$$

- Then there exists a parity-function on at most d bits that has non-trivial correlation with f

- Why? $\displaystyle\sum_{s:|s|\le d} \widehat{f}(s)^2 = \frac{1}{2^n}\sum_{x\in\{0,1\}^n} f(x)^2 = 1$ (Parseval).

  This is a sum over $\le n^d$ terms. Hence $\exists s$ with
  $$\frac{1}{n^d} \le \widehat{f}(s)^2 = |\frac{1}{2^n}\sum_{x\in\{0,1\}^n} f(x)\chi_s(x)|^2$$

- So χ_s (or its negation) has non-trivial correlation with f

(2) Learning from uniform examples

(2) Learning from uniform examples

- A Fourier coefficient is just a uniform expectation

(2) Learning from uniform examples

- A Fourier coefficient is just a uniform expectation:

$$\widehat{f}(s) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)\chi_s(x)$$

(2) Learning from uniform examples

- A Fourier coefficient is just a uniform expectation:

$$\widehat{f}(s) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)\chi_s(x) = \text{Exp}_x[f(x)\chi_s(x)]$$

(2) Learning from uniform examples

- A Fourier coefficient is just a uniform expectation:

$$\widehat{f}(s) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)\chi_s(x) = \mathsf{Exp}_x[f(x)\chi_s(x)]$$

- We can approximate this given uniformly random examples $(x^1, f(x^1)), \ldots, (x^m, f(x^m))$:

(2) Learning from uniform examples

- A Fourier coefficient is just a uniform expectation:

$$\widehat{f}(s) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)\chi_s(x) = \mathsf{Exp}_x[f(x)\chi_s(x)]$$

- We can approximate this given uniformly random examples $(x^1, f(x^1)), \ldots, (x^m, f(x^m))$:
$$\frac{1}{m} \sum_{i=1}^{m} f(x^i)\chi_s(x^i)$$

(2) Learning from uniform examples

- A Fourier coefficient is just a uniform expectation:

$$\widehat{f}(s) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)\chi_s(x) = \mathrm{Exp}_x[f(x)\chi_s(x)]$$

- We can approximate this given uniformly random examples $(x^1, f(x^1)), \ldots, (x^m, f(x^m))$:
$$\frac{1}{m}\sum_{i=1}^{m} f(x^i)\chi_s(x^i) \to \widehat{f}(s)$$

(2) Learning from uniform examples

- A Fourier coefficient is just a uniform expectation:

$$\widehat{f}(s) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)\chi_s(x) = \text{Exp}_x[f(x)\chi_s(x)]$$

- We can approximate this given uniformly random examples $(x^1, f(x^1)), \ldots, (x^m, f(x^m))$:
$$\frac{1}{m}\sum_{i=1}^{m} f(x^i)\chi_s(x^i) \to \widehat{f}(s)$$

- Converges fast if $|\widehat{f}(s)|$ is not too small (Chernoff)

(2) Learning from uniform examples

- A Fourier coefficient is just a uniform expectation:

$$\widehat{f}(s) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)\chi_s(x) = \mathsf{Exp}_x[f(x)\chi_s(x)]$$

- We can approximate this given uniformly random examples $(x^1, f(x^1)), \ldots, (x^m, f(x^m))$:
$$\frac{1}{m}\sum_{i=1}^{m} f(x^i)\chi_s(x^i) \rightarrow \widehat{f}(s)$$
- Converges fast if $|\widehat{f}(s)|$ is not too small (Chernoff)
- Hence we can quickly learn (approximate) an unknown function f that is dominated by a few large coefficients

(2) Learning from uniform examples

- A Fourier coefficient is just a uniform expectation:

$$\widehat{f}(s) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)\chi_s(x) = \text{Exp}_x[f(x)\chi_s(x)]$$

- We can approximate this given uniformly random examples $(x^1, f(x^1)), \ldots, (x^m, f(x^m))$:
$$\frac{1}{m}\sum_{i=1}^{m} f(x^i)\chi_s(x^i) \to \widehat{f}(s)$$
- Converges fast if $|\widehat{f}(s)|$ is not too small (Chernoff)
- Hence we can quickly learn (approximate) an unknown function f that is dominated by a few large coefficients (example from LMN 89: AC_0-circuits)

