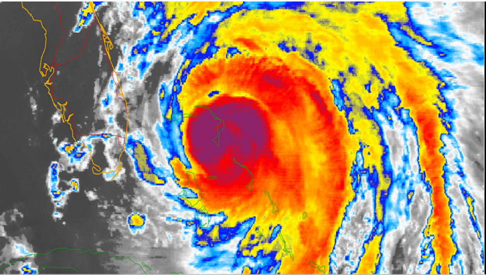
Visualization-Aware Sampling for Very Large Databases

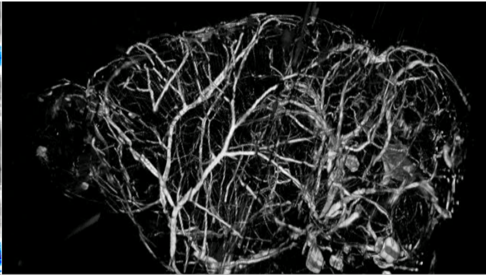
Yongjoo Park
Michael Cafarella
Barzan Mozafari

University of Michigan, Ann Arbor

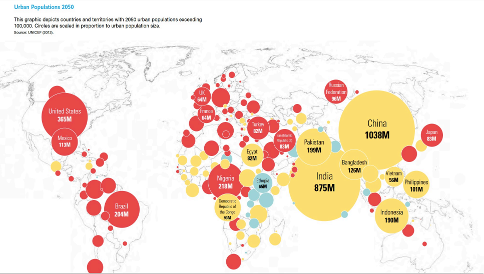
Prevalence of Viz-centric Data Analysis



Weather



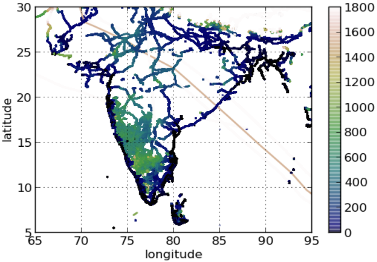
Brain



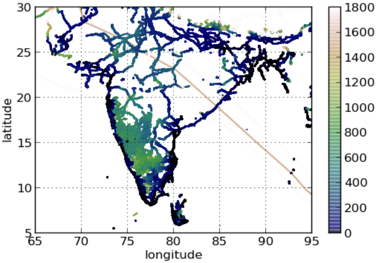
Population

Genome

Big Data, Poor Latency

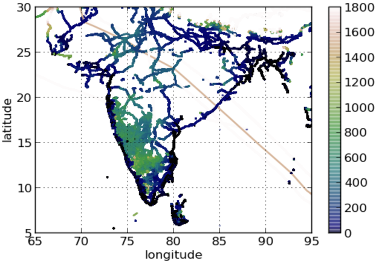


Big Data, Poor Latency



A part of 2 billion points

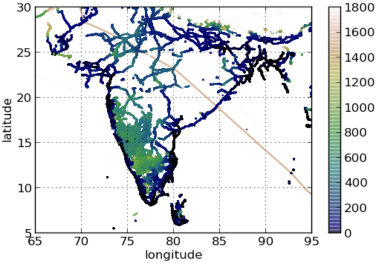
Big Data, Poor Latency



A part of 2 billion points

71 mins !! (`matplotlib`)

Big Data, Poor Latency

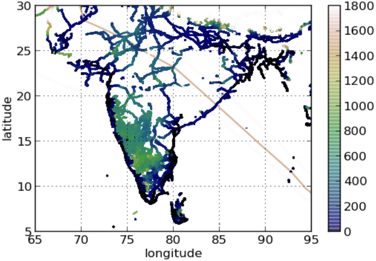


A part of 2 billion points

71 mins !! (`matplotlib`)

`MathGL` took 2+ hours for 100M points.

Big Data, Poor Latency



A part of 2 billion points

71 mins !! (`matplotlib`)

`MathGL` took 2+ hours for 100M points.

`Tableau` crashed for 100 million points
on a machine with `122GB` memory (`r3.4xlarge`).

Fast Response is Important

Fast Response is Important

Five times more interactions as the response time was reduced [3].

> *The most skilled user went from 800 interactions per hour with a 1.5-second response time up to 4,300 interactions per hour with a 0.4-second response time.*

Fast Response is Important

Five times more interactions as the response time was reduced [3].

> *The most skilled user went from 800 interactions per hour with a 1.5-second response time up to 4,300 interactions per hour with a 0.4-second response time.*

+500ms latency $\rightarrow$ 50% less data exploration subconsciously [1]

> *6 out of 16 subjects did not report a noticeable difference in terms of system responsiveness.*

Fast Response is Important

Five times more interactions as the response time was reduced [3].

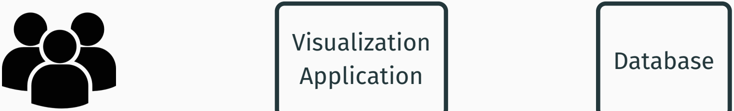
> *The most skilled user went from 800 interactions per hour with a 1.5-second response time up to 4,300 interactions per hour with a 0.4-second response time.*

+500ms latency → 50% less data exploration subconsciously [1]

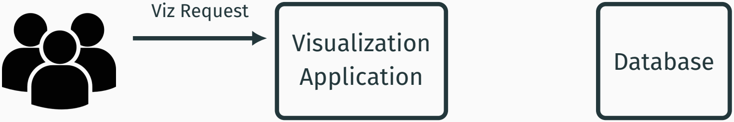
> *6 out of 16 subjects did not report a noticeable difference in terms of system responsiveness.*

A good reason for my coffee break.

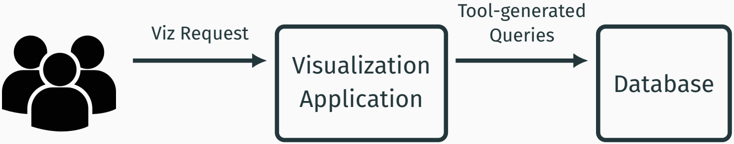
Why is Visualizing Big Data Slow?



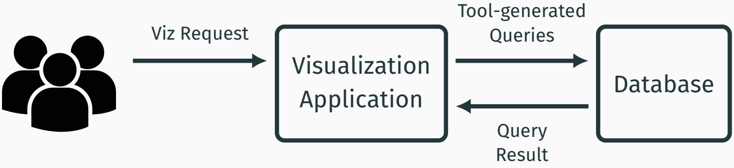
Why is Visualizing Big Data Slow?



