Fast and Near-Optimal Algorithms for Approximating Distributions by Histograms

Jayadev Acharya [1] Ilias Diakonikolas [2] Chinmay Hegde [1]
Jerry Li [1] Ludwig Schmidt [1]

[1]MIT

[2]University of Edinburgh

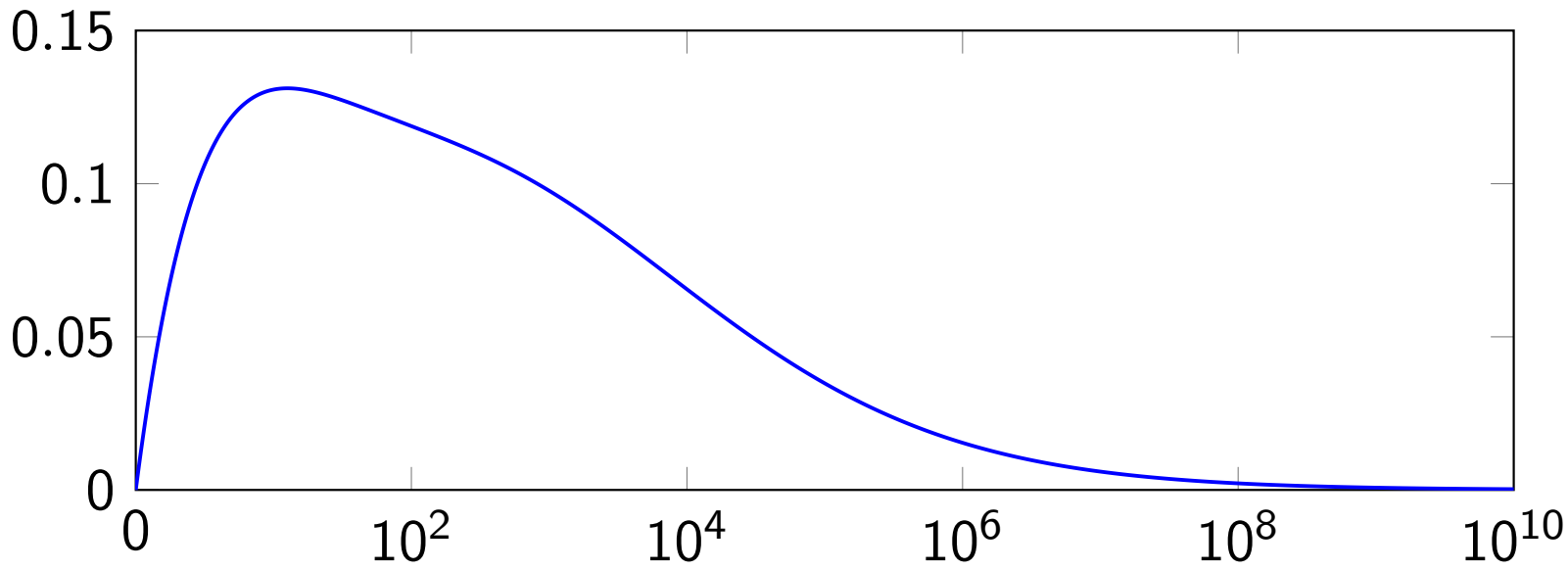
June 1, 2015

Motivating Example

- You want a representation of $f(i)$, the fraction of the world's population whose annual salary is i dollars.

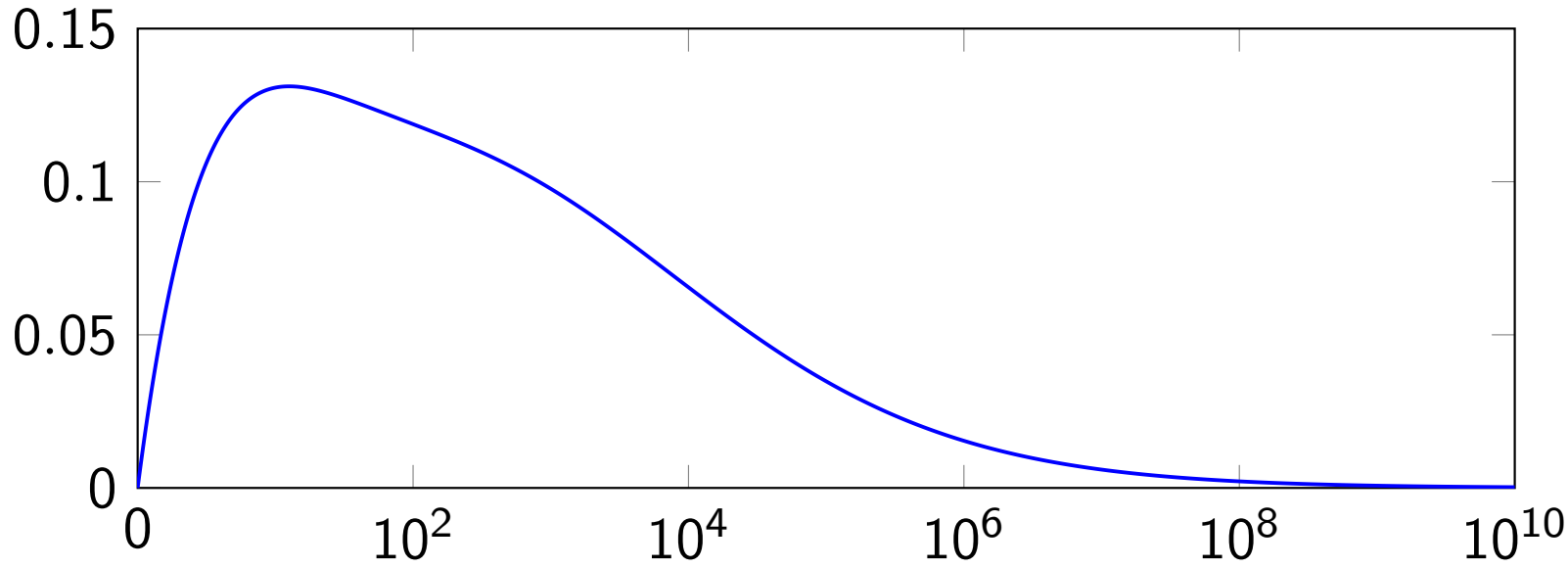
Motivating Example

- You want a representation of $f(i)$, the fraction of the world's population whose annual salary is i dollars.



Motivating Example

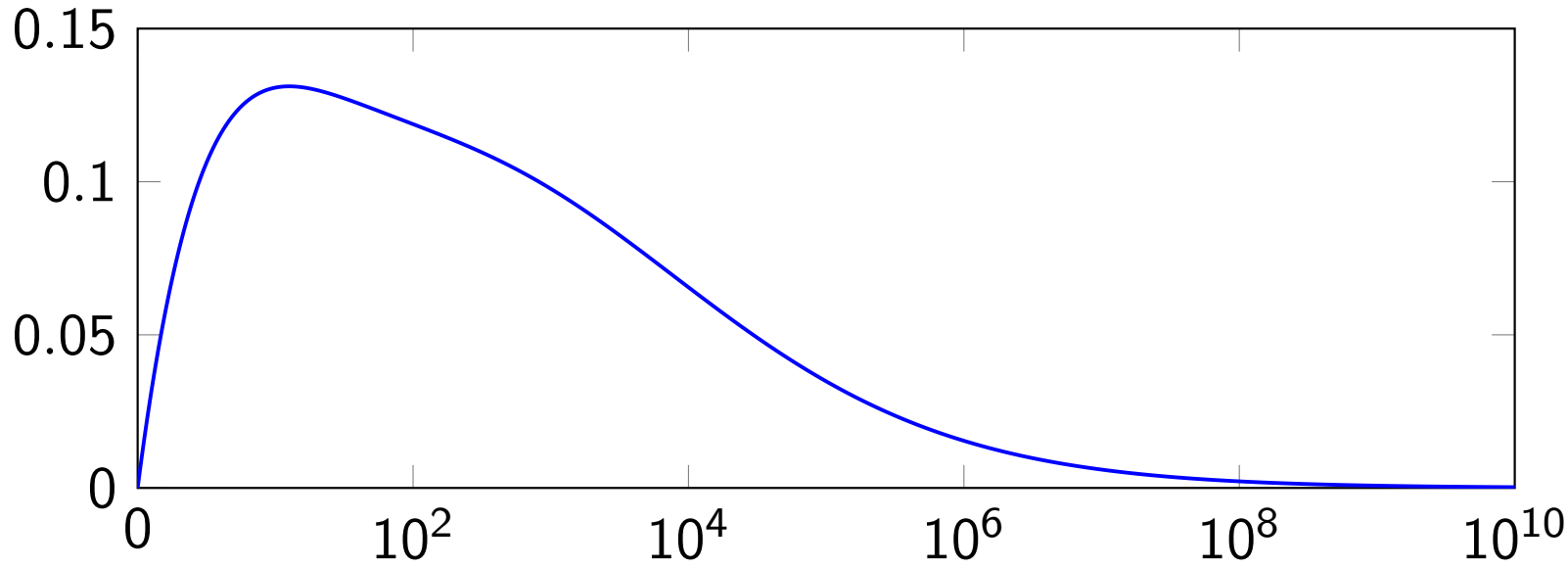
- You want a representation of $f(i)$, the fraction of the world's population whose annual salary is i dollars.



- **Problem:** Don't want to store $\sim 10^{10}$ elements.

Motivating Example

- You want a representation of $f(i)$, the fraction of the world's population whose annual salary is i dollars.



- **Problem:** Don't want to store $\sim 10^{10}$ elements.
- **Problem:** Too many people in the world, so we can't get all the data

Motivating Example (cont.)

Problem: Don't want to store $\sim 10^{10}$ elements.

Motivating Example (cont.)

Problem: Don't want to store $\sim 10^{10}$ elements.

Solution: Store a *concise representation* as a k-histogram.

Motivating Example (cont.)

Problem: Don't want to store $\sim 10^{10}$ elements.

Solution: Store a *concise representation* as a k-histogram.

Definition

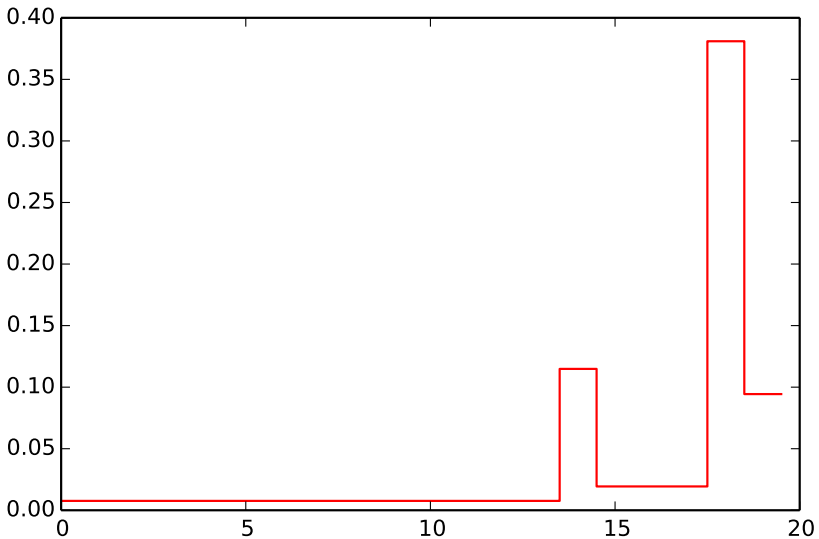
A k-histogram is a k-piecewise flat function $h : \{1, \ldots, n\} \to \mathbb{R}$.

Motivating Example (cont.)

