CORESETS FOR SIGNAL PROCESSING

Dan Feldman
Assistant Professor

אוניברסיטת חיפה
University of Haifa

Motivation

New computation models

- Big Data
- Streaming real-time data
- Distributed data

Limited hardware

- Computation: IoT, GPU
- Energy: smartphones, drones

Common solution

- New optimization algorithms

Big Data

- Volume: huge amount of data points
- Variety: huge number of sensors
- Velocity: data arrive in real-time streaming

Need:

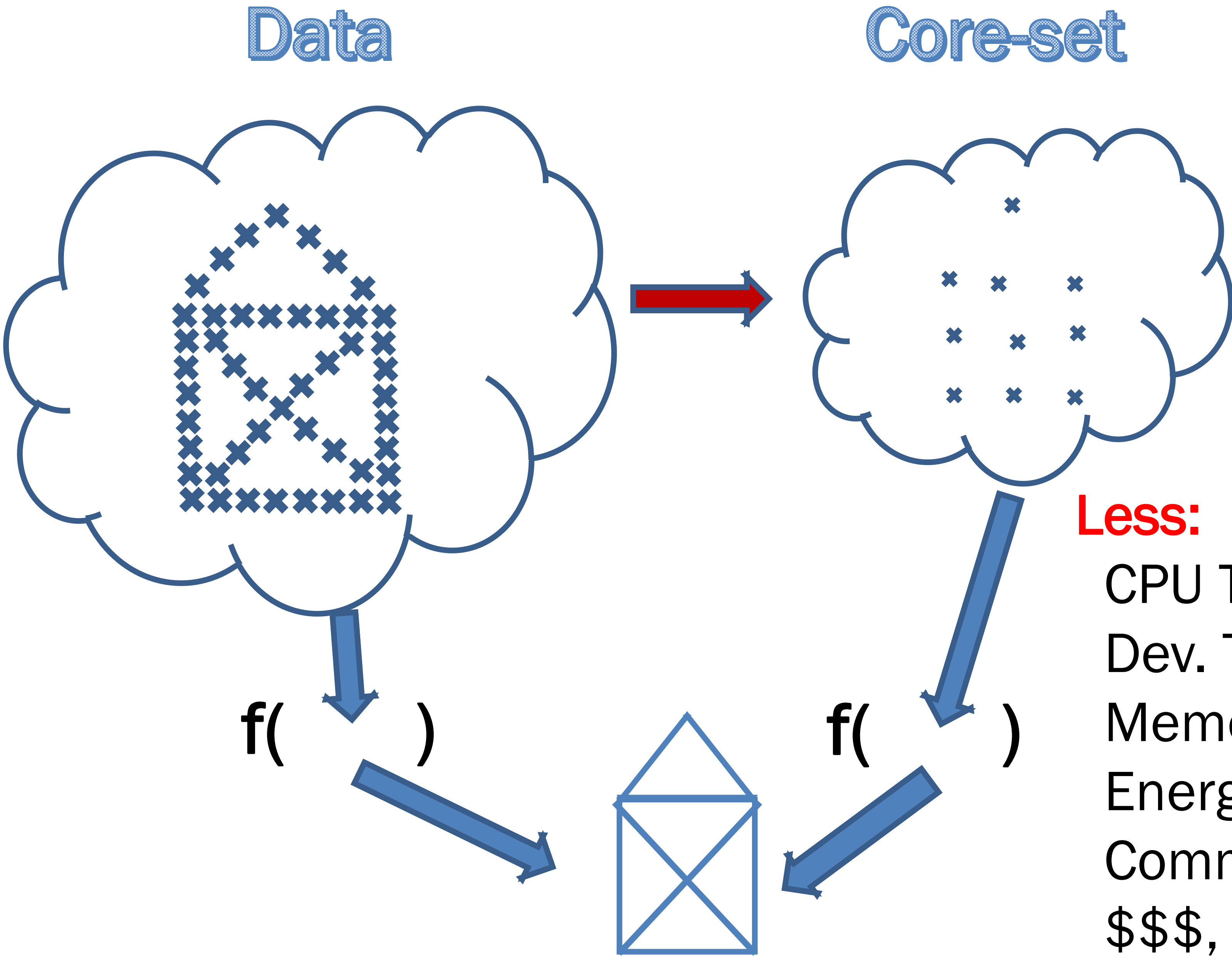
- *Streaming algorithms (use logarithmic memory)*
- *Parallel algorithms (use networks, clouds)*
- *Simple computations (use GPUs)*
- *No assumption on order of points*

Big Data Computation model

- *= Streaming + Parallel computation*
- *Input: infinite stream of vectors*
- $n =$ *vectors seen so far*
- $\sim\log n$ *memory*
- *M processors*
- *~log (n)/M insertion time per point*

(Embarrassingly parallel)

Focus on ~~optimization~~ summarization

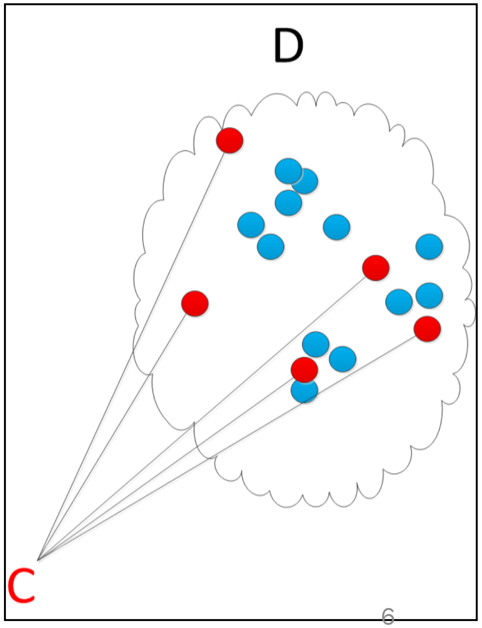


Less:
- CPU Time
- Dev. Time
- Memory
- Energy
- Comm.
- $$$, ...

Challenge:
Find RIGHT data from Big Data

Given data *D* and Algorithm *A* with *A*(*D*) intractable, can we efficiently reduce *D* to *C* so that *A*(C) fast and *A*(C)~*A*(D)?

Provable guarantees on approximation with respect to the size of C



Generic Coreset definition

Let

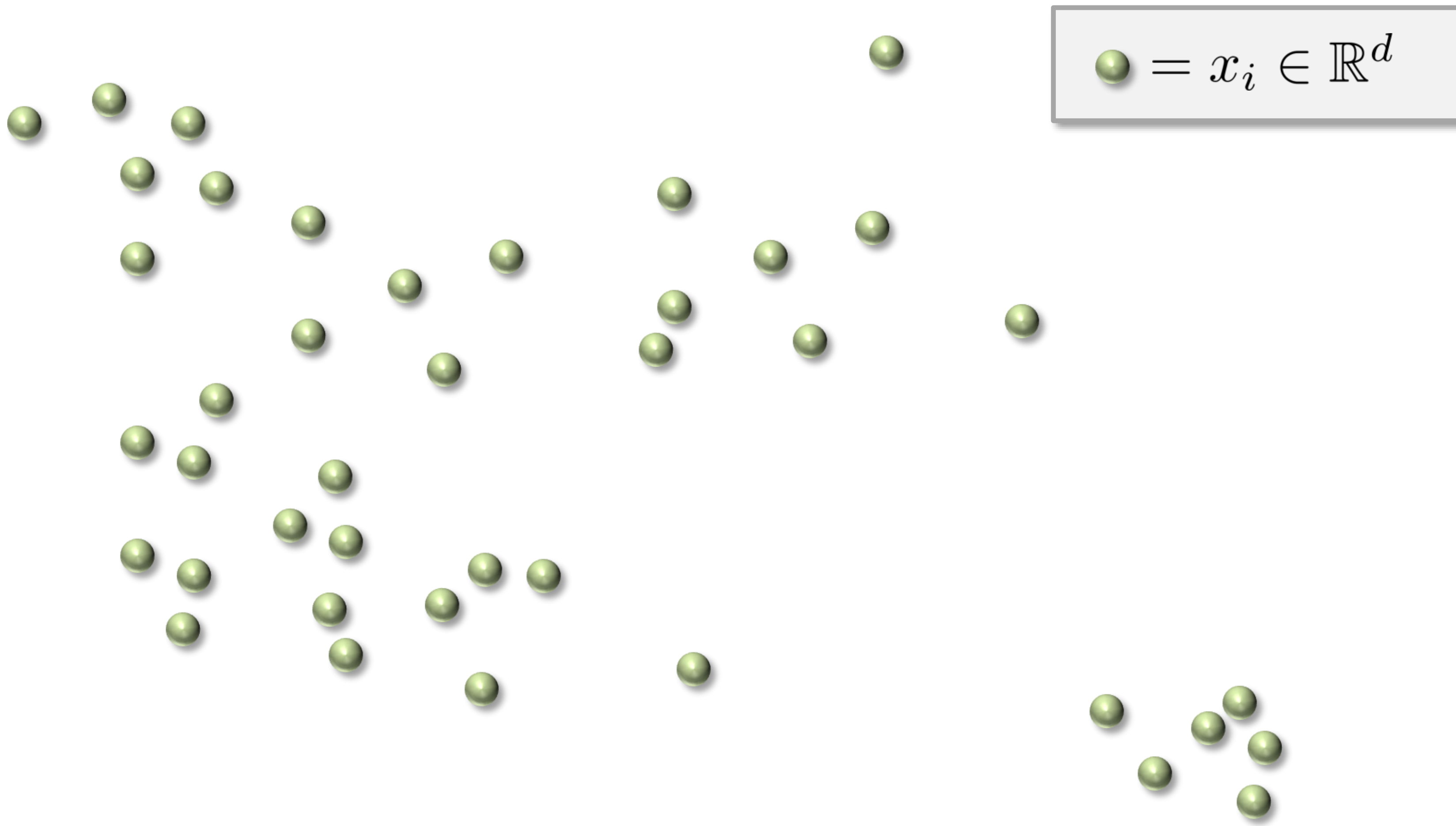
- X be a set, called *points set*
- Q be a set, called *query set*
- $\mathrm{cost}: 2^X \times Q \rightarrow [0, \infty)$ be a function that maps every set $P \subseteq X$ and query $q \in Q$ into a non-negative number $\mathrm{cost}(P, q)$

For a given $\epsilon > 0$ and $P \subseteq X$,
the set $C \subseteq X$ is a *corset* if
for every $q \in Q$ we have

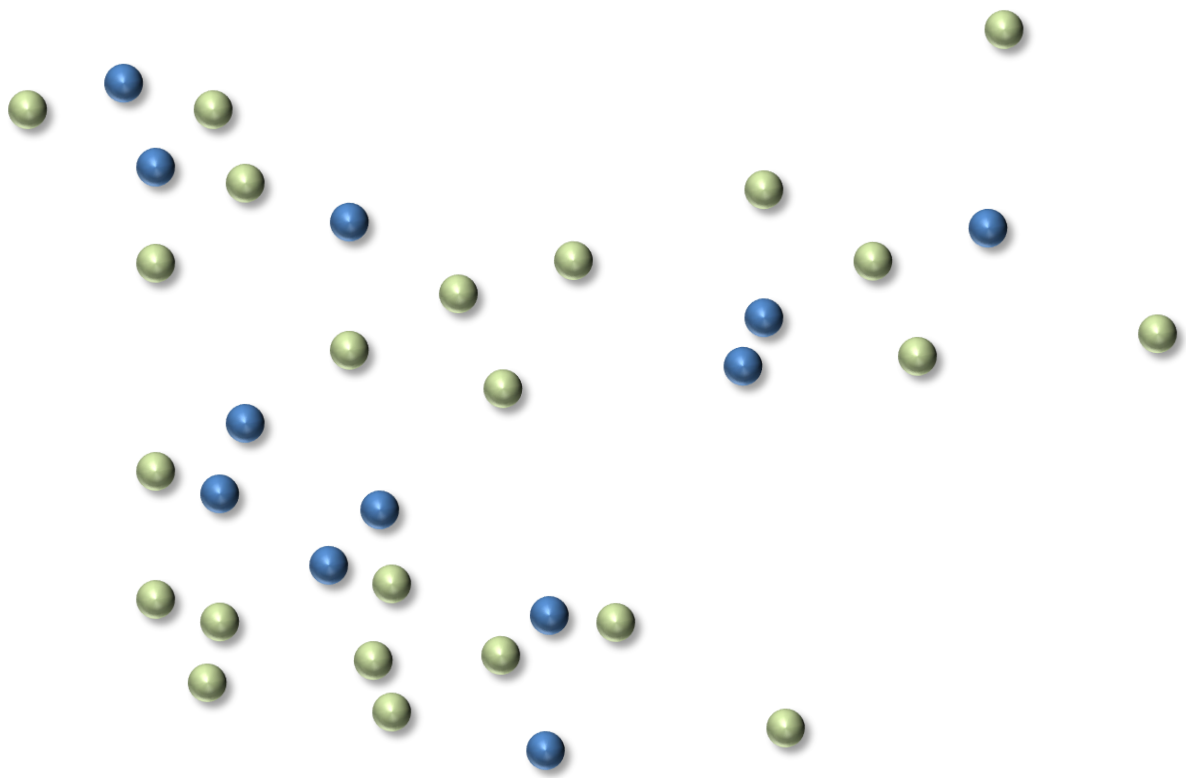
$$\mathrm{cost}(P, q) \sim cost(C, q)$$

Up to $(1 \pm \epsilon)$ multiplicative error

Naïve Uniform Sampling

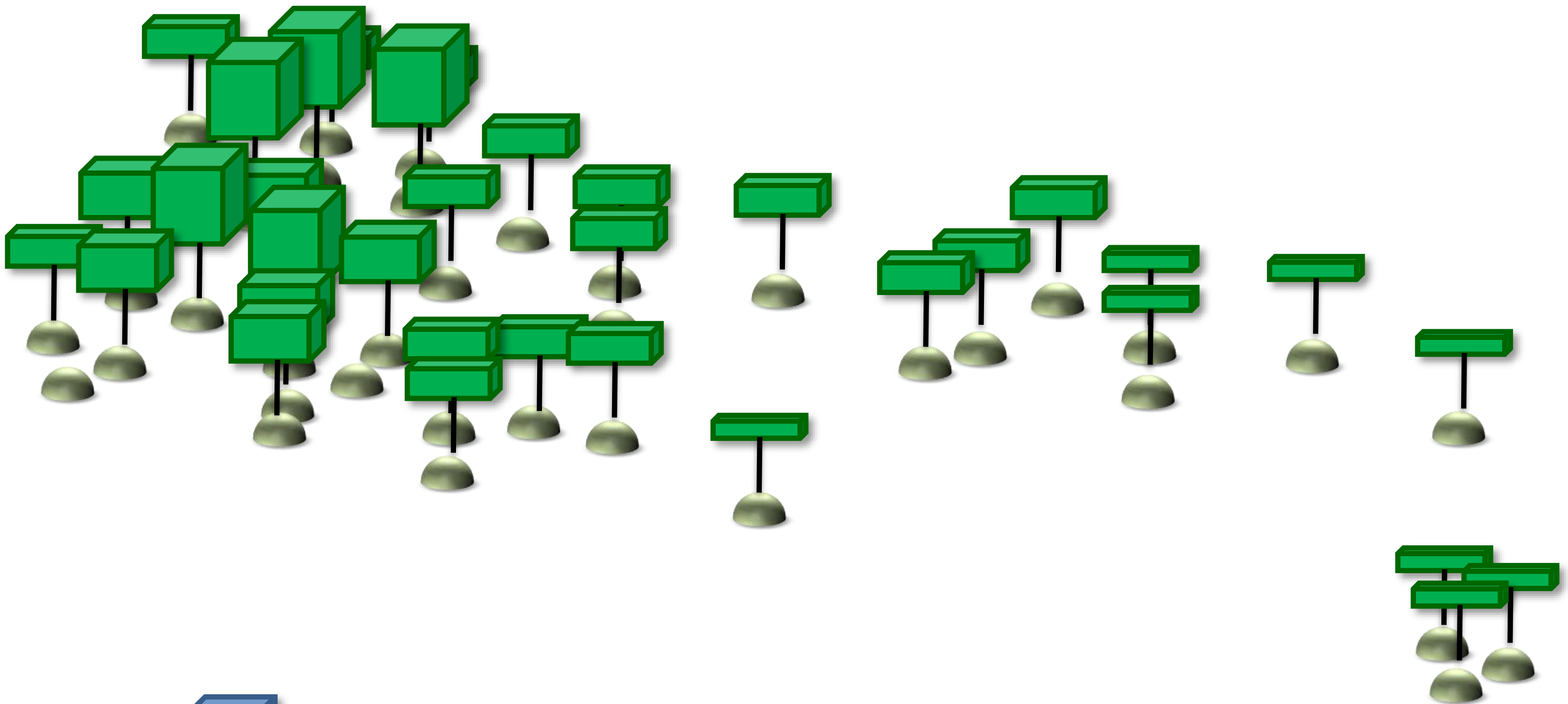


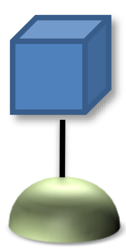
Naïve Uniform Sampling



Sample a set *U* of *m* points uniformly

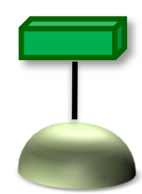
Importance Weights





$Sensitivity(p)$

Sampling distribution



$$\frac{1}{Sensitivity(p)}$$

Weights

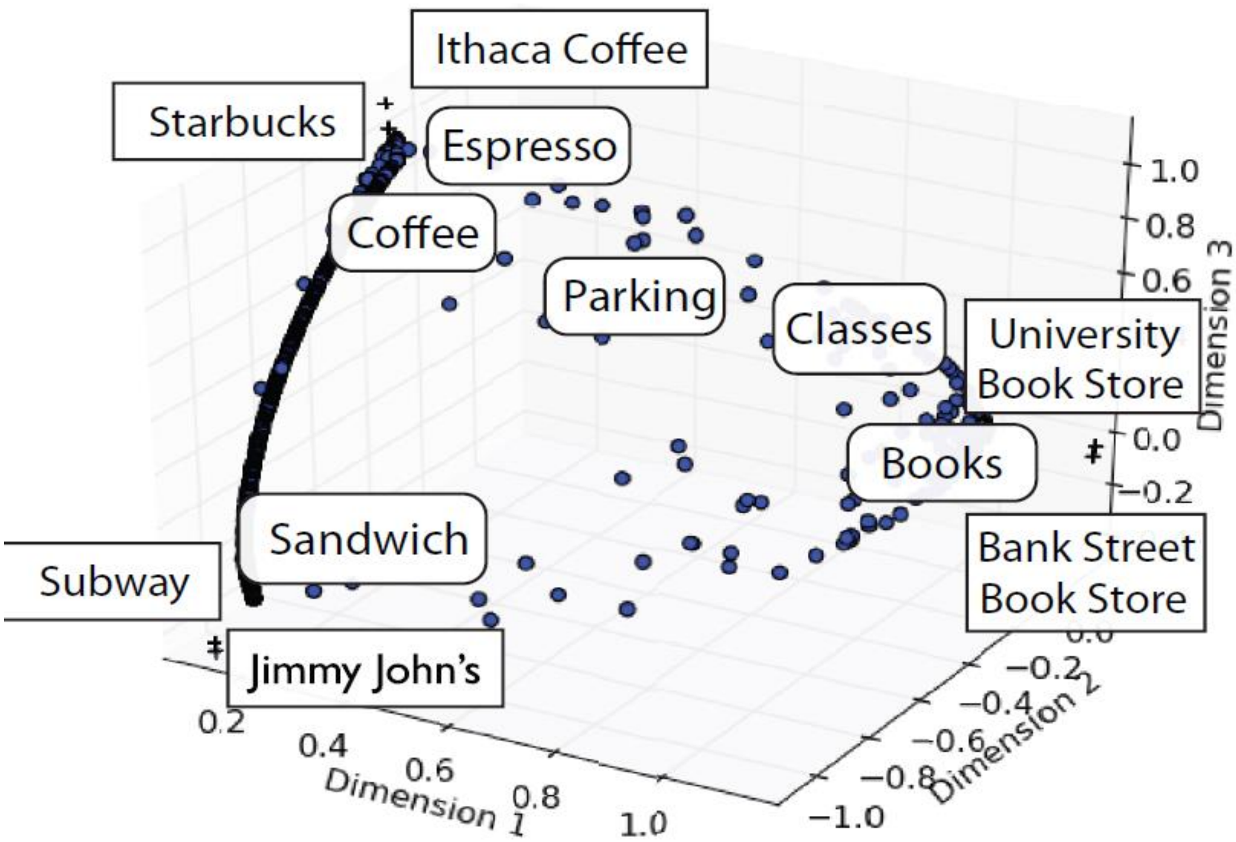
Coreset for Image Denoising

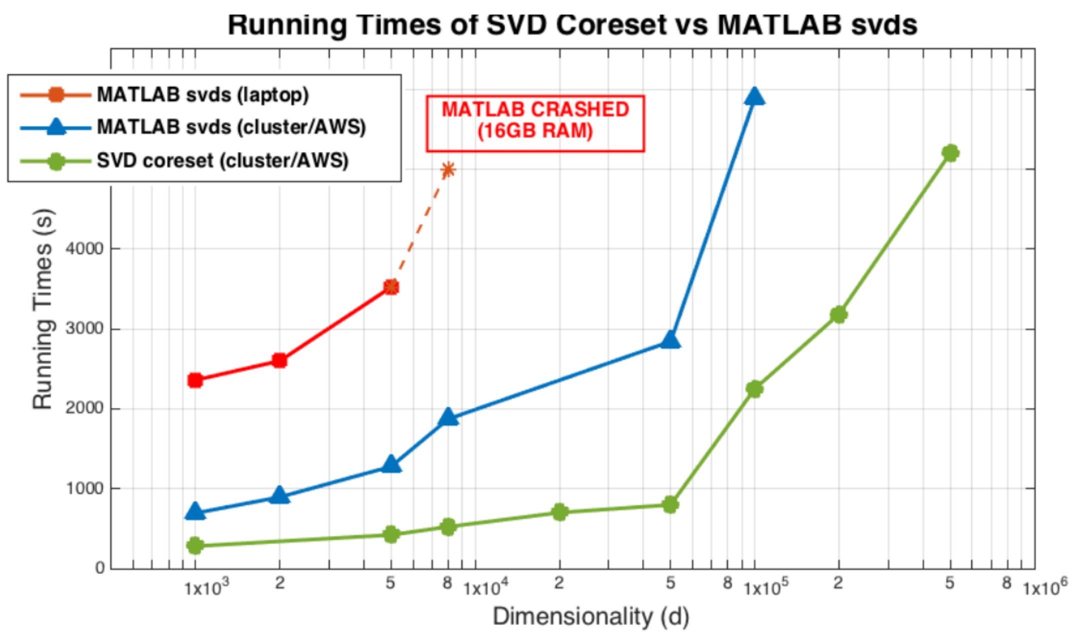
[F, Feigin , Sochen [SSVM’13]

