



# Learning and Fourier Analysis

**Grigory Yaroslavtsev**

<http://grigory.us>

Slides at

<http://grigory.us/cis625/lecture2.pdf>

**CIS 625: Computational Learning Theory**

# Fourier Analysis and Learning

- Powerful tool for PAC-style learning under **uniform distribution** over  $\{0,1\}^n$
- Sometimes requires **queries** of the form  $f(x)$
- Works for learning many classes of functions, e.g:
  - Monotone, DNF, decision trees, low-degree polynomials
  - Small circuits, halfspaces, k-linear, juntas (depend on small # of variables)
  - Submodular functions (analog of convex/concave)
  - ...
- Can be extended to **product** distributions over  $\{0,1\}^n$ , i.e.  $D = D_1 \times D_2 \times \cdots \times D_n$  where  $\times$  means that draws are independent

# Recap: Fourier Analysis

- Functions as vectors form a vector space:

$$\mathbf{f}: \{-1,1\}^n \rightarrow \mathbb{R} \Leftrightarrow \mathbf{f} \in \mathbb{R}^{2^n}$$

- Inner product on functions = “correlation”:

$$\langle \mathbf{f}, \mathbf{g} \rangle = 2^{-n} \sum_{x \in \{-1,1\}^n} \mathbf{f}(x) \mathbf{g}(x) = \mathbb{E}_{x \sim \{-1,1\}^n} [\mathbf{f}(x) \mathbf{g}(x)]$$

- **Thm:** Every function  $\mathbf{f}: \{-1,1\} \rightarrow \mathbb{R}$  can be **uniquely** represented as a multilinear polynomial

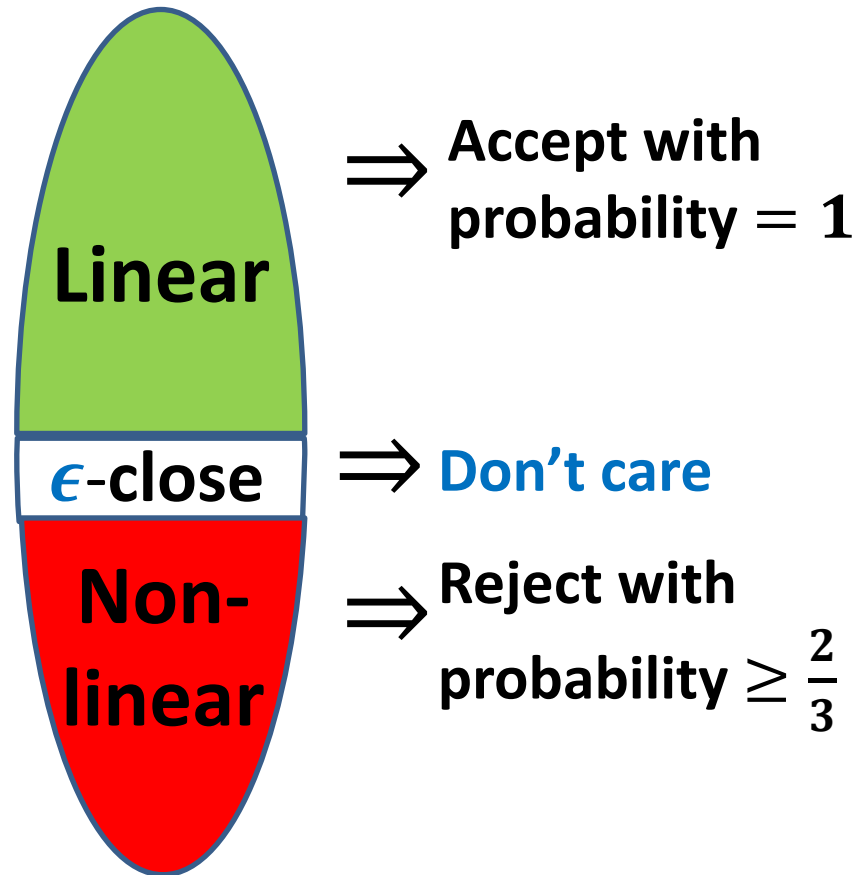
$$\mathbf{f}(x_1, \dots, x_n) = \sum_{\mathbf{S} \subseteq [n]} \hat{\mathbf{f}}(\mathbf{S}) \chi_{\mathbf{S}}(x)$$

