DS-563/CS-543 Lecture 16 / 2023-03-23

Today:

- wrap up learning arbitrary discrete distributions (see previous notes)
- learning of monotone discrete distributions

Monotone distributions:

D on $[n]$ where p_i = probability of i

is monotone if $p_1 \geq p_2 \geq p_3 \geq \dots \geq p_n$

How many samples from D are needed to output D' such that $d_{TV}(D, D') \leq \varepsilon$ with probability $4/5$?

We will show $O(\varepsilon^{-3} \log n)$

Partition $[n]$ into buckets $\{a_i, a_i+1, \ldots, b_i\}$

$$\underset{\{1,2,3,\ldots,n\}}{[n]}$$

$a_1 = b_1 = 1$

then

$$a_{i+1} = b_i + 1$$

$$b_{i+1} = \lceil b_i(1+\varepsilon) \rceil$$

(truncate when n is reached)

Algorithm:

- collect $t = C \cdot \varepsilon^{-3} \log n$ independent samples from D
- output D' such that probability of $i \in \{a_j, \ldots, b_j\}$ equals $(\#\text{samples in } \{a_j, \ldots, b_j\}) / (b_j - a_j + 1)$

Intuition: replace each bucket with the uniform distribution on the bucket of the same total probability

Claim: monotone distributions close to distributions uniform on each bucket $\{a_i, a_i+1, a_i+2, \ldots, b_i\}$

D – monotone distribution on $[n]$

p_i = probability of i in D

D_* – distribution created by making each bucket of D uniform

q_i = probability of i in D_*

For $i \in \{a_j, \ldots, b_j\}$, $q_i = \dfrac{\sum_{k=a_j}^{b_j} p_k}{a_j - b_j + 1}$

$$d_{TV}(D, D_*) \leq \varepsilon$$

Proof: Let's bound $\|p - q\|_1 \overset{\text{Homework 2}}{=} 2\, d_{TV}(D, D_*)$

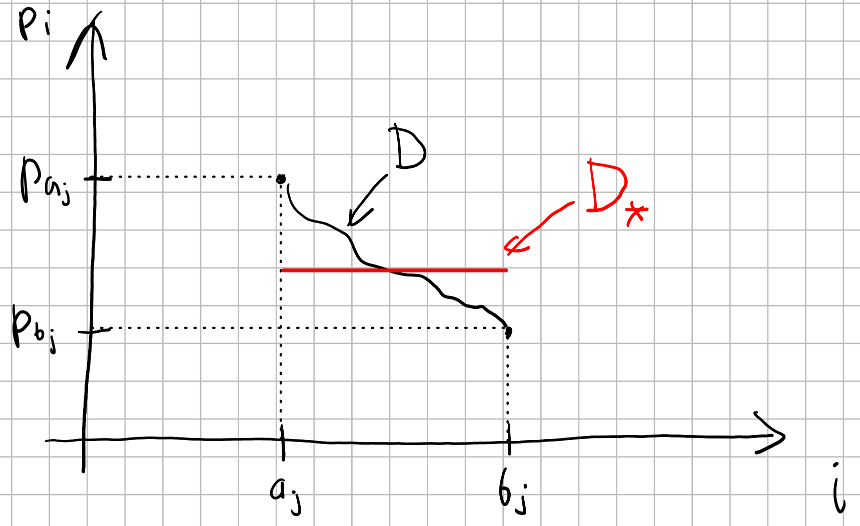
where
$$p = (p_1, \ldots, p_n) \text{ and } q = (q_1, \ldots, q_n)$$

Consider bucket j: $\{a_j, a_j+1, \ldots, b_j\}$

If $a_j = b_j$: D equals D_* on this bucket

If $a_j < b_j$:

$$b_j = \lceil b_{j-1}(1+\varepsilon) \rceil > a_j = b_{j-1} + 1$$
$$\Downarrow$$
$$b_{j-1}\varepsilon > 1$$



$$\sum_{k=a_j}^{b_j} |p_k - q_k| \le (b_j - a_j + 1)\cdot(p_{a_j} - p_{b_j})$$
$$\le \lceil \varepsilon b_{j-1} \rceil \cdot (p_{a_j} - p_{b_j})$$
$$\le (\varepsilon b_{j-1} + 1)(p_{a_j} - p_{b_j})$$
$$\le 2\varepsilon b_{j-1}(p_{a_j} - p_{b_j})$$