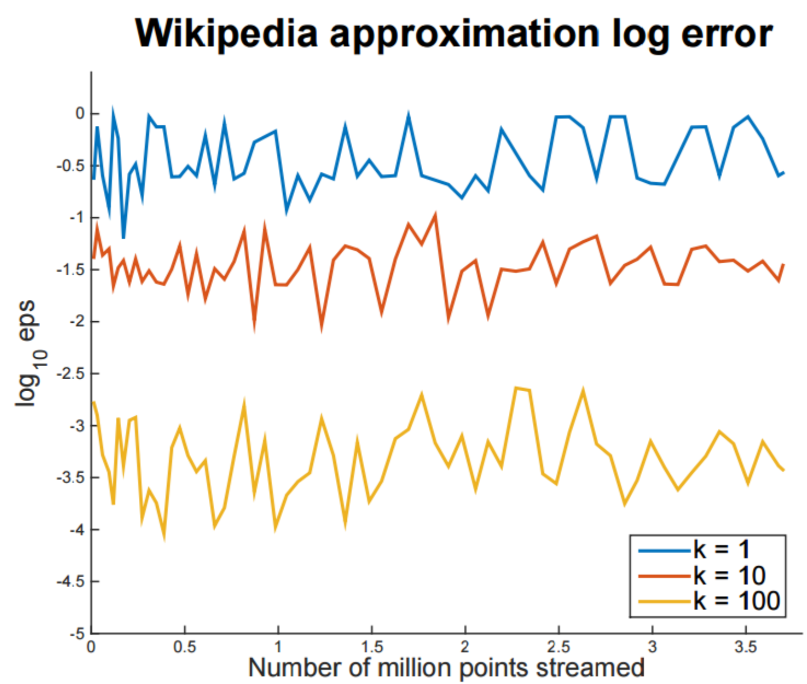
Uniform sample=
only green/white points

First Provable Latent Semantic Analysis on Wikipedia (now Twitter)

[SDM'16, with Artem Barger,
NIPS'16, with Prof. Daniela Rus]

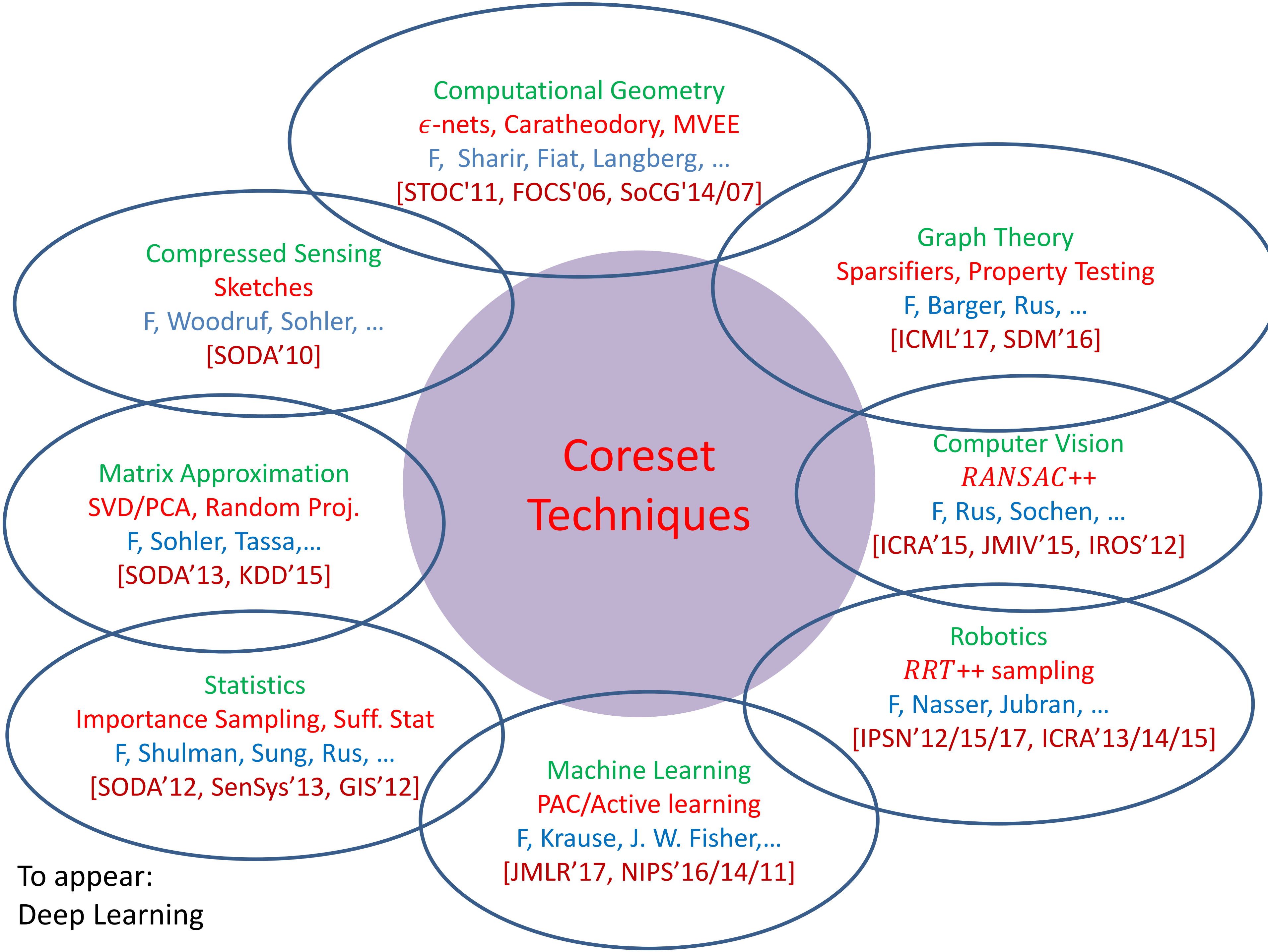



Running Times of SVD Coreset vs MATLAB svds
MATLAB svds (laptop)
MATLAB svds (cluster/AWS)
SVD coreset (cluster/AWS)
MATLAB CRASHED
(16GB RAM)
Running Times (s)
Dimensionality (d)


Wikipedia approximation log error
log10 eps
Number of million points streamed
k = 1
k = 10
k = 100

Example Coresets

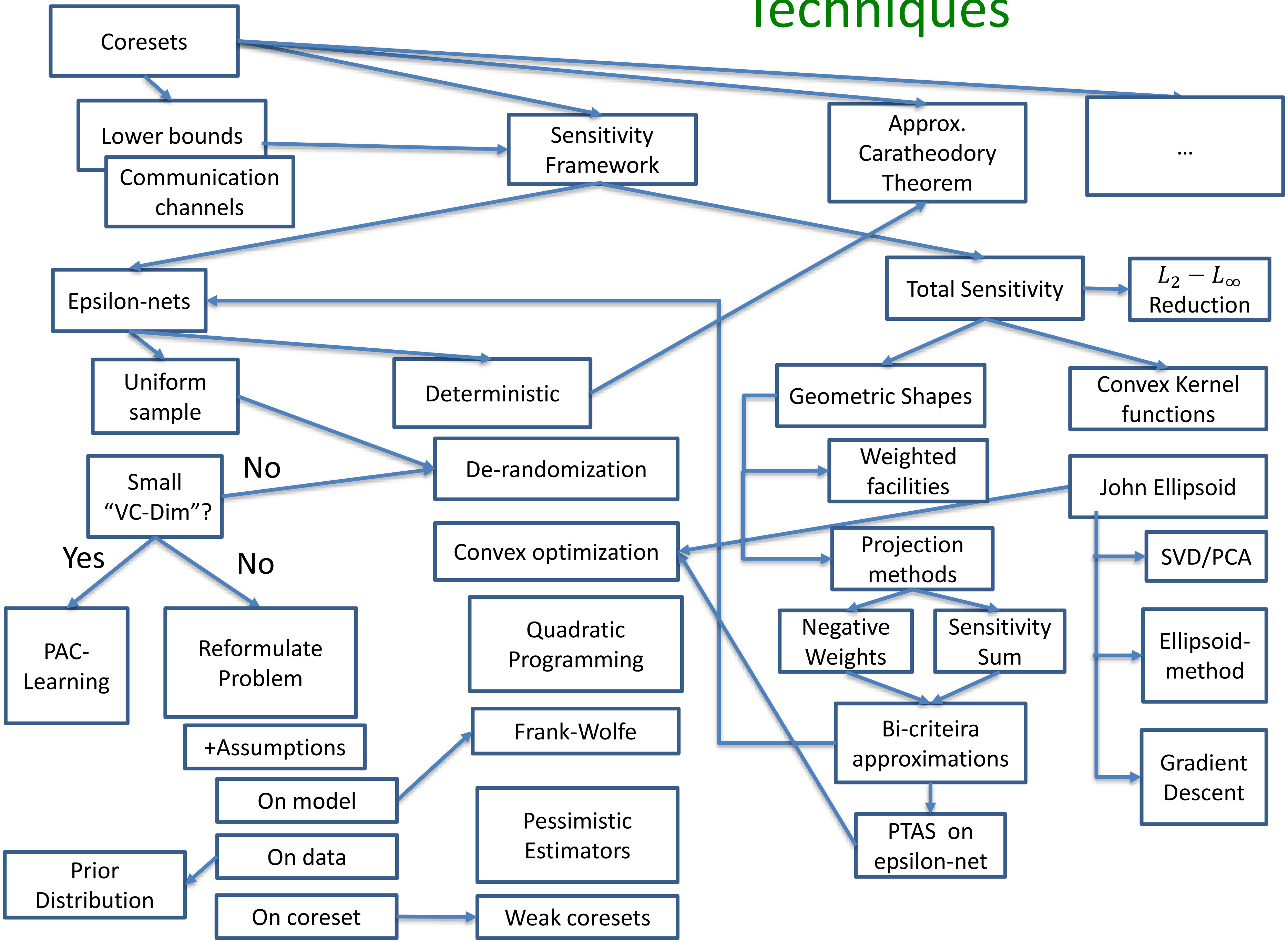
- Deep Learning [F, E. Tolichensky, submitted]
- Logistic Regression [F, Sedat, Murad, submitted]
- Mixture of Gaussians [F, Krause, etc JMLR'17]
- PCA/SVD [F, Rus, and Volkob, NIPS'16]
- k-Means [F, Barger, SDM'16]
- Non-Negative Matrix Factorization [F, Tassa, KDD15]
- Pose Estimatoin [F, Cindy, Rus, ICRA'15]
- Robots Coverage [F, Gil, Rus, ICRA'13]
- Signal Segmentation [F, Rosman, Rus, NIPS'14]
- ....
- k-Line Means [F, Fiat, Sharir, FOCS'06]


Computational Geometry
ϵ-nets, Caratheodory, MVEE
F, Sharir, Fiat, Langberg, …
[STOC'11, FOCS'06, SoCG'14/07]
Compressed Sensing
Sketches
F, Woodruf, Sohler, …
[SODA'10]
Graph Theory
Sparsifiers, Property Testing
F, Barger, Rus, …
[ICML'17, SDM'16]
Coreset Techniques
Matrix Approximation
SVD/PCA, Random Proj.
F, Sohler, Tassa,…
[SODA'13, KDD'15]
Computer Vision
$RANSAC$++
F, Rus, Sochen, …
[ICRA'15, JMIV'15, IROS'12]
Robotics
RRT++ sampling
F, Nasser, Jubran, …
[IPSN'12/15/17, ICRA'13/14/15]
Statistics
Importance Sampling, Suff. Stat
F, Shulman, Sung, Rus, …
[SODA'12, SenSys'13, GIS'12]
Machine Learning
PAC/Active learning
F, Krause, J. W. Fisher,…
[JMLR'17, NIPS'16/14/11]
To appear:
Deep Learning

Related techniques

- Sketch matrix
  - Random projections (JL Lemma, compressed sensing)
  - Usually lost sparsity of input
  - Cons: usually points on a grid
  - Pros: Support update of entries
- Sparse approximations (e.g. Frank-Wolfe)
  - Not composable coreset (does not support streaming)
- Property testing:
  - Construction takes sub-linear time
  - Binary answer (testing)

Techniques



Coresets
Lower bounds
Communication channels
Sensitivity Framework
Approx. Caratheodory Theorem
...
Epsilon-nets
Total Sensitivity
$L_2 - L_\infty$ Reduction
Uniform sample
Deterministic
Geometric Shapes
Convex Kernel functions
Small “VC-Dim”?
No
De-randomization
Weighted facilities
John Ellipsoid
Yes
No
Convex optimization
Projection methods
SVD/PCA
PAC-Learning
Reformulate Problem
Quadratic Programming
Negative Weights
Sensitivity Sum
Ellipsoid-method
+Assumptions
Frank-Wolfe
Bi-criteira approximations
Gradient Descent
On model
Pessimistic Estimators
PTAS on epsilon-net
Prior Distribution
On data
On coreset
Weak coresets

Example: k-means clustering

Arguably most common clustering technique in academy and industry

State of the art uses coreset in theory and practice

State of the art: theory and practice

Random

Coreset Size	Authors
$\left(\frac{k}{\varepsilon}\right)^{O(1)}$	F, Sohler, Schmidt,'13
$\left(\frac{dk}{\varepsilon}\right)^{O(1)}$	F, Sohler…'07
$\left(\frac{dk\log n}{\varepsilon}\right)^{O(1)}$	Ke-Chen'06

Extension	Authors
Dynamic Data	F, Gil, Rus
Weak coresets for k-Median	F, Tassa
Sketches	Indyk, Woodruf,…
Outliers Handling	F, Schulman

Deterministic

	Coreset Size	Authors
	$\left(\frac{k}{\varepsilon}\right)^{O(d)}$	Har-Peled, Kushal,'05
Coreset, 4 pages	$\left(\frac{k\log n}{\varepsilon}\right)^{O(d)}$	Har-Peled, Mazumdar,'04
Solution Set, 40 pages	$n(\log n)^{\left(\frac{dk}{\varepsilon}\right)^{O(1)}}$	Matousek,'00

$\frac{10}{\varepsilon^2}, k=1$	F, Sedat, Rus'17
$k^{O(1/\epsilon^2)}$	Barger, F, '16
$k^{O(k/\epsilon)}$	F, Sohler, Schmidt,'13

EXACT CORESETS

- Input: P in R^d (usually finite)
- Queries set: Q (possibly infinite)
- Cost function: $f: P \times Q \to [0, \infty)$

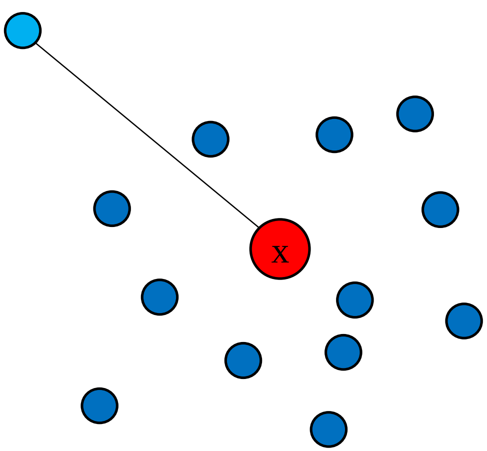
- **Exact coreset:** C is an exact coreset (usually C in P), if for every q in Q we have that the sum of the cost function on P with query q is the same as the sum of the cost function on C with query q.

$$\forall \boldsymbol{q} \in \boldsymbol{Q}: \sum_{\boldsymbol{p} \in \boldsymbol{P}} \boldsymbol{f}(\boldsymbol{p}, \boldsymbol{q}) = \sum_{\boldsymbol{c} \in \boldsymbol{C}} \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q})$$

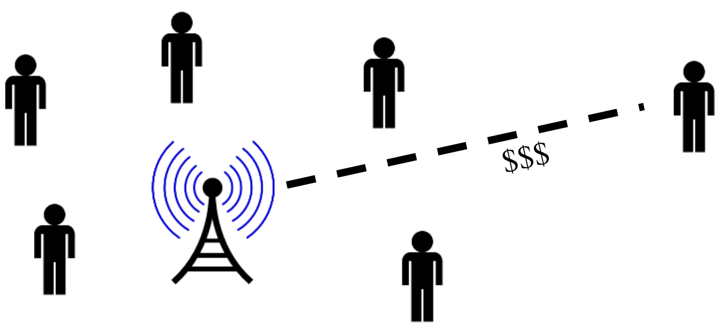
1-CENTER / MINIMUM ENCLOSING BALL

- Given a set of n points P in R^d, find the point $x \in R^d$ that minimizes:

$$far(P, x) = \max_{p \in P} \|p - x\|$$



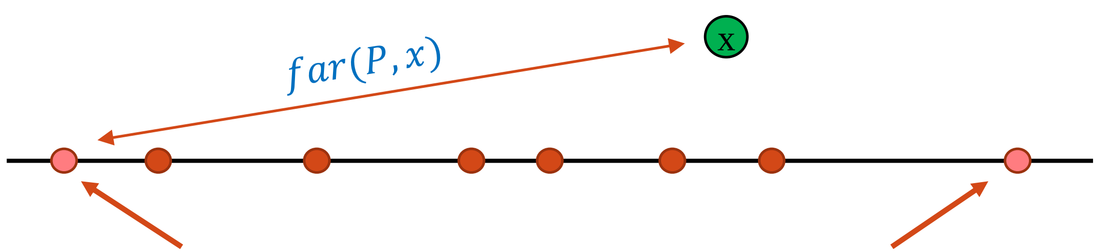
Motivation:
Where should we place an antenna if the price paid is the antenna's distance to the farthest customer?



1-CENTER QUERIES

- Input: $P = \{p_1, p_2, \dots, p_n\}$ in R^d
- Query: a point $x \in R^d$
- Result: $\boldsymbol{far(P,x) = \max_{p \in P} \|p - x\|^2}$

Exact coreset for $\boldsymbol{1\text{-}center}$ queries when $\boldsymbol{P}$ in $\boldsymbol{R}$ and $\boldsymbol{x \in R^d}$:



The farthest point from every query $\boldsymbol{x \in R^d}$ is one of the edge points!

1-MEAN QUERIES

- Input: P in R^d
- Query: a point $x \in R^d$
- Cost: $\boldsymbol{dist^2(P, x) = \sum_{p \in P} \|p - x\|^2}$

1-MEAN QUERIES

- Input: P in R^d
- Query: a point $x \in R^d$
- Cost: $\boldsymbol{dist^2(P,x)} = \sum_{\boldsymbol{p \in P}} \|\boldsymbol{p} - \boldsymbol{x}\|^2$

Exact coreset for 1-*mean* queries using 3 first moments:

$$\sum_{\boldsymbol{p\in P}} \|\boldsymbol{p} - \boldsymbol{x}\|^2 = \sum_{p\in P} (\|p\|^2 + \|x\|^2 - 2p^T x) = \sum_{p \in P} \|p\|^2 + \sum_{p\in P} \|x\|^2 - 2\sum_{p\in P} p^T x$$

$$= \sum_{p\in P} \|p\|^2 + \boldsymbol{n} \cdot \|\boldsymbol{x}\|^2 - \boldsymbol{2}\left(\sum_{p\in P} p^T\right)\boldsymbol{x}$$

Store those in memory!

1-MEAN QUERIES

Solution #3:

$$\sum_{\boldsymbol{p}\in \boldsymbol{P}} \|\boldsymbol{p}-\boldsymbol{x}\|^{2} = \sum_{p\in P} \|p\|^{2} + \boldsymbol{n} * \|\boldsymbol{x}\|^{2} - \mathbf{2}\left(\sum_{p\in P} p^{T}\right)\boldsymbol{x}$$

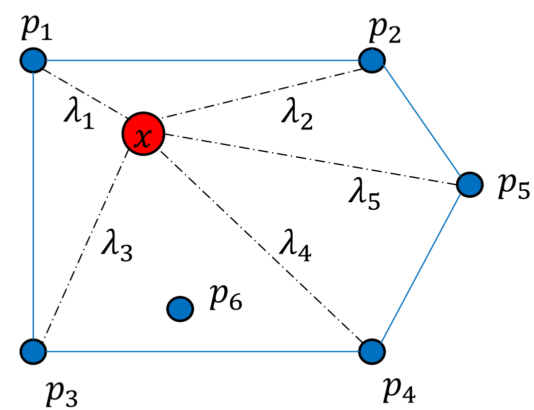
1) Build new vectors in R^{d+2}:

$$\boldsymbol{p}_{\boldsymbol{i}}' = \begin{pmatrix} \boldsymbol{p}_{\boldsymbol{i}} \\ \|\boldsymbol{p}_{\boldsymbol{i}}\|^{2} \\ \mathbf{1} \end{pmatrix}$$

2) Use $Caratheodory's\ theorem$ to compute a weighted subset C' of the vectors that has the same 3 $first\ moments$.

CONVEX COMBINATION

- A *convex combination* is a linear combination of points where all coefficients are non-negative and sum to 1.
- A *convex region* is a region where, for every pair of points within the region, every point on the straight line segment that joins the pair of points is also within the region.
- A *convex hull* of a set P is is the smallest convex set that contains P.
- Every point x in a convex hull of a set of points P can be written as a convex combination of a finite number of points in P.



$$x = \sum_{i=1}^{5} \lambda_i p_i \,,$$

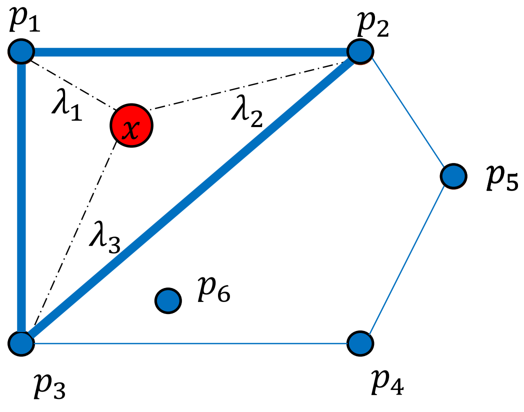
$$\lambda_i \geq 0, \sum_{i=1}^{5} \lambda_i = 1$$

CARATHEODORY'S THEOREM

- "If a point $x \in R^d$ lies in the convex hull of a set P, there is a subset P' of P consisting of $d + 1$ or fewer points such that x lies in the convex hull of P'."

$$x = \sum_{i=1}^{3} \lambda_i p_i \,,$$

$$\lambda_i \geq 0, \sum_{i=1}^{3} \lambda_i = 1$$



CARATHEODORY'S THEOREM - INTUITION

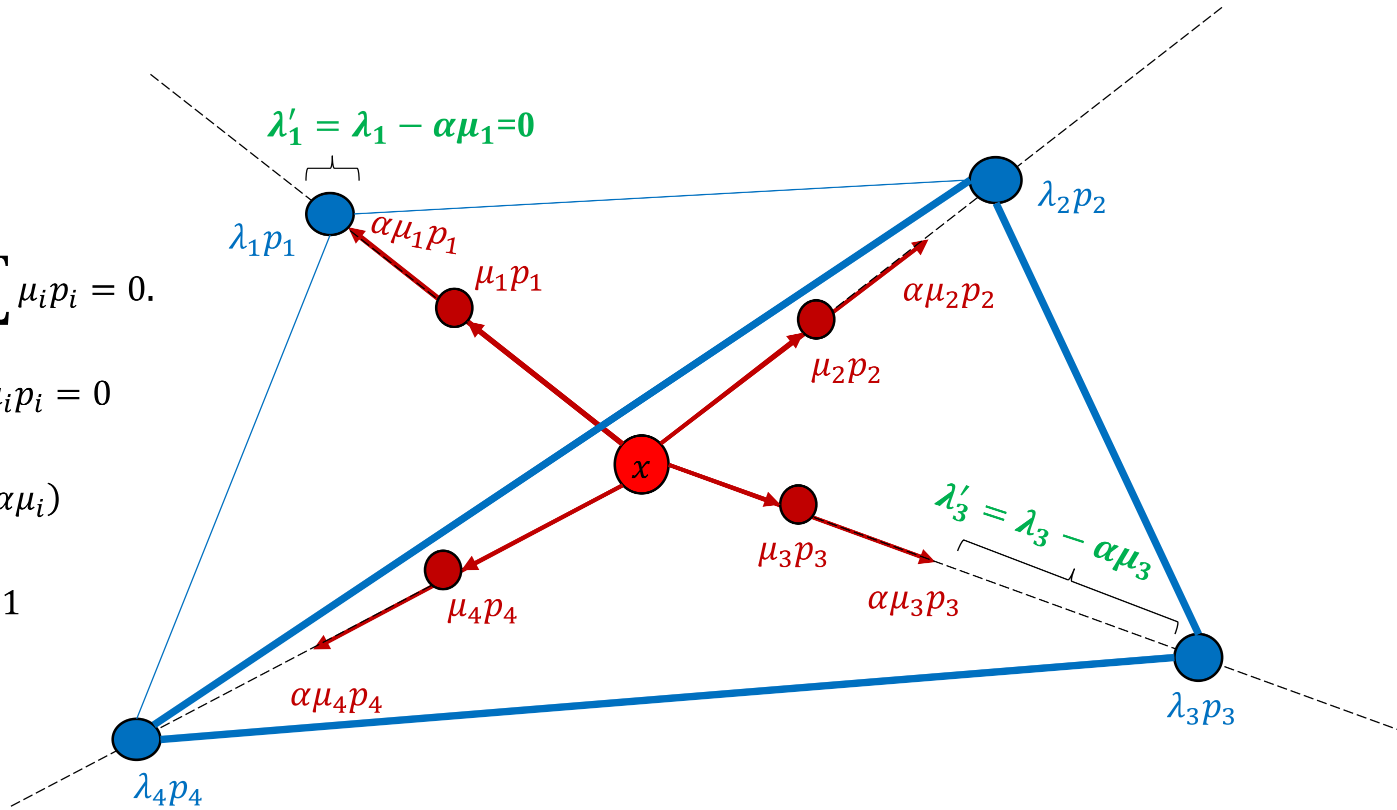
Assume that x is the origin.