- $\hat{\mathbf{f}}(\mathbf{S}) \equiv$  Fourier coefficient of  $\mathbf{f}$  on  $\mathbf{S} = \langle \mathbf{f}, \chi_{\mathbf{S}} \rangle$

# Linearity Testing

- $f: \{0,1\}^n \rightarrow \{0,1\}$
- $\mathcal{P}$  = class of linear functions
- $dist(f, \mathcal{P}) = \min_{g \in \mathcal{P}} dist(f, g)$
- $\epsilon$ -close:  $dist(f, \mathcal{P}) \leq \epsilon$

## Linearity Tester



# Local Correction

- Learning linear functions takes  $n$  queries
- **Lem:** If  $f$  is  $\epsilon$ -close to linear function  $\chi_S$  then for **every**  $x$  one can compute  $\chi_S(x)$  w.p.  $1 - 2\epsilon$  as:
  - Pick  $y \sim \{0,1\}^n$
  - Output  $f(y) \oplus f(x \oplus y)$

- **Proof:**

$$\Pr[f(y) \neq \chi_S(y)] = \Pr[f(x \oplus y) \neq \chi_S(x \oplus y)] = \epsilon$$

By union bound:

$$\Pr[f(y) = \chi_S(y), f(x \oplus y) = \chi_S(x \oplus y)] \geq 1 - 2\epsilon$$

$$\text{Then } f(y) \oplus f(x \oplus y) = \chi_S(y) \oplus \chi_S(x \oplus y) = \chi_S(x)$$

# Recap: PAC-style learning

- **PAC**-learning under uniform distribution: for a class of functions  $\mathcal{C}$ , given access to  $f \in \mathcal{C}$  and  $\epsilon$  find a hypothesis  $h$  such that  $dist(f, h) \leq \epsilon$
- Two query access models:
  - Random samples  $(x, f(x))$ , where  $x \sim \{-1, 1\}^n$
  - Queries:  $(x, f(x))$ , for any  $x \in \{-1, 1\}^n$
- Fourier analysis helps because of **sparsity** in Fourier spectrum
  - Low-degree concentration
  - Concentration on a small number of Fourier coefficients

# Fourier Analysis and Learning

- **Def (Fourier Concentration):** Fourier spectrum of  $f: \{-1,1\}^n \rightarrow \mathbb{R}$  is  $\epsilon$ -concentrated on a collection of subsets  $\mathbb{F}$  if:

$$\sum_{S \subseteq [n], S \in \mathbb{F}} \hat{f}(S)^2 \geq 1 - \epsilon$$

- **Thm (Sparse Fourier Algorithm):** Given  $\mathbb{F}$  on which  $f: \{-1,1\}^n \rightarrow \{-1,1\}$  is  $\epsilon/2$ -concentrated there is an algorithm that PAC-learns  $f$  with  $O(|\mathbb{F}| \log |\mathbb{F}| / \epsilon)$  random samples

# Estimating Fourier Coefficients

- **Lemma:** Given  $\mathcal{S}$  and  $O\left(\log\frac{1}{\delta}/\epsilon^2\right)$  random samples from  $f: \{-1,1\}^n \rightarrow \{-1,1\}$  there is an algorithm that gives  $\tilde{f}(\mathcal{S})$  such that with prob.  $\geq 1 - \delta$ :
$$|\tilde{f}(\mathcal{S}) - \hat{f}(\mathcal{S})| \leq \epsilon$$
- **Proof:**  $\hat{f}(\mathcal{S}) = \mathbb{E}_x[f(x)\chi_{\mathcal{S}}(x)]$
- Given  $k = O\left(\log\frac{1}{\delta}/\epsilon^2\right)$  random samples  $(x_i, f(x_i))$
- Empirical average  $\frac{1}{k} \times \sum_i^k f(x_i)\chi_{\mathcal{S}}(x_i)$   $\epsilon$ -close by a Chernoff bound with prob.  $\geq 1 - \delta$