(3) List-decoding of Hadamard code

(3) List-decoding of Hadamard code

- Error-correcting code: $E : \{0,1\}^n \to \{0,1\}^m$

(3) List-decoding of Hadamard code

- Error-correcting code: $E : \{0,1\}^n \to \{0,1\}^m$
- If all codewords have distance $d(E(x), E(y)) \geq 2e + 1$, then we can uniquely recover x from corrupted codeword $w \in \{0,1\}^m$ with e errors

(3) List-decoding of Hadamard code

- Error-correcting code: $E : \{0,1\}^n \to \{0,1\}^m$
- If all codewords have distance $d(E(x), E(y)) \geq 2e + 1$, then we can uniquely recover x from corrupted codeword $w \in \{0,1\}^m$ with e errors ($d(w, E(x)) = e$)

(3) List-decoding of Hadamard code

- Error-correcting code: $E : \{0,1\}^n \to \{0,1\}^m$
- If all codewords have distance $d(E(x), E(y)) \geq 2e + 1$, then we can uniquely recover x from corrupted codeword $w \in \{0,1\}^m$ with e errors ($d(w, E(x)) = e$)
- Hadamard code ($m = 2^n$):

(3) List-decoding of Hadamard code

- Error-correcting code: $E : \{0,1\}^n \to \{0,1\}^m$
- If all codewords have distance $d(E(x), E(y)) \geq 2e + 1$, then we can uniquely recover x from corrupted codeword $w \in \{0,1\}^m$ with e errors ($d(w, E(x)) = e$)
- Hadamard code ($m = 2^n$): $E(x)_y = x \cdot y$ mod 2

(3) List-decoding of Hadamard code

- Error-correcting code: $E : \{0,1\}^n \to \{0,1\}^m$
- If all codewords have distance $d(E(x), E(y)) \geq 2e + 1$, then we can uniquely recover x from corrupted codeword $w \in \{0,1\}^m$ with e errors ($d(w, E(x)) = e$)
- Hadamard code ($m = 2^n$): $E(x)_y = x \cdot y$ mod 2
- All codewords are at distance $m/2$

(3) List-decoding of Hadamard code

- Error-correcting code: $E : \{0,1\}^n \to \{0,1\}^m$
- If all codewords have distance $d(E(x), E(y)) \geq 2e + 1$, then we can uniquely recover x from corrupted codeword $w \in \{0,1\}^m$ with e errors ($d(w, E(x)) = e$)
- Hadamard code ($m = 2^n$): $E(x)_y = x \cdot y$ mod 2
- All codewords are at distance $m/2$ $\Rightarrow$ given w with $e < m/4$ errors, there is a unique x with $d(w, E(x)) \leq e$

(3) List-decoding of Hadamard code

- Error-correcting code: $E : \{0,1\}^n \to \{0,1\}^m$
- If all codewords have distance $d(E(x), E(y)) \geq 2e + 1$, then we can uniquely recover x from corrupted codeword $w \in \{0,1\}^m$ with e errors ($d(w, E(x)) = e$)
- Hadamard code ($m = 2^n$): $E(x)_y = x \cdot y$ mod 2
- All codewords are at distance $m/2$ $\Rightarrow$ given w with $e < m/4$ errors, there is a unique x with $d(w, E(x)) \leq e$
- Problem: if $e \geq m/4$ errors, then there may be many different x with $d(w, E(x)) \leq e$

(3) List-decoding of Hadamard code

- Error-correcting code: $E : \{0,1\}^n \to \{0,1\}^m$
- If all codewords have distance $d(E(x), E(y)) \geq 2e + 1$, then we can uniquely recover x from corrupted codeword $w \in \{0,1\}^m$ with e errors ($d(w, E(x)) = e$)
- Hadamard code ($m = 2^n$): $E(x)_y = x \cdot y$ mod 2
- All codewords are at distance $m/2$ $\Rightarrow$ given w with $e < m/4$ errors, there is a unique x with $d(w, E(x)) \leq e$
- Problem: if $e \geq m/4$ errors, then there may be many different x with $d(w, E(x)) \leq e$
- Example: $w = 0^{3m/4}1^{m/4}$ could've come from codewords $E(0^n) = 0^m$ or $E(10^{n-1}) = 0^{m/2}1^{m/2}$