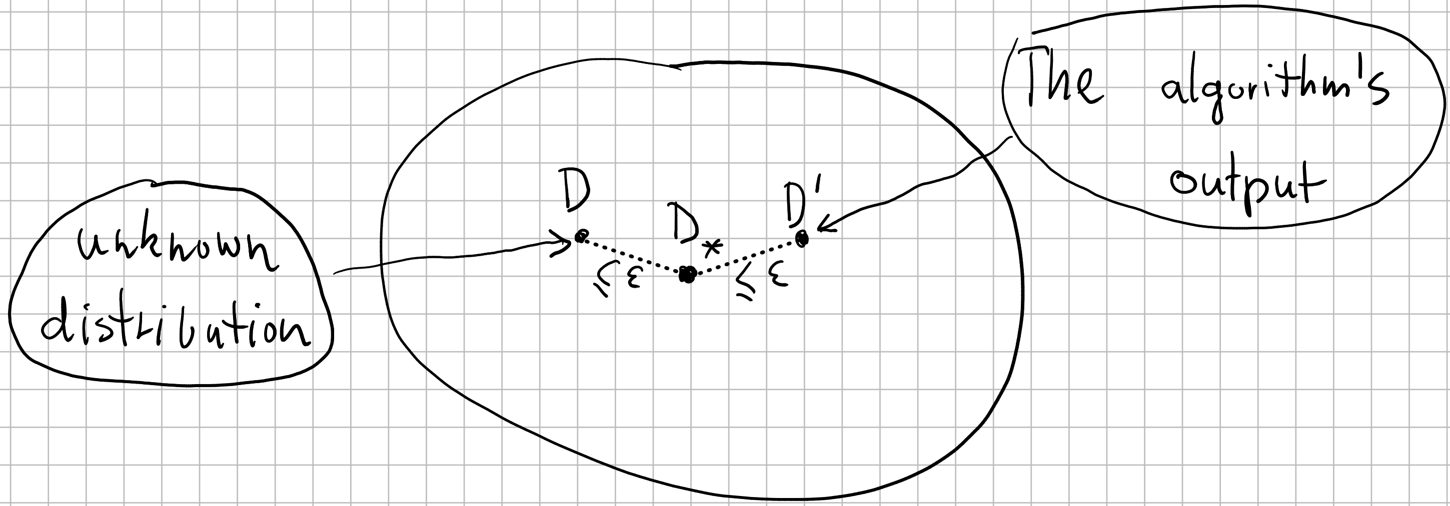
Let B = number of buckets

$$\|p - q\|_1 \le \sum_{j=2}^{B} 2\varepsilon b_{j-1} (p_{a_j} - p_{b_j})$$
$$= 2\varepsilon \sum_{j=2}^{B} \left((p_{a_j} - p_{b_j}) \sum_{k=1}^{j-1} (b_k - a_k + 1) \right)$$
$$= 2\varepsilon \sum_{k=1}^{B-1} \left((b_k - a_k + 1) \sum_{j=k+1}^{B} (p_{a_j} - p_{b_j}) \right)$$
$$\le 2\varepsilon \sum_{k=1}^{B-1} (b_k - a_k + 1)\, p_{a_{k+1}}$$
$$\le 2\varepsilon \sum_{k=1}^{B-1} \sum_{i=a_k}^{b_k} p_i \le 2\varepsilon \sum_{i=1}^{n} p_i = 2\varepsilon$$

$\boxed{16-4}$

This implies $d_{TV}(D, D_*) \leq \varepsilon$ (via Homework 2) ∎

Why the algorithm works:



$$d_{TV}(D, D') \leq 2\varepsilon \text{ with probability } 99/100$$

because

$$\underbrace{d_{TV}(D_*, D') \leq \text{ with probability } 99/100}$$

Why this holds?