$$\sum \lambda_i = 1, \sum \mu_i = 0, \sum \mu_i p_i = 0.$$

$$\rightarrow \sum \alpha\mu_i p_i = \alpha \sum \mu_i p_i = 0$$

$$\rightarrow \sum \lambda_i{}' = \sum (\lambda_i - \alpha\mu_i)$$

$$= \sum \lambda_i - \alpha \sum \mu_i = 1$$



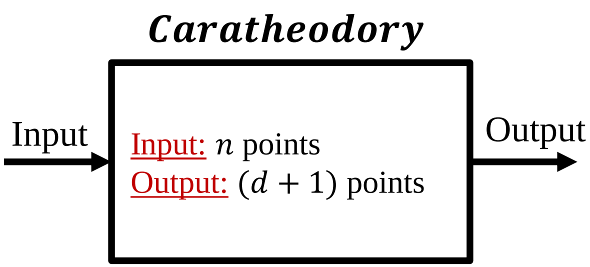
1-MEAN QUERIES WITH INSERTIONS

- Input: P in R^d - inserted one at a time!
- Query: a point $x \in R^d$
- Result: $\boldsymbol{dist}(\boldsymbol{P}, \boldsymbol{x}) = \sum_{\boldsymbol{p} \in \boldsymbol{P}} \|\boldsymbol{p} - \boldsymbol{x}\|^2$

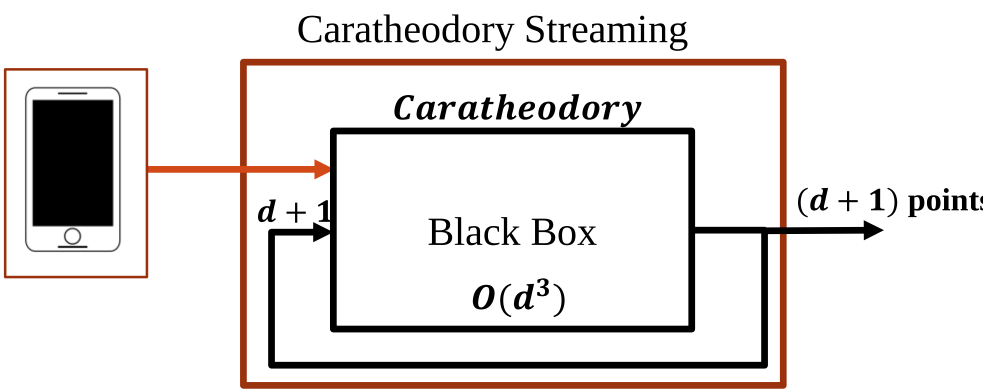
Solution:

$Caratheodory's\ Theorem$ – **the streaming version**.

STREAMING USING CORESETS



$$O\Big(\underbrace{(n-d-1)}_{\#iterations}\underbrace{nd^2}_{Each\ iteration}\Big) = O(n^2d^2)$$



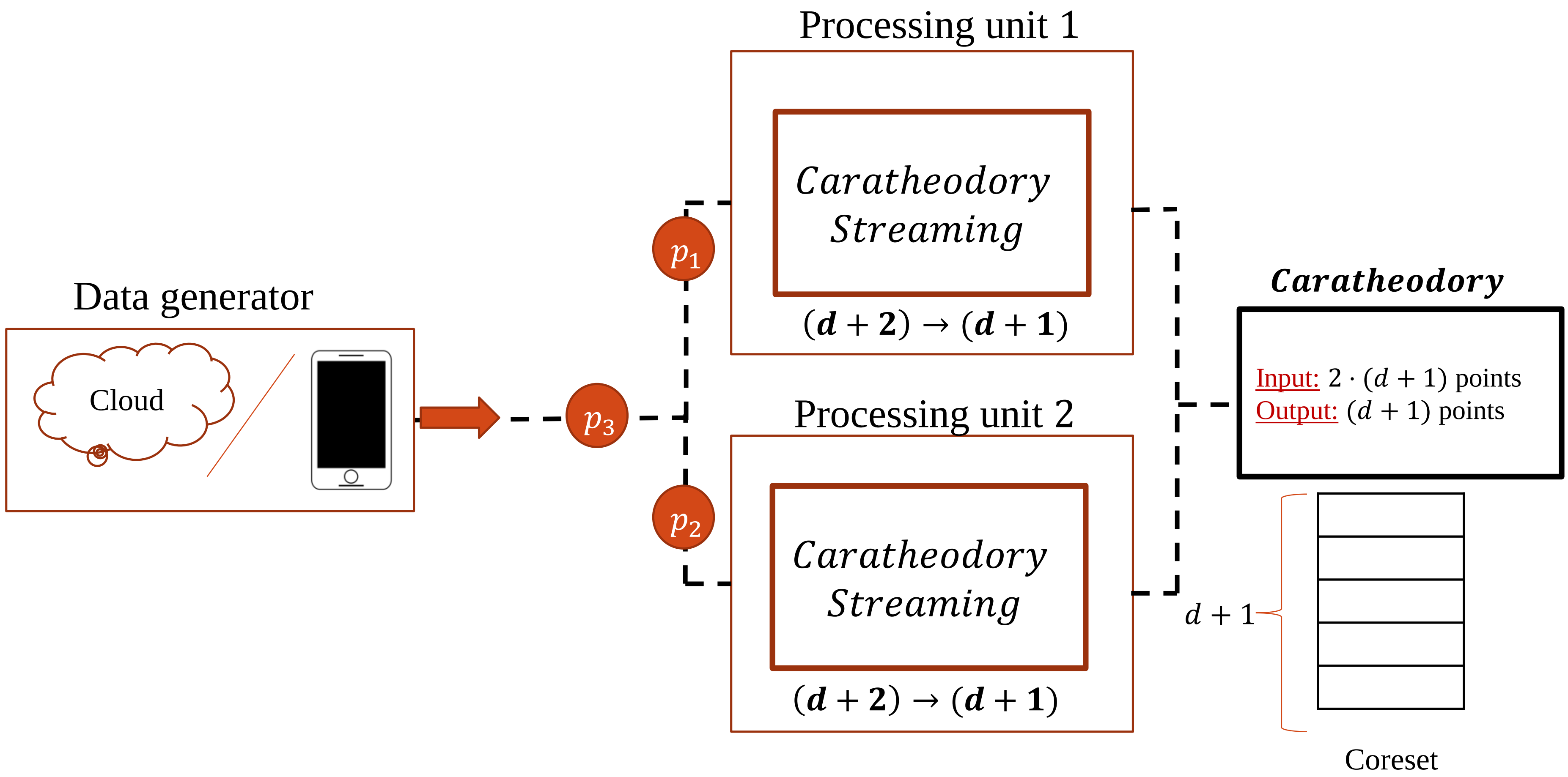
$$O\big(\underbrace{(n-d-1)}_{} * \underbrace{d^3}_{}\big) = O(nd^3)$$

At each iteration we reduce number of points from $k \to k-1$
start: $k = n$, finish: $k = d+1$

To compute the linear combination

- Step 1: Read new point p.
- Step 2: Compress using black box.
- Step 3: Repeat.

STREAMING + DISTRIBUTED (CLOUD, IOT)



CORESET FOR PCA/SVD

- Input: $A = \begin{bmatrix} -a_1- \\ \vdots \\ -a_n- \end{bmatrix} \in R^{n\times d}$ (n points in R^d)

- Query space: $S = \{x | x \in R^d\}$ (Hyperplanes in R^d)

- Output: $f(A,x) = ||Ax||^2$

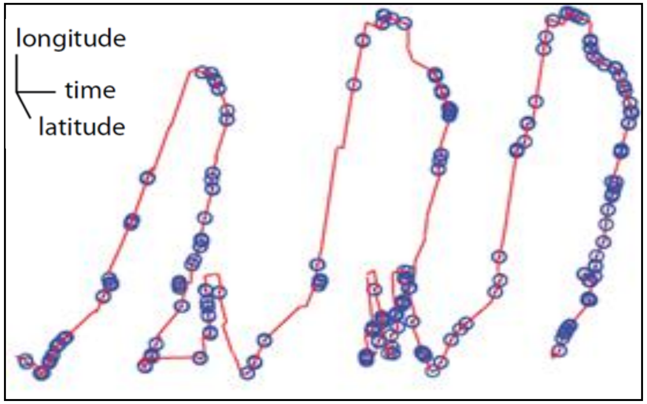
SVD IS A CORESET FOR SVD

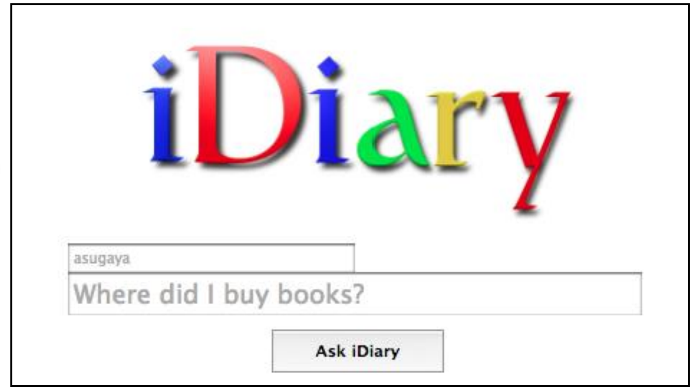
- Factorization $\boldsymbol{A} = \boldsymbol{QR}$ where $\boldsymbol{Q}^T\boldsymbol{Q} = I$,
- $\boldsymbol{Q} \in R^{n\times d}$, $\boldsymbol{R} \in R^{n\times d}$, $A = U\,DV^T = \;Q\;R$
- For every $x \in S$ it holds that:

$$\boldsymbol{f}(\boldsymbol{A},\boldsymbol{x}) = \|\boldsymbol{Ax}\|^2 = \|\boldsymbol{QRx}\|^2 = \|\boldsymbol{Rx}\|^2 = \boldsymbol{f}(\boldsymbol{R},\boldsymbol{x})$$

$$A = QR \qquad Q^TQ = I$$

- $\forall x \in S\colon \|Ax\|^2 = \|Rx\|^2$





Co-authors:
Privacy: E. Zhang
Server Code: C. Sung
Smartphone Code: M. Vo-Thanh

Text mining: Rishabh Kabra
Web-site: A. Sugaya
GPUs: Micha Feigin

Robots code: S. Gil
Kuka Robots: R. Knepper
Quadrobots: B. Julian

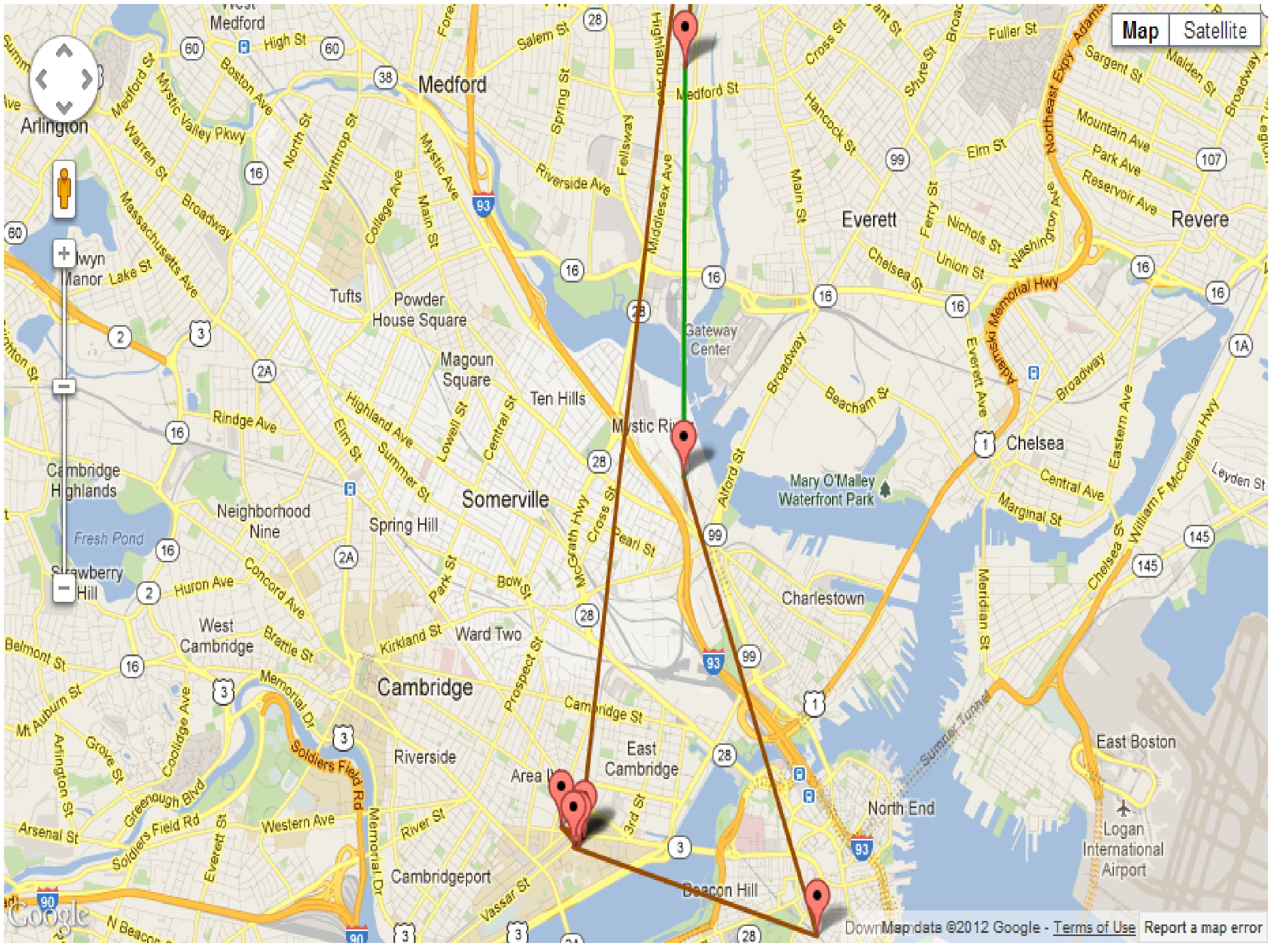
- First text search application on GPS data
- Other GPS managers:
  - Foursquare: manual check-ins
  - Google Latitude: no text search

iDiary

Search my history | Get suggestions

Welcome, David
Logout

Location resolution:



<< | May 15, 2012 | >>

- You left home at 9:17 AM.
- You arrived at Maiden Center Station at 9:26 AM, after traveling by foot for 9 minutes
- You arrived at Kendall Station at 9:52 AM, after traveling by public transportation for 26 minutes
- You arrived at work at 9:57 AM, after traveling by foot for 5 minutes.
- You stayed at work for 3 hours, leaving at 1:03 PM.
- You arrived at Quiznos for lunch at 1:09 PM, after traveling by foot for 6 minutes.
- You stayed at Quiznos for 27 minutes, leaving at 1:36 PM.
- You arrived at work at 1:43 PM, after traveling by foot for 7 minutes.
- You stayed at work for 5 hours, leaving at

Diary | Summery | Search | Friends

iDiary

restaurants July 11 | Search my history | Get suggestions

Welcome, Gary
Logout

Restaurants you visited on July 11th, 2012

1. ## Anna's Taqueria

   You were here on July 11th from 7:03 PM to 7:31 PM, with John Smith, Foo Bar, and 3 others.

   You have been here 142 other times.

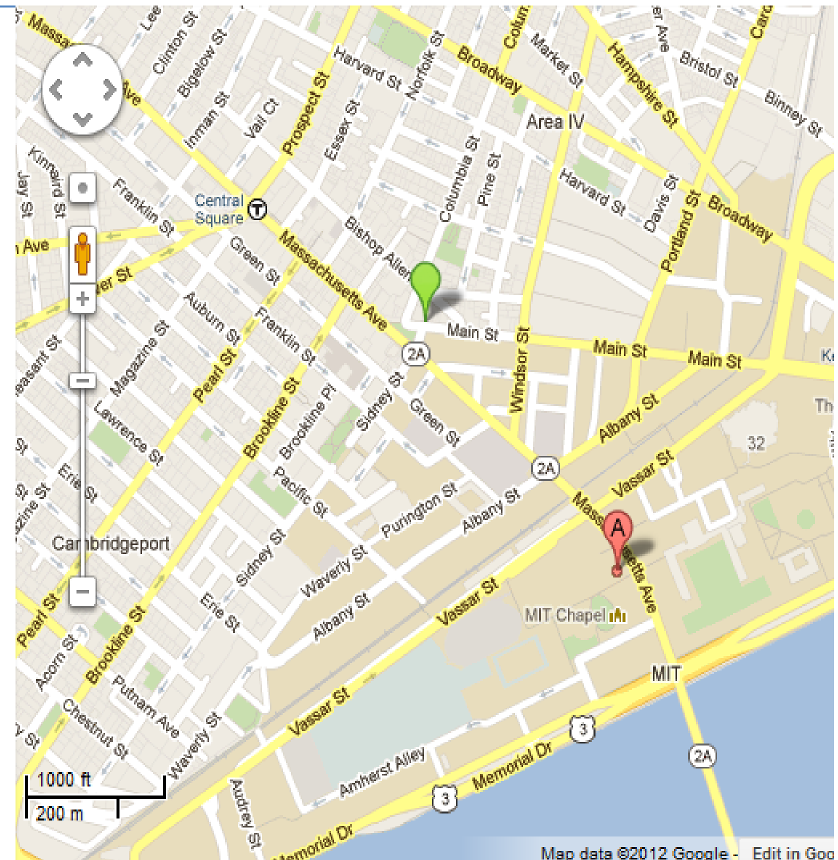
   View similar restaurants

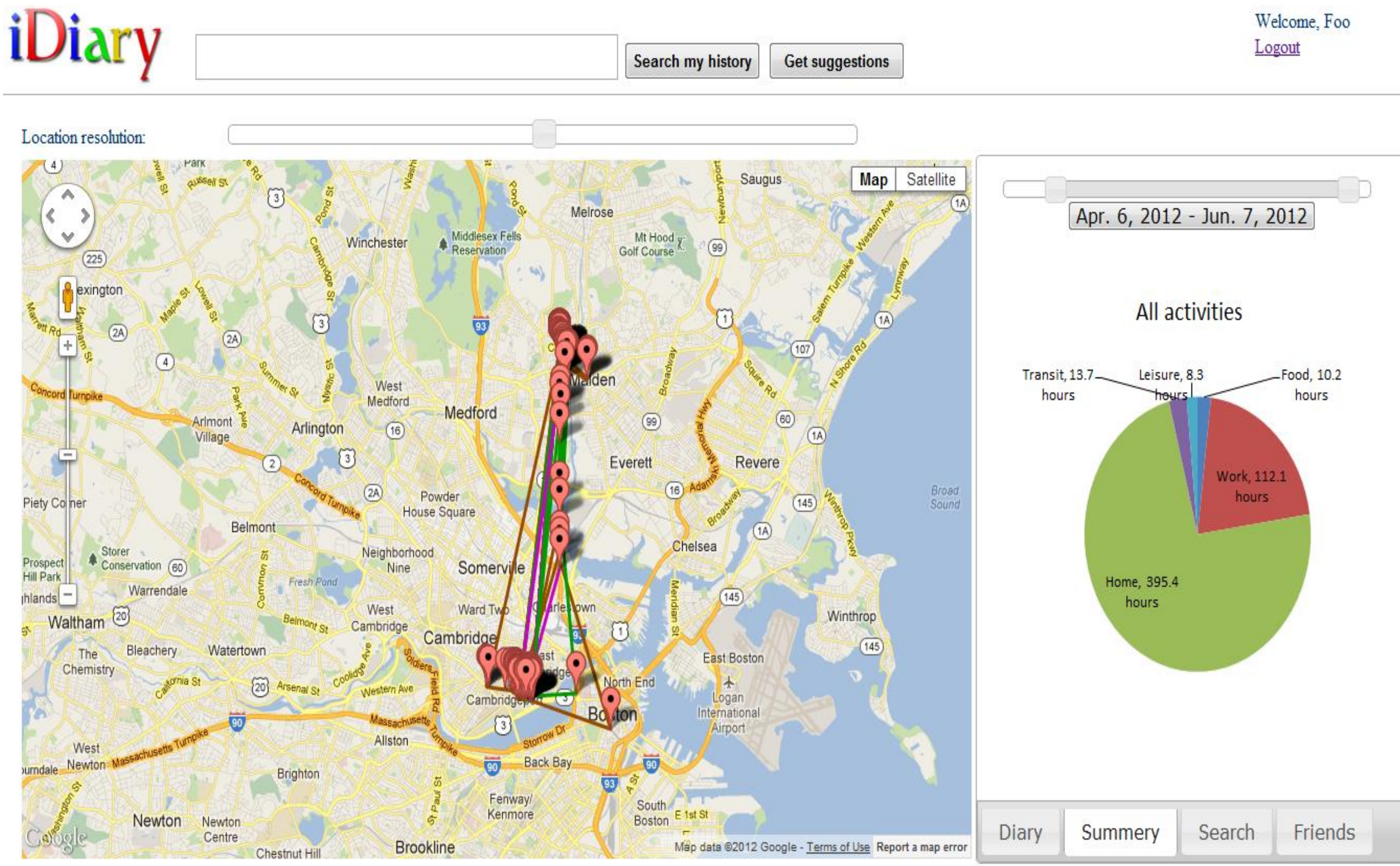
2. ## Toscanini's Ice Cream

   You were here on July 11th from 7:44 PM to 7:58 PM, with Tim Yang, John Smith, and 4 others.

   You have been here 17 other times.