(3) List-decoding of Hadamard code

- Error-correcting code: $E : \{0,1\}^n \to \{0,1\}^m$
- If all codewords have distance $d(E(x), E(y)) \geq 2e+1$, then we can uniquely recover x from corrupted codeword $w \in \{0,1\}^m$ with e errors ($d(w, E(x)) = e$)
- Hadamard code ($m = 2^n$): $E(x)_y = x \cdot y$ mod 2
- All codewords are at distance $m/2$ $\Rightarrow$ given w with $e < m/4$ errors, there is a unique x with $d(w, E(x)) \leq e$
- Problem: if $e \geq m/4$ errors, then there may be many different x with $d(w, E(x)) \leq e$
- Example: $w = 0^{3m/4}1^{m/4}$ could've come from codewords $E(0^n) = 0^m$ or $E(10^{n-1}) = 0^{m/2}1^{m/2}$
- List-decoding: output the whole list (hopefully small)

List-decoding of Hadamard code (cntd)

List-decoding of Hadamard code (cntd)

- List-decoding: given corrupted codeword $w \in \{0,1\}^m$ and error bound e, output list $\{x : d(w, E(x)) \leq e\}$

List-decoding of Hadamard code (cntd)

- List-decoding: given corrupted codeword $w \in \{0,1\}^m$ and error bound e, output list $\{x : d(w, E(x)) \leq e\}$
- For Hadamard code: if $e \leq (1/2 - \varepsilon)m$, then this list has only $O(1/\varepsilon^2)$ elements!

List-decoding of Hadamard code (cntd)

- List-decoding: given corrupted codeword $w \in \{0,1\}^m$ and error bound e, output list $\{x : d(w, E(x)) \leq e\}$
- For Hadamard code: if $e \leq (1/2 - \varepsilon)m$, then this list has only $O(1/\varepsilon^2)$ elements!
- Why?

List-decoding of Hadamard code (cntd)

- List-decoding: given corrupted codeword $w \in \{0,1\}^m$ and error bound e, output list $\{x : d(w, E(x)) \leq e\}$
- For Hadamard code: if $e \leq (1/2 - \varepsilon)m$, then this list has only $O(1/\varepsilon^2)$ elements!
- Why? Fourier analysis!

List-decoding of Hadamard code (cntd)

- List-decoding: given corrupted codeword $w \in \{0,1\}^m$ and error bound e, output list $\{x : d(w, E(x)) \le e\}$
- For Hadamard code: if $e \le (1/2 - \varepsilon)m$, then this list has only $O(1/\varepsilon^2)$ elements!
- Why? Fourier analysis!

  1. View w as function $w : \{0,1\}^n \to \{\pm 1\}$, and $E(s) = \chi_s$

List-decoding of Hadamard code (cntd)

- List-decoding: given corrupted codeword $w \in \{0,1\}^m$ and error bound e, output list $\{x : d(w, E(x)) \leq e\}$
- For Hadamard code: if $e \leq (1/2 - \varepsilon)m$, then this list has only $O(1/\varepsilon^2)$ elements!
- Why? Fourier analysis!
  1. View w as function $w : \{0,1\}^n \to \{\pm 1\}$, and $E(s) = \chi_s$
  2. If $d(w, E(s)) \leq (1/2 - \varepsilon)m$, then $\widehat{w}(s) \geq 2\varepsilon$

List-decoding of Hadamard code (cntd)

- List-decoding: given corrupted codeword $w \in \{0,1\}^m$ and error bound e, output list $\{x : d(w, E(x)) \leq e\}$
- For Hadamard code: if $e \leq (1/2 - \varepsilon)m$, then this list has only $O(1/\varepsilon^2)$ elements!
- Why? Fourier analysis!
  1. View w as function $w : \{0,1\}^n \to \{\pm 1\}$, and $E(s) = \chi_s$
  2. If $d(w, E(s)) \leq (1/2 - \varepsilon)m$, then $\widehat{w}(s) \geq 2\varepsilon$
  3. $\sum_s \widehat{w}(s)^2 = 1$ (by Parseval)

List-decoding of Hadamard code (cntd)