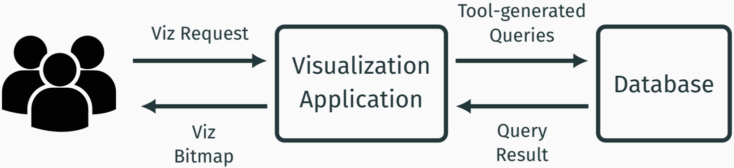
Why is Visualizing Big Data Slow?



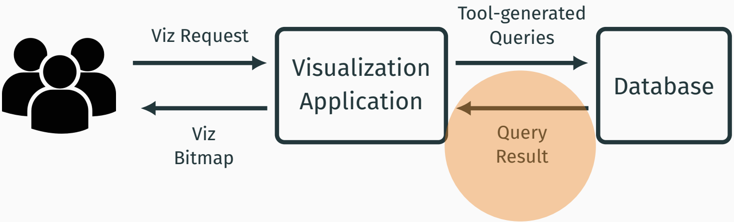
Why is Visualizing Big Data Slow?



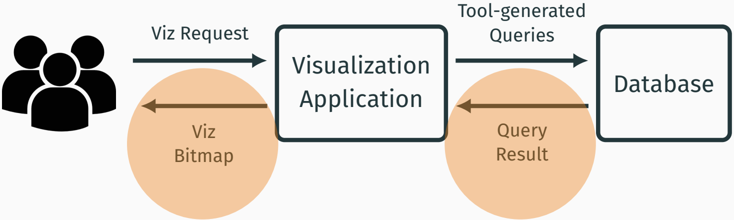
Why is Visualizing Big Data Slow?



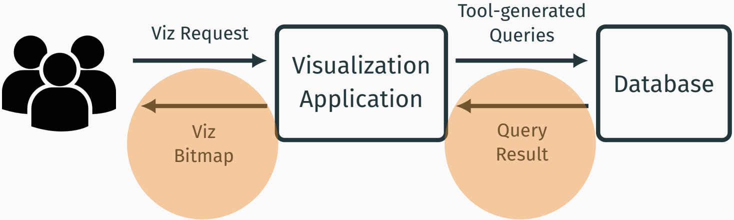
Why is Visualizing Big Data Slow?



Why is Visualizing Big Data Slow?



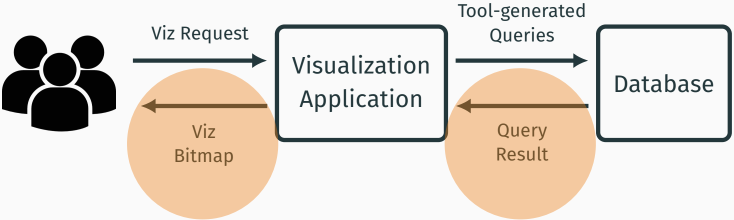
Why is Visualizing Big Data Slow?



We want to

Reduce computational efforts

Why is Visualizing Big Data Slow?



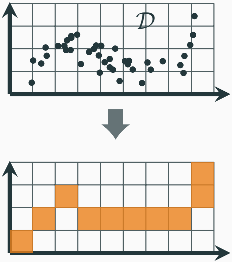
We want to

Reduce computational efforts

Without affecting visual perception

Two Approaches to Fast Viz

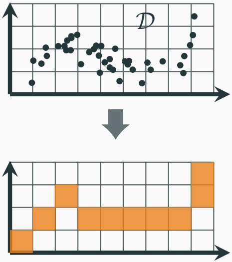
Two Approaches to Fast Viz



(# of bins < $|\mathcal{D}|$) $\rightarrow$ faster

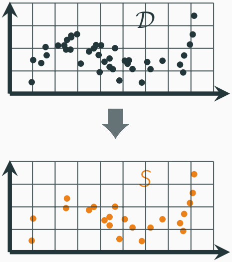
Pre-aggregation

Two Approaches to Fast Viz



(# of bins < $|\mathcal{D}|$) $\rightarrow$ faster

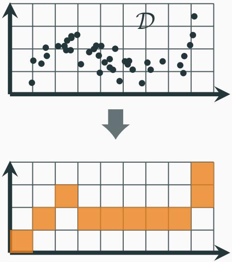
Pre-aggregation



($|S| < |\mathcal{D}|$) $\rightarrow$ faster

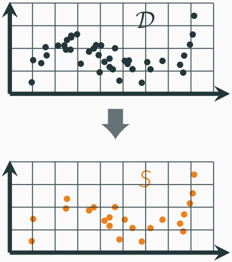
Data Reduction

Two Approaches to Fast Viz



(# of bins < $|\mathcal{D}|$) $\rightarrow$ faster

Pre-aggregation



($|S| < |\mathcal{D}|$) $\rightarrow$ faster

Data Reduction

Our Approach. Good when:

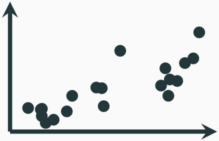
1. You do not want pre-aggregations (no knowledge, lost info).
2. You want no modifications on Viz Application / DB.

Our Concrete Goal: Fast Viz of Scatter Plots

Why Scatter Plots?

Our Concrete Goal: Fast Viz of Scatter Plots

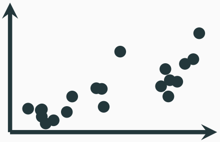
Why Scatter Plots?



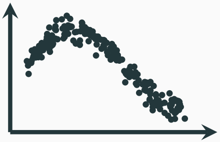
1. Correlation

Our Concrete Goal: Fast Viz of Scatter Plots

Why Scatter Plots?



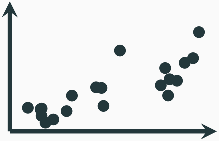
1. Correlation



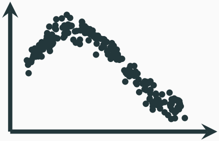
2. Trend Analysis

Our Concrete Goal: Fast Viz of Scatter Plots

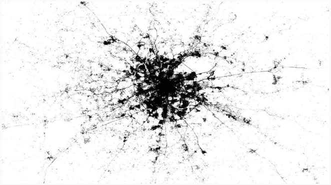
Why Scatter Plots?



1. Correlation



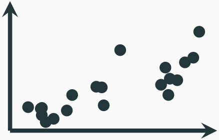
2. Trend Analysis



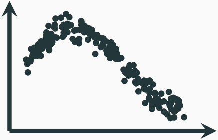
3. Geographic Info

Our Concrete Goal: Fast Viz of Scatter Plots

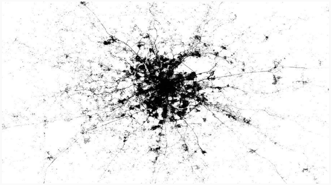
Why Scatter Plots?



