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 $\boldsymbol{f}(x)$
- 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:
$$\boldsymbol{f}: \{-1,1\}^n \to \mathbb{R} \Leftrightarrow \boldsymbol{f} \in \mathbb{R}^{2^n}$$
- Inner product on functions = "correlation":
$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle = 2^{-n} \sum_{x \in \{-1,1\}^n} \boldsymbol{f}(x)\boldsymbol{g}(x) = \mathbb{E}_{x \sim \{-1,1\}^n}[\boldsymbol{f}(x)\boldsymbol{g}(x)]$$
- **Thm**: Every function $\boldsymbol{f}: \{-1,1\} \to \mathbb{R}$ can be **uniquely** represented as a multilinear polynomial
$$\boldsymbol{f}(x_1, \ldots, x_n) = \sum_{\boldsymbol{S} \subseteq [n]} \hat{\boldsymbol{f}}(\boldsymbol{S}) \boldsymbol{\chi}_{\boldsymbol{S}}(x)$$
- $\hat{\boldsymbol{f}}(\boldsymbol{S})$ $\equiv$ Fourier coefficient of $\boldsymbol{f}$ on $\boldsymbol{S}$ $= \langle \boldsymbol{f}, \boldsymbol{\chi}_{\boldsymbol{S}} \rangle$

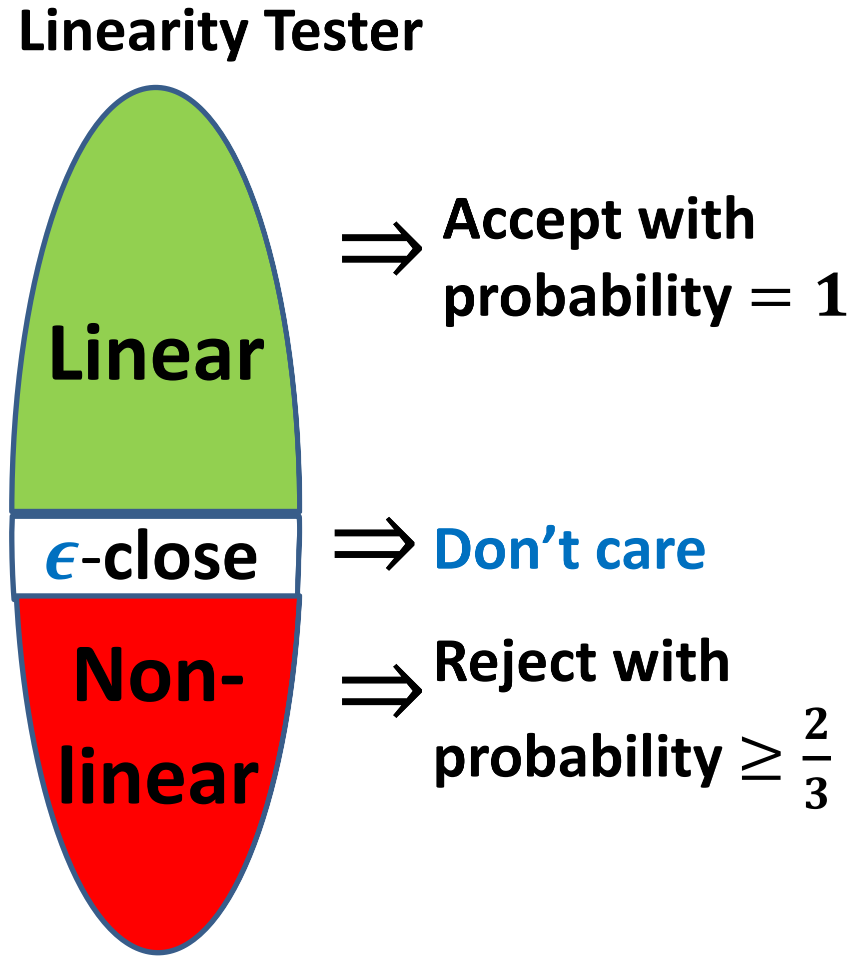
Recap:Convolution

- **Def.:** For $x, y \in \{-1,1\}^n$ define $x \odot y \in \{-1,1\}^n$:
$$(x \odot y)_i = x_i y_i$$
  - **Def.:** For $\boldsymbol{f}, \boldsymbol{g}: \{-1,1\}^n \to \mathbb{R}$ their **convolution**
$$\boldsymbol{f} * \boldsymbol{g}: \{-1,1\}^n \to \mathbb{R}:$$
$$\boldsymbol{f} * \boldsymbol{g}(x) \equiv \mathbb{E}_{y \sim \{-1,1\}^n}[\boldsymbol{f}(y)\boldsymbol{g}(x \odot y)]$$

- Properties:
  1. $\boldsymbol{f} * \boldsymbol{g} = \boldsymbol{g} * \boldsymbol{f}$
  2. $\boldsymbol{f} * (\boldsymbol{g} * \boldsymbol{h}) = (\boldsymbol{f} * \boldsymbol{g}) * \boldsymbol{h}$
  3. For all $\boldsymbol{S} \subseteq [n]$: $\widehat{\boldsymbol{f} * \boldsymbol{g}}(\boldsymbol{S}) = \hat{\boldsymbol{f}}(\boldsymbol{S})\hat{\boldsymbol{g}}(\boldsymbol{S})$