Problem: Don't want to store $\sim 10^{10}$ elements.
Solution: Store a *concise representation* as a k-histogram.

Definition

A k-histogram is a k-piecewise flat function $h : \{1, \ldots, n\} \to \mathbb{R}$.



Motivating Example (cont.)

Problem: Don't want to store $\sim 10^{10}$ elements.

Solution: Store a *concise representation* as a k-histogram.

Definition

A k-histogram is a k-piecewise flat function $h : \{1, \ldots, n\} \to \mathbb{R}$.

Instead of storing f, efficiently find and store k-histogram h (for some small value of k) so that

$$\|f - h\|_2^2 \text{ is small,}$$

where $\|g\|_2^2 = \sum_{i=1}^{n} g(i)^2$ is the Sum-Squared Error or *V-optimal* error [IP95].

Motivating Example (cont.)

Problem: Too many people in the world, so we can't get all the data

Motivating Example (cont.)

Problem: Too many people in the world, so we can't get all the data

Solution: Uniformly at random query people for their yearly income, and output a good k-histogram approximation to f using these answers.

Motivating Example (cont.)

Problem: Too many people in the world, so we can't get all the data

Solution: Uniformly at random query people for their yearly income, and output a good k-histogram approximation to f using these answers.

i.e. Draw *independent samples* from f, and use them to recover a good k-histogram approximation to f

Formal Problem Statement

Given $k \in \mathbb{N}, \epsilon > 0$, and independent samples from an unknown distribution f supported on $[n] = \{1, \ldots, n\}$, recover a k-histogram h so that w.h.p.,

$$\|f - h\|_2^2 \leq \mathrm{OPT}_k(f) + \epsilon \,,$$

where for any function $q : [n] \to \mathbb{R}$,

$$\mathrm{OPT}_k(q) = \inf_{\text{h: } k\text{-histogram}} \|q - h\|_2^2 \,.$$

Formal Problem Statement

Given $k \in \mathbb{N}, \epsilon > 0$, and independent samples from an unknown distribution f supported on $[n] = \{1, \ldots, n\}$, recover a $\alpha \cdot k$-histogram h so that w.h.p.,

$$\|f - h\|_2^2 \leq \beta \cdot \mathrm{OPT}_k(f) + \epsilon \,,$$

where for any function $q : [n] \to \mathbb{R}$,

$$\mathrm{OPT}_k(q) = \inf_{\text{h: } k\text{-histogram}} \|q - h\|_2^2 \,.$$

Formal Problem Statement

Given $k \in \mathbb{N}, \epsilon > 0$, and independent samples from an unknown distribution f supported on $[n] = \{1, \ldots, n\}$, recover a $\alpha \cdot k$-histogram h so that w.h.p.,

$$\|f - h\|_2^2 \leq \beta \cdot \mathrm{OPT}_k(f) + \epsilon \,,$$

where for any function $q : [n] \to \mathbb{R}$,

$$\mathrm{OPT}_k(q) = \inf_{\text{h: } k\text{-histogram}} \|q - h\|_2^2 \,.$$

Sample Complexity: How many samples does our algorithm need?

Formal Problem Statement

Given $k \in \mathbb{N}, \epsilon > 0$, and independent samples from an unknown distribution f supported on $[n] = \{1, \ldots, n\}$, recover a $\alpha \cdot k$-histogram h so that w.h.p.,

$$\|f - h\|_2^2 \leq \beta \cdot \mathrm{OPT}_k(f) + \epsilon \,,$$

where for any function $q : [n] \to \mathbb{R}$,

$$\mathrm{OPT}_k(q) = \inf_{\text{h: } k\text{-histogram}} \|q - h\|_2^2 \,.$$

Sample Complexity: How many samples does our algorithm need?

Time Complexity: How fast does our algorithm run?

Formal Problem Statement

Given $k \in \mathbb{N}, \epsilon > 0$, and independent samples from an unknown distribution f supported on $[n] = \{1, \ldots, n\}$, recover a $\alpha \cdot k$-histogram h so that w.h.p.,

$$\|f - h\|_2^2 \leq \beta \cdot \mathrm{OPT}_k(f) + \epsilon \,,$$

where for any function $q : [n] \to \mathbb{R}$,

$$\mathrm{OPT}_k(q) = \inf_{\text{h: } k\text{-histogram}} \|q - h\|_2^2 \,.$$

Sample Complexity: How many samples does our algorithm need?

Time Complexity: How fast does our algorithm run?

Gold Standard: Algorithm which achieves $\alpha = \beta = 1$ which takes an *information-theoretically* optimal number of samples, and runs in time *linear* in the number of samples taken.

Previous Work

Previous Work

When given complete access to f:

Algorithm	Runtime	α	β
Basic DP [JKM+98]	$O(kn^2)$	1	1
Greedy Dual [JKM+98]	$O\left(n \log \mathrm{OPT}_k(f)\right)$	3	3
Smart DPs [TGIK02, GGI+02, GKS06]	$O\left(n + \frac{k^3 \log^2 n}{\delta^2}\right)$	$(1+\delta)$	1

Previous Work

When given complete access to f:

Algorithm	Runtime	α	β
Basic DP [JKM+98]	$O(kn^2)$	1	1
Greedy Dual [JKM+98]	$O\left(n \log \mathrm{OPT}_k(f)\right)$	3	3
Smart DPs [TGIK02, GGI+02, GKS06]	$O\left(n + \frac{k^3 \log^2 n}{\delta^2}\right)$	$(1+\delta)$	1
This Work	$O(n)$	5	2

Previous Work

When given complete access to f:

Algorithm	Runtime	α	β
Basic DP [JKM+98]	$O(kn^2)$	1	1
Greedy Dual [JKM+98]	$O\left(n \log \mathrm{OPT}_k(f)\right)$	3	3
Smart DPs [TGIK02, GGI+02, GKS06]	$O\left(n + \frac{k^3 \log^2 n}{\delta^2}\right)$	$(1+\delta)$	1
This Work	$O(n)$	5	2

When given sample access to f:
[ILR12]: $\tilde{O}\left(\frac{k^2}{\epsilon^2} \log n\right)$ samples, $\tilde{O}\left(\frac{k^5}{\epsilon^4} \log^2 n\right)$ time, $\alpha = O(\log \frac{1}{\epsilon})$, $\beta = 1$.

Previous Work

When given complete access to f:

Algorithm	Runtime	α	β
Basic DP [JKM+98]	$O(kn^2)$	1	1
Greedy Dual [JKM+98]	$O\left(n \log \mathrm{OPT}_k(f)\right)$	3	3
Smart DPs [TGIK02, GGI+02, GKS06]	$O\left(n + \frac{k^3 \log^2 n}{\delta^2}\right)$	$(1+\delta)$	1
This Work	$O(n)$	5	2

When given sample access to f:

[ILR12]: $\tilde{O}\left(\frac{k^2}{\epsilon^2} \log n\right)$ samples, $\tilde{O}\left(\frac{k^5}{\epsilon^4} \log^2 n\right)$ time, $\alpha = O(\log \frac{1}{\epsilon})$, $\beta = 1$.

This work: $O(\frac{1}{\epsilon})$ samples, $O(\frac{1}{\epsilon})$ time, $\alpha = 5$, $\beta = 2$.

Previous Work

When given complete access to f:

Algorithm	Runtime	α	β
Basic DP [JKM+98]	$O(kn^2)$	1	1
Greedy Dual [JKM+98]	$O\left(n \log \mathrm{OPT}_k(f)\right)$	3	3
Smart DPs [TGIK02, GGI+02, GKS06]	$O\left(n + \frac{k^3 \log^2 n}{\delta^2}\right)$	$(1+\delta)$	1
This Work	$O(n)$	5	2

When given sample access to f:
[ILR12]: $\tilde{O}\left(\frac{k^2}{\epsilon^2} \log n\right)$ samples, $\tilde{O}\left(\frac{k^5}{\epsilon^4} \log^2 n\right)$ time, $\alpha = O(\log \frac{1}{\epsilon})$, $\beta = 1$.
This work: $O(\frac{1}{\epsilon})$ samples, $O(\frac{1}{\epsilon})$ time, $\alpha = 5$, $\beta = 2$.

Our algorithm is sample optimal (up to constants) and runs in linear time.

Outline of Rest of Talk

Outline of Rest of Talk

An $O(1/\epsilon)$ Sample Upper Bound

Let $\hat{f}_m$ denote the empirical distribution after drawing m samples $X_1, \ldots, X_m$ from f.

An $O(1/\epsilon)$ Sample Upper Bound

Let $\hat{f}_m$ denote the empirical distribution after drawing m samples $X_1, \ldots, X_m$ from f.

$$\hat{f}_m(i) = \frac{\#\{j : X_j = i\}}{m} .$$

An $O(1/\epsilon)$ Sample Upper Bound

Let $\hat{f}_m$ denote the empirical distribution after drawing m samples $X_1, \ldots, X_m$ from f.

$$\hat{f}_m(i) = \frac{\#\{j : X_j = i\}}{m} .$$

Key lemma:

Lemma

If $m = O(\frac{1}{\epsilon})$, then $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ with probability $99/100$.

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, *then* $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ *with probability* $99/100$.

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, *then* $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ *with probability* $99/100$.

Proof.

We will show $\mathbb{E}[\|f - \hat{f}_m\|_2^2] \leq \epsilon$.

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, *then* $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ *with probability* $99/100$.

Proof.

We will show $\mathbb{E}[\|f - \hat{f}_m\|_2^2] \leq \epsilon$.

$$\mathbb{E}[\|f - \hat{f}_m\|_2^2] = \mathbb{E}\left[\sum_{i=1}^{n}(f(i) - \hat{f}_m(i))^2\right]$$

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, *then* $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ *with probability* $99/100$.

Proof.

We will show $\mathbb{E}[\|f - \hat{f}_m\|_2^2] \leq \epsilon$.

$$\begin{aligned}\mathbb{E}[\|f - \hat{f}_m\|_2^2] &= \mathbb{E}\left[\sum_{i=1}^{n}(f(i) - \hat{f}_m(i))^2\right] \\ &= \sum_{i=1}^{n}\mathbb{E}\left[(f(i) - \hat{f}_m(i))^2\right]\end{aligned}$$

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, *then* $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ *with probability* $99/100$.

Proof.

We will show $\mathbb{E}[\|f - \hat{f}_m\|_2^2] \leq \epsilon$.

$$\begin{aligned}
\mathbb{E}[\|f - \hat{f}_m\|_2^2] &= \mathbb{E}\left[\sum_{i=1}^{n}(f(i) - \hat{f}_m(i))^2\right] \\
&= \sum_{i=1}^{n}\mathbb{E}\left[(f(i) - \hat{f}_m(i))^2\right] \\
&= \sum_{i=1}^{n}\mathrm{Var}\left[\hat{f}_m(i)\right] \ .
\end{aligned}$$

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, *then* $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ *with probability* $99/100$.

Proof.

$$\mathbb{E}[\|f - \hat{f}_m\|_2^2] = \sum_{i=1}^{n} \mathrm{Var}\left[\hat{f}_m(i)\right] .$$

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, then $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ with probability 99/100.

Proof.

$$\mathbb{E}[\|f - \hat{f}_m\|_2^2] = \sum_{i=1}^{n} \mathrm{Var}\left[\hat{f}_m(i)\right] .$$

But $\hat{f}_m(i) \sim \frac{1}{m}\mathrm{Bin}(m, f(i))$, and $\mathrm{Var}(\mathrm{Bin}(n,p)) = np(1-p)$.

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, then $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ with probability 99/100.