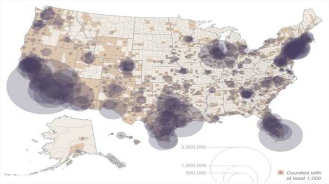
1. Correlation



2. Trend Analysis

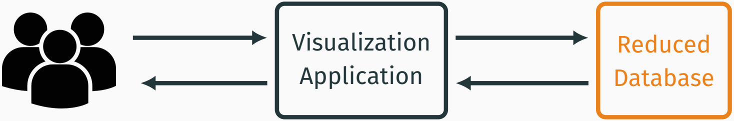


3. Geographic Info

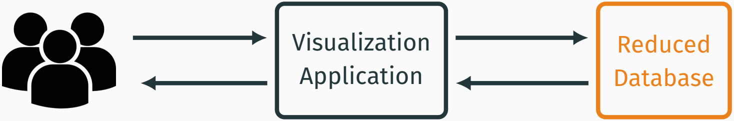


4. Density Estimation

Simple Architecture if you have a Good Sample

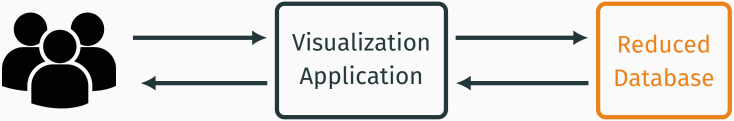


Simple Architecture if you have a Good Sample



Once you generate a sample *offline* $\rightarrow$ Enjoy fast viz at *query time*

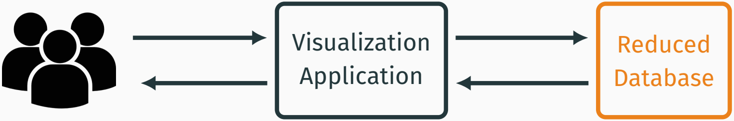
Simple Architecture if you have a Good Sample



Once you generate a sample *offline* $\longrightarrow$ Enjoy fast viz at *query time*

Just like B-tree used to speed up queries

Simple Architecture if you have a Good Sample

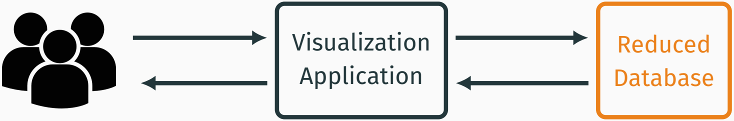


Once you generate a sample *offline* $\longrightarrow$ Enjoy fast viz at *query time*

Just like B-tree used to speed up queries

Question 1: What is a good Sample?

Simple Architecture if you have a Good Sample



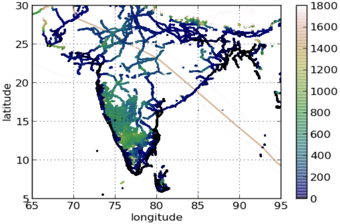
Once you generate a sample *offline* → Enjoy fast viz at *query time*

Just like B-tree used to speed up queries

Question 1: What is a good Sample?

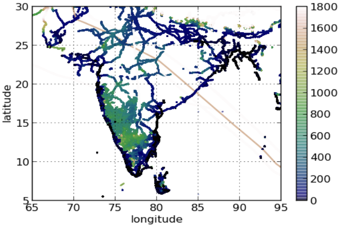
Question 2: How to obtain such a good Sample?

Existing Samplings Fail

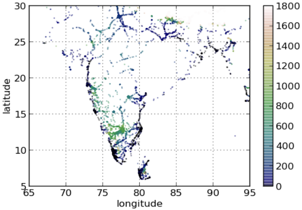


Original (2 billion points, 71 mins)

Existing Samplings Fail



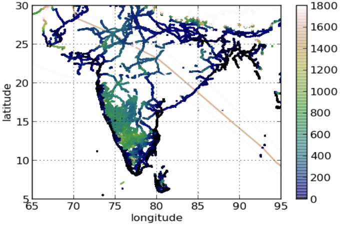
Original (2 billion points, 71 mins)



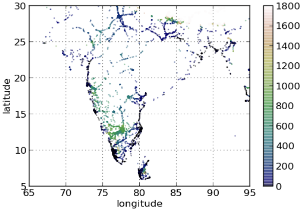
1. Uniform random sample

(1 million, 3 secs)

Existing Samplings Fail

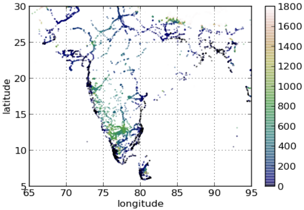


Original (2 billion points, 71 mins)



1. Uniform random sample

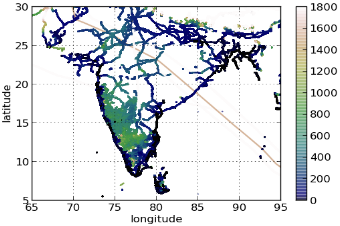
(1 million, 3 secs)



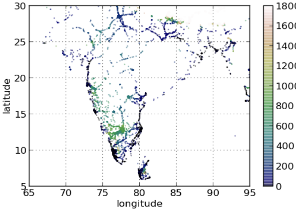
2. Stratified sample

(1 million, 3 secs)

Existing Samplings Fail

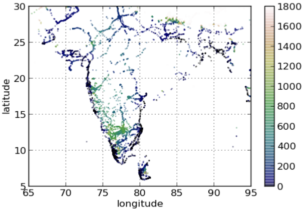


Original (2 billion points, 71 mins)



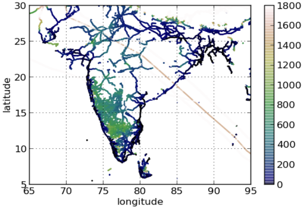
1. Uniform random sample

(1 million, 3 secs)



2. Stratified sample

(1 million, 3 secs)



3. Ours

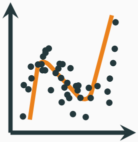
(1 million, 3 secs)

Good Sample determined by User Performance

Three common visualization-driven tasks [4]:

Good Sample determined by User Performance

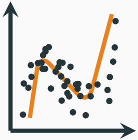
Three common visualization-driven tasks [4]:



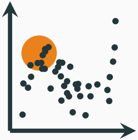
1. Trend Analysis

Good Sample determined by User Performance

Three common visualization-driven tasks [4]:



1. Trend Analysis



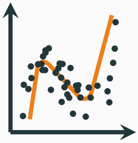
2. Density Est.