- List-decoding: given corrupted codeword $w \in \{0,1\}^m$ and error bound e, output list $\{x : d(w, E(x)) \leq e\}$
- For Hadamard code: if $e \leq (1/2 - \varepsilon)m$, then this list has only $O(1/\varepsilon^2)$ elements!
- Why? Fourier analysis!
  1. View w as function $w : \{0,1\}^n \to \{\pm 1\}$, and $E(s) = \chi_s$
  2. If $d(w, E(s)) \leq (1/2 - \varepsilon)m$, then $\widehat{w}(s) \geq 2\varepsilon$
  3. $\sum_s \widehat{w}(s)^2 = 1$ (by Parseval), hence at most $\frac{1}{4\varepsilon^2}$ different s satisfy $\widehat{w}(s) \geq 2\varepsilon$

List-decoding of Hadamard code (cntd)

- List-decoding: given corrupted codeword $w \in \{0,1\}^m$ and error bound e, output list $\{x : d(w, E(x)) \le e\}$
- For Hadamard code: if $e \le (1/2 - \varepsilon)m$, then this list has only $O(1/\varepsilon^2)$ elements!
- Why? Fourier analysis!
  1. View w as function $w : \{0,1\}^n \to \{\pm 1\}$, and $E(s) = \chi_s$
  2. If $d(w, E(s)) \le (1/2 - \varepsilon)m$, then $\widehat{w}(s) \ge 2\varepsilon$
  3. $\sum_s \widehat{w}(s)^2 = 1$ (by Parseval), hence at most $\frac{1}{4\varepsilon^2}$ different s satisfy $\widehat{w}(s) \ge 2\varepsilon$
- Goldreich and Levin show how to find this list efficiently

List-decoding of Hadamard code (cntd)

- List-decoding: given corrupted codeword $w \in \{0,1\}^m$ and error bound e, output list $\{x : d(w, E(x)) \leq e\}$
- For Hadamard code: if $e \leq (1/2 - \varepsilon)m$, then this list has only $O(1/\varepsilon^2)$ elements!
- Why? Fourier analysis!
  1. View w as function $w : \{0,1\}^n \to \{\pm 1\}$, and $E(s) = \chi_s$
  2. If $d(w, E(s)) \leq (1/2 - \varepsilon)m$, then $\widehat{w}(s) \geq 2\varepsilon$
  3. $\sum_s \widehat{w}(s)^2 = 1$ (by Parseval), hence at most $\frac{1}{4\varepsilon^2}$ different s satisfy $\widehat{w}(s) \geq 2\varepsilon$
- Goldreich and Levin show how to find this list efficiently
- There are codes with much better rate that are still efficiently list-decodable

List-decoding of Hadamard code (cntd)

- List-decoding: given corrupted codeword $w \in \{0,1\}^m$ and error bound e, output list $\{x : d(w, E(x)) \le e\}$
- For Hadamard code: if $e \le (1/2 - \varepsilon)m$, then this list has only $O(1/\varepsilon^2)$ elements!
- Why? Fourier analysis!
  1. View w as function $w : \{0,1\}^n \to \{\pm 1\}$, and $E(s) = \chi_s$
  2. If $d(w, E(s)) \le (1/2 - \varepsilon)m$, then $\widehat{w}(s) \ge 2\varepsilon$
  3. $\sum_s \widehat{w}(s)^2 = 1$ (by Parseval), hence at most $\frac{1}{4\varepsilon^2}$ different s satisfy $\widehat{w}(s) \ge 2\varepsilon$
- Goldreich and Levin show how to find this list efficiently
- There are codes with much better rate that are still efficiently list-decodable (e.g. Reed-Solomon)

(4) Influence of variables

(4) Influence of variables

- Consider a Boolean function $f : \{0,1\}^n \to \{0,1\}$

(4) Influence of variables

- Consider a Boolean function $f : \{0,1\}^n \to \{0,1\}$
- The influence of variable i is the probability that x_i determines the function value

(4) Influence of variables

- Consider a Boolean function $f : \{0,1\}^n \to \{0,1\}$
- The influence of variable i is the probability that x_i determines the function value:

$$\mathrm{Inf}_f(i) = \Pr_{x \in \{0,1\}^n}[f(x) \neq f(x \oplus e_i)]$$

(4) Influence of variables

- Consider a Boolean function $f : \{0,1\}^n \to \{0,1\}$
- The influence of variable i is the probability that x_i determines the function value:

$$\mathsf{Inf}_f(i) = \Pr_{x \in \{0,1\}^n} [f(x) \neq f(x \oplus e_i)]$$

- For things like voting and distributed coin-flipping: would like to find a balanced f where each $\mathsf{Inf}_f(i) \approx 1/n$