# Rounding real-valued approximations

- **Lem:** If  $\mathbf{f}: \{-1,1\}^n \rightarrow \{-1,1\}$ ,  $\mathbf{g}: \{-1,1\}^n \rightarrow \mathbb{R}$  such that  $\mathbb{E}_x \left[ \|\mathbf{f} - \mathbf{g}\|_2^2 \right] \leq \epsilon$ . For  $\mathbf{h}: \{-1,1\}^n \rightarrow \{-1,1\}$  defined as  $\mathbf{h}(x) = \text{sign}(\mathbf{g}(x))$ :

$$\text{dist}(\mathbf{f}, \mathbf{h}) \leq \epsilon$$

- **Proof:**  $\mathbf{f}(x) \neq \mathbf{g}(x) \Rightarrow |\mathbf{f}(x) - \mathbf{g}(x)|^2 \geq 1$

$$\begin{aligned} \text{dist}(\mathbf{f}, \mathbf{h}) &= \Pr_x[\mathbf{f}(x) \neq \mathbf{h}(x)] = \\ &= \mathbb{E}_x \left[ \mathbf{1}_{\mathbf{f}(x) \neq \text{sign}(\mathbf{g}(x))} \right] \leq \\ &= \mathbb{E}_x \left[ \|\mathbf{f} - \mathbf{g}\|_2^2 \right] \leq \epsilon \end{aligned}$$

# Sparse Fourier Algorithm

- **Thm (Sparse Fourier Algorithm):**

Given  $\mathbb{F}$  such that  $f : \{-1,1\}^n \rightarrow \{-1,1\}$  is  $\epsilon/2$ -concentrated on  $\mathbb{F}$  there is a **Sparse Fourier Algorithm** which PAC-learns  $f$  with  $O(|\mathbb{F}| \log |\mathbb{F}| / \epsilon)$  random samples.

- For each  $\mathbf{S} \in \mathbb{F}$  get  $\tilde{f}(\mathbf{S})$  with prob.  $1 - 1/10|\mathbb{F}|$ :

$$|\tilde{f}(\mathbf{S}) - \hat{f}(\mathbf{S})| \leq \sqrt{\epsilon}/2\sqrt{|\mathbb{F}|}$$

- **Output:**  $h = \text{sign}(g)$  where  $g = \sum_{\mathbf{S} \in \mathbb{F}} \tilde{f}(\mathbf{S}) \chi_{\mathbf{S}}$

$$\|f - g\|_2^2 = \sum_{\mathbf{S}} (\widehat{f - g})(\mathbf{S})^2 =$$

$$\sum_{\mathbf{S} \in \mathbb{F}} |\tilde{f}(\mathbf{S}) - \hat{f}(\mathbf{S})|^2 + \sum_{\mathbf{S} \in \mathbb{F}} \hat{f}(\mathbf{S})^2 \leq \sum_{\mathbf{S} \in \mathbb{F}} \left( \frac{\sqrt{\epsilon}}{2\sqrt{|\mathbb{F}|}} \right)^2 + \frac{\epsilon}{2} \leq \epsilon$$

# Low-Degree Algorithm

- Some classes are  $\epsilon$ -concentrated on low degree Fourier coefficients:  $\mathbb{F} = \{\mathbf{S}: |\mathbf{S}| \leq k\}$ ,  $k \ll n$
- $|\mathbb{F}| \leq n^k$
- Monotone functions:  $k = O(\sqrt{n}/\epsilon)$ 
  - Learning complexity:  $n^{\tilde{O}(\sqrt{n}/\epsilon)}$
- Size- $s$  decision trees:  $k = O((\log s)/\epsilon)$ 
  - Learning complexity:  $n^{O((\log s)/\epsilon)}$
- Depth- $d$  decision trees:  $k = O(d/\epsilon)$ 
  - Learning complexity:  $n^{O(d/\epsilon)}$

# Restrictions

- **Def:** For a partition  $(J, \bar{J})$  of  $[n]$  and  $\mathbf{z} \in \{-1, 1\}^{\bar{J}}$  let the **restriction**  $f_{J|\mathbf{z}}: \{-1, 1\}^{|J|} \rightarrow \mathbb{R}$  be

$$f_{J|\mathbf{z}}(y) = f(y, \mathbf{z})$$

where  $(y, \mathbf{z})$  is a string composed of  $y$  and  $\mathbf{z}$ .