Good Sample determined by User Performance

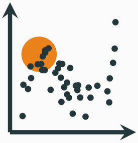
Three common visualization-driven tasks [4]:

Good Sample determined by User Performance

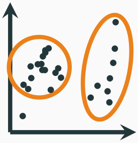
Three common visualization-driven tasks [4]:



1. Trend Analysis



2. Density Est.

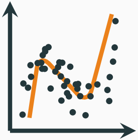


3. Clustering

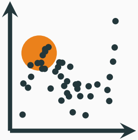
Other tasks: (1) shape viz, (2) classification, (3) hierarchy understanding, (4) community detection, etc.

Good Sample determined by User Performance

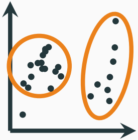
Three common visualization-driven tasks [4]:



1. Trend Analysis



2. Density Est.



3. Clustering

What we focus

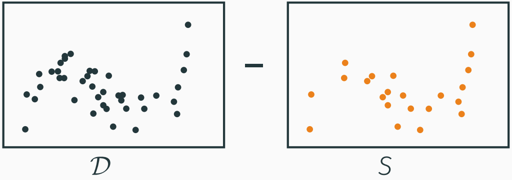
Other tasks: (1) shape viz, (2) classification, (3) hierarchy understanding, (4) community detection, etc.

Good Sample leads to High Fidelity

What is a good sample (S) of the original dataset ($\mathcal{D}$)?

Good Sample leads to High Fidelity

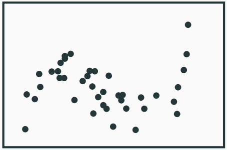
What is a good sample (S) of the original dataset ($\mathcal{D}$)?



Good Sample leads to High Fidelity

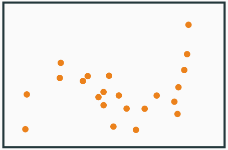
What is a good sample (S) of the original dataset ($\mathcal{D}$)?

$\text{Loss}(S) \; =$



$\mathcal{D}$

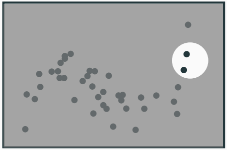
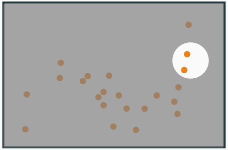
$-$



S

Good Sample leads to High Fidelity

What is a good sample (S) of the original dataset ($\mathcal{D}$)?

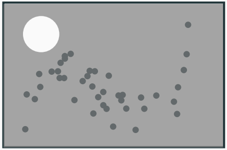
$\text{subloss}(x) =$  $-$ 

$\mathcal{D}$ S

Good Sample leads to High Fidelity

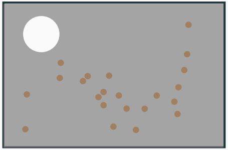
What is a good sample (S) of the original dataset ($\mathcal{D}$)?

subloss(x) =



$\mathcal{D}$

−

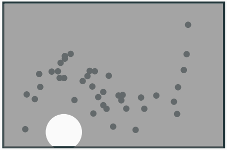


S

Good Sample leads to High Fidelity

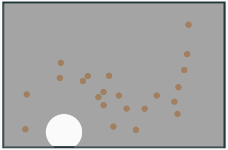
What is a good sample (S) of the original dataset ($\mathcal{D}$)?

subloss(x) =



$\mathcal{D}$

−

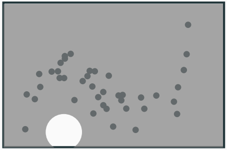


S

Good Sample leads to High Fidelity

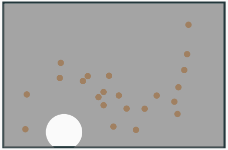
What is a **good** sample (S) of the original dataset ($\mathcal{D}$)?

subloss(x) =



$\mathcal{D}$

−



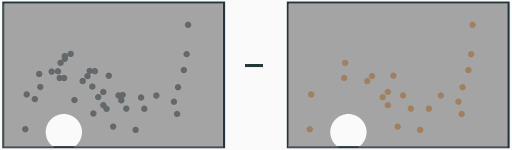
S

If the visual difference for **every circle** is small,
$\rightarrow$ two viz will look similar.

Good Sample leads to High Fidelity

What is a good sample (S) of the original dataset ($\mathcal{D}$)?

subloss(x) =



$\mathcal{D}$ S

If the visual difference for every circle is small,
$\rightarrow$ two viz will look similar.

Our Goal

To minimize:

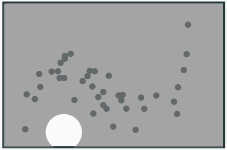
$$\text{Loss}(S) = \int \text{subloss}(x)\, dx$$

where subloss(x) is the viz distance for the circle centered at x.

Good Sample leads to High Fidelity

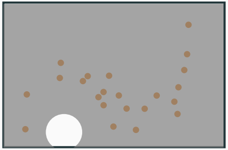
What is a **good** sample (S) of the original dataset ($\mathcal{D}$)?

subloss(x) =



$\mathcal{D}$

−



S

If the visual difference for **every circle** is small,
$\rightarrow$ two viz will look similar.

Our Goal

To minimize:

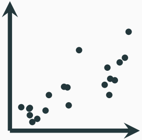
$$\text{Loss}(S) = \int \textbf{subloss}(x)\, dx$$

where subloss(x) is the viz distance for the circle centered at x.

What function to use for subloss(x)?