(4) Influence of variables

- Consider a Boolean function $f : \{0,1\}^n \to \{0,1\}$
- The influence of variable i is the probability that x_i determines the function value:

$$\mathsf{Inf}_f(i) = \Pr_{x \in \{0,1\}^n}[f(x) \neq f(x \oplus e_i)]$$

- For things like voting and distributed coin-flipping: would like to find a balanced f where each $\mathsf{Inf}_f(i) \approx 1/n$
- KKL 88: if f is balanced, then there always is an i with $\mathsf{Inf}_f(i) \geq \log(n)/n$

(4) Influence of variables

- Consider a Boolean function $f : \{0,1\}^n \to \{0,1\}$
- The influence of variable i is the probability that x_i determines the function value:

$$\mathsf{Inf}_f(i) = \Pr_{x \in \{0,1\}^n}[f(x) \neq f(x \oplus e_i)]$$

- For things like voting and distributed coin-flipping: would like to find a balanced f where each $\mathsf{Inf}_f(i) \approx 1/n$
- KKL 88: if f is balanced, then there always is an i with $\mathsf{Inf}_f(i) \geq \log(n)/n$
- This implies there is a set of $O(n/\log(n))$ variables that controls f with high probability

The KKL proof

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i)$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\mathrm{Inf}_f(i)$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\text{Inf}_f(i) = \Pr[f_i \neq 0]$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\text{Inf}_f(i) = \Pr[f_i \neq 0] = \text{Exp}[f_i^2]$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\text{Inf}_f(i) = \Pr[f_i \neq 0] = \text{Exp}[f_i^2] = \sum_s \widehat{f_i}(s)^2$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\text{Inf}_f(i) = \Pr[f_i \neq 0] = \text{Exp}[f_i^2] = \sum_s \widehat{f_i}(s)^2 = 4 \sum_{s:s_i=1} \widehat{f}(s)^2$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\text{Inf}_f(i) = \Pr[f_i \neq 0] = \text{Exp}[f_i^2] = \sum_s \widehat{f_i}(s)^2 = 4 \sum_{s:s_i=1} \widehat{f}(s)^2$
- If $L = \sum_{s:|s|>\log n} \widehat{f}(s)^2 \geq 1/3$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\text{Inf}_f(i) = \Pr[f_i \neq 0] = \text{Exp}[f_i^2] = \sum_s \widehat{f_i}(s)^2 = 4 \sum_{s:s_i=1} \widehat{f}(s)^2$
- If $L = \sum_{s:|s|>\log n} \widehat{f}(s)^2 \geq 1/3$, then $\sum_{i=1}^n \text{Inf}_f(i)$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\text{Inf}_f(i) = \Pr[f_i \neq 0] = \text{Exp}[f_i^2] = \sum_s \widehat{f_i}(s)^2 = 4 \sum_{s:s_i=1} \widehat{f}(s)^2$
- If $L = \sum_{s:|s|>\log n} \widehat{f}(s)^2 \geq 1/3$, then $\sum_{i=1}^n \text{Inf}_f(i)$
  $= 4 \sum_s |s| \widehat{f}(s)^2$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\text{Inf}_f(i) = \Pr[f_i \neq 0] = \text{Exp}[f_i^2] = \sum_s \widehat{f_i}(s)^2 = 4 \sum_{s:s_i=1} \widehat{f}(s)^2$
- If $L = \sum_{s:|s|>\log n} \widehat{f}(s)^2 \geq 1/3$, then $\sum_{i=1}^n \text{Inf}_f(i)$
  $= 4\sum_s |s|\widehat{f}(s)^2 \geq \Omega(\log n)$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\text{Inf}_f(i) = \Pr[f_i \neq 0] = \text{Exp}[f_i^2] = \sum_s \widehat{f_i}(s)^2 = 4 \sum_{s:s_i=1} \widehat{f}(s)^2$
- If $L = \sum_{s:|s|>\log n} \widehat{f}(s)^2 \geq 1/3$, then $\sum_{i=1}^n \text{Inf}_f(i)$
$= 4 \sum_s |s| \widehat{f}(s)^2 \geq \Omega(\log n) \ \Rightarrow \max_i \text{Inf}_f(i) \geq \Omega(\log(n)/n)$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\text{Inf}_f(i) = \Pr[f_i \neq 0] = \text{Exp}[f_i^2] = \sum_s \widehat{f_i}(s)^2 = 4 \sum_{s:s_i=1} \widehat{f}(s)^2$
- If $L = \sum_{s:|s|>\log n} \widehat{f}(s)^2 \geq 1/3$, then $\sum_{i=1}^n \text{Inf}_f(i)$
  $= 4\sum_s |s|\widehat{f}(s)^2 \geq \Omega(\log n) \;\Rightarrow \max_i \text{Inf}_f(i) \geq \Omega(\log(n)/n)$
- If $L < 1/3$, then use KKL inequality (special case of Bonami-Beckner):