   View similar restaurants





iDiary
Search my history
Get suggestions
Welcome, Foo
Logout
Location resolution:
Map
Satellite
Apr. 6, 2012 - Jun. 7, 2012
All activities
Transit, 13.7 hours
Leisure, 8.3 hours
Food, 10.2 hours
Work, 112.1 hours
Home, 395.4 hours
Diary
Summery
Search
Friends
Saugus
Melrose
Winchester
Middlesex Fells Reservation
Mt Hood Golf Course
Lexington
West Medford
Medford
Malden
Arlington
Arlmont Village
Everett
Revere
Powder House Square
Belmont
Chelsea
Neighborhood Nine
Somerville
Storer Conservation
Prospect Hill Park
Warrendale
Waltham
West Cambridge
Ward Two
Cambridge
Winthrop
Watertown
Bleachery
The Chemistry
East Boston
North End
Logan International Airport
Boston
Allston
Back Bay
West Newton
Brighton
Fenway/ Kenmore
South Boston
Newton
Newton Centre
Brookline
Chestnut Hill
Broad Sound
Map data ©2012 Google - Terms of Use
Report a map error

iDiary

user:timyang

Search my history

Get suggestions

Welcome, Gary

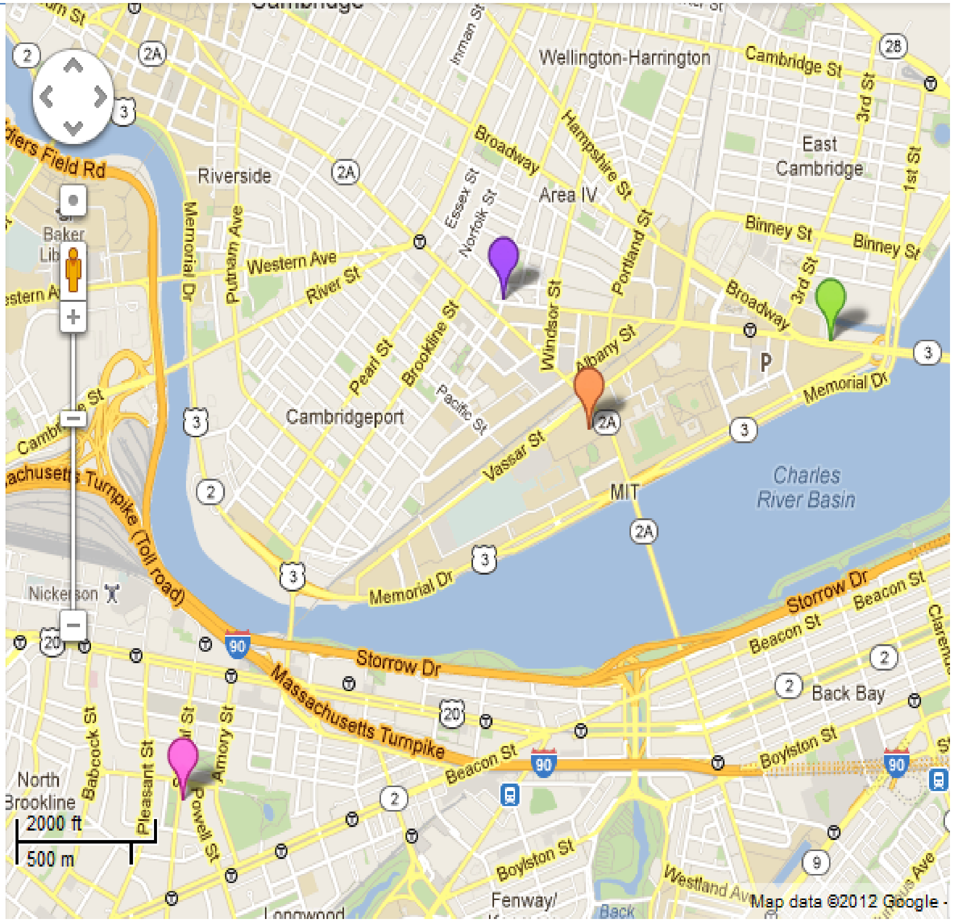
Logout

You and Tim Yang

Recent encounters:

1. Toscanini's Ice Cream – July 11th
2. Home – July 7th
3. Home – July 1st
4. MIT Student Center – June 25th
5. 123 Main Street – June 24th
6. Home – June 23rd

See more…



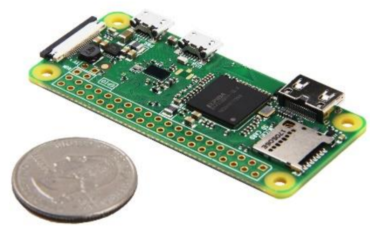
Time analysis

1. Home – 5 hours/week
   - Weekend evenings
2. Leisure – 3 hours/week
   - Weekday afternoons

GPS tracking using coreset

Data collection | Data reduction | Sent results to cloud

GPS Hat for Raspberry Pi



Raspberry Pi Zero W
IoT Device
•1GHz, Single-core CPU.
•512MB RAM

T2.micro AWS instance
Collecting data
No additional computational power needed

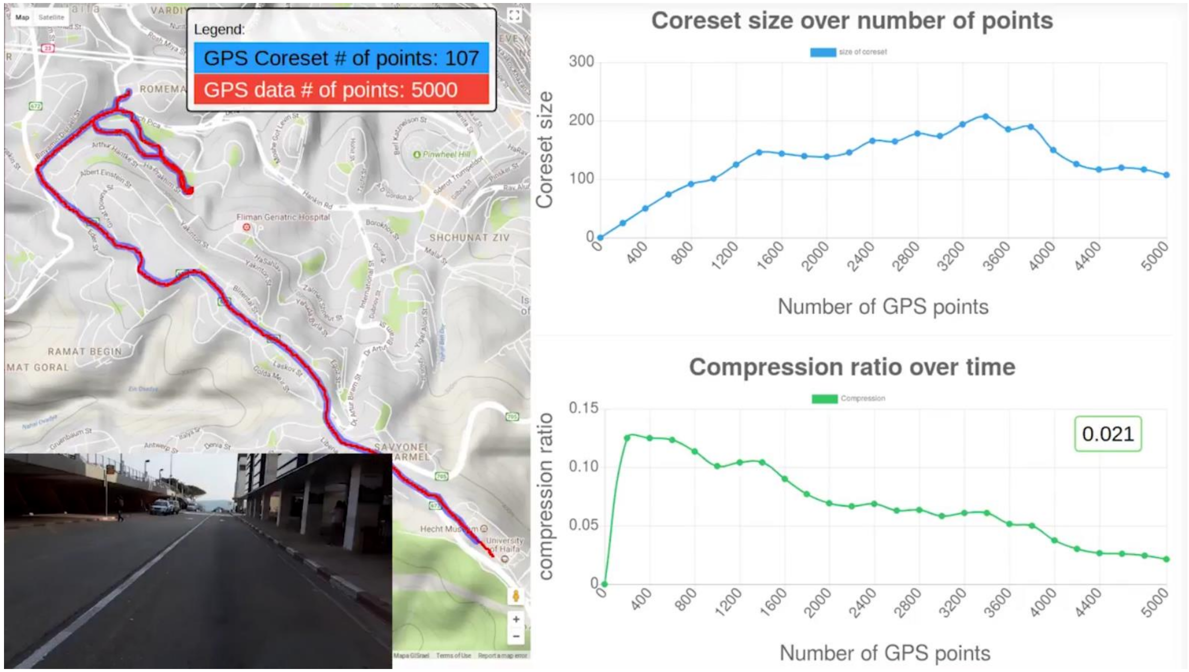
GPS core-set

Results:

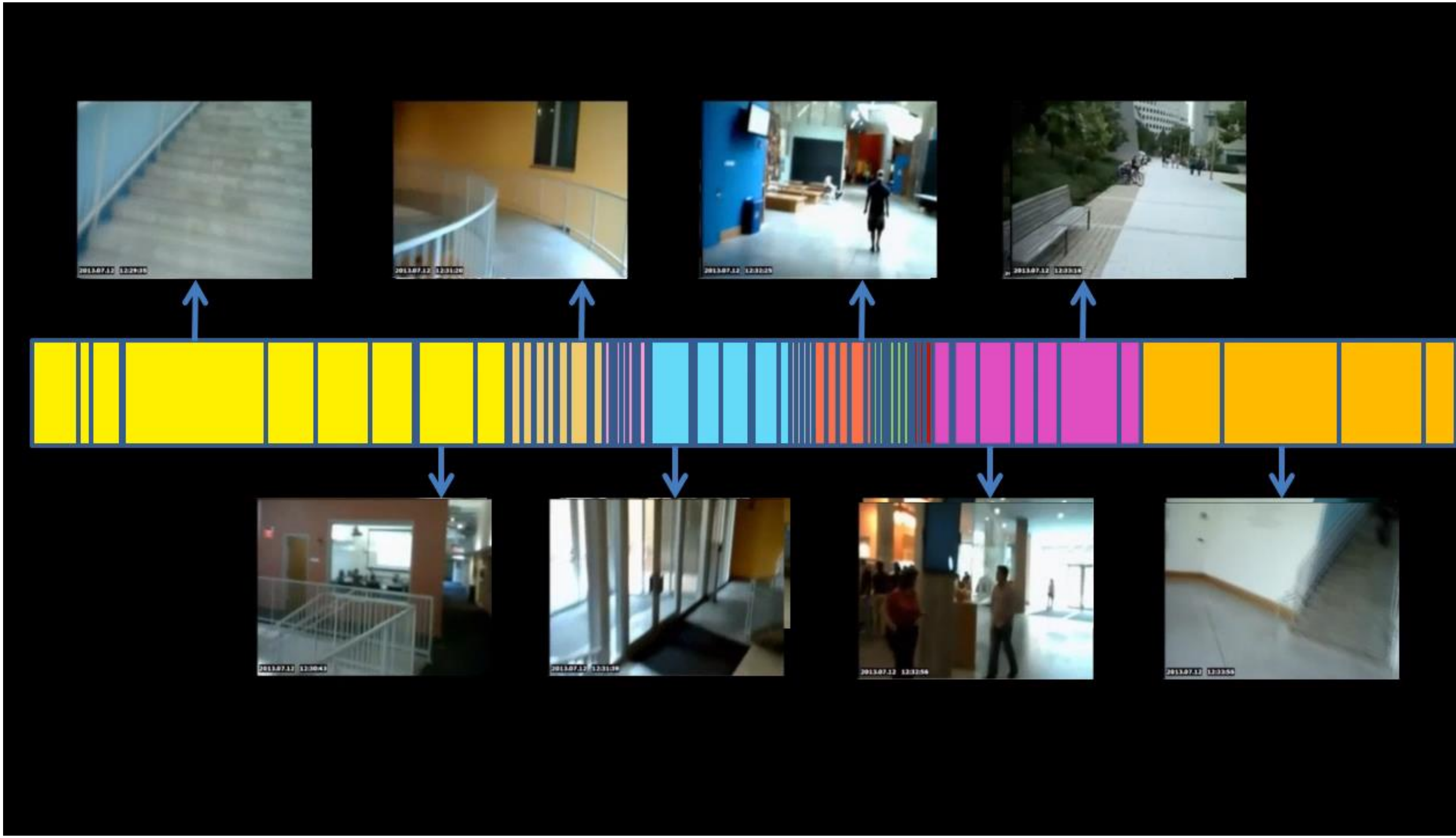
Original data size: 5000

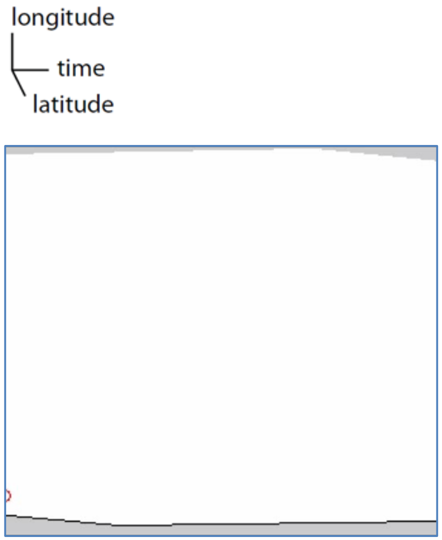
Compressed with our technique data size: **107**

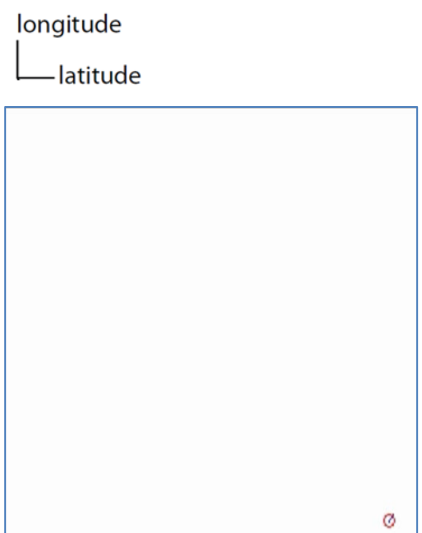
Compression ratio: 0.021 (0.02% from original)

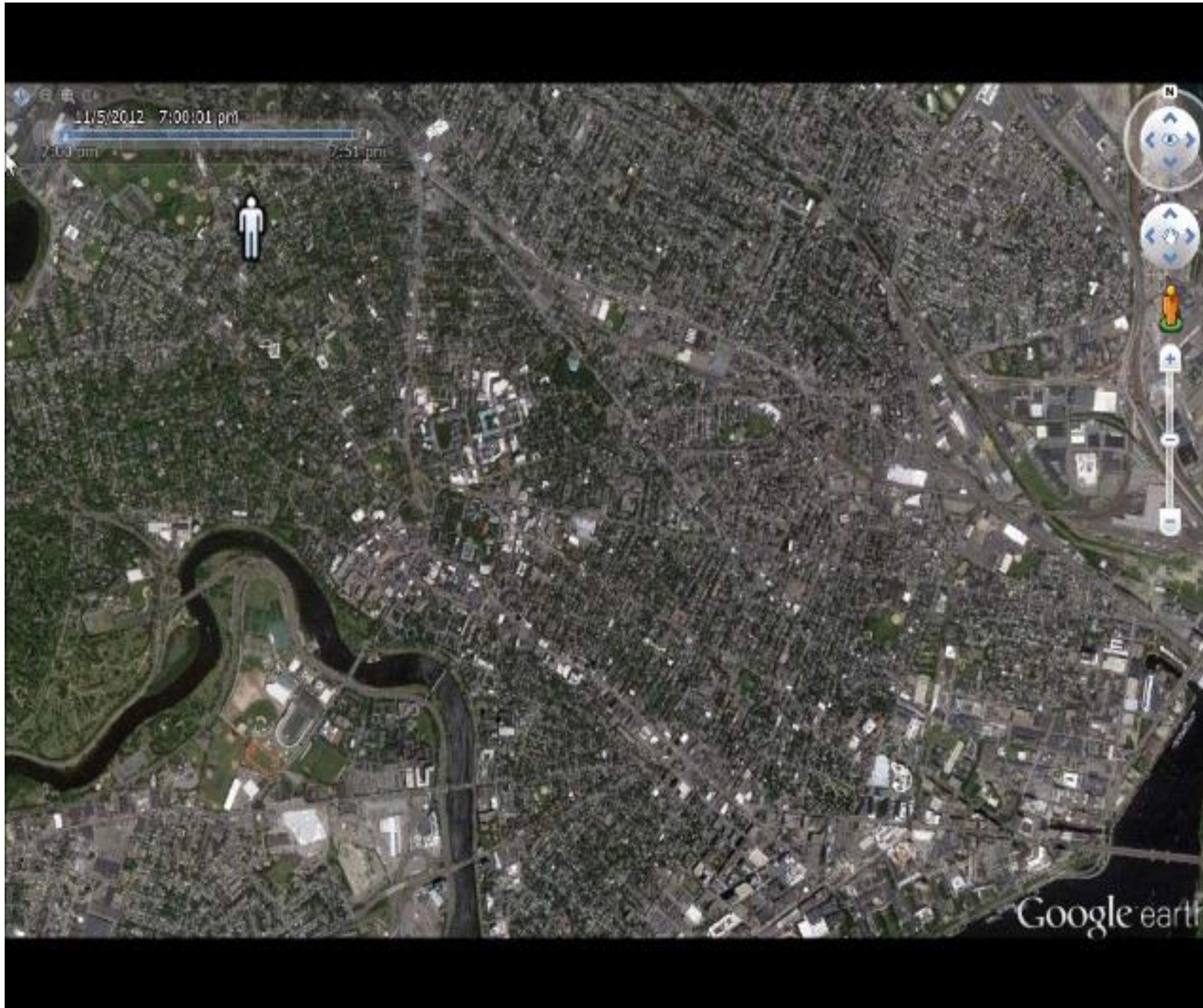


Map
Satellite
speed x20
Legend:
GPS Coreset # of points: 25
GPS data # of points: 200
Coreset size over number of points
size of coreset
Coreset size
30
20
10
0
0
200
Number of GPS points
Compression ratio over time
Compression
compression ratio
0.15
0.10
0.05
0
0
200
0.125
Number of GPS points

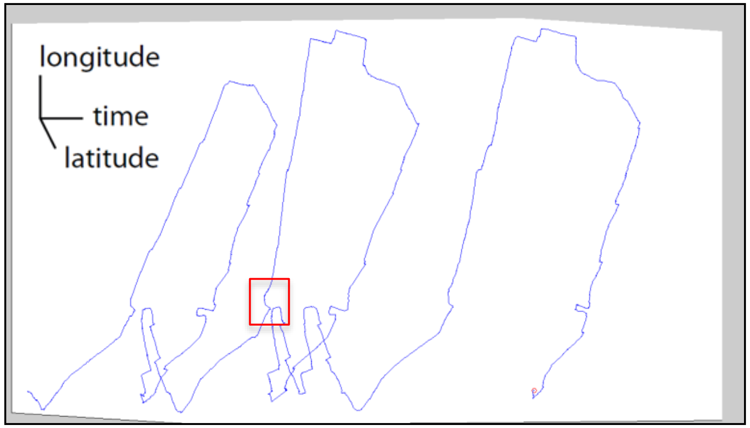


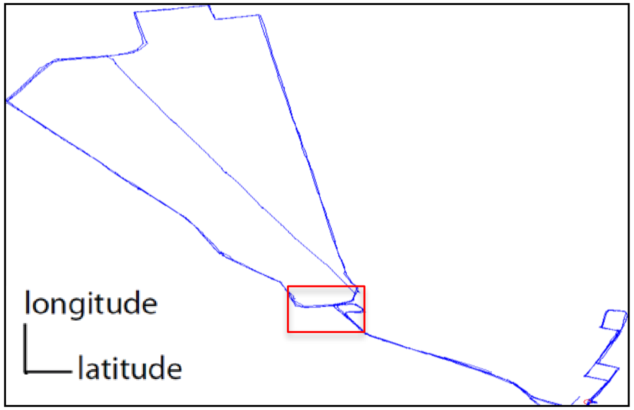

longitude
time
latitude

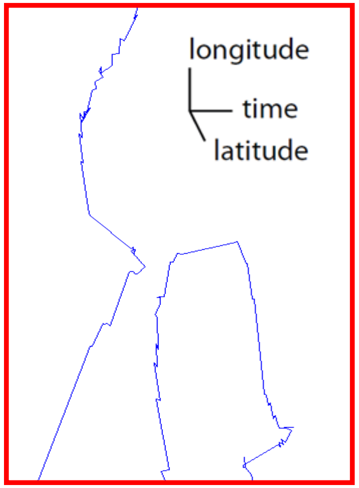

longitude
latitude


11/5/2012 7:00:01 pm
7:51 pm
Google earth

Big Data – Big Noise



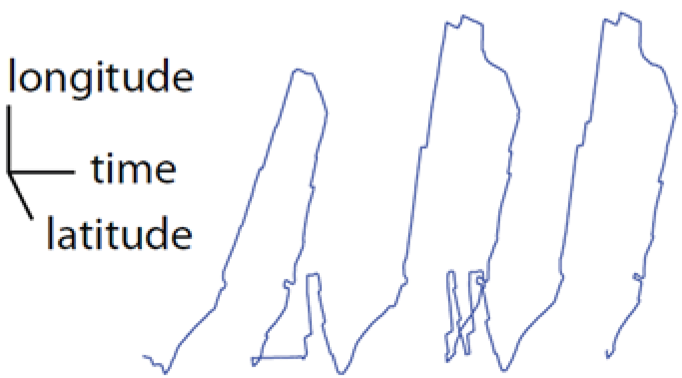


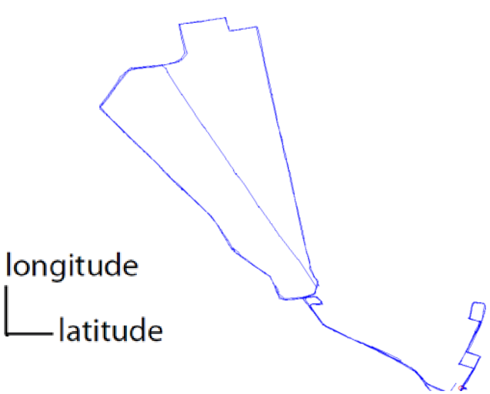


Input

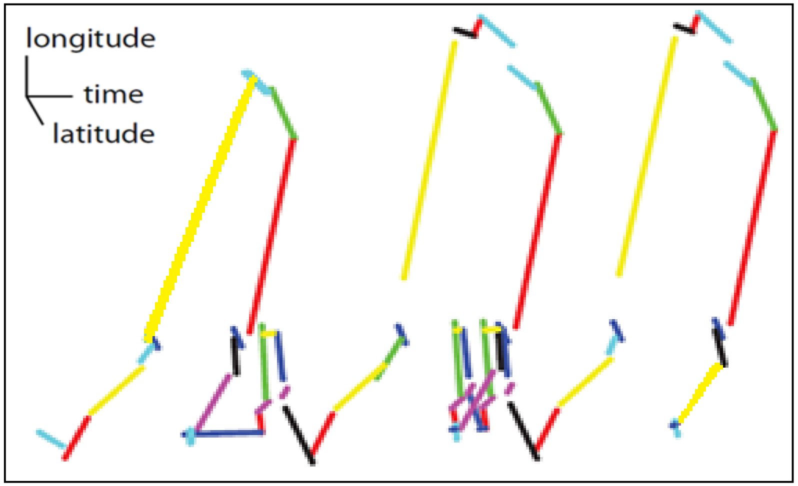
- GPS-point = (latitude, longitude, time)

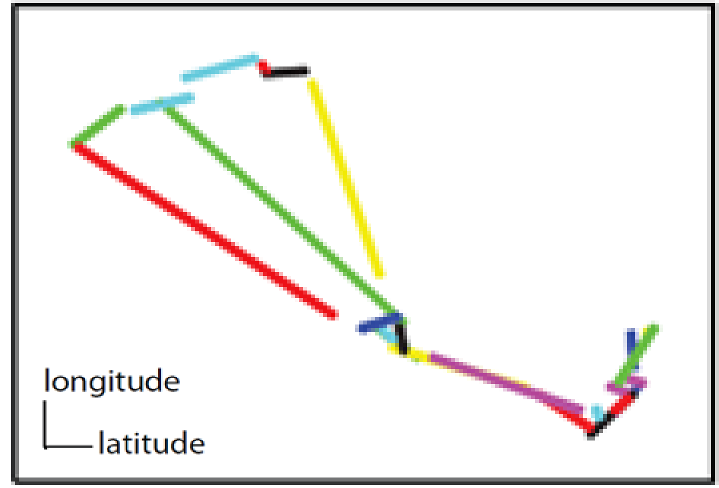
latitude	longitude	time
1.295783	103.7816	8:44:57
1.295785	103.7816	8:44:59
1.295782	103.7816	8:45:00
1.295782	103.7816	8:45:01
1.29579	103.7817	8:45:04
1.295802	103.7817	8:45:05
1.295915	103.7818	8:45:08
1.29598	103.7819	8:45:09
1.296015	103.7819	8:45:10
1.296057	103.782	8:45:11
...	...	...