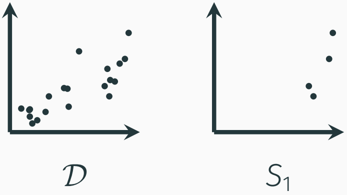
Our Suggestion for subloss

Our Suggestion for subloss

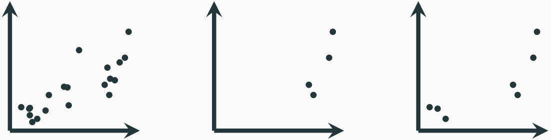


$\mathcal{D}$

Our Suggestion for subloss

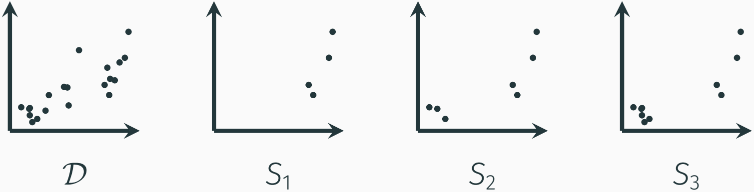


Our Suggestion for subloss



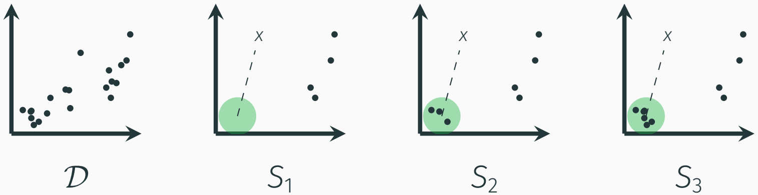
$\mathcal{D}$ $\qquad$ S_1 $\qquad$ S_2

Our Suggestion for subloss



Our Suggestion for subloss

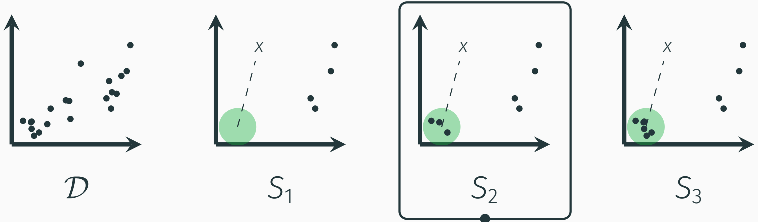
Let's see what happens **around** the location x.



$\mathcal{D}$ S_1 S_2 S_3

Our Suggestion for subloss

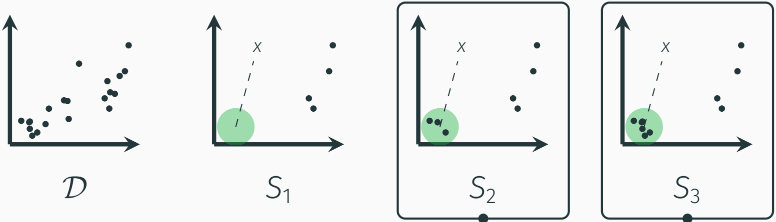
Let's see what happens **around** the location x.



Initially, visual distance around x reduces fast

Our Suggestion for subloss

Let's see what happens **around** the location *x*.

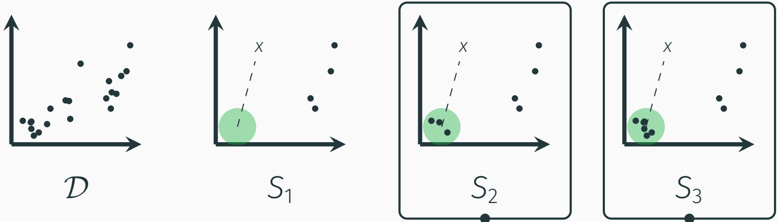


Initially, visual distance around *x* reduces fast

Visual distance around *x* NO LONGER reduces much

Our Suggestion for subloss

Let's see what happens **around** the location x.



Initially, visual distance around x reduces fast

Visual distance around x NO LONGER reduces much

Our Choice

$$\textbf{subloss}(x) = \frac{1}{\sum_{s_i \in S} \kappa(x, s_i)}$$

where κ captures the *proximity* between two coordinates.

Good Viz by Solving VAS Problem

Given a budget $|S| = K$, we want

$$\underset{\substack{S \\ \text{s.t. } S \subseteq \mathcal{D} \,\wedge\, |S|=K}}{\arg\min} \; \boxed{\text{Loss}(S)} \dashrightarrow \int \frac{1}{\sum_{s_i \in S} \kappa(x, s_i)} \; dx$$

Good Viz by Solving VAS Problem

Given a budget $|S| = K$, we want

$$\underset{\substack{S \\ \text{s.t. } S \subseteq \mathcal{D} \,\wedge\, |S|=K}}{\arg\min} \; \boxed{\mathrm{Loss}(S)} \dashrightarrow \int \frac{1}{\sum_{s_i \in S} \kappa(x, s_i)} \, dx$$

After some approximations and calculations,

Problem VAS

$$\underset{\substack{S \\ \text{s.t. } S \subseteq \mathcal{D} \,\wedge\, |S|=K}}{\arg\min} \; \boxed{\sum_{s_i, s_j \in S;\, i<j} \tilde{\kappa}(s_i, s_j)} \dashrightarrow Loss(S)$$

where

$$\tilde{\kappa}(s_i, s_j) = \int \kappa(x, s_i)\, \kappa(x, s_j) \, dx$$

Good Viz by Solving VAS Problem

Given a budget $|S| = K$, we want

$$\underset{\substack{S \\ \text{s.t. } S \subseteq \mathcal{D} \,\wedge\, |S|=K}}{\arg\min} \; \boxed{\mathrm{Loss}(S)} \dashrightarrow \int \frac{1}{\sum_{s_j \in S} \kappa(x, s_j)} \, dx$$

After some approximations and calculations,

Problem VAS

$$\underset{\substack{S \\ \text{s.t. } S \subseteq \mathcal{D} \,\wedge\, |S|=K}}{\arg\min} \; \boxed{\sum_{s_i, s_j \in S;\, i<j} \tilde{\kappa}(s_i, s_j)} \dashrightarrow Loss(S)$$

where

$$\tilde{\kappa}(s_i, s_j) = \int \kappa(x, s_i)\, \kappa(x, s_j)\, dx$$

<u>NP-hard (evaluated later); How can we solve?</u>

1. Motivation
2. Formal Definition of Good Sample
3. Approximation Algorithm to VAS Problem
4. Large-Scale User Study
5. Offline Runtime Analysis

Basic (Slow) Approach

Nemhauser, et al. [2] algorithm (comes with an error guarantee)

Basic (Slow) Approach

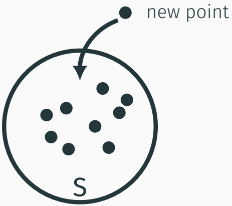
Nemhauser, et al. [2] algorithm (comes with an error guarantee)

As scanning over a dataset,

Basic (Slow) Approach

Nemhauser, et al. [2] algorithm (comes with an error guarantee)

As scanning over a dataset,

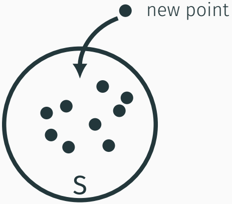


When $|S| < K$

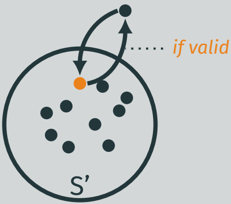
Basic (Slow) Approach

Nemhauser, et al. [2] algorithm (comes with an error guarantee)