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\mathrm{Inf}_f(i) = \Pr[f_i \neq 0] = \mathrm{Exp}[f_i^2] = \sum_s \widehat{f_i}(s)^2 = 4 \sum_{s:s_i=1} \widehat{f}(s)^2$
- If $L = \sum_{s:|s|>\log n} \widehat{f}(s)^2 \geq 1/3$, then $\sum_{i=1}^n \mathrm{Inf}_f(i)$ $= 4 \sum_s |s| \widehat{f}(s)^2 \geq \Omega(\log n) \;\Rightarrow \max_i \mathrm{Inf}_f(i) \geq \Omega(\log(n)/n)$
- If $L < 1/3$, then use KKL inequality (special case of Bonami-Beckner): $\forall g : \{0,1\}^n \to \{-1, 0, +1\}$, $\delta \in [0, 1]$

$$\sum_{s \in \{0,1\}^n} \delta^{|s|} \widehat{g}(s)^2 \leq \Pr[g \neq 0]^{2/(1+\delta)}$$

The KKL proof

- Define $f_i(x) = f(x) - f(x \oplus e_i) \in \{-1, 0, +1\}$
- Then $\widehat{f_i}(s) = 2\widehat{f}(s)$ if $s_i = 1$, and $\widehat{f_i}(s) = 0$ if $s_i = 0$
- $\mathrm{Inf}_f(i) = \Pr[f_i \neq 0] = \mathrm{Exp}[f_i^2] = \sum_s \widehat{f_i}(s)^2 = 4 \sum_{s:s_i=1} \widehat{f}(s)^2$
- If $L = \sum_{s:|s|>\log n} \widehat{f}(s)^2 \geq 1/3$, then $\sum_{i=1}^n \mathrm{Inf}_f(i)$ $= 4 \sum_s |s| \widehat{f}(s)^2 \geq \Omega(\log n) \;\Rightarrow \max_i \mathrm{Inf}_f(i) \geq \Omega(\log(n)/n)$
- If $L < 1/3$, then use KKL inequality (special case of Bonami-Beckner): $\forall g : \{0,1\}^n \to \{-1, 0, +1\}$, $\delta \in [0, 1]$

$$\sum_{s \in \{0,1\}^n} \delta^{|s|} \widehat{g}(s)^2 \leq \Pr[g \neq 0]^{2/(1+\delta)}$$

A calculation shows $\max_i \mathrm{Inf}_f(i) \geq \Omega(\log(n)/n)$

Summary

Summary

- Fourier analysis of Boolean functions is an increasingly prominent tool in theoretical computer science

Summary

- Fourier analysis of Boolean functions is an increasingly prominent tool in theoretical computer science
- We showed a few simple but beautiful examples:

Summary

- Fourier analysis of Boolean functions is an increasingly prominent tool in theoretical computer science
- We showed a few simple but beautiful examples:

1. Approximating low-degree functions by parities

Summary

- Fourier analysis of Boolean functions is an increasingly prominent tool in theoretical computer science

- We showed a few simple but beautiful examples:

1. Approximating low-degree functions by parities

2. List-decoding of Hadamard codes

Summary

- Fourier analysis of Boolean functions is an increasingly prominent tool in theoretical computer science
- We showed a few simple but beautiful examples:

1. Approximating low-degree functions by parities
2. List-decoding of Hadamard codes
3. Learning under the uniform distribution

Summary

- Fourier analysis of Boolean functions is an increasingly prominent tool in theoretical computer science
- We showed a few simple but beautiful examples:

1. Approximating low-degree functions by parities
2. List-decoding of Hadamard codes
3. Learning under the uniform distribution
4. The influence of variables on Boolean functions

Warning: these are powerful techniques!