Proof.

$$\mathbb{E}[\|f - \hat{f}_m\|_2^2] = \sum_{i=1}^{n} \mathrm{Var}\left[\hat{f}_m(i)\right] .$$

But $\hat{f}_m(i) \sim \frac{1}{m}\mathrm{Bin}(m, f(i))$, and $\mathrm{Var}(\mathrm{Bin}(n,p)) = np(1-p)$.

$$\mathbb{E}[\|f - \hat{f}_m\|_2^2] = \sum_{i=1}^{n} \frac{1}{m^2} mf(i)(1 - f(i))$$

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, then $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ with probability 99/100.

Proof.

$$\mathbb{E}[\|f - \hat{f}_m\|_2^2] = \sum_{i=1}^{n} \mathrm{Var}\left[\hat{f}_m(i)\right] .$$

But $\hat{f}_m(i) \sim \frac{1}{m}\mathrm{Bin}(m, f(i))$, and $\mathrm{Var}(\mathrm{Bin}(n,p)) = np(1-p)$.

$$\begin{aligned}\mathbb{E}[\|f - \hat{f}_m\|_2^2] &= \sum_{i=1}^{n} \frac{1}{m^2} mf(i)(1 - f(i)) \\ &\leq \frac{1}{m}\sum_{i=1}^{n} f(i)\end{aligned}$$

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, then $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ with probability 99/100.

Proof.

$$\mathbb{E}[\|f - \hat{f}_m\|_2^2] = \sum_{i=1}^{n} \mathrm{Var}\left[\hat{f}_m(i)\right] .$$

But $\hat{f}_m(i) \sim \frac{1}{m}\mathrm{Bin}(m, f(i))$, and $\mathrm{Var}(\mathrm{Bin}(n,p)) = np(1-p)$.

$$\begin{aligned}\mathbb{E}[\|f - \hat{f}_m\|_2^2] &= \sum_{i=1}^{n} \frac{1}{m^2} mf(i)(1 - f(i)) \\ &\leq \frac{1}{m}\sum_{i=1}^{n} f(i) = \frac{1}{m}\end{aligned}$$

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, then $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ with probability 99/100.

Proof.

$$\mathbb{E}[\|f - \hat{f}_m\|_2^2] = \sum_{i=1}^{n} \mathrm{Var}\left[\hat{f}_m(i)\right] .$$

But $\hat{f}_m(i) \sim \frac{1}{m}\mathrm{Bin}(m, f(i))$, and $\mathrm{Var}(\mathrm{Bin}(n,p)) = np(1-p)$.

$$\begin{aligned}\mathbb{E}[\|f - \hat{f}_m\|_2^2] &= \sum_{i=1}^{n} \frac{1}{m^2} mf(i)(1 - f(i)) \\ &\leq \frac{1}{m}\sum_{i=1}^{n} f(i) = \frac{1}{m} \leq \epsilon .\end{aligned}$$

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, *then* $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ *with probability* $99/100$.

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, then $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ with probability 99/100.

Corollary

Let $m = O(1/\epsilon)$, and let h be so that $\|h - \hat{f}_m\|_2^2 \leq \beta \cdot OPT_k(\hat{f}_m)$. Then w.h.p.

$$\|h - f\|_2^2 \leq \beta \cdot OPT_k(f) + \epsilon \ .$$

An $O(1/\epsilon)$ Sample Upper Bound (cont.)

Lemma

If $m = O(\frac{1}{\epsilon})$, then $\|f - \hat{f}_m\|_2^2 \leq \epsilon$ with probability 99/100.

Corollary

Let $m = O(1/\epsilon)$, and let h be so that $\|h - \hat{f}_m\|_2^2 \leq \beta \cdot OPT_k(\hat{f}_m)$. Then w.h.p.

$$\|h - f\|_2^2 \leq \beta \cdot OPT_k(f) + \epsilon \, .$$

This reduces to a completely deterministic problem!

Outline of Rest of Talk

Main Result

Main Result

Main Algorithmic Result

An algorithm which, given $k \in \mathbb{N}$ and $q : [n] \to \mathbb{R}$ supported on m elements, runs in time $O(m)$, and outputs a $5k$-histogram h so that

$$\|h - q\|_2^2 \leq 2 \cdot \mathrm{OPT}_k(q) \; .$$

Main Result

Main Algorithmic Result

An algorithm which, given $k \in \mathbb{N}$ and $q : [n] \to \mathbb{R}$ supported on m elements, runs in time $O(m)$, and outputs a $5k$-histogram h so that

$$\|h - q\|_2^2 \leq 2 \cdot \mathrm{OPT}_k(q) \; .$$

Corollary

An algorithm for learning histogram approximations that takes $O(1/\epsilon)$ samples and runs in time $O(1/\epsilon)$ which achieves $\alpha = 5$ and $\beta = 2$.

Main Result

Main Algorithmic Result

An algorithm which, given $k \in \mathbb{N}$ and $q : [n] \to \mathbb{R}$ supported on m elements, runs in time $O(m)$, and outputs a $5k$-histogram h so that

$$\|h - q\|_2^2 \leq 2 \cdot \mathrm{OPT}_k(q) \; .$$

Corollary

An algorithm for learning histogram approximations that takes $O(1/\epsilon)$ samples and runs in time $O(1/\epsilon)$ which achieves $\alpha = 5$ and $\beta = 2$.

Proof.

- Draw $m = O(1/\epsilon)$ samples and form the empirical $\hat{f}_m$.

Main Result

Main Algorithmic Result

An algorithm which, given $k \in \mathbb{N}$ and $q : [n] \to \mathbb{R}$ supported on m elements, runs in time $O(m)$, and outputs a $5k$-histogram h so that

$$\|h - q\|_2^2 \leq 2 \cdot \mathrm{OPT}_k(q) \; .$$

Corollary

An algorithm for learning histogram approximations that takes $O(1/\epsilon)$ samples and runs in time $O(1/\epsilon)$ which achieves $\alpha = 5$ and $\beta = 2$.

Proof.

- Draw $m = O(1/\epsilon)$ samples and form the empirical $\hat{f}_m$.
- Run the above algorithm on $\hat{f}_m$. □

Flattening

Flattening

Definition

Let $q : [n] \to \mathbb{R}$, and and let $I \subseteq [n]$ be an interval.

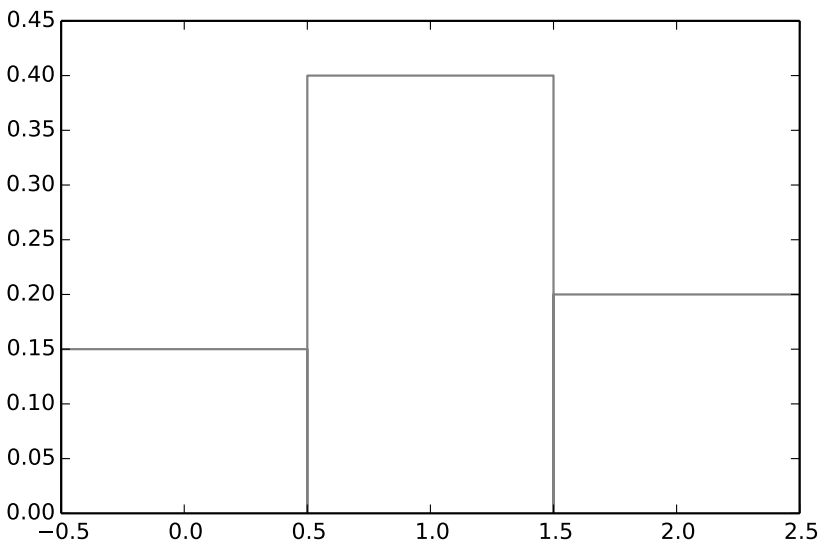
- Let $\overline{q}_I$ be the constant function on I which is identically $\frac{1}{|I|} \sum_{i \in I} q(I)$. We call this the *flattening* of q over I.
- Let $\text{flat-err}_q(I) = \sum_{i \in I} (q(i) - \overline{q}_I(i))^2$.

Flattening

Definition

Let $q : [n] \to \mathbb{R}$, and and let $I \subseteq [n]$ be an interval.

- Let $\overline{q}_I$ be the constant function on I which is identically $\frac{1}{|I|} \sum_{i \in I} q(I)$. We call this the *flattening* of q over I.
- Let $\text{flat-err}_q(I) = \sum_{i \in I} (q(i) - \overline{q}_I(i))^2$.

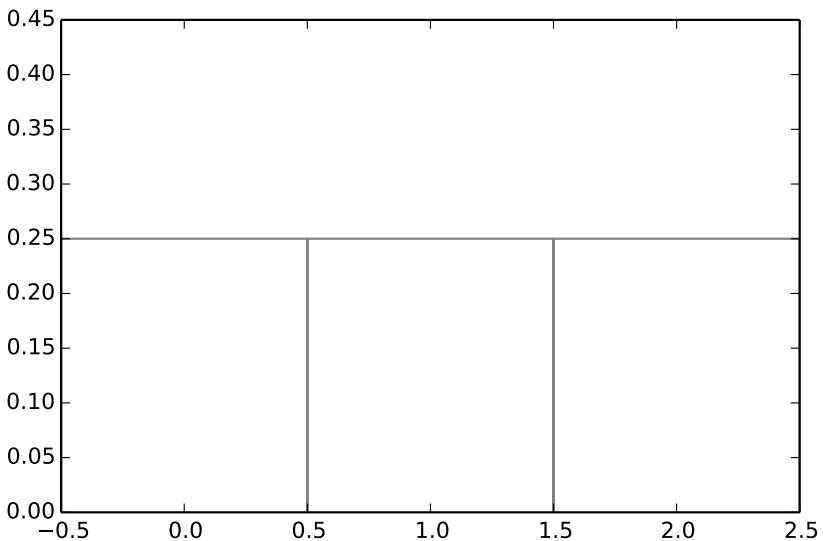


Flattening

Definition

Let $q : [n] \to \mathbb{R}$, and and let $I \subseteq [n]$ be an interval.

- Let $\overline{q}_I$ be the constant function on I which is identically $\frac{1}{|I|} \sum_{i \in I} q(I)$. We call this the *flattening* of q over I.
- Let $\text{flat-err}_q(I) = \sum_{i \in I} (q(i) - \overline{q}_I(i))^2$.



Flattening

Definition

Let $q : [n] \to \mathbb{R}$, and and let $I \subseteq [n]$ be an interval.

- Let $\overline{q}_I$ be the constant function on I which is identically $\frac{1}{|I|} \sum_{i \in I} q(I)$. We call this the *flattening* of q over I.
- Let $\text{flat-err}_q(I) = \sum_{i \in I} (q(i) - \overline{q}_I(i))^2$.

Lemma

For any flat function ϕ on I, $\mathit{flat\text{-}err}_q(I) \leq \sum_{i \in I} (q(i) - \phi(i))^2$.