- **Example:**

$$\mathit{min}(x_1, x_2) = -\frac{1}{2} + \frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_1x_2$$

$$J = \{1\}, \bar{J} = \{2\}, \mathbf{z} = 1 \Rightarrow$$

$$f_{J|\mathbf{z}}: \{-1, 1\} \rightarrow \{-1, 1\} = x_1$$

# Fourier coefficients of restrictions

- Fourier coefficients of  $f_{J|z}$  can be obtained from the multilinear polynomial by substitution

- $\hat{f}_{J|z}(S) = \sum_{T \subseteq \bar{J}} \hat{f}(S \cup T) \chi_T(z)$

- $\mathbb{E}_z[\hat{f}_{J|z}(S)] = \hat{f}(S)$

Take  $T = \emptyset$ , otherwise  $\mathbb{E}_z[\chi_T(z)] = 0$

- $\mathbb{E}_z[\hat{f}_{J|z}(S)^2] = \sum_{T \subseteq \bar{J}} \hat{f}(S \cup T)^2$

$$\mathbb{E}_z[\hat{f}_{J|z}(S)^2] = \mathbb{E}_z \left[ \left( \sum_{T \subseteq \bar{J}} \hat{f}(S \cup T) \chi_T(z) \right)^2 \right] = \sum_{T \subseteq \bar{J}} \hat{f}(S \cup T)^2$$

since  $\mathbb{E}_z[\chi_T(z) \chi_{T'}(z)] = 0$

# Goldreich-Levin/Kushilevitz-Mansour

- **Thm (GL/KM):** Given query access to  $f: \{-1,1\}^n \rightarrow \{-1,1\}$  and  $0 < \tau \leq 1$  GL/KM-algorithm w.h.p. outputs  $L = \{U_1, \dots, U_\ell\}$ :
  - $|\hat{f}(U)| \geq \tau \Rightarrow U \in L$
  - $U \in L \Rightarrow |\hat{f}(U)| \geq \tau/2$
- **Exercise: GL/KM + Sparse Fourier Algorithm:** A class  $\mathcal{C}$  which is  $\epsilon$ -concentrated on at most  $M$  sets can be learned using  $\text{poly}\left(M, \frac{1}{\epsilon}, n\right)$  queries
  - Every large coefficient  $|\hat{f}(U)| \geq 1/\sqrt{M}$
- **Corollary:** Size- $s$  decision trees are learnable with  $\text{poly}\left(n, s, \frac{1}{\epsilon}\right)$  queries

# Estimating Fourier Weight via Restrictions

- Recall:  $\mathbb{E}_{\mathbf{z}} [\hat{f}_{J|\mathbf{z}}(\mathbf{S})^2] = \sum_{T \subseteq \bar{J}} \hat{f}(\mathbf{S} \cup T)^2$
- **Lemma:**  $\sum_{T \subseteq \bar{J}} \hat{f}(\mathbf{S} \cup T)^2$  can be estimated from  $O(1/\epsilon^2 \log 1/\delta)$  random samples w.p.  $1 - \delta$
- $\sum_{T \subseteq \bar{J}} \hat{f}(\mathbf{S} \cup T)^2 = \mathbb{E}_{\mathbf{z}} [\hat{f}_{J|\mathbf{z}}(\mathbf{S})^2] =$   
 $= \mathbb{E}_{\mathbf{z} \in \{-1,1\}^{\bar{J}}} \left[ \mathbb{E}_{\mathbf{y} \in \{-1,1\}^J} [f(\mathbf{y}, \mathbf{z}) \chi_{\mathbf{S}}(\mathbf{y})]^2 \right]$   
 $= \mathbb{E}_{\mathbf{z} \in \{-1,1\}^{\bar{J}}} \left[ \mathbb{E}_{\mathbf{y}, \mathbf{y}' \in \{-1,1\}^J} [f(\mathbf{y}, \mathbf{z}) \chi_{\mathbf{S}}(\mathbf{y}) f(\mathbf{y}', \mathbf{z}) \chi_{\mathbf{S}}(\mathbf{y}')] \right]$
- $f(\mathbf{y}, \mathbf{z}) \chi_{\mathbf{S}}(\mathbf{y}) f(\mathbf{y}', \mathbf{z}) \chi_{\mathbf{S}}(\mathbf{y}')$  is a  $\pm 1$  random variable  
 $\Rightarrow O(1/\epsilon^2 \log 1/\delta)$  samples suffice to estimate