Output





k trajectories

Begin time	End time	Location ID	Speed
8:44:57	8:48:57	c	30
8:49:59	8:51:59	d	24
8:52:00	8:54:00	g	24
8:54:01	8:55:01	q	11
8:56:57	8:57:57	r	120
8:58:57	8:59:57	m	55
…	…	…	65

m locations

Location ID	Begin Point	End Point
a	(42.374,-71.120)	(42.374,-71.120)
b	(42.386, -71.130)	(42.386,-71.130)
c	(42.391,-71.128)	(42.391,-71.128)
d	(42.393,-71.130)	(42.394,-71.129)
…	…	…

From Semantic Database Text Mining

m locations

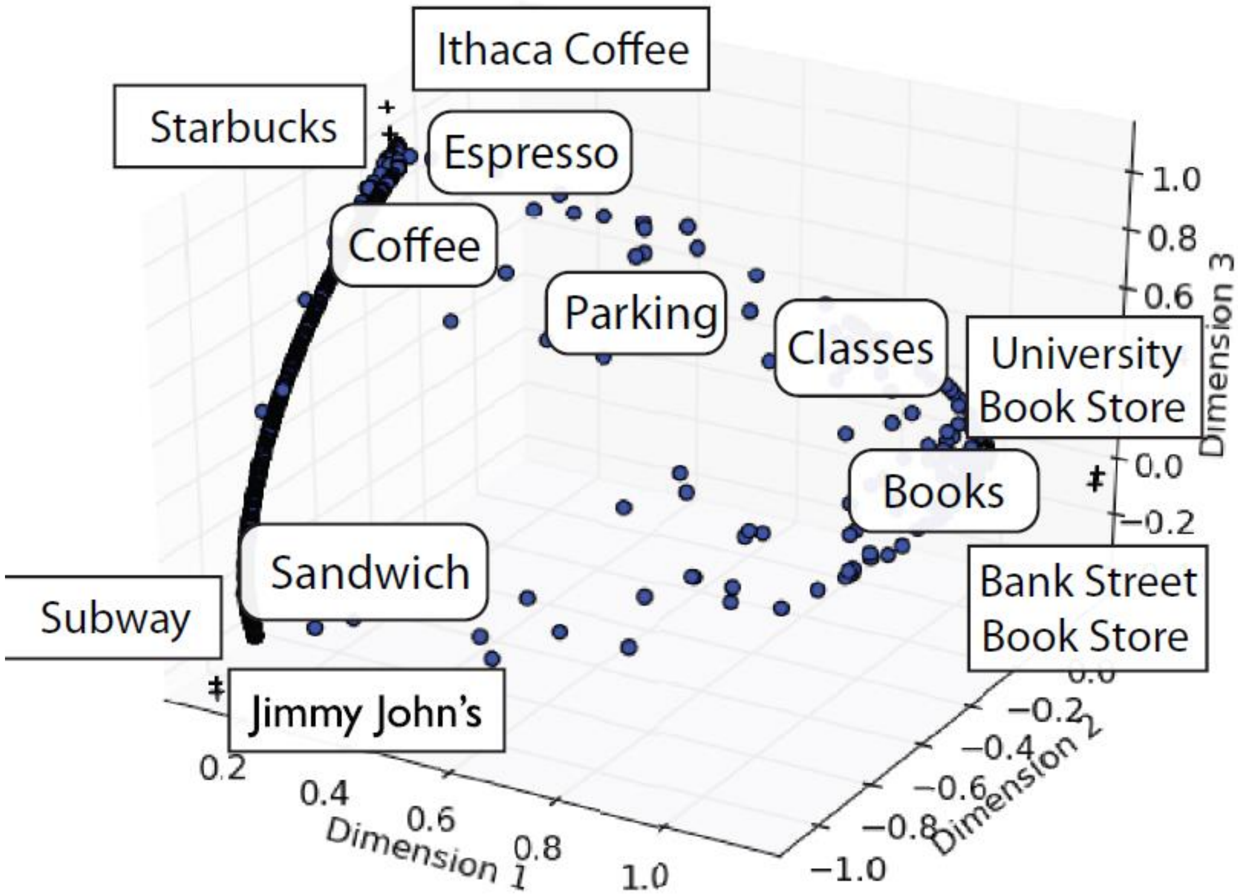
Location ID	Begin Point	End Point
	(42.374,-71.120)	(42.374,-71.120)
2	(42.386, -71.130)	(42.386,-71.130)
3	(42.391,-71.128)	(42.391,-71.128)
4	(42.393,-71.130)	(42.394,-71.129)
5	(42.385,-71.132)	(42.384,-71.130)
6	(42.358, -71.091)	(42.358,-71.098)
…	…	…

Google maps, mapQuest …

Reverse Geo-coding

Starbucks, Harvard Square
225 Walden St.
Peabody Preschool
55-92 Rice St.
128-157 Garden St.
130-169 Vassar St, Cambridge
…

Latent Semantic Analysis (PCA) on Yelp reviews [SODA'13, with Sohler and Schmit]



Now we can use traditional algorithms…

- String Compression (e.g. zip files)
- Data mining (decision tree, k-means, NN)
- Motion Prediction (e.g. to save Battery life)
- Social Network analysis on User/Locations matrix

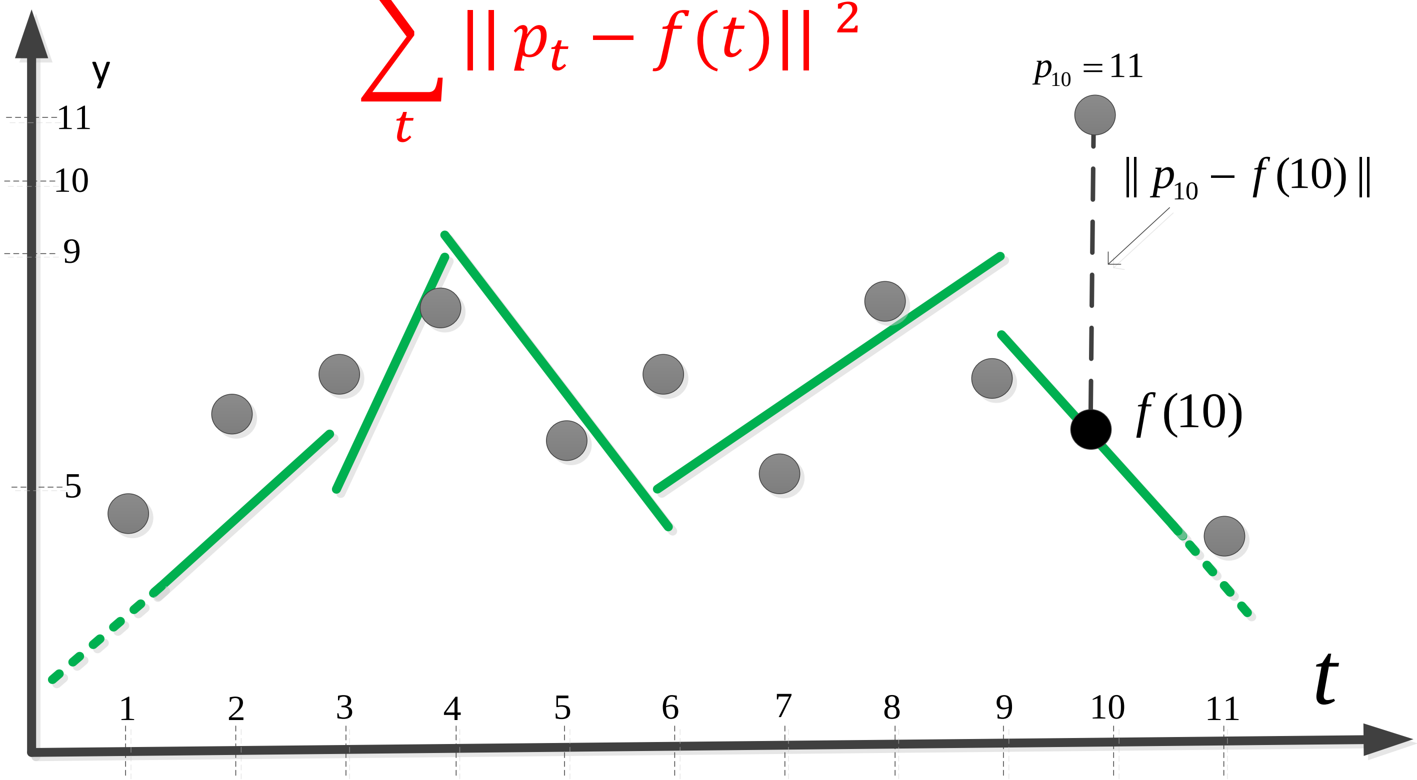
	a	b	c	d	e	f
1	0	1	1	0	0	0
2	1	0	0	1	0	0
3	1	0	0	1	0	0
4	0	1	1	0	1	0
5	0	0	0	1	0	1
6	0	0	0	0	1	0

Adjacency Matrix

k-Segment mean

The k-segment f^* that minimizes the fitting cost from points to a d-dimensional signal

$$\sum_t \|p_t - f(t)\|^2$$



$k = 5$

$d = 1$

Related Work

Provable Guarantee:

- Exact solution in $O(n^2 k)$ time and $O(n^2)$ space [Bellman'68]
- For monotonic sequences [Abam, De-berg, Hachenberg, 2010]

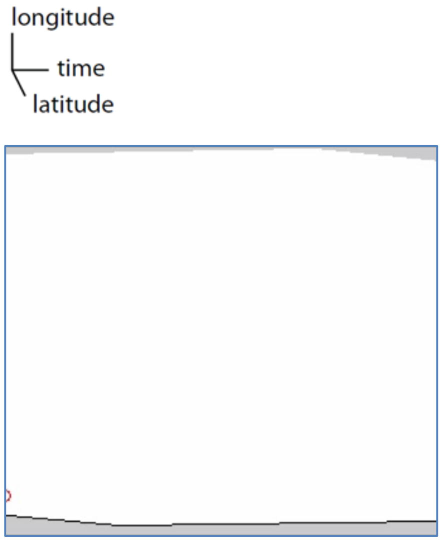
Numerous heuristics:

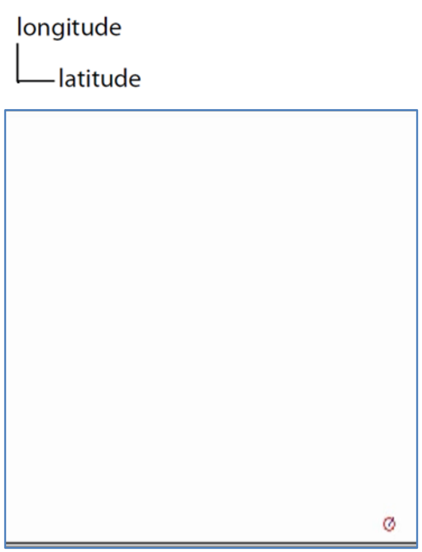
- Off-line [Douglas, Peucker'73, Kaminka et al.'10]
- Streaming [Cao, O. Wolfson, and G. Trajcevski.]
- In Matlab, Oracle, …

Theorem [with Sung &Rus, GIS'12]

A $(1+\epsilon)$ approximation to the k-segment mean w.h.p. in the big data computation model

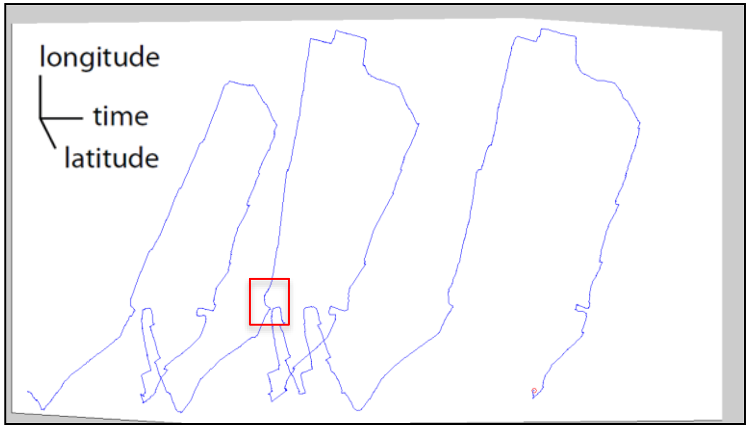
- $\sim\log n$ memory
- M processors
- *~log (n)/M* insertion time per point

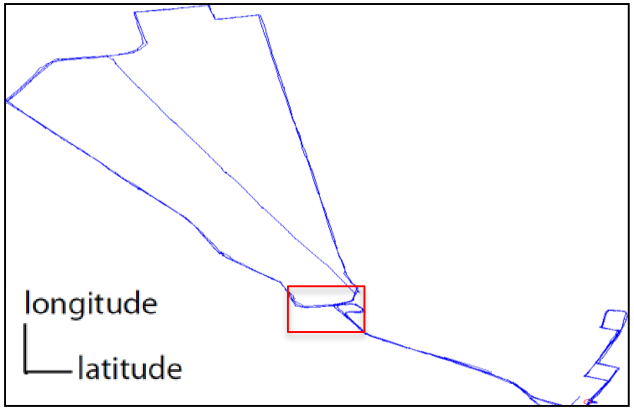

longitude
time
latitude

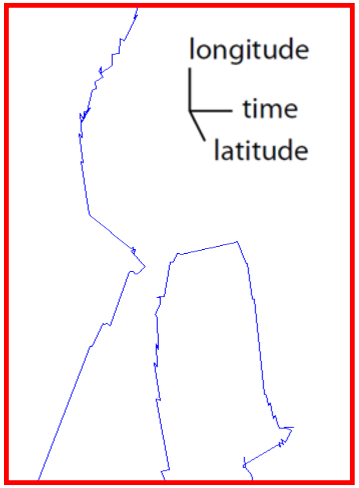

longitude
latitude


11/5/2012 7:00:01 pm
Google earth

Big Data — Big Noise

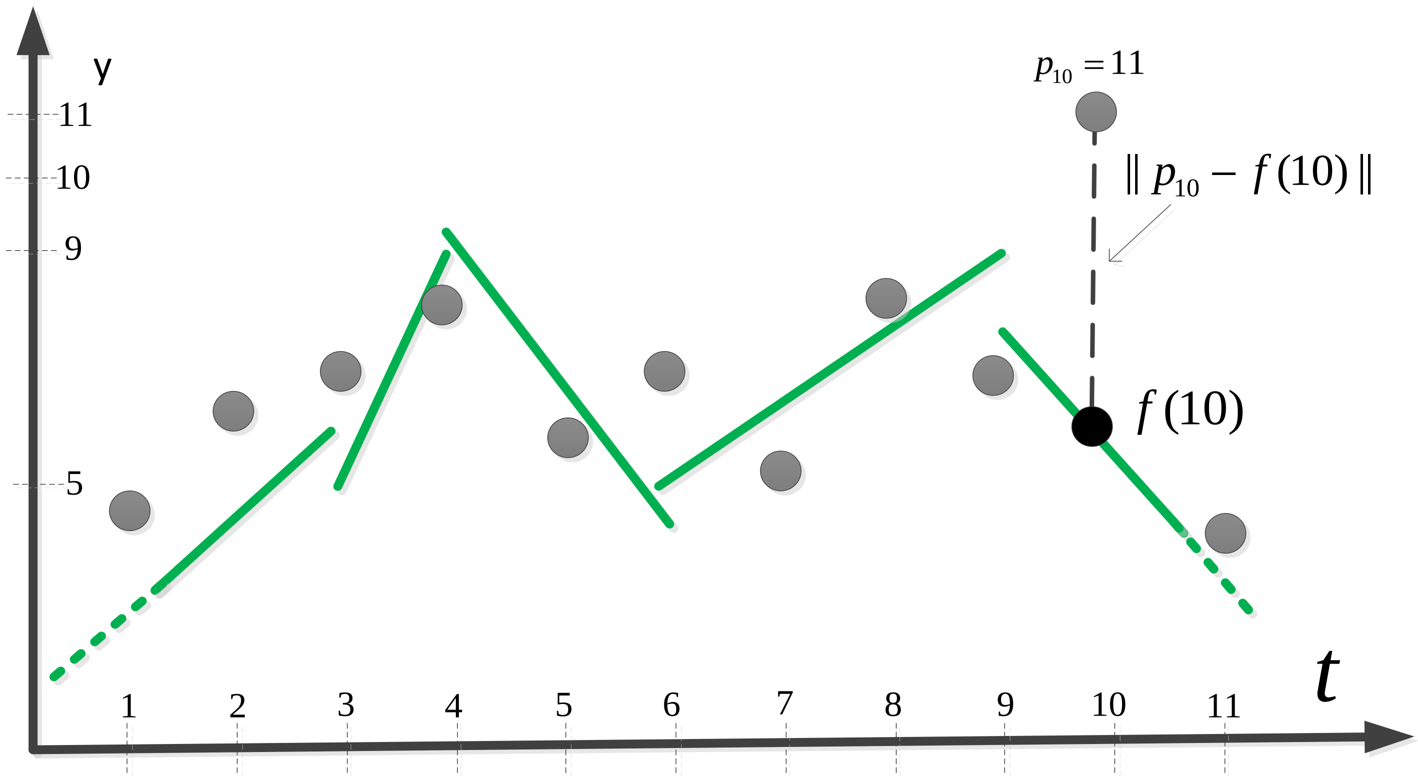






k-Segment mean

The k-piecewise linear function f^* that minimizes the fitting cost from points to a d-dimensional signal



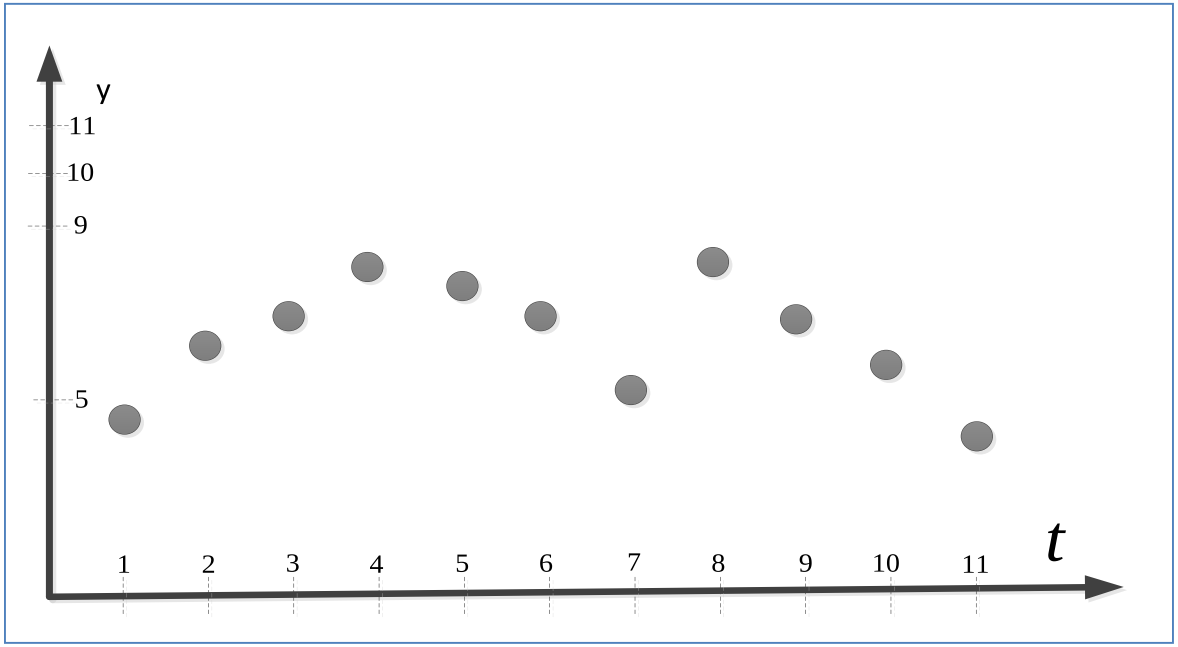
$k = 5$

$d = 1$

$$cost(P, f) = \sum_t || p_t - f(t) ||^2$$

k – Segment Queries

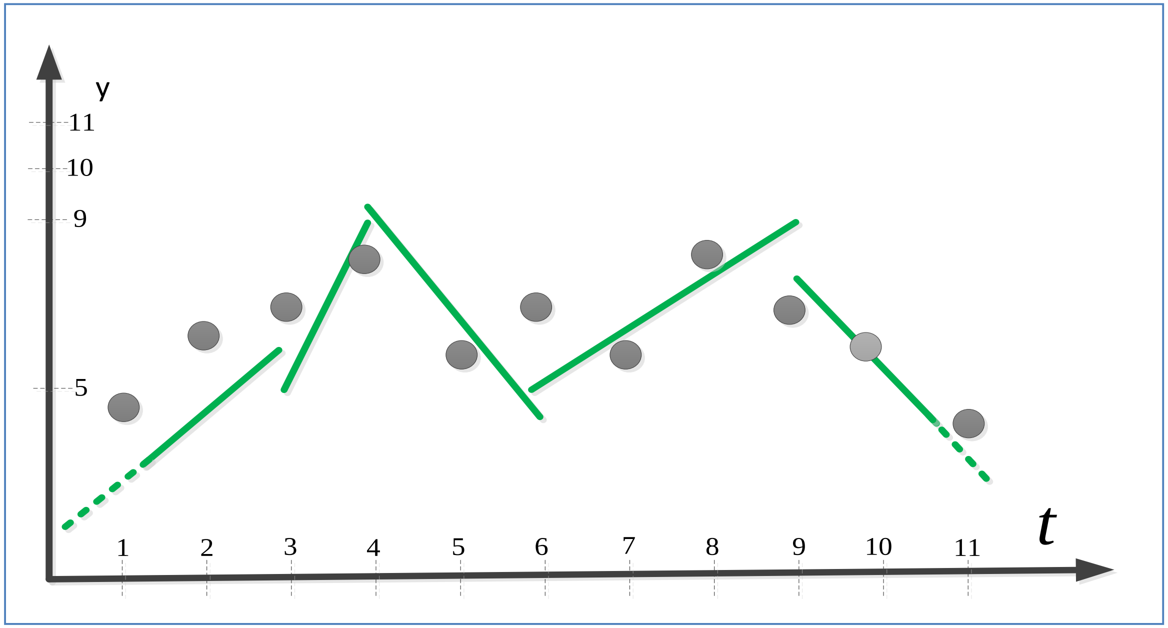
Input: *d*-dimensional signal *P* over time



k – Segment Queries

Input: *d*-dimensional signal *P* over time

Query: *k* segments over time



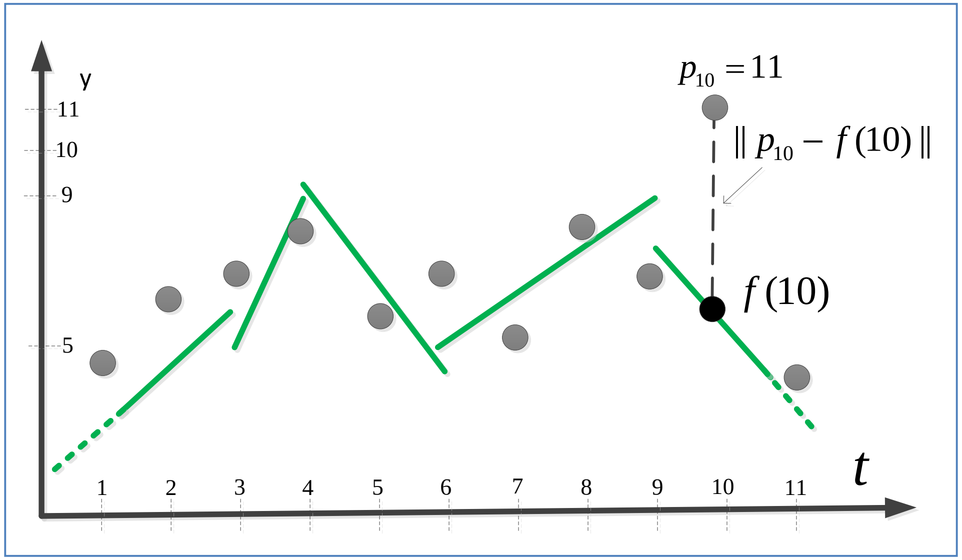
k-Piecewise linear function *f* over *t*

k – Segment Queries

Input: *d*-dimensional signal *P* over time

Query: *k* segments over time

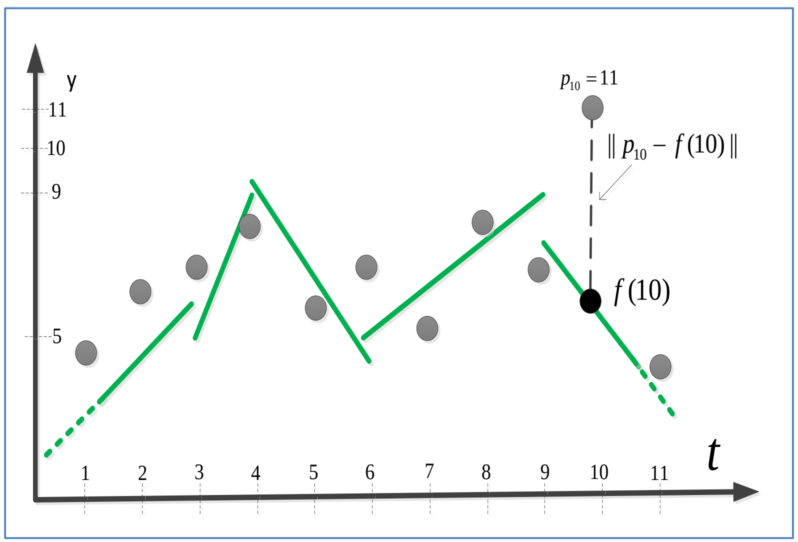
Output: Sum of squared distances from *P*