Partitions

Partitions

Definition

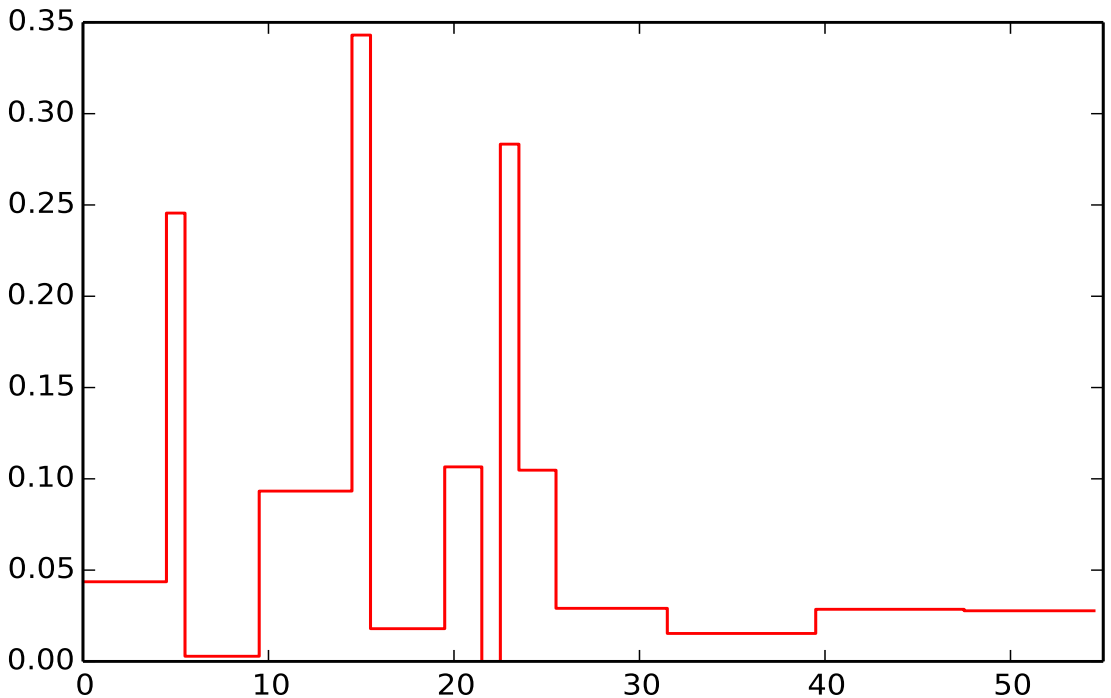
A *partition* of $[n]$ is a set of disjoint intervals $I_1, \ldots, I_r$ so that $\bigcup I_i = [n]$.

Partitions

Definition

A *partition* of $[n]$ is a set of disjoint intervals $I_1, \ldots, I_r$ so that $\bigcup I_i = [n]$.

- Any k-histogram induces a partition of $[n]$.

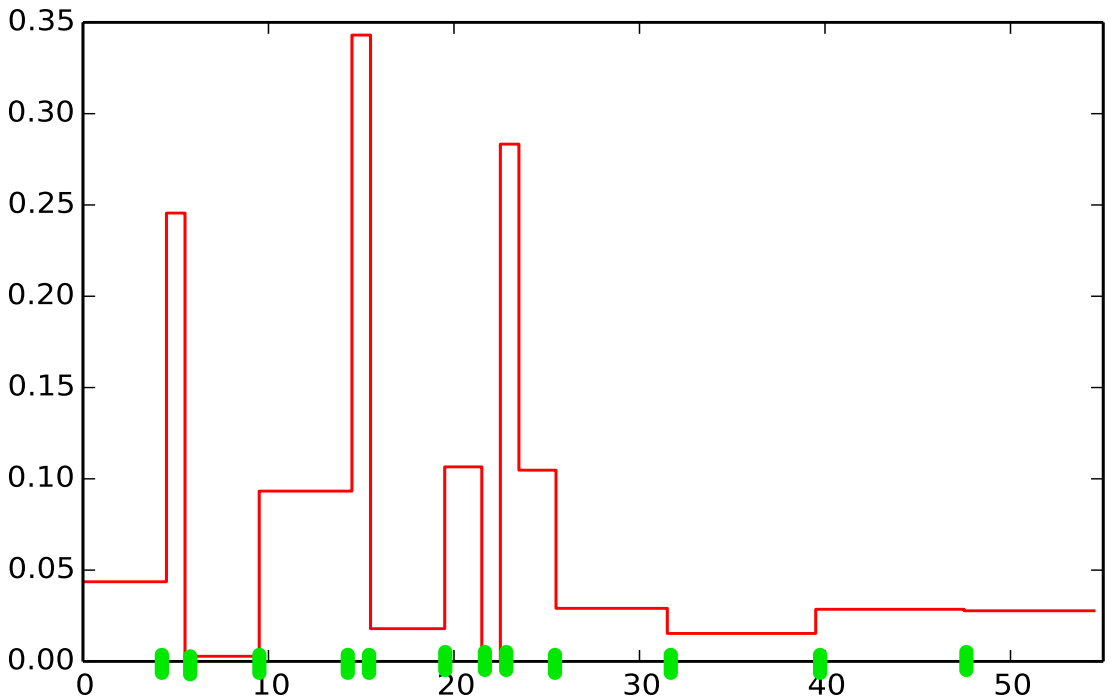


Partitions

Definition

A *partition* of $[n]$ is a set of disjoint intervals $I_1, \ldots, I_r$ so that $\bigcup I_i = [n]$.

- Any k-histogram induces a partition of $[n]$.

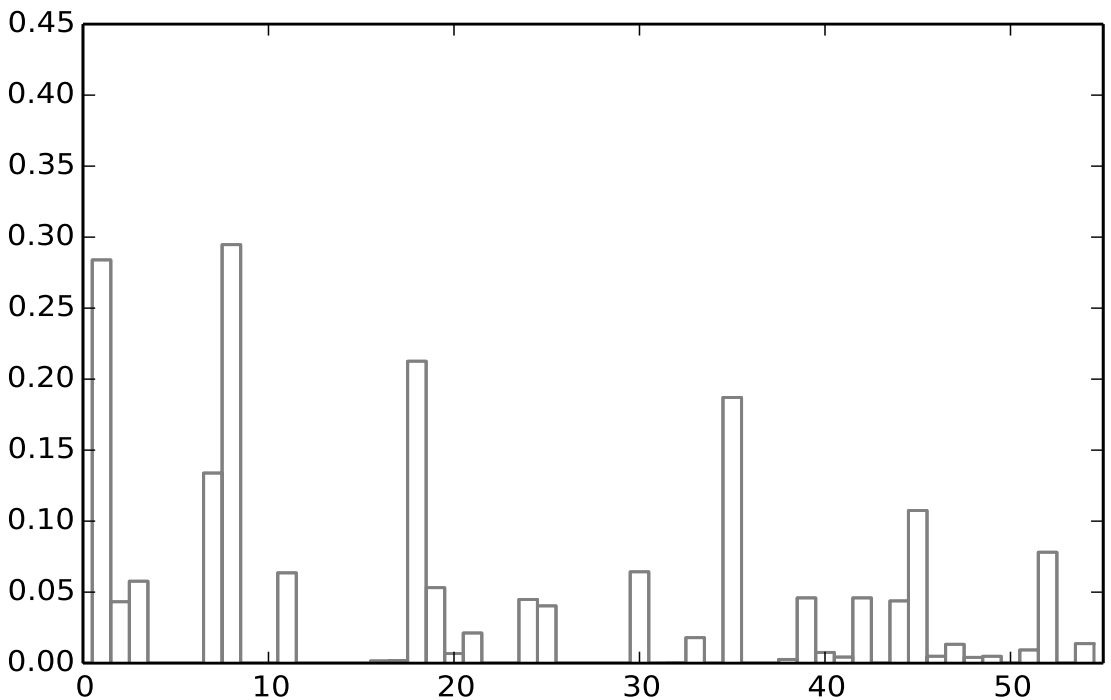


Partitions

Definition

A *partition* of $[n]$ is a set of disjoint intervals $I_1, \ldots, I_r$ so that $\bigcup I_i = [n]$.

- Any partition induces a unique k-histogram (for our purposes)

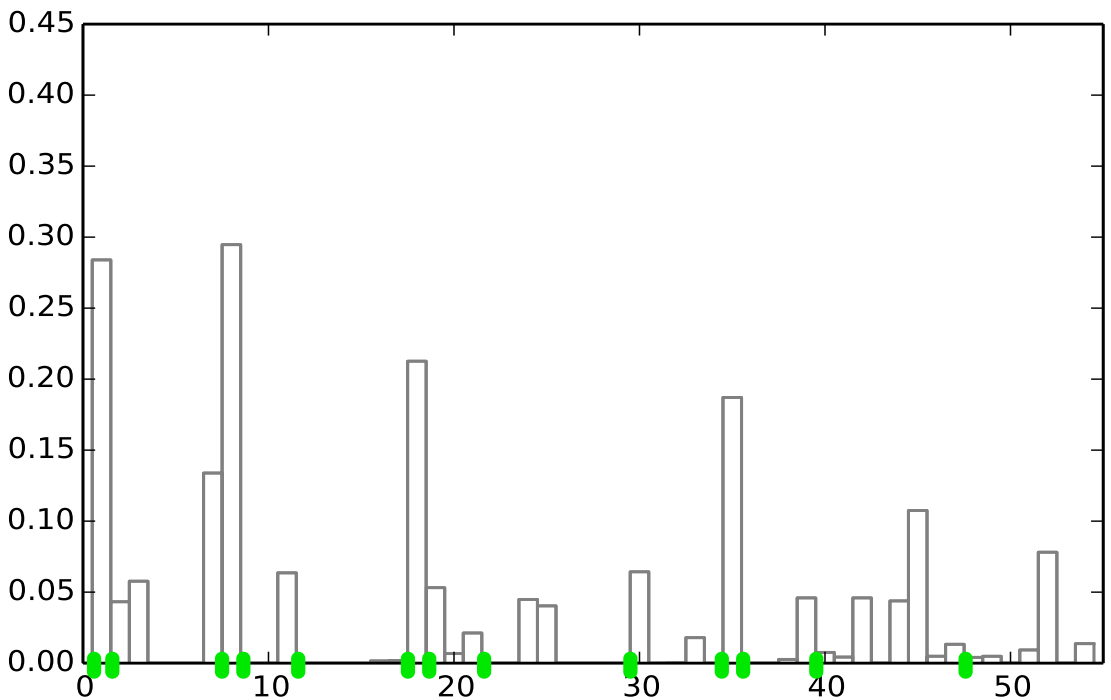


Partitions

Definition

A *partition* of $[n]$ is a set of disjoint intervals $I_1, \ldots, I_r$ so that $\bigcup I_i = [n]$.

- Any partition induces a unique k-histogram (for our purposes)

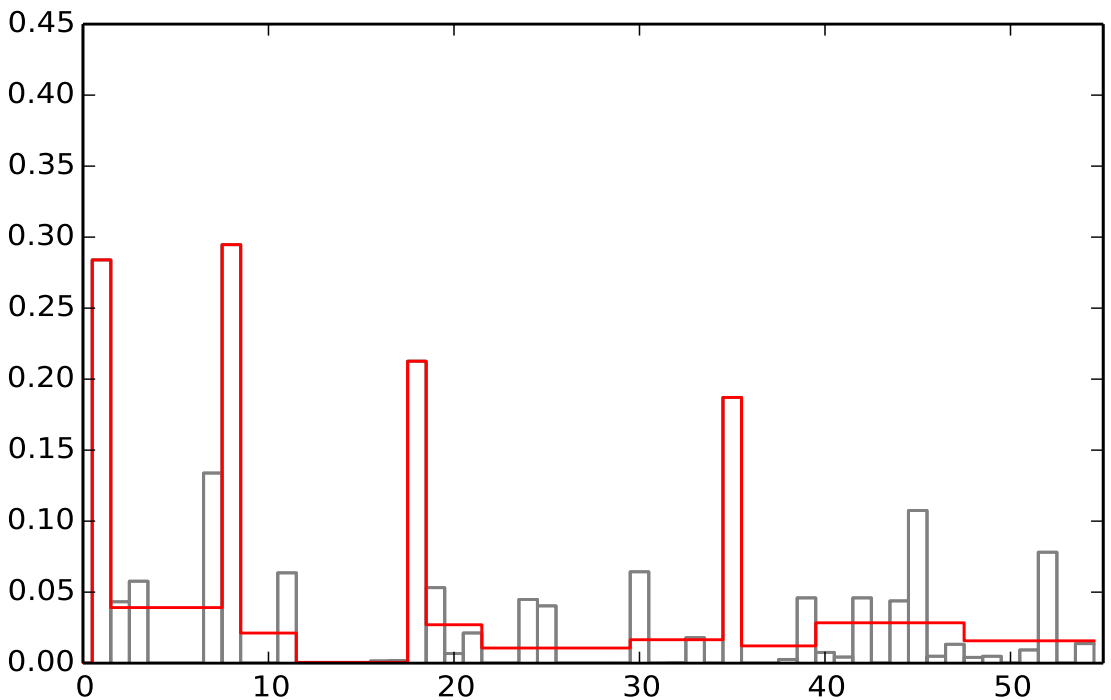


Partitions

Definition

A *partition* of $[n]$ is a set of disjoint intervals $I_1, \ldots, I_r$ so that $\bigcup I_i = [n]$.