As scanning over a dataset,



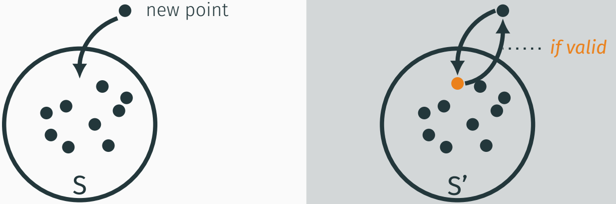
When $|S| < K$



When $|S| = K$

Nemhauser, et al. [2] algorithm (comes with an error guarantee)

As scanning over a dataset,



When $|S| < K$

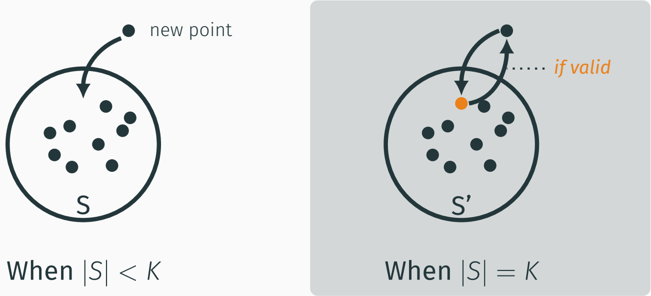
When $|S| = K$

Valid replacement: Loss(S') < Loss(S)

Basic (Slow) Approach

Nemhauser, et al. [2] algorithm (comes with an error guarantee)

As scanning over a dataset,



Valid replacement: Loss(S') < Loss(S)

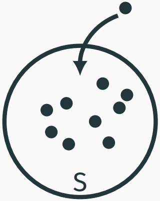
Testing for *valid replacements* is **too Slow**: $O(K^3)$ for every point

Our Fast Approach

Our algorithm: Expand/Shrink operation

Our Fast Approach

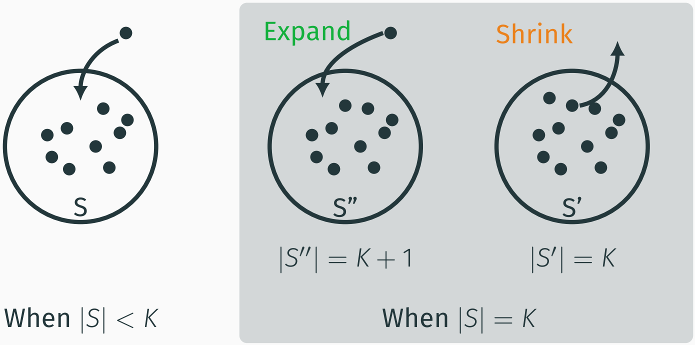
Our algorithm: Expand/Shrink operation



When $|S| < K$

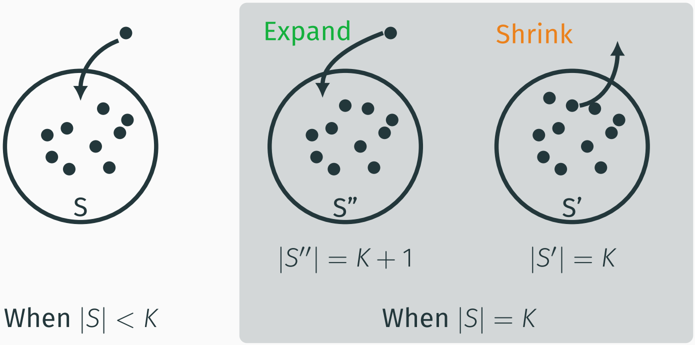
Our Fast Approach

Our algorithm: Expand/Shrink operation



Our Fast Approach

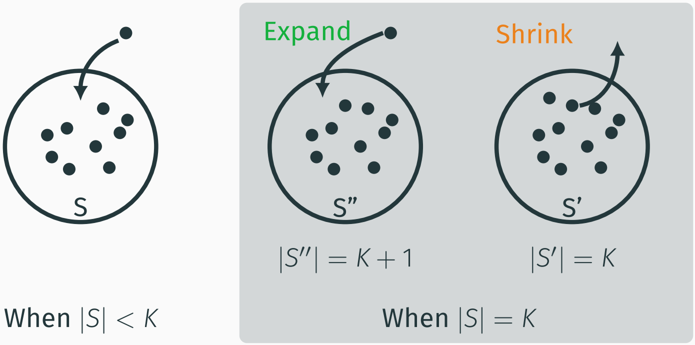
Our algorithm: Expand/Shrink operation



Expand/Shrink is Fast: $O(K)$ for every point $\rightarrow$ $\mathbf{O(K^2)}$ **times faster!!**

Our Fast Approach

Our algorithm: Expand/Shrink operation



Expand/Shrink is **Fast**: $O(K)$ for every point $\rightarrow$ $\mathbf{O(K^2)}$ **times faster!!**

The result is exactly same as the Nemhauser's algorithm.

Outline

1. Motivation
2. Formal Definition of Good Sample
3. Approximation Algorithm to VAS Problem
4. Large-Scale User Study
5. Offline Runtime Analysis

Large-scale User Study: Setup

Large-scale User Study: Setup

Questions: User performance, Validity of VAS Problem

Large-scale User Study: Setup

Questions: User performance, Validity of VAS Problem

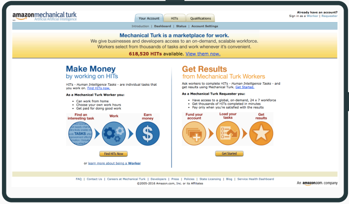
Tasks	Questions	Datasets
trend analysis, density est, clustering	#: 72 - 80 with different sample sizes	· 22 million GPS logs [5] · Synthetic

Large-scale User Study: Setup

Questions: User performance, Validity of VAS Problem

Tasks	Questions	Datasets
trend analysis, density est, clustering	#: 72 - 80 with different sample sizes	· 22 million GPS logs [5] · Synthetic

mechanical turk



× 40 unique

Large-scale User Study: Example

Amazon Mechanical Turk

https://workersandbox.mturk.com/mturk/preview?requesterId=&canAccept=true&hitId=3HEM8MA6H9DV7EZBLC9WFCPQ6AWQPH...

Where is the Densest?

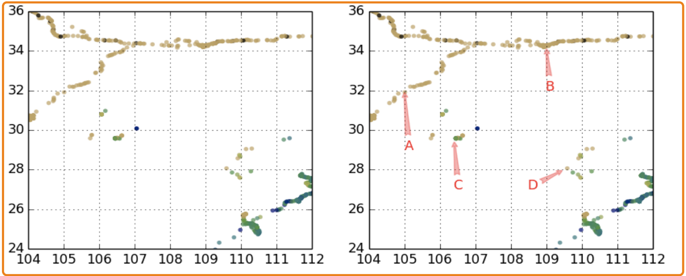
In the following figure, density is indicated by either

- Putting more dots in denser areas, or
- Using larger dots for denser areas.