$(1+\epsilon)$-Corset for k-segment queries

A weighted set $C \subseteq P$ such that
for every k-segment f:

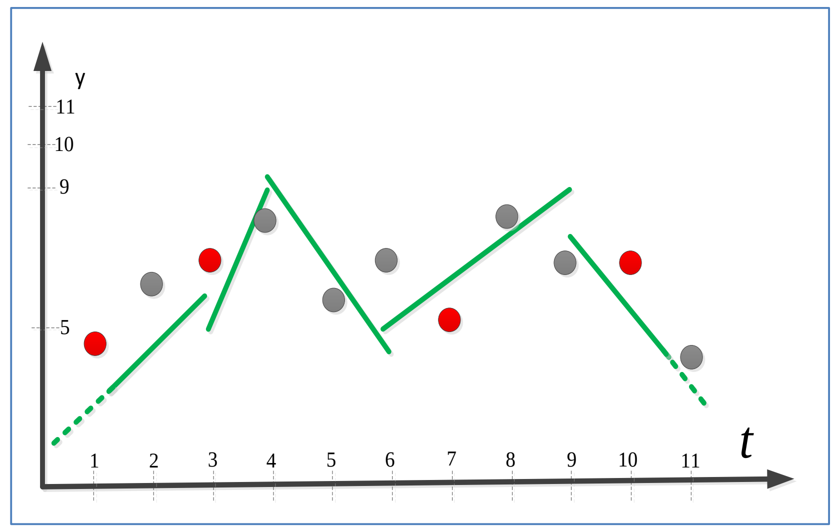
$$\mathrm{cost}(P,f) \sim \mathrm{cost}_{\mathrm{w}}(C,f)$$



$$\sum_t \|f(t) - pt\|$$

$\sim$

$(1 \pm \epsilon)$

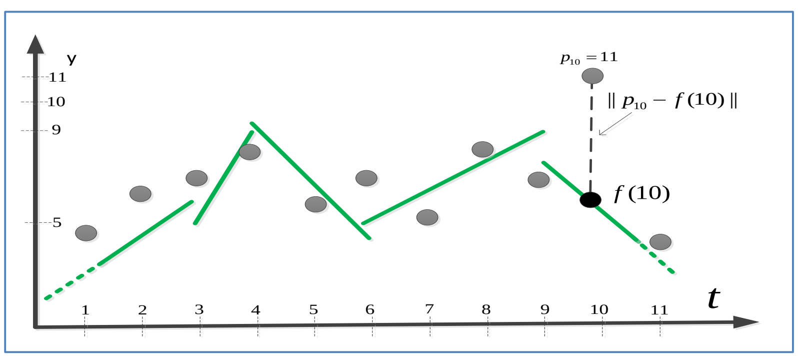


$$\sum_{p_t \in C} w(p_t) \cdot \|f(t) - pt\|$$

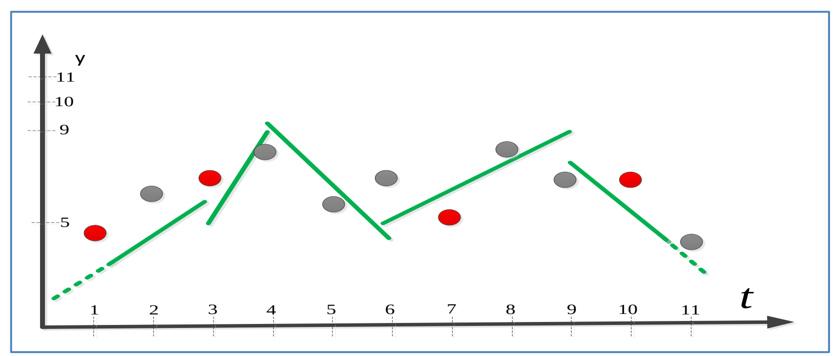
From Big Data to Small Data

Suppose that we can compute such a corset C of size $\frac{1}{\epsilon}$ for every set P of n points

- in time n^5,
- off-line, non-parallel, non-streaming algorithm



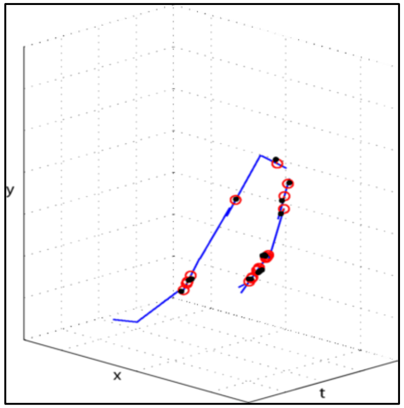
$\sim$



$(1 \pm \epsilon)$

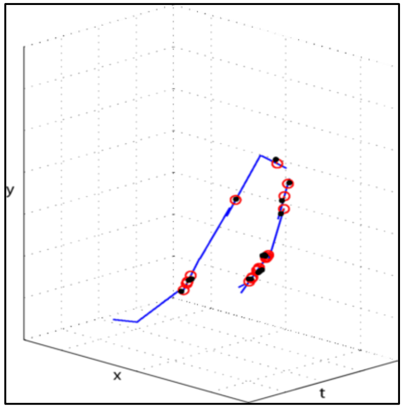
Read the first $\frac{2}{\epsilon}$ streaming points and reduce them into $\frac{1}{\epsilon}$ weighted points in time $\left(\frac{2}{\epsilon}\right)^5$

$1 + \epsilon$ corset for P_1

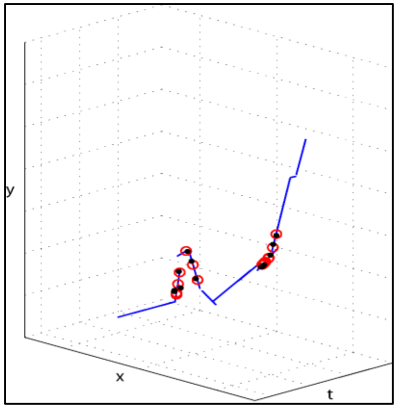


Read the next $\frac{2}{\epsilon}$ streaming point and reduce them into $\frac{1}{\epsilon}$ weighted points in time $\left(\frac{2}{\epsilon}\right)^5$

$1 + \epsilon$ corset for P_1

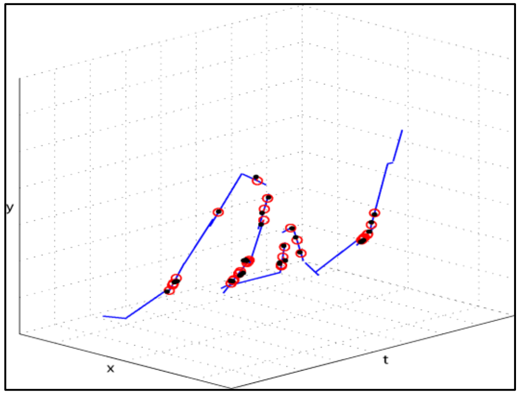


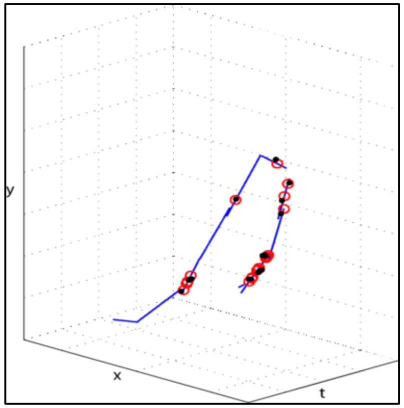
$1 + \epsilon$ corset for P_2

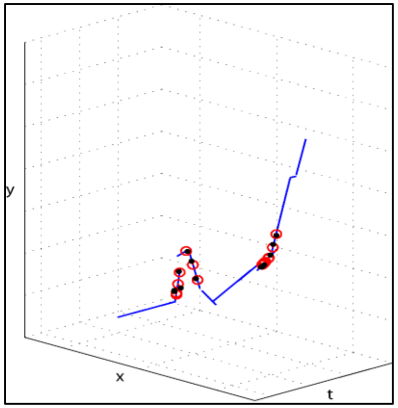


Merge the pair of ϵ-coresets into an ϵ-corset of $\frac{2}{\epsilon}$ weighted points

$1+\epsilon$-corset for $P_1 \cup P_2$

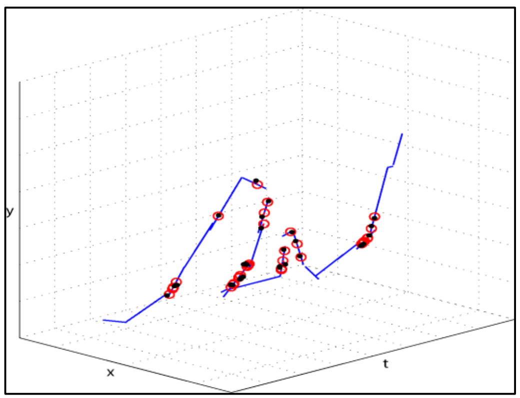


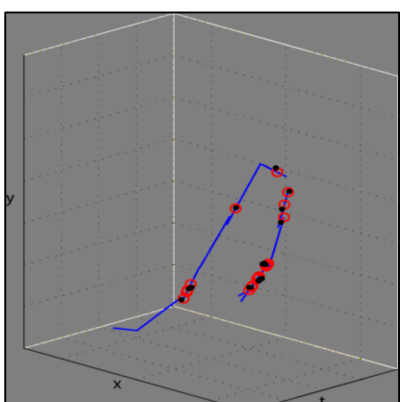


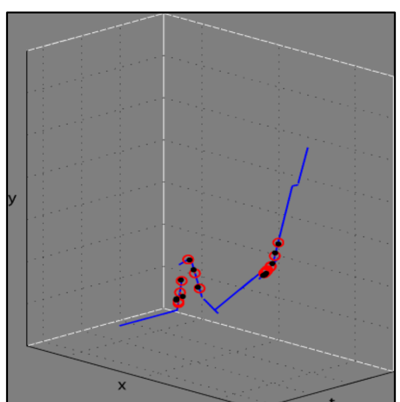


Delete the pair of original coresets from memory

$1 + \epsilon$-corset for $P_1 \cup P_2$



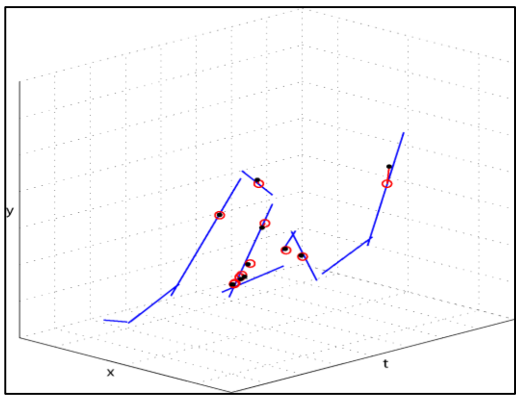


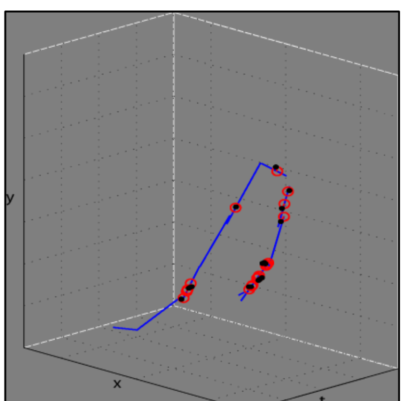


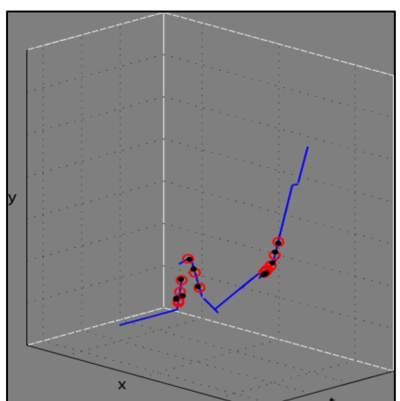
Reduce the $\frac{2}{\epsilon}$ weighted points into $\frac{1}{\epsilon}$ weighted points by constructing their coreset

$1 + \epsilon$-corset for

$1 + \epsilon$-corset for $P_1 \cup P_2$



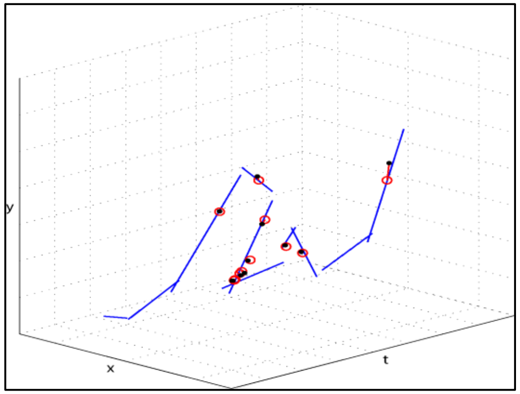




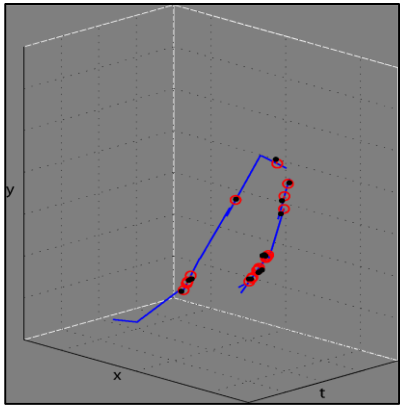
Reduce the $\frac{2}{\epsilon}$ weighted points into $\frac{1}{\epsilon}$ weighted points by constructing their coreset

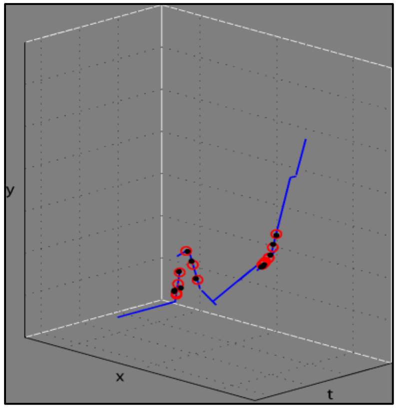
$1+\epsilon$-corset for

$1+\epsilon$-corset for $P_1 \cup P_2$

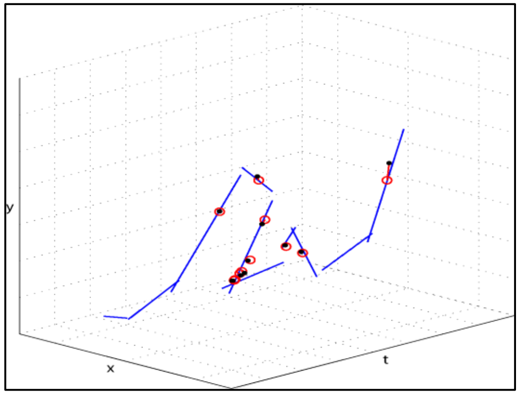


$=(1+\epsilon)^2$-corset for $P_1 \cup P_2$

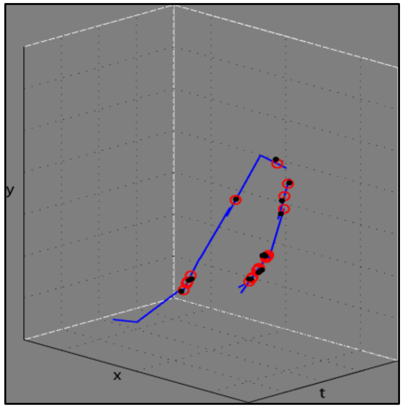


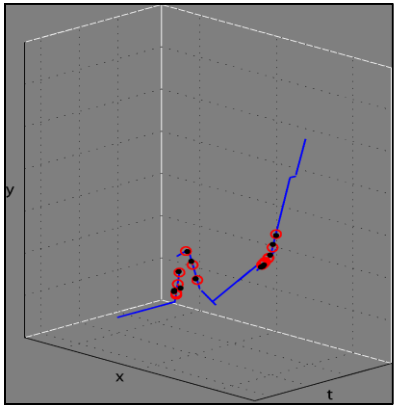


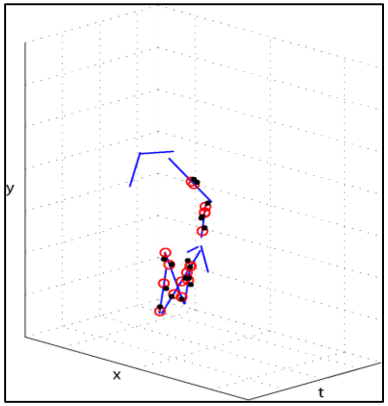
$(1+\epsilon)^2$-corset for $P_1 \cup P_2$



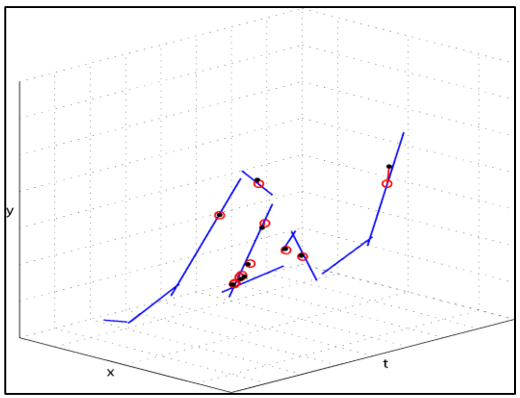
$(1+\epsilon)$-corset for P_3

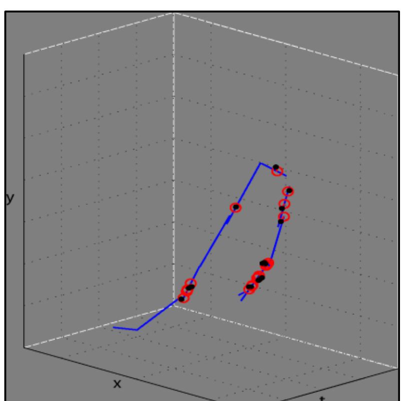


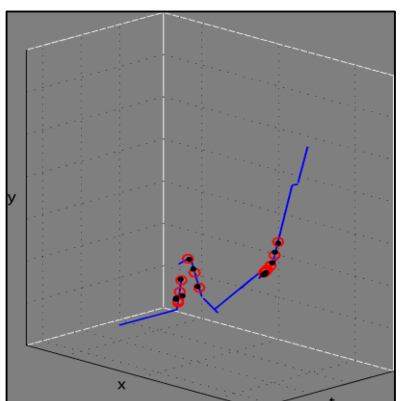




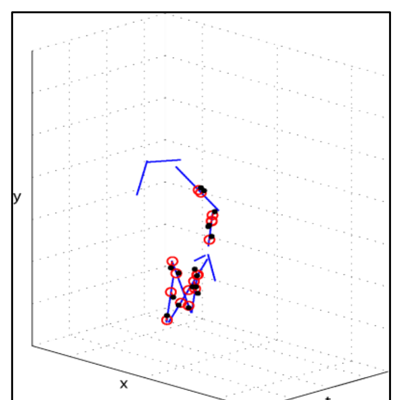
$(1+\epsilon)^2$-corset for $P_1 \cup P_2$