- Any partition induces a unique k-histogram (for our purposes)



Algorithm Description

Algorithm Description

- q is m-sparse $\Rightarrow$ q is an $O(m)$-histogram. Let $\mathcal{I}$ be the partition of $[n]$ that the jumps of q induce.

Algorithm Description

- q is m-sparse $\Rightarrow$ q is an $O(m)$-histogram. Let $\mathcal{I}$ be the partition of $[n]$ that the jumps of q induce.
- While $|\mathcal{I}| \geq 5k$:

Algorithm Description

- q is m-sparse $\Rightarrow$ q is an $O(m)$-histogram. Let $\mathcal{I}$ be the partition of $[n]$ that the jumps of q induce.
- While $|\mathcal{I}| \geq 5k$:
  - Let $\mathcal{I} = \{I_1, \ldots, I_r\}$ where the I_j are in order.

Algorithm Description

- q is m-sparse $\Rightarrow$ q is an $O(m)$-histogram. Let $\mathcal{I}$ be the partition of $[n]$ that the jumps of q induce.
- While $|\mathcal{I}| \geq 5k$:
  - Let $\mathcal{I} = \{I_1, \ldots, I_r\}$ where the I_j are in order.
  - Form $J_1 = (I_1 \cup I_2), J_2 = (I_3 \cup I_4), \ldots, J_{r/2} = (I_{r-1} \cup I_r)$

Algorithm Description

- q is m-sparse $\Rightarrow$ q is an $O(m)$-histogram. Let $\mathcal{I}$ be the partition of $[n]$ that the jumps of q induce.
- While $|\mathcal{I}| \geq 5k$:
  - Let $\mathcal{I} = \{I_1, \ldots, I_r\}$ where the I_j are in order.
  - Form $J_1 = (I_1 \cup I_2), J_2 = (I_3 \cup I_4), \ldots, J_{r/2} = (I_{r-1} \cup I_r)$
  - For $\ell = 1, \ldots, r/2$, compute $e_\ell = \text{flat-err}_q(J_\ell)$. Let $L \subseteq \{1, \ldots, r/2\}$ be the set of $2k$ indices with largest e_ℓ.

Algorithm Description

- q is m-sparse $\Rightarrow$ q is an $O(m)$-histogram. Let $\mathcal{I}$ be the partition of $[n]$ that the jumps of q induce.
- While $|\mathcal{I}| \geq 5k$:
  - Let $\mathcal{I} = \{I_1, \ldots, I_r\}$ where the I_j are in order.
  - Form $J_1 = (I_1 \cup I_2), J_2 = (I_3 \cup I_4), \ldots, J_{r/2} = (I_{r-1} \cup I_r)$
  - For $\ell = 1, \ldots, r/2$, compute $e_\ell = \text{flat-err}_q(J_\ell)$. Let $L \subseteq \{1, \ldots, r/2\}$ be the set of $2k$ indices with largest e_ℓ.
  - Form $\mathcal{I}'$ by:

Algorithm Description

- q is m-sparse $\Rightarrow$ q is an $O(m)$-histogram. Let $\mathcal{I}$ be the partition of $[n]$ that the jumps of q induce.
- While $|\mathcal{I}| \geq 5k$:
  - Let $\mathcal{I} = \{I_1, \ldots, I_r\}$ where the I_j are in order.
  - Form $J_1 = (I_1 \cup I_2), J_2 = (I_3 \cup I_4), \ldots, J_{r/2} = (I_{r-1} \cup I_r)$
  - For $\ell = 1, \ldots, r/2$, compute $e_\ell = \text{flat-err}_q(J_\ell)$. Let $L \subseteq \{1, \ldots, r/2\}$ be the set of $2k$ indices with largest e_ℓ.
  - Form $\mathcal{I}'$ by:
    - For $\ell \in L$, include $I_{2\ell-1}$ and $I_{2\ell}$.

Algorithm Description

- q is m-sparse $\Rightarrow$ q is an $O(m)$-histogram. Let $\mathcal{I}$ be the partition of $[n]$ that the jumps of q induce.
- While $|\mathcal{I}| \geq 5k$:
  - Let $\mathcal{I} = \{I_1, \ldots, I_r\}$ where the I_j are in order.
  - Form $J_1 = (I_1 \cup I_2), J_2 = (I_3 \cup I_4), \ldots, J_{r/2} = (I_{r-1} \cup I_r)$
  - For $\ell = 1, \ldots, r/2$, compute $e_\ell = \text{flat-err}_q(J_\ell)$. Let $L \subseteq \{1, \ldots, r/2\}$ be the set of $2k$ indices with largest e_ℓ.
  - Form $\mathcal{I}'$ by:
    - For $\ell \in L$, include $I_{2\ell-1}$ and $I_{2\ell}$.
    - For $\ell \notin L$, include J_ℓ.

Algorithm Description

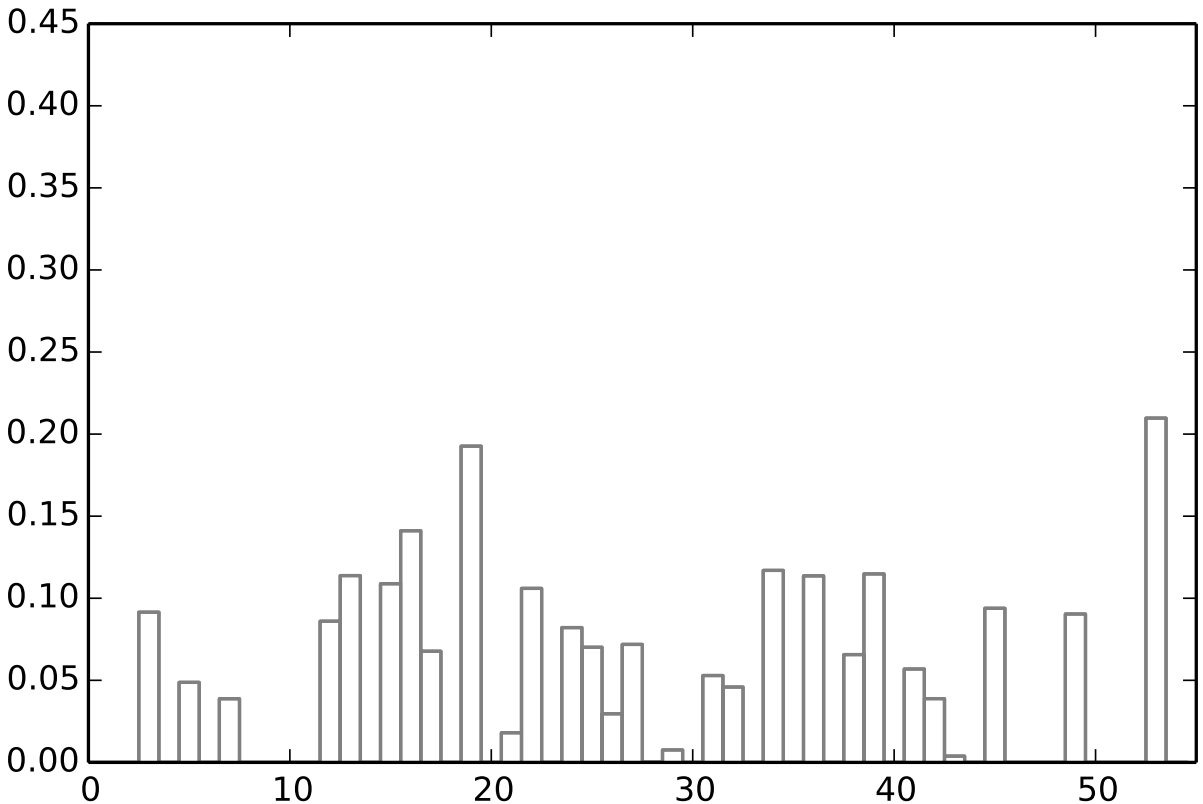
- q is m-sparse $\Rightarrow$ q is an $O(m)$-histogram. Let $\mathcal{I}$ be the partition of $[n]$ that the jumps of q induce.
- While $|\mathcal{I}| \geq 5k$:
  - Let $\mathcal{I} = \{I_1, \ldots, I_r\}$ where the I_j are in order.
  - Form $J_1 = (I_1 \cup I_2), J_2 = (I_3 \cup I_4), \ldots, J_{r/2} = (I_{r-1} \cup I_r)$
  - For $\ell = 1, \ldots, r/2$, compute $e_\ell = \text{flat-err}_q(J_\ell)$. Let $L \subseteq \{1, \ldots, r/2\}$ be the set of $2k$ indices with largest e_ℓ.
  - Form $\mathcal{I}'$ by:
    - For $\ell \in L$, include $I_{2\ell-1}$ and $I_{2\ell}$.
    - For $\ell \notin L$, include J_ℓ.
  - Set $\mathcal{I} \leftarrow \mathcal{I}'$

Algorithm Description

- q is m-sparse $\Rightarrow$ q is an $O(m)$-histogram. Let $\mathcal{I}$ be the partition of $[n]$ that the jumps of q induce.
- While $|\mathcal{I}| \geq 5k$:
  - Let $\mathcal{I} = \{I_1, \ldots, I_r\}$ where the I_j are in order.
  - Form $J_1 = (I_1 \cup I_2), J_2 = (I_3 \cup I_4), \ldots, J_{r/2} = (I_{r-1} \cup I_r)$
  - For $\ell = 1, \ldots, r/2$, compute $e_\ell = \text{flat-err}_q(J_\ell)$. Let $L \subseteq \{1, \ldots, r/2\}$ be the set of $2k$ indices with largest e_ℓ.
  - Form $\mathcal{I}'$ by:
    - For $\ell \in L$, include $I_{2\ell-1}$ and $I_{2\ell}$.
    - For $\ell \notin L$, include J_ℓ.
  - Set $\mathcal{I} \leftarrow \mathcal{I}'$
- Output the flattening of q over the intervals in $\mathcal{I}$.