- D_* and D' are both uniform on each bucket
- l_1-distance between D_* and D'

$$= \sum_{i=1}^{B} \Big| (\text{probability of bucket } i \text{ in } D_*) - (\text{probability of bucket } i \text{ in } D') \Big|$$

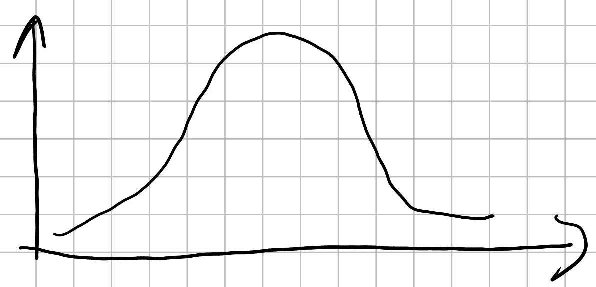
- hence it suffices to estimate the distribution on $B = O\left(\frac{1}{\varepsilon} \log n\right)$ buckets up to $d_{TV}(\ldots) \leq \varepsilon$

- this can be achieved with $O(B/\varepsilon^2)$ samples (via general distribution learning) with probability $99/100$, i.e., what the algorithm does for sufficiently large constant C

Note: $\Omega(\varepsilon^{-3} \log n)$ is needed

the lower bound for general learning can be applied to learning bucket

Learning unimodal distributions (e.g. discretized Gaussians)

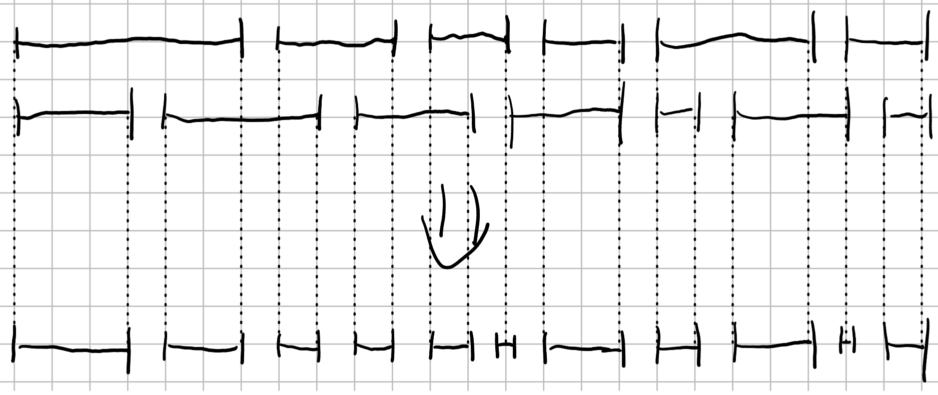


$$p_1 \leq p_2 \leq p_3 \leq \dots \leq \underset{\substack{\uparrow \\ \text{peak}}}{p_k} \geq p_{k+1} \geq p_{k+2} \geq \dots \geq p_n$$

Sketch of the general idea

- Sample $O(1/\varepsilon)$ elements
- Divide $[n]$ into $O(1/\varepsilon)$ ranges, each containing $O(1)$ samples
- The ranges correspond roughly to $\Theta(\varepsilon)$ mass of the distribution
- One of the ranges contains the peak
- For each range as the "peak candidate", create $O\left(\frac{1}{\varepsilon}\log n\right)$ buckets for ranges before and after as in learning monotone distributions (one increasing, one decreasing)
- Total of $O(\varepsilon^{-2}\log n)$ intersecting buckets
- Take their intersections to create $O(\varepsilon^{-2}\log n)$ non-intersecting buckets

Example:



- Learn the distribution on the resulting $O(\varepsilon^{-2} \log n)$ buckets up to ε in total variation distance with $O((\varepsilon^{-2} \log n)/\varepsilon^2) = O(\varepsilon^{-4} \log n)$ samples
- Output distribution uniform on each bucket with the learned probabilities of buckets