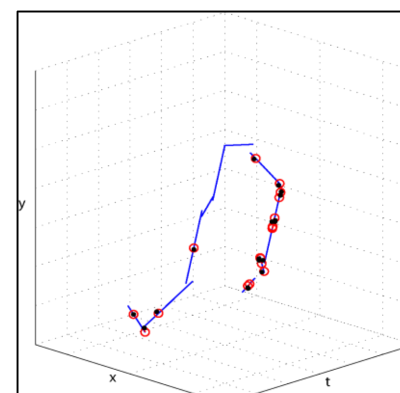




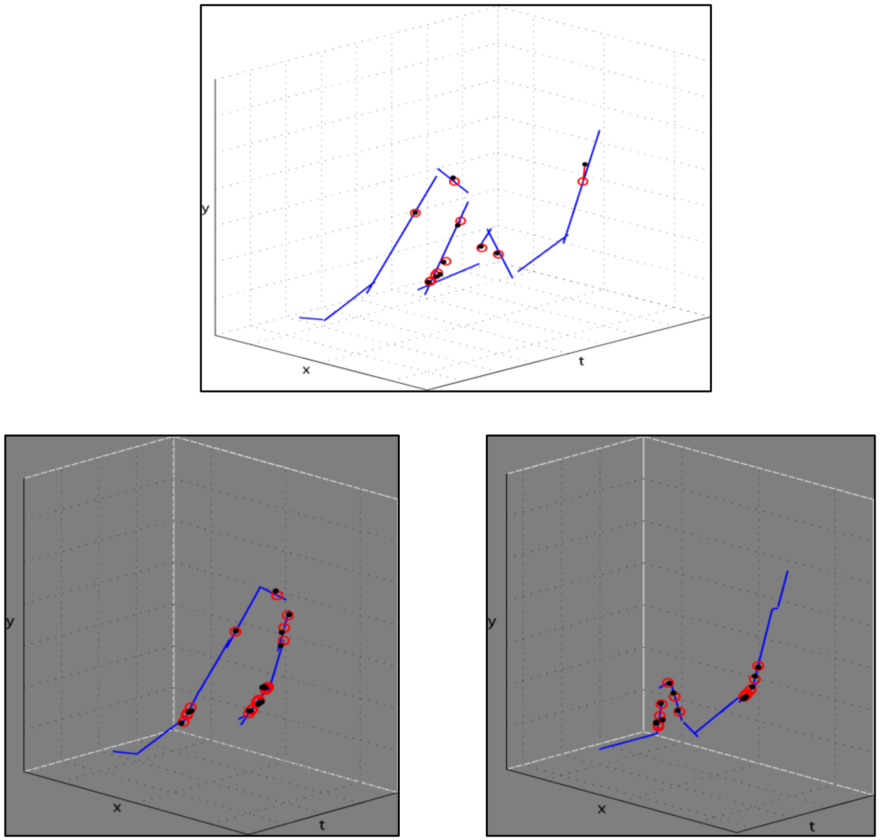
$(1+\epsilon)$-corset for P_3



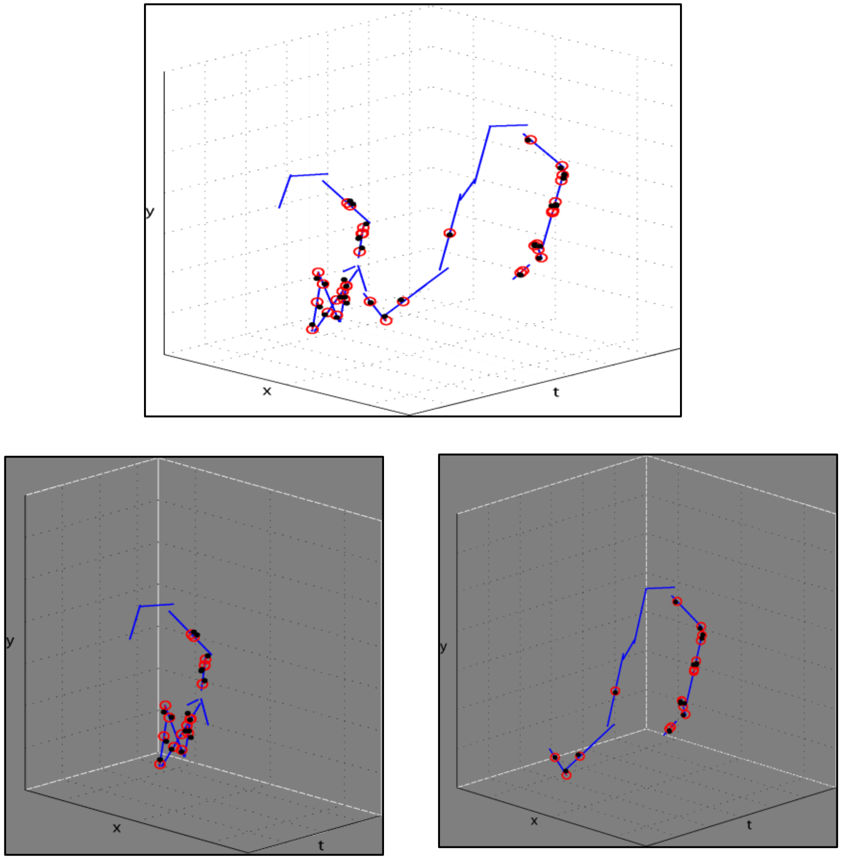
$(1+\epsilon)$-corset for P_4



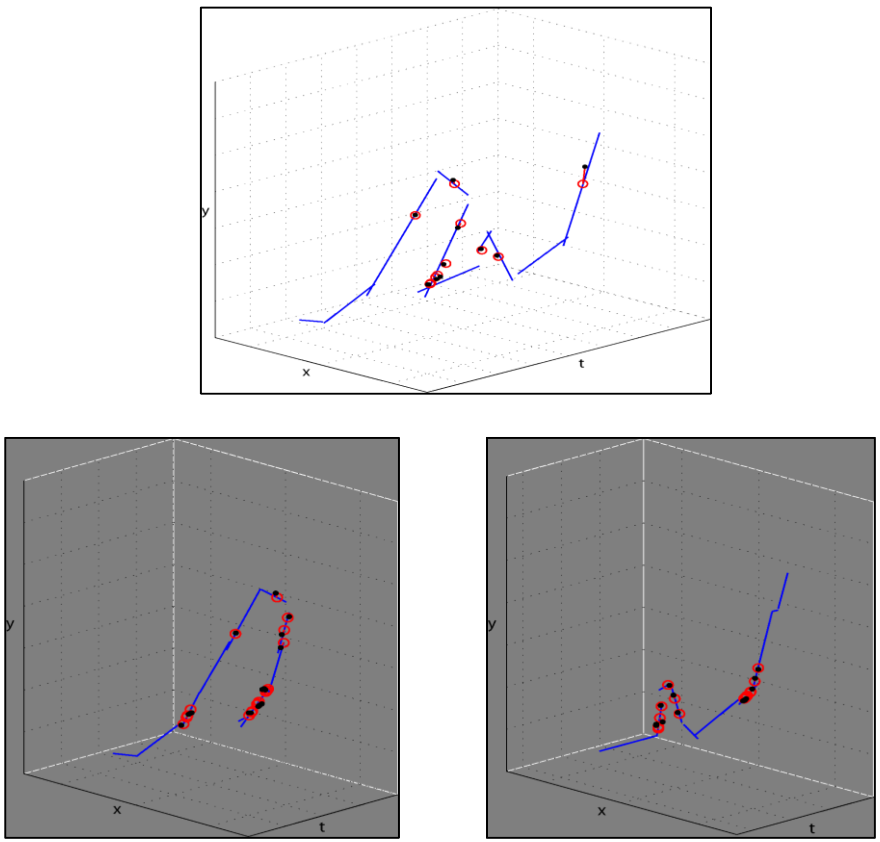
$(1+\epsilon)^2$-corset for $P_1 \cup P_2$



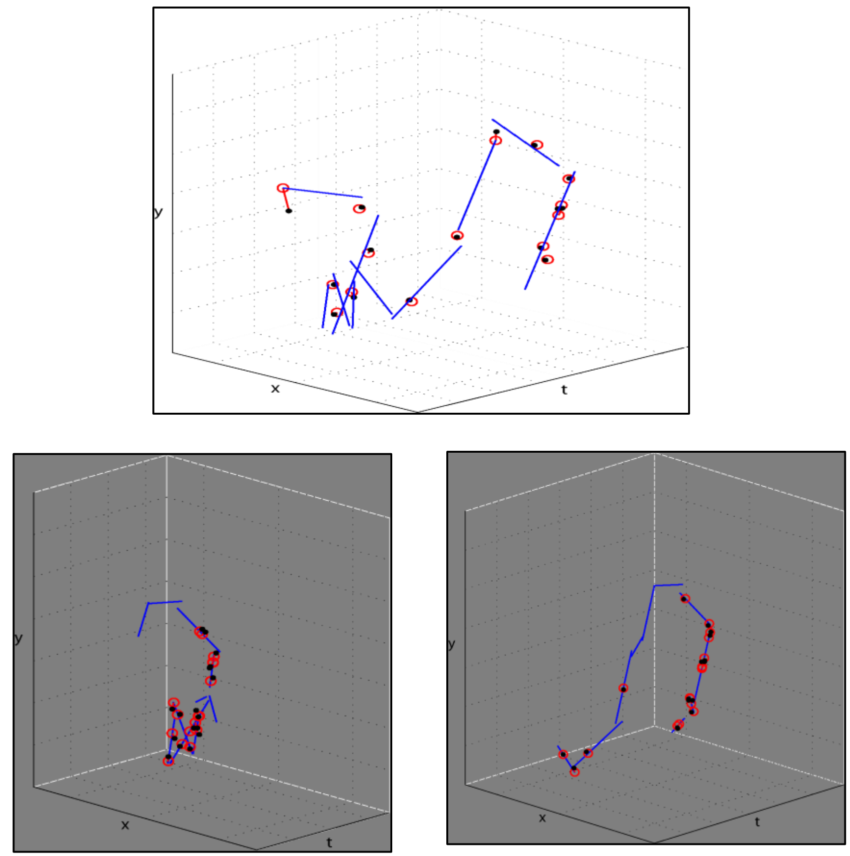
$(1+\epsilon)$-corset for $P_3 \cup P_4$

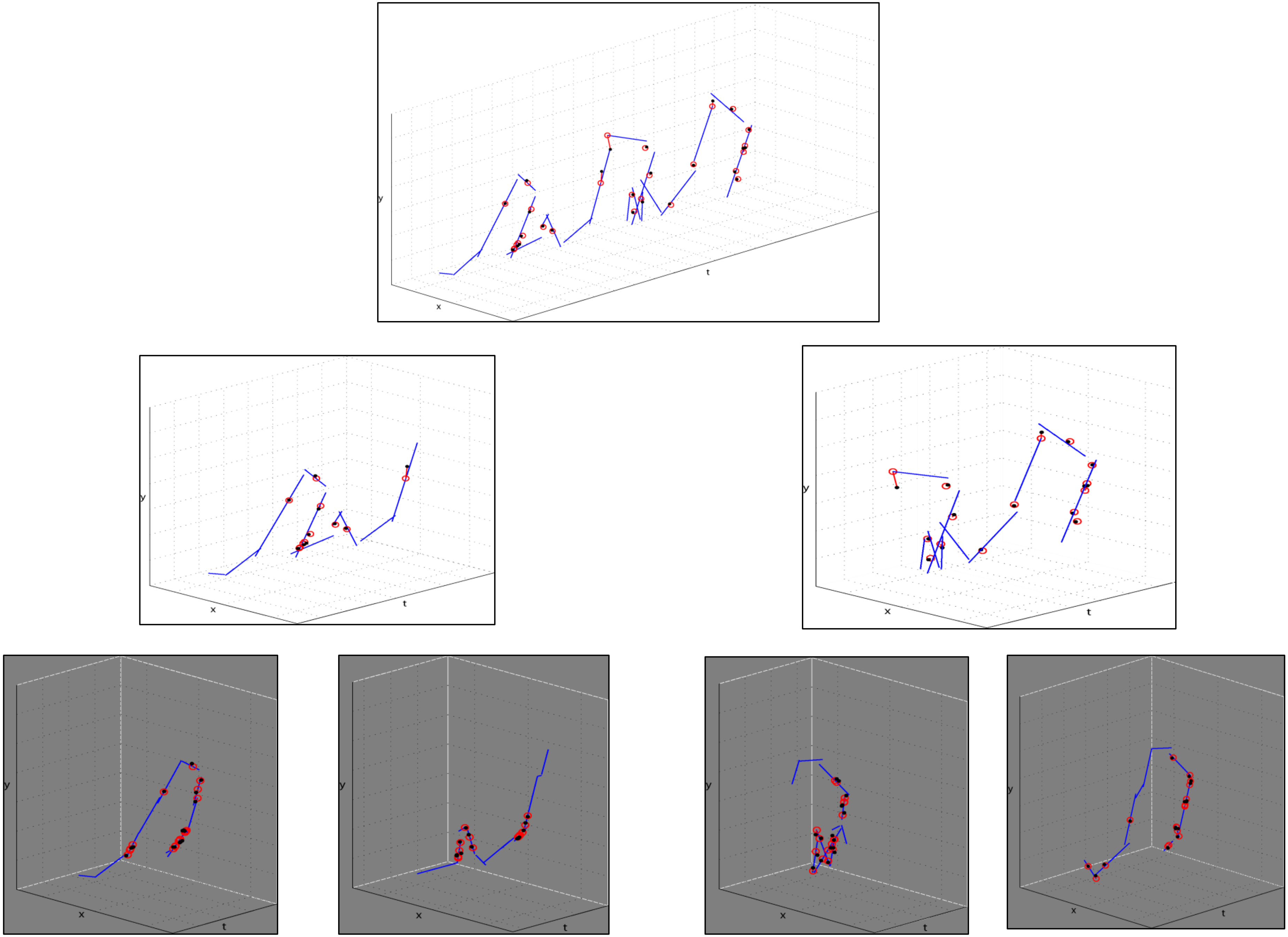


$(1+\epsilon)^2$-corset for $P_1 \cup P_2$



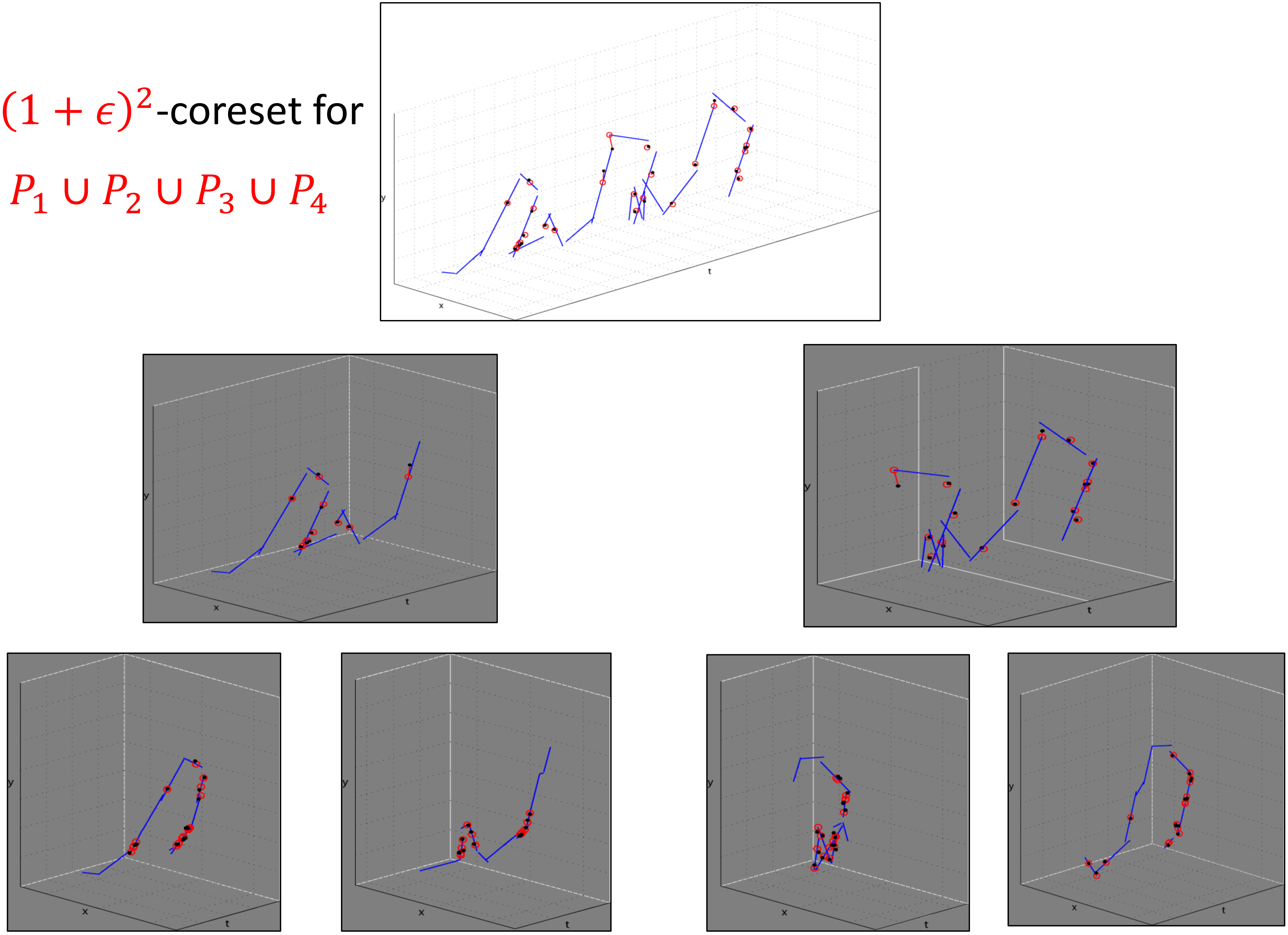
$(1+\epsilon)^2$-corset for $P_3 \cup P_4$




y
x
t
y
x
t
y
x
t
y
x
t
y
x
t
y
x
t
y
x
t

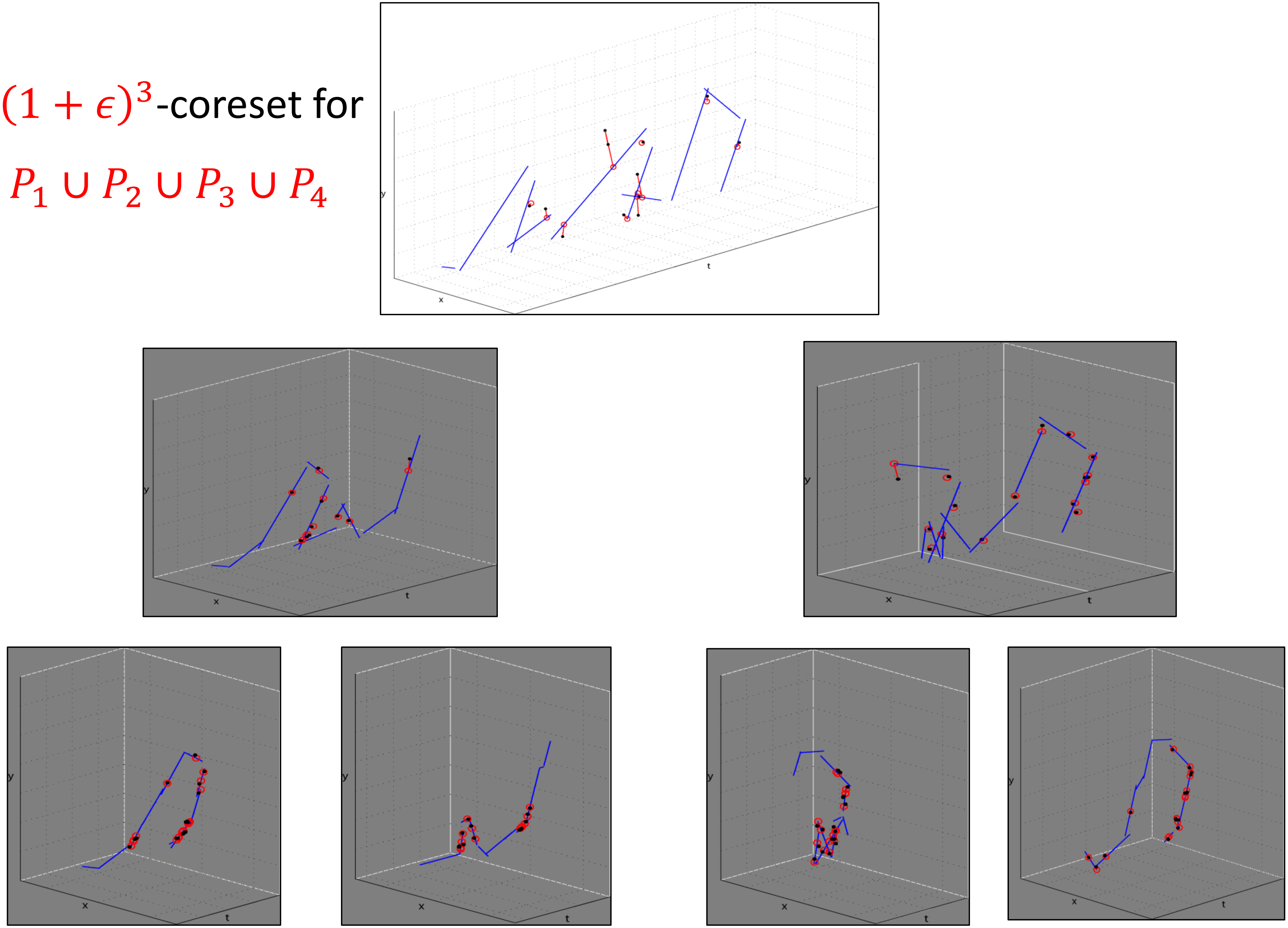
$(1+\epsilon)^2$-coreset for

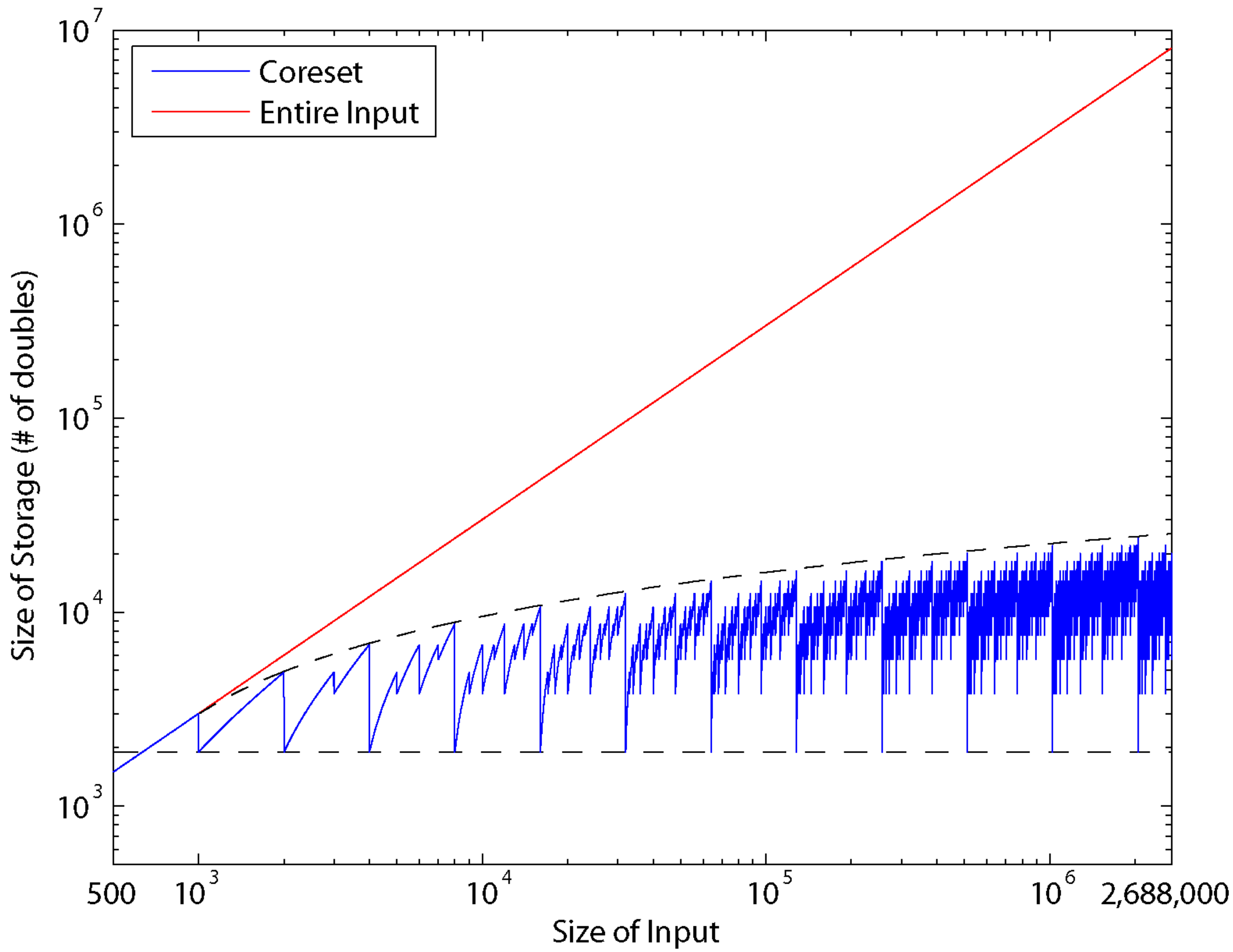
$P_1 \cup P_2 \cup P_3 \cup P_4$



$(1+\epsilon)^3$-coreset for

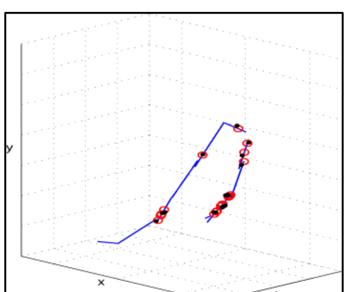
$P_1 \cup P_2 \cup P_3 \cup P_4$

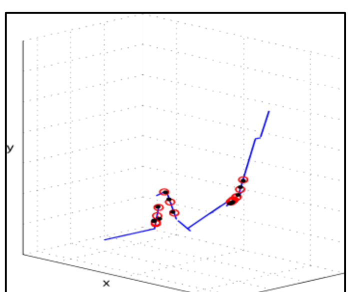


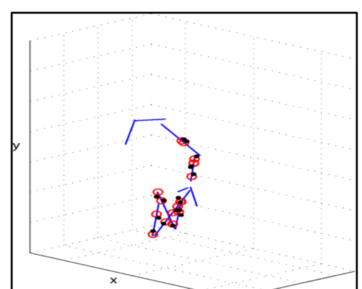


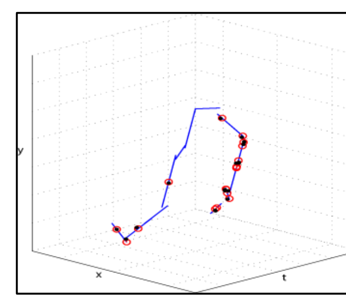
Coreset
Entire Input
Size of Storage (# of doubles)
Size of Input
10^7
10^6
10^5
10^4
10^3
500
10^3
10^4
10^5
10^6
2,688,000