Linearity Testing

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



Local Correction

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

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

By union bound:

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

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

Recap: PAC-style learning

- **PAC**-learning under uniform distribution: for a class of functions $\boldsymbol{C}$, given access to $\boldsymbol{f} \in \boldsymbol{C}$ and $\boldsymbol{\epsilon}$ find a hypothesis $\boldsymbol{h}$ such that $dist(\boldsymbol{f}, \boldsymbol{h}) \leq \boldsymbol{\epsilon}$
- Two query access models:
  – Random samples $(x, \boldsymbol{f}(x))$, where $x \sim \{-1,1\}^n$
  – Queries: $(x, \boldsymbol{f}(x))$, for any $x \in \{-1,1\}^n$

Fourier Analysis and Learning

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

Rounding real-valued approximations

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

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

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

$$dist(\boldsymbol{f}, \boldsymbol{h}) = \Pr_{\mathrm{x}}[\boldsymbol{f}(x) \neq \boldsymbol{h}(x)] =$$
$$\mathbb{E}_x\left[1_{\boldsymbol{f}(x) \neq sign(\boldsymbol{g}(x))}\right] \leq$$
$$\mathbb{E}_x\left[||\boldsymbol{f}-\boldsymbol{g}||_2^2\right] \leq \boldsymbol{\epsilon}$$

Sparse Fourier Algorithm

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

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

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

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

- **Output:** $\boldsymbol{h} = sign(\boldsymbol{g})$ where $\boldsymbol{g} = \sum_{\boldsymbol{S} \in \mathbb{F}} \tilde{\boldsymbol{f}}(\boldsymbol{S}) \boldsymbol{\chi}_{\boldsymbol{S}}$

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

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

Low-Degree Algorithm

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

Restrictions

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

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

where (y, z) is a string composed of y and z.

- **Example:**

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

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

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

Fourier coefficients of restrictions

- Fourier coefficients of $f_{J|z}$ can be obtained from the multilinear polynomial by subsitution
- $\hat{f}_{J|z}(S) = \sum_{T\subseteq \bar{J}} \hat{f}(S \cup T)\chi_T(z)$
- $\mathbb{E}_z\left[\hat{f}_{J|z}(S)\right] = \hat{f}(S)$

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

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

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

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

Goldreich-Levin/Kushilevitz-Mansour

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

Estimating Fourier Weight via Restrictions

- Recall: $\mathbb{E}_{z}\left[\hat{f}_{J|z}(S)^2\right] = \sum_{T \subseteq \bar{J}} \hat{f}(S \cup T)^2$
- **Lemma:** $\sum_{T \subseteq \bar{J}} \hat{f}(S \cup T)^2$ can be estimated from $O(1/\epsilon^2 \log 1/\delta)$ random samples w.p. $1 - \delta$
- $\sum_{T \subseteq \bar{J}} \hat{f}(S \cup T)^2 = \mathbb{E}_{z}\left[\hat{f}_{J|z}(S)^2\right] =$

$$= \mathbb{E}_{z \in \{-1,1\}^{\bar{J}}}\left[\mathbb{E}_{y \in \{-1,1\}^{J}}[f(y,z)\chi_S(y)]^2\right]$$

$$= \mathbb{E}_{z \in \{-1,1\}^{\bar{J}}}\left[\mathbb{E}_{y,y' \in \{-1,1\}^{J}}[f(y,z)\chi_S(y)f(y',z)\chi_S(y')]\right]$$

- $f(y,z)\chi_S(y)f(y',z)\chi_S(y')$ is a ± 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 $\sum_{U \in \mathfrak{B}_i} \hat{\boldsymbol{f}}(U)^2$ up to $\boldsymbol{\tau^2}/4$ for of 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 $\sum_{U \in \mathfrak{B}_i} \hat{\boldsymbol{f}}(U)^2$ up to $\boldsymbol{\tau}^2/4$ for each $\mathfrak{B}_i$
  - Discard $\mathfrak{B}_1$ or $\mathfrak{B}_2$ if its weight is $\leq \frac{\boldsymbol{\tau}^2}{2}$
- Output all buckets that contain a single set

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

- $|\hat{\boldsymbol{f}}(U)| \geq \boldsymbol{\tau} \Rightarrow U \in L$: no bucket with weight $\geq \boldsymbol{\tau}^2$ discarded
- $U \in L \Rightarrow |\hat{\boldsymbol{f}}(U)| \geq \boldsymbol{\tau}/2$: buckets with weight $\leq \boldsymbol{\tau}^2/4$ 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 $\sum_{U \in \mathfrak{B}_i} \hat{\boldsymbol{f}}(U)^2$ up to $\boldsymbol{\tau}^2/4$ for each $\mathfrak{B}_i$
  - Discard $\mathfrak{B}_1$ or $\mathfrak{B}_2$ if its weight is $\leq \frac{\boldsymbol{\tau}^2}{2}$
- Output all buckets that contain a single set

- By Parseval $\leq 4/\boldsymbol{\tau}^2$ active buckets at any time
- Bucket can be split at most n times
- At most $4n/\boldsymbol{\tau}^2$ steps to finish

GL/KM-Algorithm: Bucketing

- $\mathfrak{B}_{k,S} = \{S \cup \mathrm{T}: T \subseteq \{k+1, \dots, n\}\}$, $|\mathfrak{B}_{k,S}| = 2^{n-k}$
- 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!