Given the map below, your task is to estimate the most or least dense area.

The left figure is without marks, and the right figure is with marks.

instruction



figures from a sample

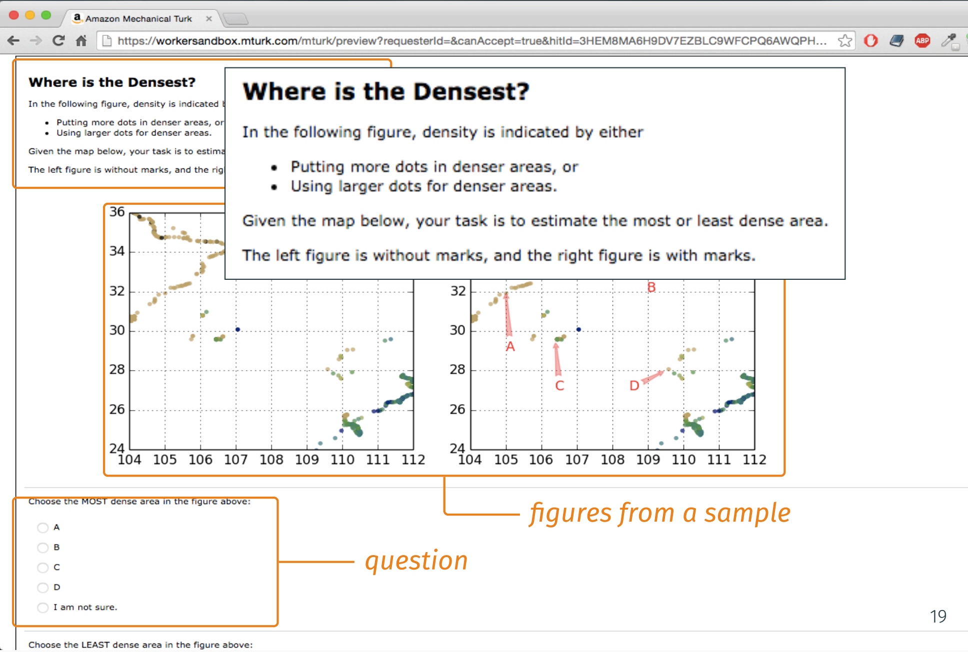
Choose the MOST dense area in the figure above:

- A
- B
- C
- D
- I am not sure.

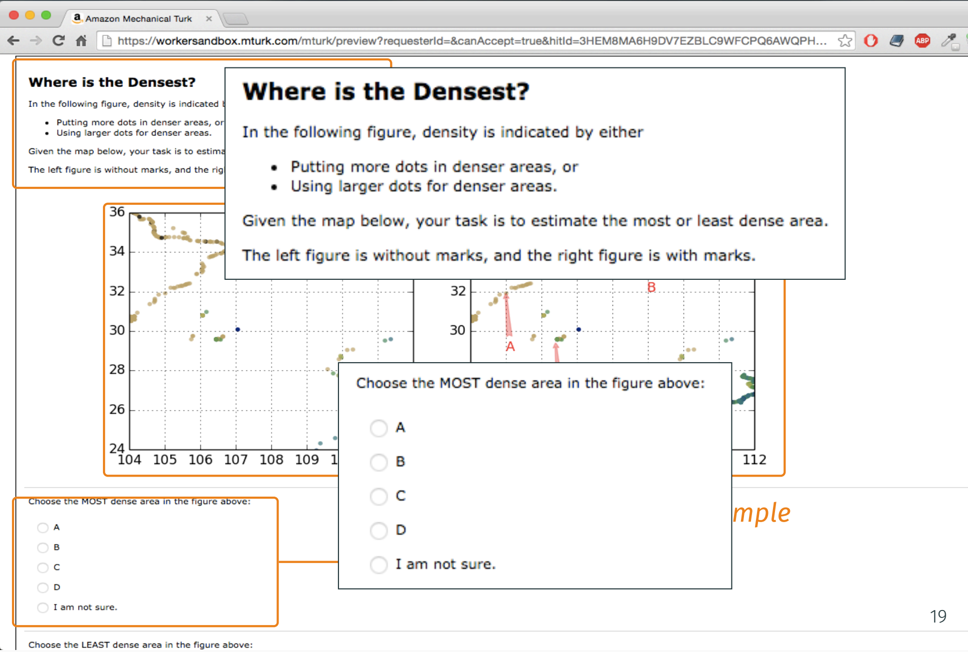
question

Choose the LEAST dense area in the figure above:

Large-scale User Study: Example



Large-scale User Study: Example



Superior User Performance by VAS

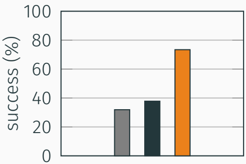
Average user performance

1. Uniform Random Sampling
2. Stratified Sampling
3. Visualization-Aware Sampling (VAS)

Superior User Performance by VAS

Average user performance

1. Uniform Random Sampling
2. Stratified Sampling
3. Visualization-Aware Sampling (VAS)



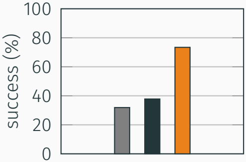
Trend Analysis

success: if a user answers a question correctly.

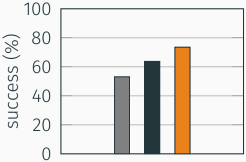
Superior User Performance by VAS

Average user performance

1. Uniform Random Sampling
2. Stratified Sampling
3. Visualization-Aware Sampling (VAS)



Trend Analysis



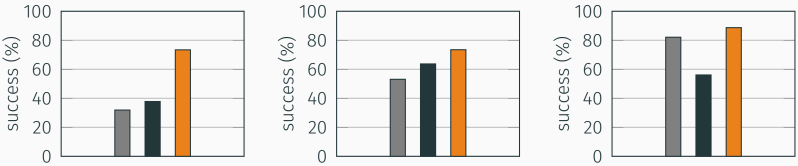
Density

success: if a user answers a question correctly.