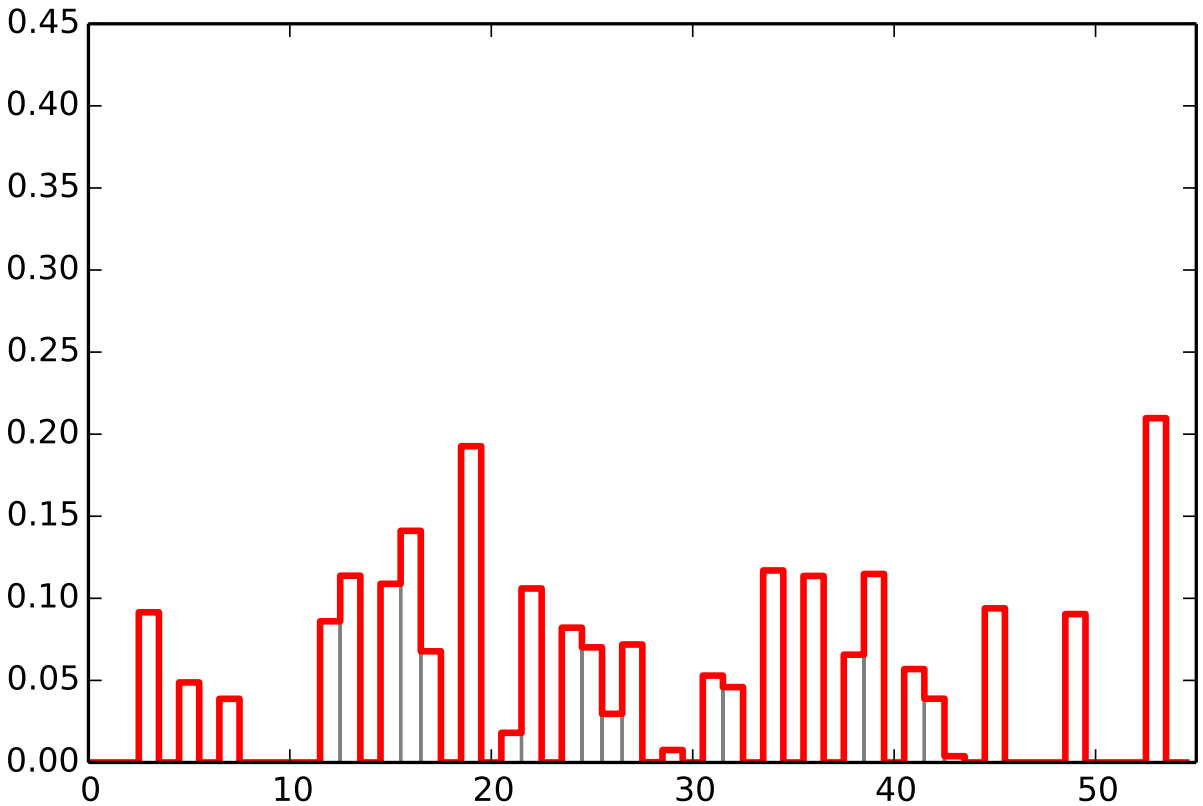
Example ($k = 2$)

Input:



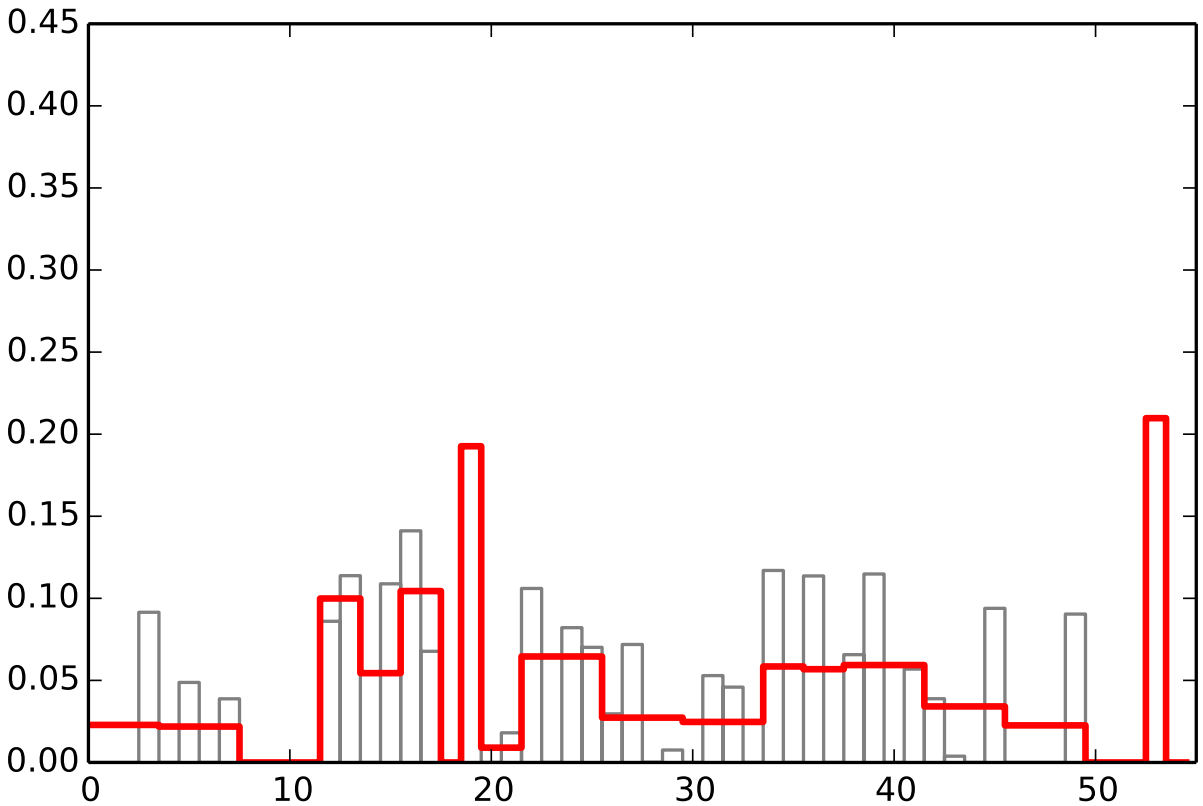
Example ($k = 2$)

Iteration 0:



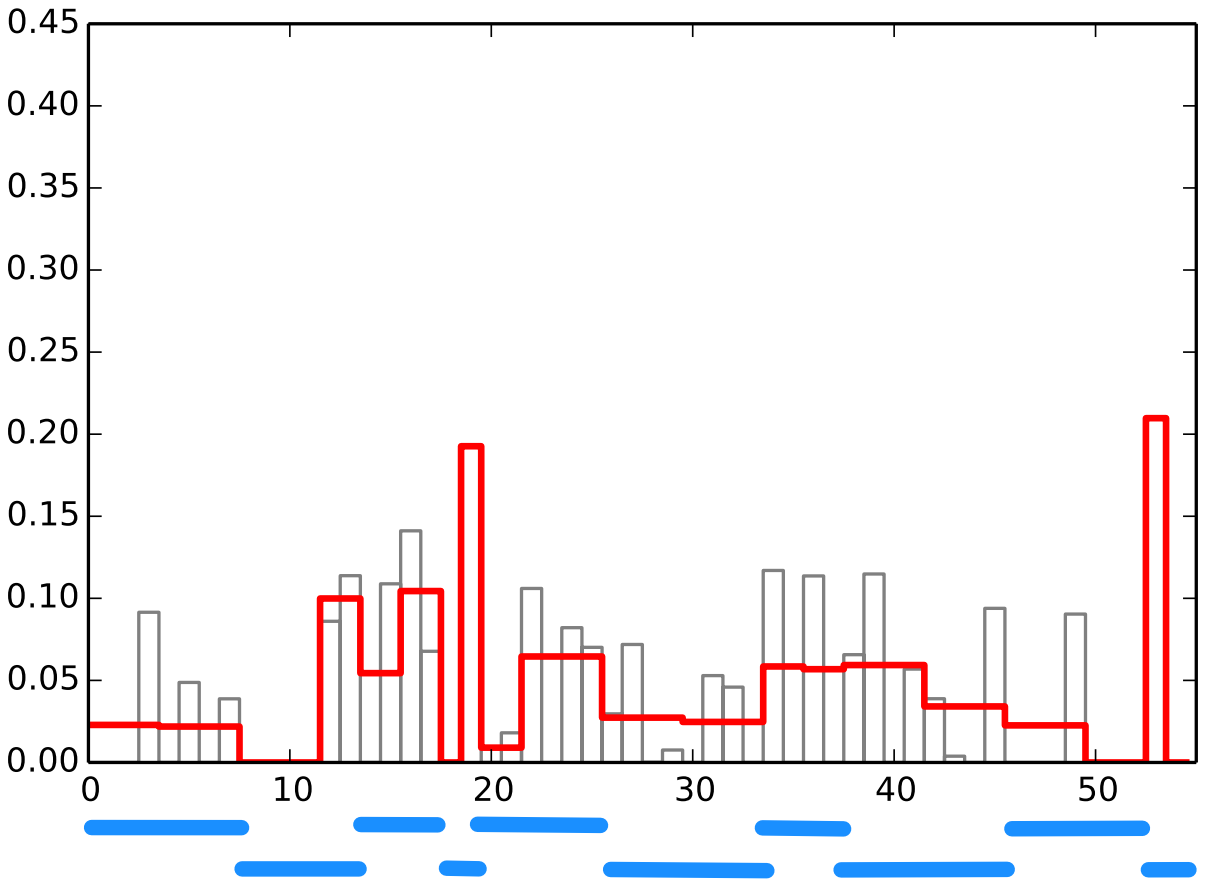
Example ($k = 2$)

Iteration i:



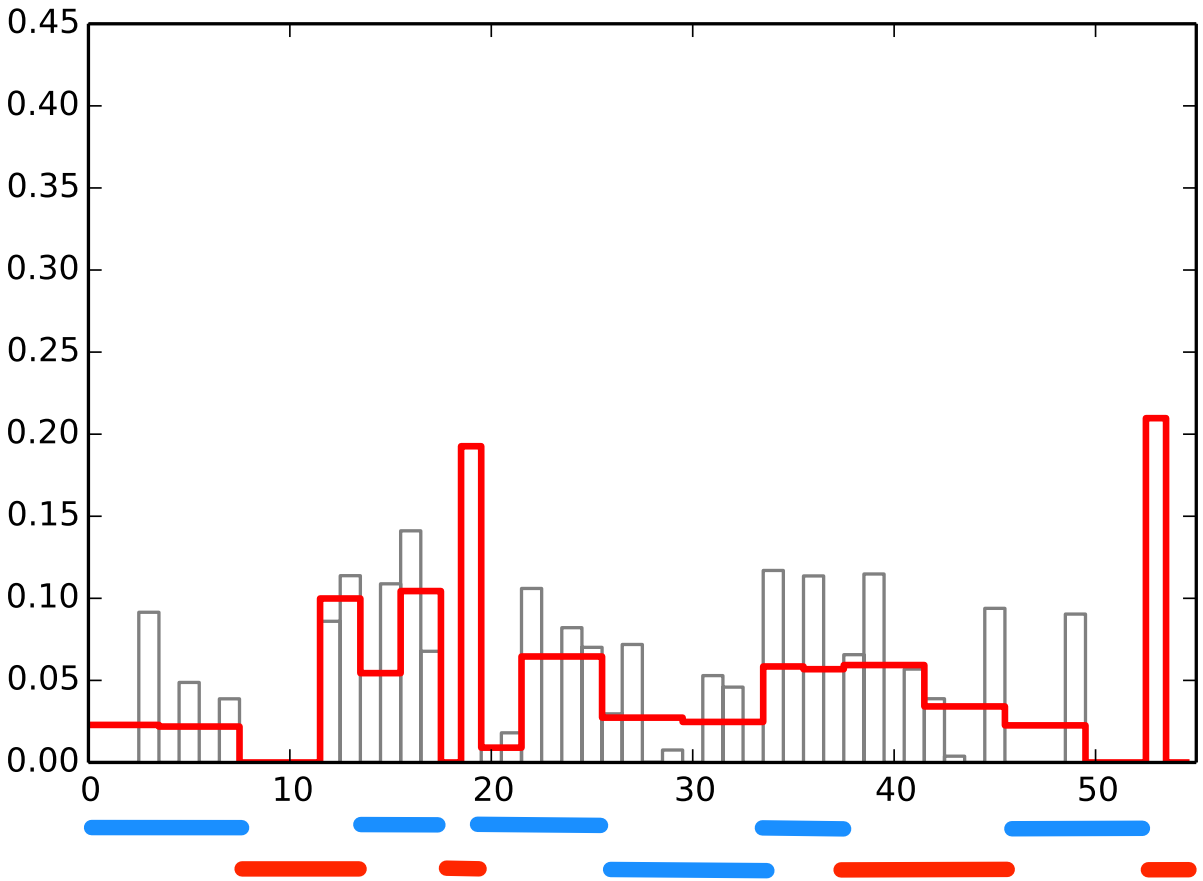
Example ($k = 2$)