# GL/KM-Algorithm

- Put all  $2^n$  subsets of  $[n]$  into a single “bucket”
- At each step:
  - Select any bucket  $\mathfrak{B}$  containing  $2^m$  sets,  $m \geq 1$
  - Split  $\mathfrak{B}$  into  $\mathfrak{B}_1, \mathfrak{B}_2$  of  $2^{m-1}$  sets each
  - Estimate Fourier weight  $\sum_{U \in \mathfrak{B}_i} \hat{f}(U)^2$  up to  $\tau^2/4$  for each  $\mathfrak{B}_i$
  - Discard  $\mathfrak{B}_1$  or  $\mathfrak{B}_2$  if its weight is  $\leq \frac{\tau^2}{2}$
- Output all buckets that contain a single set



# GL/KM-Algorithm: Correctness

- Put all  $2^n$  subsets of  $[n]$  into a single “bucket”
- At each step:
  - Select any bucket  $\mathfrak{B}$  containing  $2^m$  sets,  $m \geq 1$
  - Split  $\mathfrak{B}$  into  $\mathfrak{B}_1, \mathfrak{B}_2$  of  $2^{m-1}$  sets each
  - Estimate Fourier weight  $\sum_{U \in \mathfrak{B}_i} \hat{f}(U)^2$  up to  $\tau^2/4$  for each  $\mathfrak{B}_i$
  - Discard  $\mathfrak{B}_1$  or  $\mathfrak{B}_2$  if its weight is  $\leq \frac{\tau^2}{2}$
- Output all buckets that contain a single set

**Correctness** (assuming all estimates up to  $\tau^2/4$  w.h.p):

- $|\hat{f}(U)| \geq \tau \Rightarrow U \in L$ : no bucket with weight  $\geq \tau^2$  discarded
- $U \in L \Rightarrow |\hat{f}(U)| \geq \tau/2$ : buckets with weight  $\leq \tau^2/4$  discarded

# GL/KM-Algorithm: Complexity

- Put all  $2^n$  subsets of  $[n]$  into a single “bucket”
  - At each step:
    - Select any bucket  $\mathfrak{B}$  containing  $2^m$  sets,  $m \geq 1$
    - Split  $\mathfrak{B}$  into  $\mathfrak{B}_1, \mathfrak{B}_2$  of  $2^{m-1}$  sets each
    - Estimate Fourier weight  $\sum_{U \in \mathfrak{B}_i} \hat{f}(U)^2$  up to  $\tau^2/4$  for each  $\mathfrak{B}_i$
    - Discard  $\mathfrak{B}_1$  or  $\mathfrak{B}_2$  if its weight is  $\leq \frac{\tau^2}{2}$
  - Output all buckets that contain a single set
- 
- By Parseval  $\leq 4/\tau^2$  active buckets at any time
  - Bucket can be split at most  $n$  times
  - At most  $4n/\tau^2$  steps to finish

# GL/KM-Algorithm: Bucketing

- $\mathfrak{B}_{k,S} = \{S \cup T : T \subseteq \{k+1, \dots, n\}\}$ ,  $|\mathfrak{B}_{k,S}| = 2^{n-k}$
- Initial bucket  $\mathfrak{B}_{0,\emptyset}$
- Split  $\mathfrak{B}_{k,S}$  into  $\mathfrak{B}_{k+1,S}$  and  $\mathfrak{B}_{k+1,S \cup \{k+1\}}$
- Fourier weight of  $\mathfrak{B}_{k,S} = \sum_{T \subseteq \{k+1, \dots, n\}} \hat{f}(S \cup T)^2$
- $\sum_{T \subseteq \{k+1, \dots, n\}} \hat{f}(S \cup T)^2$  estimated via restrictions
- Estimate each up to  $\pm \frac{\tau^2}{4}$  with prob.  $1 - \frac{\tau^2}{80n'}$   
complexity  $O\left(\frac{1}{\tau^4} \log\left(\frac{n}{\tau}\right)\right)$
- All estimates are up to  $\pm \frac{\tau^2}{4}$  with prob. 9/10

Thanks!