Superior User Performance by VAS

Average user performance

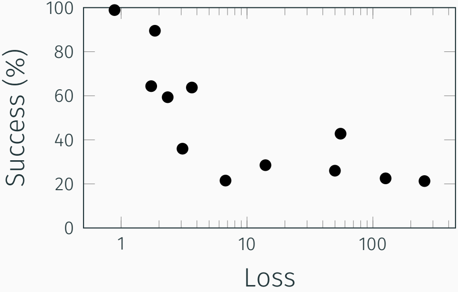
1. Uniform Random Sampling
2. Stratified Sampling
3. Visualization-Aware Sampling (VAS)



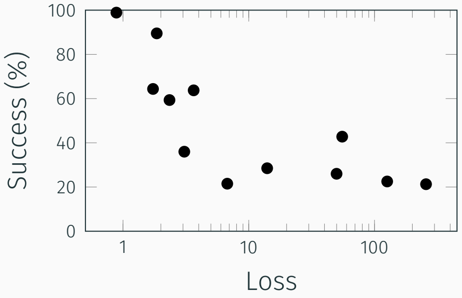
Trend Analysis   Density   Clustering

success: if a user answers a question correctly.

Strong Correlation (between User Performance and Loss)

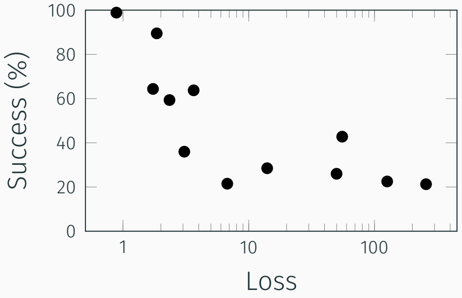


Strong Correlation (between User Performance and Loss)



Smaller Loss(S) $\rightarrow$ More successes

Strong Correlation (between User Performance and Loss)

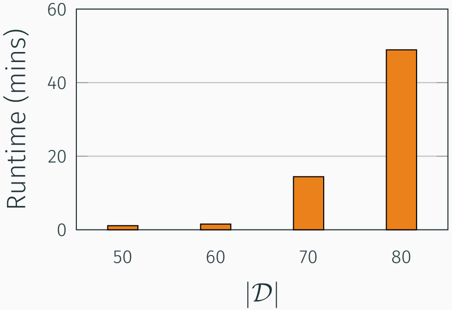


Smaller Loss(S) $\rightarrow$ More successes

(Spearman's rank) correlation coefficient = -0.85

Exact Algorithm is Slow

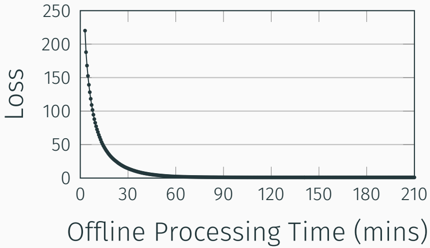
We tested Mixed Integer Programming (MIP) for exact solutions.



Exact approach (MIP) is prohibitively slow even for small data.

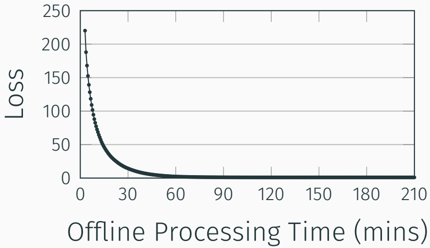
Fast Offline Sampling over Big Data

Ran our VAS algorithm over 2 billion points.



Fast Offline Sampling over Big Data

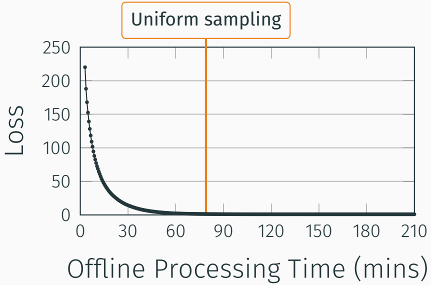
Ran our VAS algorithm over 2 billion points.



Minimizes Loss fast.

Fast Offline Sampling over Big Data

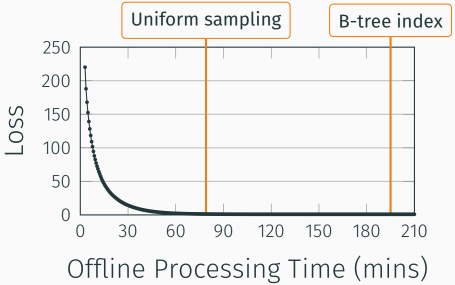
Ran our VAS algorithm over 2 billion points.



Minimizes Loss fast.

Fast Offline Sampling over Big Data

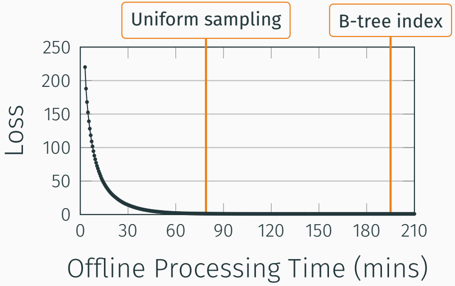
Ran our VAS algorithm over 2 billion points.



Minimizes Loss fast.

Fast Offline Sampling over Big Data

Ran our VAS algorithm over 2 billion points.

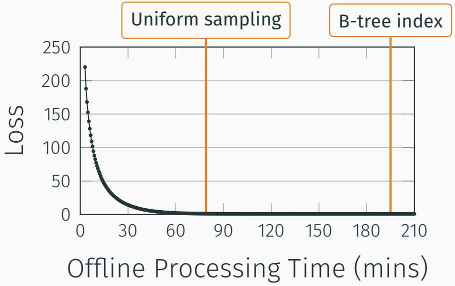


Minimizes Loss **fast**.

High-quality viz even at early steps

Fast Offline Sampling over Big Data

Ran our VAS algorithm over 2 billion points.



Minimizes Loss fast.

High-quality viz even at early steps

Gradually improves if more time is allowed

Demo

Also available at `yongjoopark.com/vas`

Conclusion

Conclusion

We formulated an important problem: Sampling for Visualization.

Conclusion

We formulated an important problem: Sampling for Visualization.

We proposed an efficient algorithm to VAS problem.

Conclusion

We formulated an important problem: Sampling for Visualization.

We proposed an efficient algorithm to VAS problem.

We demonstrated users achieve superior performance with VAS.

Thank You!

References I

- Z. Liu and J. Heer.
  The effects of interactive latency on exploratory visual analysis.

  TVCG, 2014.
- G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher.
  An analysis of approximations for maximizing submodular set functions—i.
  Mathematical Programming, 1978.
- B. Shneiderman.
  Response time and display rate in human performance with computers.
  CSUR, 1984.

References II

B. Shneiderman.
The eyes have it: A task by data type taxonomy for information visualizations.
In *Symposium on Visual Languages*, 1996.

Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma.
Understanding mobility based on gps data.
In *UbiComp*, 2008.