Iteration i:



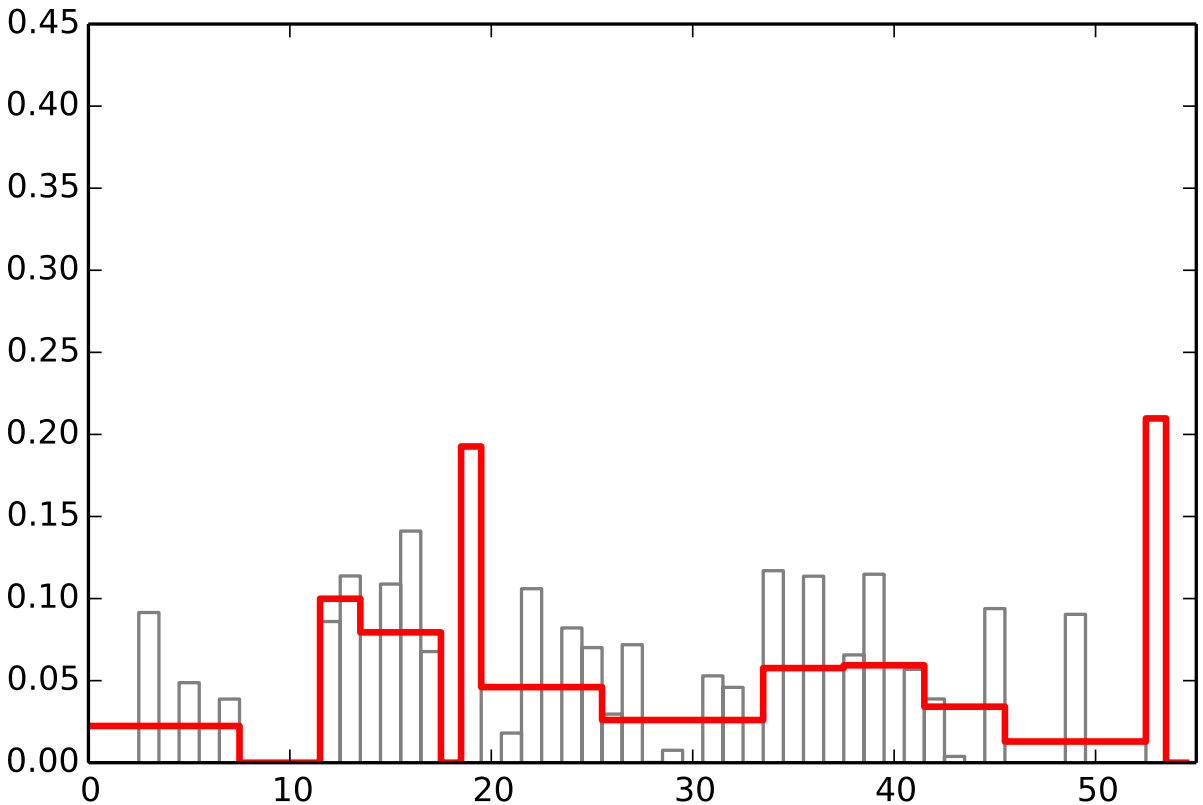
Example ($k = 2$)

Iteration i:



Example ($k = 2$)

Iteration $i + 1$:



Outline of Rest of Talk

Runtime Analysis

Runtime Analysis

Theorem

The greedy merging algorithm runs in time $O(m)$.

Runtime Analysis

Theorem

The greedy merging algorithm runs in time $O(m)$.

Proof Sketch.

Runtime Analysis

Theorem

The greedy merging algorithm runs in time $O(m)$.

Proof Sketch.

- Each iteration can be performed in time proportional to the number of intervals left in the partition in that iteration.

Runtime Analysis

Theorem

The greedy merging algorithm runs in time $O(m)$.

Proof Sketch.

- Each iteration can be performed in time proportional to the number of intervals left in the partition in that iteration.
- Let s_j be the number of intervals after the jth iteration of the algorithm.

Runtime Analysis

Theorem

The greedy merging algorithm runs in time $O(m)$.

Proof Sketch.

- Each iteration can be performed in time proportional to the number of intervals left in the partition in that iteration.
- Let s_j be the number of intervals after the jth iteration of the algorithm. Then

$$s_{j+1} = \frac{s_j - 4k}{2} + 4k = \frac{s_j + 4k}{2} \leq \frac{9}{10} s_j$$

as long as $s_j \geq 5k$.

Runtime Analysis

Theorem

The greedy merging algorithm runs in time $O(m)$.

Proof Sketch.

- Each iteration can be performed in time proportional to the number of intervals left in the partition in that iteration.
- Let s_j be the number of intervals after the jth iteration of the algorithm. Then

$$s_{j+1} = \frac{s_j - 4k}{2} + 4k = \frac{s_j + 4k}{2} \leq \frac{9}{10} s_j$$

  as long as $s_j \geq 5k$.
- Thus the runtime is dominated by the runtime of the first iteration, which runs in time $O(m)$. □

Error Analysis

Error Analysis

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

Error Analysis

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

- Let h^* be an optimal k-histogram; i.e. $\|h^* - q\|_2^2 = \mathrm{OPT}_k(q)$.

Error Analysis

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

- Let h^* be an optimal k-histogram; i.e. $\|h^* - q\|_2^2 = \mathrm{OPT}_k(q)$.
- Let $\mathcal{I} = \{I_1, \ldots, I_{5k}\}$ be the set of intervals we produce.

Partition $\mathcal{I}$:

Error Analysis

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

- Let h^* be an optimal k-histogram; i.e. $\|h^* - q\|_2^2 = \mathrm{OPT}_k(q)$.
- Let $\mathcal{I} = \{I_1, \ldots, I_{5k}\}$ be the set of intervals we produce.

Partition $\mathcal{I}$:

- Let $\mathcal{F}$ be the set of intervals in $\mathcal{I}$ on which h^* has no jumps

Error Analysis

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

- Let h^* be an optimal k-histogram; i.e. $\|h^* - q\|_2^2 = \mathrm{OPT}_k(q)$.
- Let $\mathcal{I} = \{I_1, \ldots, I_{5k}\}$ be the set of intervals we produce.

Partition $\mathcal{I}$:

- Let $\mathcal{F}$ be the set of intervals in $\mathcal{I}$ on which h^* has no jumps
- Let $\mathcal{J}$ be the set of intervals in $\mathcal{I}$ on which h^* has jumps

Error Analysis

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

- Let h^* be an optimal k-histogram; i.e. $\|h^* - q\|_2^2 = \mathrm{OPT}_k(q)$.
- Let $\mathcal{I} = \{I_1, \ldots, I_{5k}\}$ be the set of intervals we produce.

Partition $\mathcal{I}$:

- Let $\mathcal{F}$ be the set of intervals in $\mathcal{I}$ on which h^* has no jumps
- Let $\mathcal{J}$ be the set of intervals in $\mathcal{I}$ on which h^* has jumps

$$\|h - q\|_2^2 = \sum_{I \in \mathcal{F}} \text{flat-err}_q(I) + \sum_{I \in \mathcal{J}} \text{flat-err}_q(I) \ .$$

Error Analysis

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

- Let h^* be an optimal k-histogram; i.e. $\|h^* - q\|_2^2 = \mathrm{OPT}_k(q)$.
- Let $\mathcal{I} = \{I_1, \ldots, I_{5k}\}$ be the set of intervals we produce.

Partition $\mathcal{I}$:

- Let $\mathcal{F}$ be the set of intervals in $\mathcal{I}$ on which h^* has no jumps
- Let $\mathcal{J}$ be the set of intervals in $\mathcal{I}$ on which h^* has jumps

$$\|h - q\|_2^2 = \sum_{I \in \mathcal{F}} \text{flat-err}_q(I) + \sum_{I \in \mathcal{J}} \text{flat-err}_q(I) \ .$$

We will bound each term separately.

Error Analysis (cont.)

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Error Analysis (cont.)

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

Error on $\mathcal{F}$:

Error Analysis (cont.)

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

Error on $\mathcal{F}$:

- Fix $I \in \mathcal{F}$.

Error Analysis (cont.)

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

Error on $\mathcal{F}$:

- Fix $I \in \mathcal{F}$.
- Since h^* is flat on I, $\text{flat-err}_q(I) \leq \sum_{i \in I}(q(i) - h^*(I))^2$.

Error Analysis (cont.)

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

Error on $\mathcal{F}$:

- Fix $I \in \mathcal{F}$.
- Since h^* is flat on I, $\text{flat-err}_q(I) \leq \sum_{i \in I}(q(i) - h^*(I))^2$.
- Thus the squared error we have on $\mathcal{F}$ is at most the squared error of h^* on $\mathcal{F}$, i.e.

Error Analysis (cont.)

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

Error on $\mathcal{F}$:

- Fix $I \in \mathcal{F}$.
- Since h^* is flat on I, $\text{flat-err}_q(I) \leq \sum_{i \in I}(q(i) - h^*(I))^2$.
- Thus the squared error we have on $\mathcal{F}$ is at most the squared error of h^* on $\mathcal{F}$, i.e.

$$\sum_{I \in \mathcal{F}} \text{flat-err}_q(I) \leq$$

Error Analysis (cont.)

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

Error on $\mathcal{F}$:

- Fix $I \in \mathcal{F}$.
- Since h^* is flat on I, $\text{flat-err}_q(I) \leq \sum_{i \in I}(q(i) - h^*(I))^2$.
- Thus the squared error we have on $\mathcal{F}$ is at most the squared error of h^* on $\mathcal{F}$, i.e.

$$\sum_{I \in \mathcal{F}} \text{flat-err}_q(I) \leq \sum_{I \in \mathcal{F}} \sum_{i \in I} (q(i) - h^*(I))^2$$

Error Analysis (cont.)

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

Error on $\mathcal{F}$:

- Fix $I \in \mathcal{F}$.
- Since h^* is flat on I, $\text{flat-err}_q(I) \leq \sum_{i \in I}(q(i) - h^*(I))^2$.
- Thus the squared error we have on $\mathcal{F}$ is at most the squared error of h^* on $\mathcal{F}$, i.e.

$$\sum_{I \in \mathcal{F}} \text{flat-err}_q(I) \leq \sum_{I \in \mathcal{F}} \sum_{i \in I} (q(i) - h^*(I))^2 \leq \text{OPT}_k(q) \; .$$

Error Analysis (cont.)

Theorem

Let h be the output of our algorithm. Then $\|h - q\|_2^2 \leq 2 \cdot OPT_k(q)$.

Proof.

Error on $\mathcal{F}$:

- Fix $I \in \mathcal{F}$.
- Since h^* is flat on I, $\text{flat-err}_q(I) \leq \sum_{i \in I}(q(i) - h^*(I))^2$.
- Thus the squared error we have on $\mathcal{F}$ is at most the squared error of h^* on $\mathcal{F}$, i.e.

$$\sum_{I \in \mathcal{F}} \text{flat-err}_q(I) \leq \sum_{I \in \mathcal{F}} \sum_{i \in I} (q(i) - h^*(I))^2 \leq \text{OPT}_k(q) \ .$$

Notice this is true for any set of intervals on which h^* is flat.

Error Analysis (cont.)

Error Analysis (cont.)

Proof.

Error on $\mathcal{J}$:

Error Analysis (cont.)

Proof.

Error on $\mathcal{J}$:

- Fix $I \in \mathcal{J}$. WLOG assume it was merged in some iteration.

Error Analysis (cont.)

Proof.

Error on $\mathcal{J}$:

- Fix $I \in \mathcal{J}$. WLOG assume it was merged in some iteration.
- Let $J_1, \ldots, J_{2k}$ be the $2k$ candidate intervals which were not merged in that iteration.

Error Analysis (cont.)

Proof.

Error on $\mathcal{J}$:

- Fix $I \in \mathcal{J}$. WLOG assume it was merged in some iteration.
- Let $J_1, \ldots, J_{2k}$ be the $2k$ candidate intervals which were not merged in that iteration. For $\ell = 1, \ldots, 2k$, we know

$$\text{flat-err}_{\boldsymbol{q}}(I) \leq \text{flat-err}_{\boldsymbol{q}}(J_\ell) \ .$$

Error Analysis (cont.)

Proof.

Error on $\mathcal{J}$:

- Fix $I \in \mathcal{J}$. WLOG assume it was merged in some iteration.
- Let $J_1, \ldots, J_{2k}$ be the $2k$ candidate intervals which were not merged in that iteration. For $\ell = 1, \ldots, 2k$, we know

$$\text{flat-err}_q(I) \leq \text{flat-err}_q(J_\ell) \ .$$

- h^* has at most k jumps $\Rightarrow$ it has no jumps in at least k of the $J_1, \ldots, J_{2k}$. WLOG assume $J_1, \ldots, J_k$ have no jumps of h^*.

Error Analysis (cont.)

Proof.

Error on $\mathcal{J}$:

- Fix $I \in \mathcal{J}$. WLOG assume it was merged in some iteration.
- Let $J_1, \ldots, J_{2k}$ be the $2k$ candidate intervals which were not merged in that iteration. For $\ell = 1, \ldots, 2k$, we know

$$\text{flat-err}_q(I) \leq \text{flat-err}_q(J_\ell) \ .$$

- h^* has at most k jumps $\Rightarrow$ it has no jumps in at least k of the $J_1, \ldots, J_{2k}$. WLOG assume $J_1, \ldots, J_k$ have no jumps of h^*.
- Hence $\sum_{\ell=1}^{k} \text{flat-err}_q(J_\ell) \leq \text{OPT}_k(q)$

Error Analysis (cont.)

Proof.

Error on $\mathcal{J}$:

- Fix $I \in \mathcal{J}$. WLOG assume it was merged in some iteration.
- Let $J_1, \ldots, J_{2k}$ be the $2k$ candidate intervals which were not merged in that iteration. For $\ell = 1, \ldots, 2k$, we know

$$\text{flat-err}_q(I) \leq \text{flat-err}_q(J_\ell) \ .$$

- h^* has at most k jumps $\Rightarrow$ it has no jumps in at least k of the $J_1, \ldots, J_{2k}$. WLOG assume $J_1, \ldots, J_k$ have no jumps of h^*.
- Hence $\sum_{\ell=1}^{k} \text{flat-err}_q(J_\ell) \leq \text{OPT}_k(q) \Rightarrow \text{flat-err}_q(I) \leq \frac{1}{k}\text{OPT}_k(q)$.

Error Analysis (cont.)

Proof.

Error on $\mathcal{J}$:

- Fix $I \in \mathcal{J}$. WLOG assume it was merged in some iteration.
- Let $J_1, \ldots, J_{2k}$ be the $2k$ candidate intervals which were not merged in that iteration. For $\ell = 1, \ldots, 2k$, we know

$$\text{flat-err}_q(I) \leq \text{flat-err}_q(J_\ell) \ .$$

- h^* has at most k jumps $\Rightarrow$ it has no jumps in at least k of the $J_1, \ldots, J_{2k}$. WLOG assume $J_1, \ldots, J_k$ have no jumps of h^*.
- Hence $\sum_{\ell=1}^{k} \text{flat-err}_q(J_\ell) \leq \text{OPT}_k(q) \Rightarrow \text{flat-err}_q(I) \leq \frac{1}{k}\text{OPT}_k(q)$.
- So $\sum_{I \in \mathcal{J}} \text{flat-err}_q(I) \leq |\mathcal{J}| \frac{1}{k}\text{OPT}_k(q)$

Error Analysis (cont.)

Proof.

Error on $\mathcal{J}$:

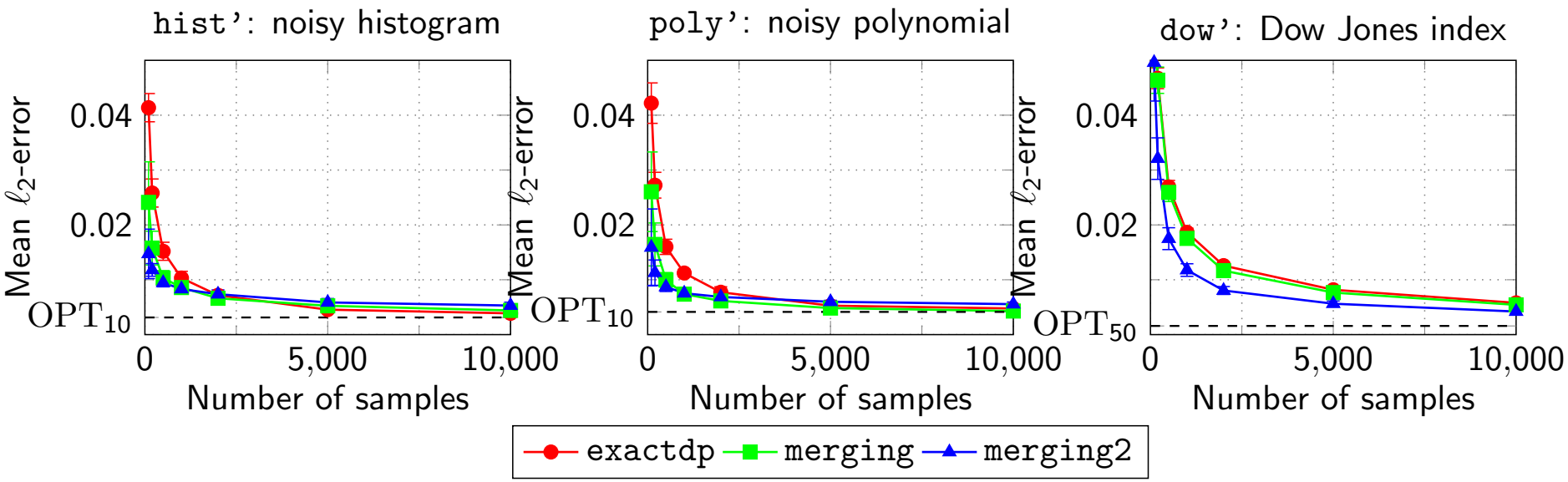
- Fix $I \in \mathcal{J}$. WLOG assume it was merged in some iteration.
- Let $J_1, \ldots, J_{2k}$ be the $2k$ candidate intervals which were not merged in that iteration. For $\ell = 1, \ldots, 2k$, we know

$$\text{flat-err}_q(I) \leq \text{flat-err}_q(J_\ell) \ .$$

- h^* has at most k jumps $\Rightarrow$ it has no jumps in at least k of the $J_1, \ldots, J_{2k}$. WLOG assume $J_1, \ldots, J_k$ have no jumps of h^*.
- Hence $\sum_{\ell=1}^{k} \text{flat-err}_q(J_\ell) \leq \text{OPT}_k(q) \Rightarrow \text{flat-err}_q(I) \leq \frac{1}{k}\text{OPT}_k(q)$.
- So $\sum_{I \in \mathcal{J}} \text{flat-err}_q(I) \leq |\mathcal{J}| \frac{1}{k}\text{OPT}_k(q) \leq \text{OPT}_k(q)$. □

Outline of Rest of Talk

Error Rate



`exactdp`: Exact DP algorithm for k-histograms.
`merging`: Merging with parameters set to produce a $2k + 1$ histogram.
`merging2`: Merging with parameters set to produce a $k + 1$ histogram.

Experimental Evaluation

		exactdp	merging	merging2	fastmerging	fastmerging2	dual
hist	Error (ℓ_2)	16.1	16.4	16.6	17.0	21.5	25.8
	Error (relative)	**1.00**	**1.02**	**1.03**	**1.06**	**1.34**	**1.60**
	Time (milliseconds)	55.391	0.038	0.037	0.020	0.014	0.108
	Time (relative)	**3,910**	**2.7**	**2.6**	**1.4**	**1.0**	**7.6**
poly	Error (ℓ_2)	105.1	85.9	111.6	85.6	111.7	124.0
	Error (relative)	**1.00**	**0.82**	**1.06**	**0.81**	**1.06**	**1.18**
	Time (milliseconds)	858.064	0.112	0.112	0.048	0.041	0.446
	Time (relative)	**20,924**	**2.7**	**2.7**	**1.2**	**1.0**	**10.9**
dow	Error (ℓ_2)	904.0	733.1	1,046.1	727.5	1,079.1	1,838.1
	Error (relative)	**1.00**	**0.81**	**1.16**	**0.80**	**1.19**	**2.03**
	Time (milliseconds)	73576.921	0.510	0.478	0.205	0.173	1.849
	Time (relative)	**425,540**	**3.0**	**2.8**	**1.2**	**1.0**	**10.7**

Outline of Rest of Talk

Extensions of the Algorithm

Extensions of the Algorithm

- What if you don't know what k to pick?

Extensions of the Algorithm

- What if you don't know what k to pick?

We give a hierarchical version of our algorithm which produces good histogram approximations for all values of $k \geq 1$.

Extensions of the Algorithm

- What if you don't know what k to pick?

We give a hierarchical version of our algorithm which produces good histogram approximations for all values of $k \geq 1$.

- What if you want more powerful representations?

Extensions of the Algorithm

- What if you don't know what k to pick?

We give a hierarchical version of our algorithm which produces good histogram approximations for all values of $k \geq 1$.

- What if you want more powerful representations?

We give a natural generalization of our algorithm which produces good piecewise polynomial approximations.

Extensions of the Algorithm

- What if you don't know what k to pick?

We give a hierarchical version of our algorithm which produces good histogram approximations for all values of $k \geq 1$.

- What if you want more powerful representations?

We give a natural generalization of our algorithm which produces good piecewise polynomial approximations.

- What about different norms?

Extensions of the Algorithm

- What if you don't know what k to pick?

We give a hierarchical version of our algorithm which produces good histogram approximations for all values of $k \geq 1$.

- What if you want more powerful representations?

We give a natural generalization of our algorithm which produces good piecewise polynomial approximations.

- What about different norms?

[ADLS15] develops a similar (but more complicated) algorithm for approximation in ℓ_1 (or total variation distance).

Conclusions

- We give the first sample-optimal, linear time algorithm for learning histogram approximations in ℓ_2^2 error.
- Reduce to giving a linear time algorithm for histogram approximation of sparse distributions.
- Do so via a simple, novel greedy merging algorithm.

Open problems:

- Streaming variants? Parallel variants?
- Is our error analysis tight?
- Can we get better α, β?

Thank you!