Parallel Computation

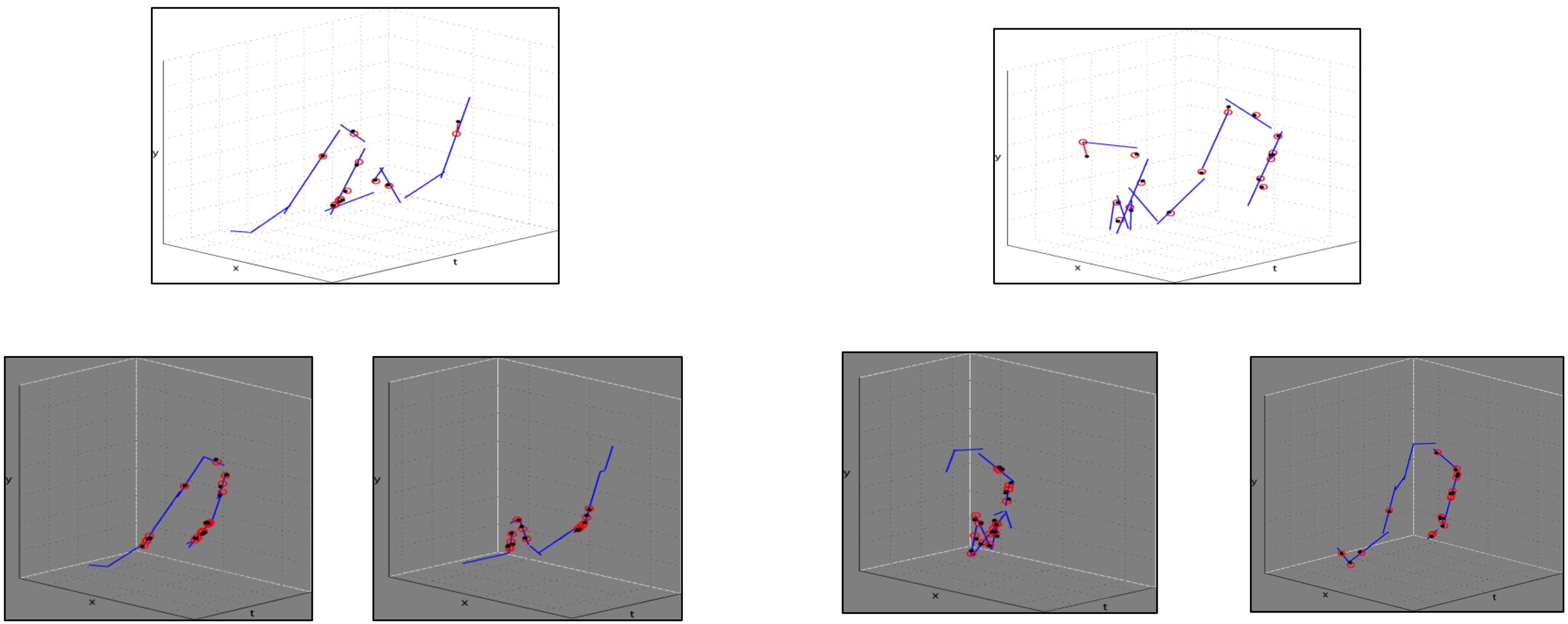






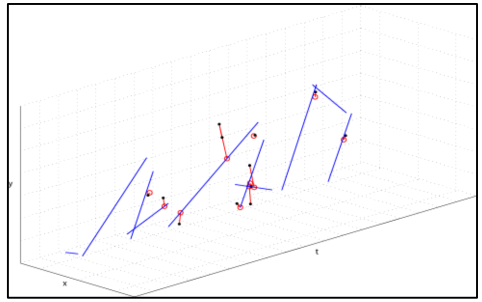


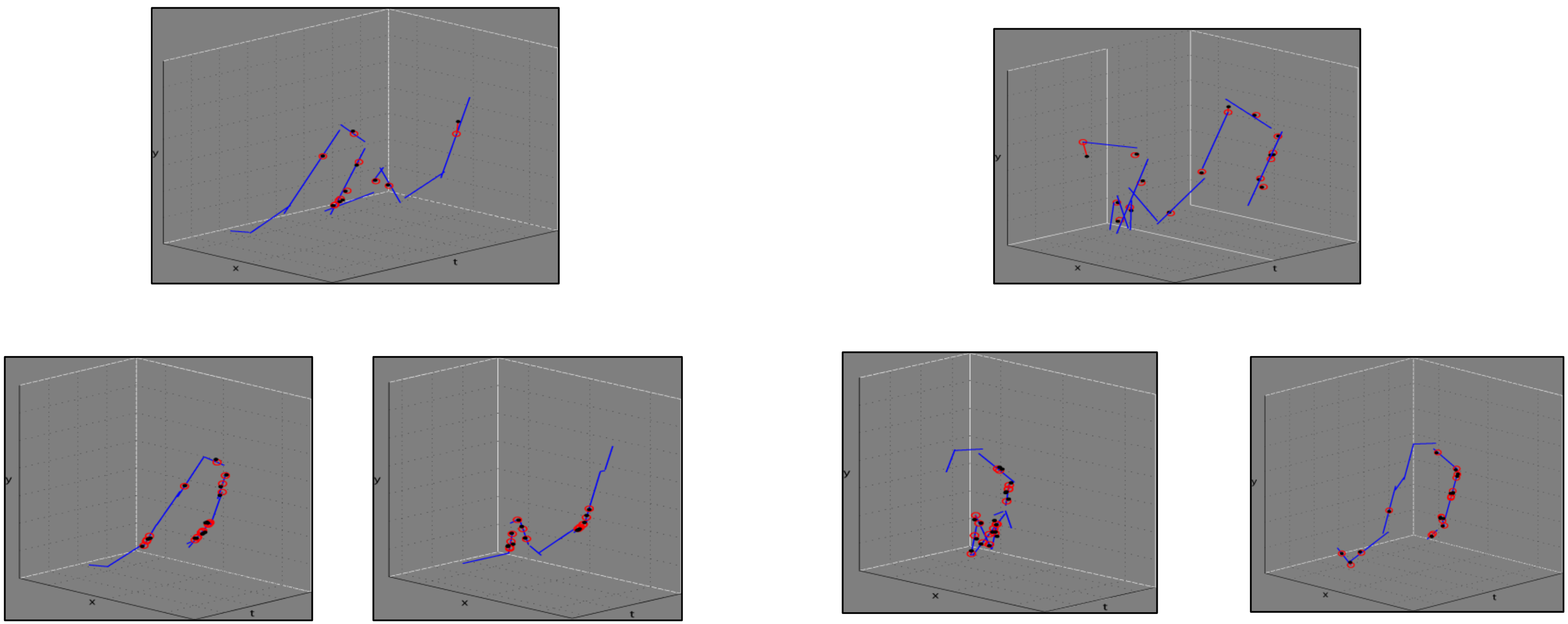
Parallel Computation



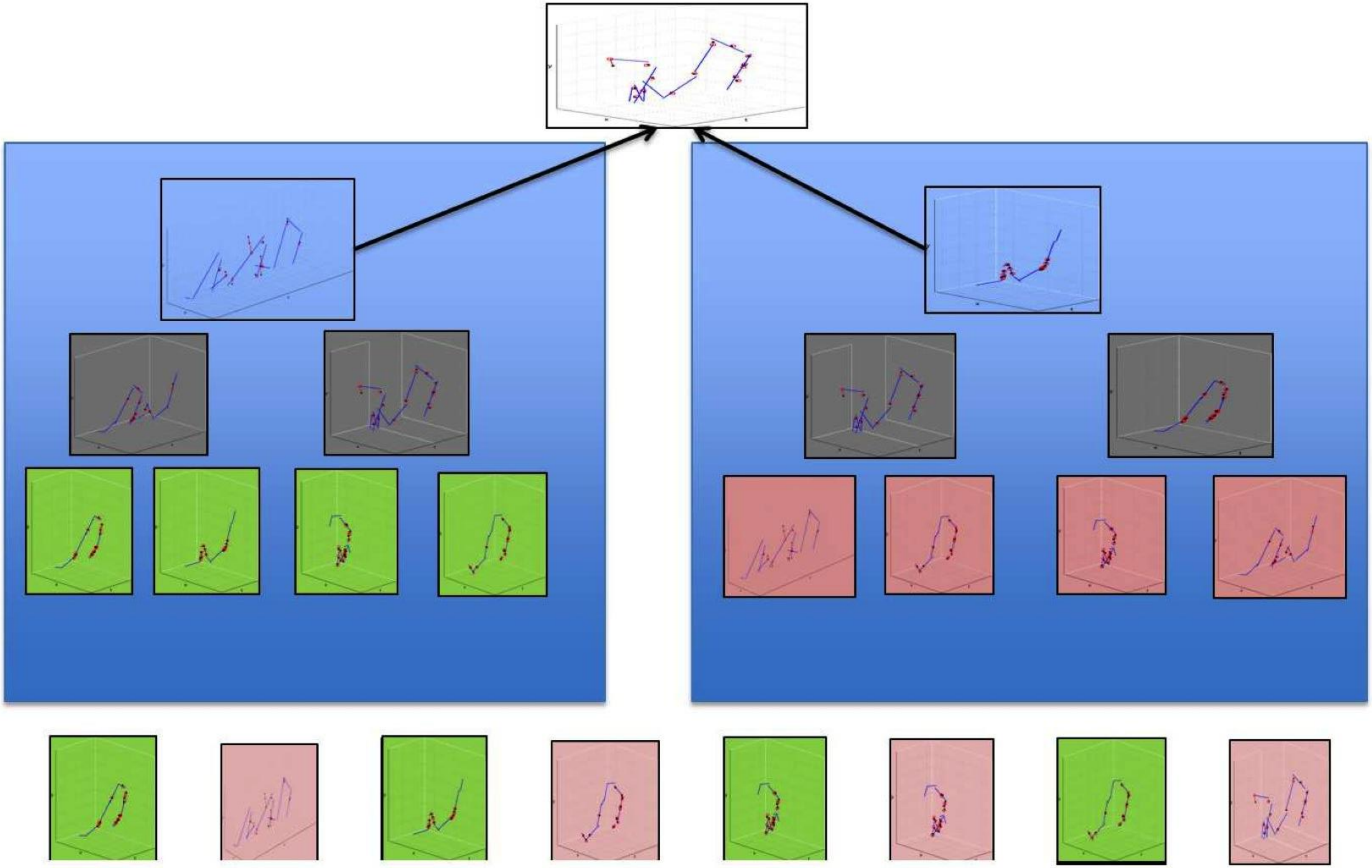
Parallel Computation

Run off-line algorithm on corset using single computer





Parallel+ Streaming Computation

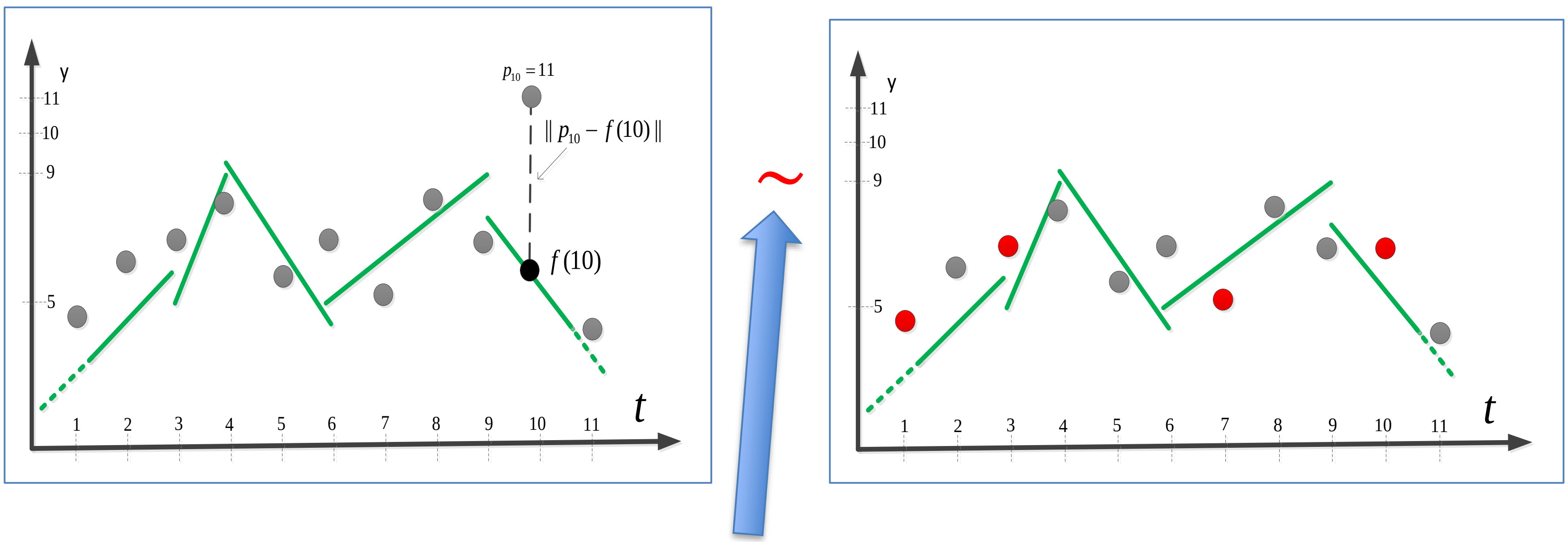


ICRA’14 (With Rus, Paul and Newman)

Coreset

A weighted set C such that for every k-segment f :

$$\mathrm{cost}(P, f) \sim \mathrm{cost}_w(C, f)$$



$(1 \pm \epsilon)$

Generic Coreset definition

Let

- X be a set, called *points set*
- Q be a set, called *query set*
- $\mathrm{cost}: 2^X \times Q \rightarrow [0, \infty)$ be a function that maps every set $P \subseteq X$ and query $q \in Q$ into a non-negative number $\mathrm{cost}(P, q)$

For a given $\epsilon > 0$ and $P \subseteq X$,
the set $C \subseteq X$ is a *corset* if
for every $q \in Q$ we have

$$\mathrm{cost}(P, q) \sim cost(C, q)$$

Up to $(1 \pm \epsilon)$ multiplicative error

Theorem [Feldman, Langberg, STOC'11]

Suppose that

$$\mathrm{cost}(P, q) := \sum_{p \in P} w(p)\mathrm{dist}(p, q)$$

where $\mathrm{dist}: P \times \; Q \to [0, \infty).$

A sample $C \subseteq P$ from the distribution

$$\mathrm{sensitivity(p)} = \max_{q \in Q} \frac{dist(p, q)}{\sum_{p'} dist(p', q)}$$

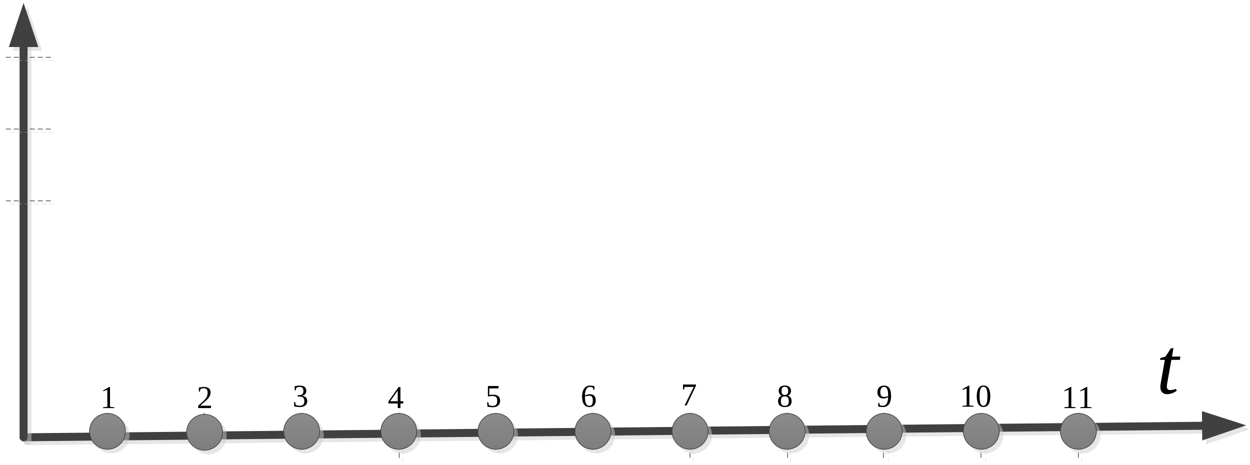
is a coreset if $|C| \geq \frac{\text{dimension of } Q}{\epsilon^2} \cdot \sum_p \mathrm{sensitibity}(p)$

No corset for *k*-segment:

If $k > 3$ there is a set P such that every weighted $(1 + \epsilon)$-coreset must be of size $|P|$

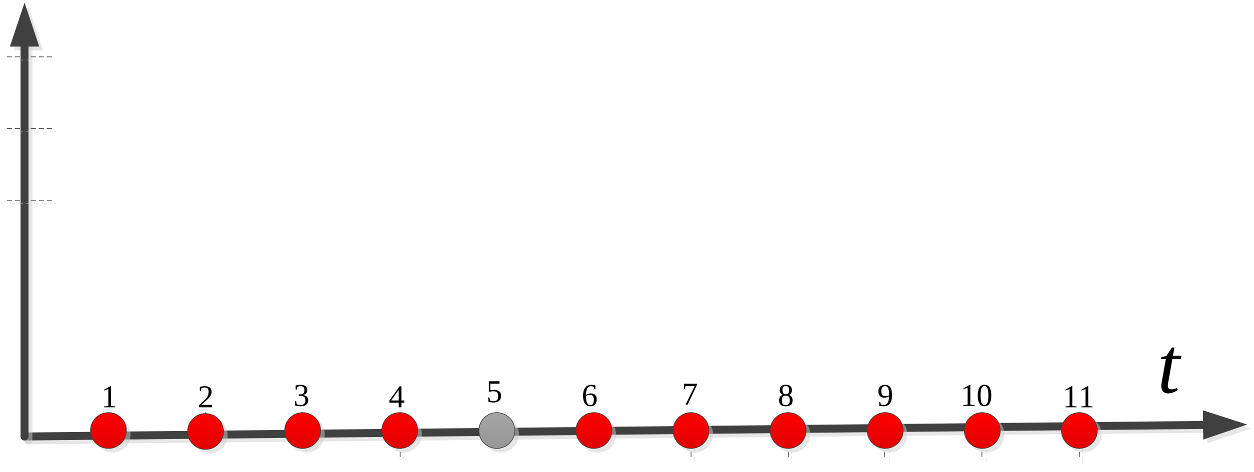
No small coreset $C \subset P$ exists

Input P: n points on the x-axis



Input P: n points on the x-axis

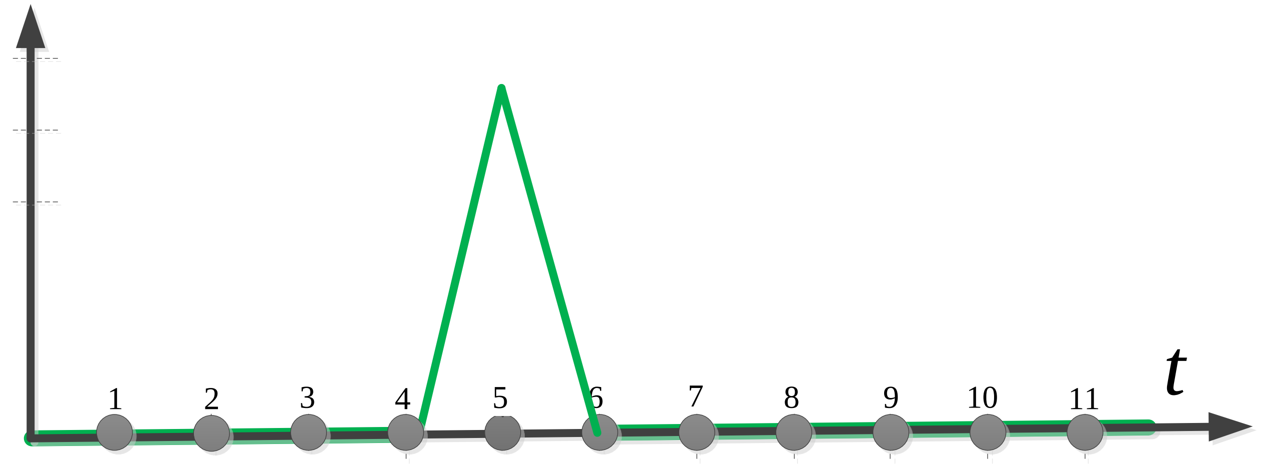
Coreset C: all points except one



Input P: n points on the x-axis

Coreset C: all points except one

Query f: covers all except this one



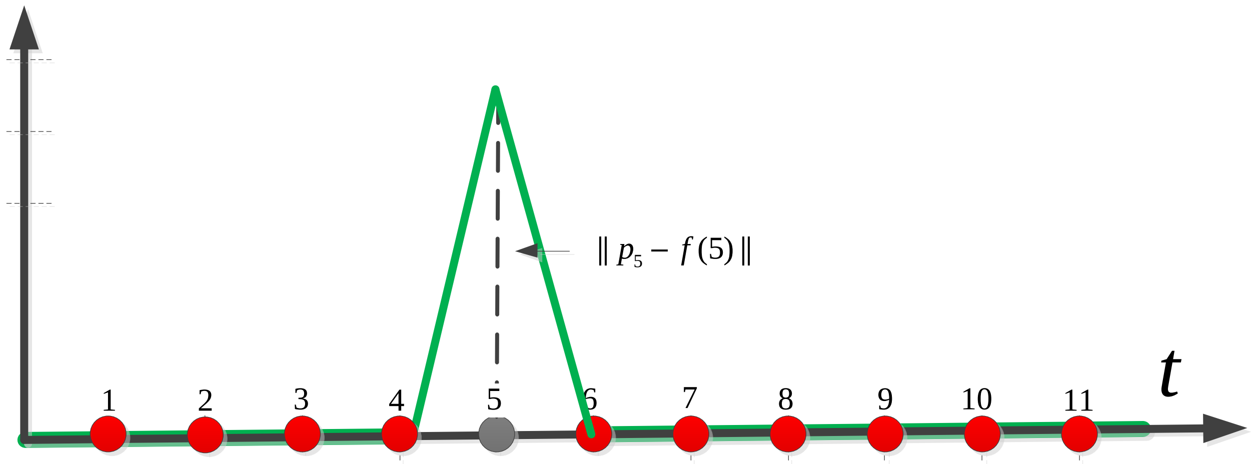
Input P: n points on the x-axis

Coreset C: all points except one

Query f: covers all except this one

$\mathrm{Cost}(P, f) > 0$

$\mathrm{Cost}(C, f) = 0$



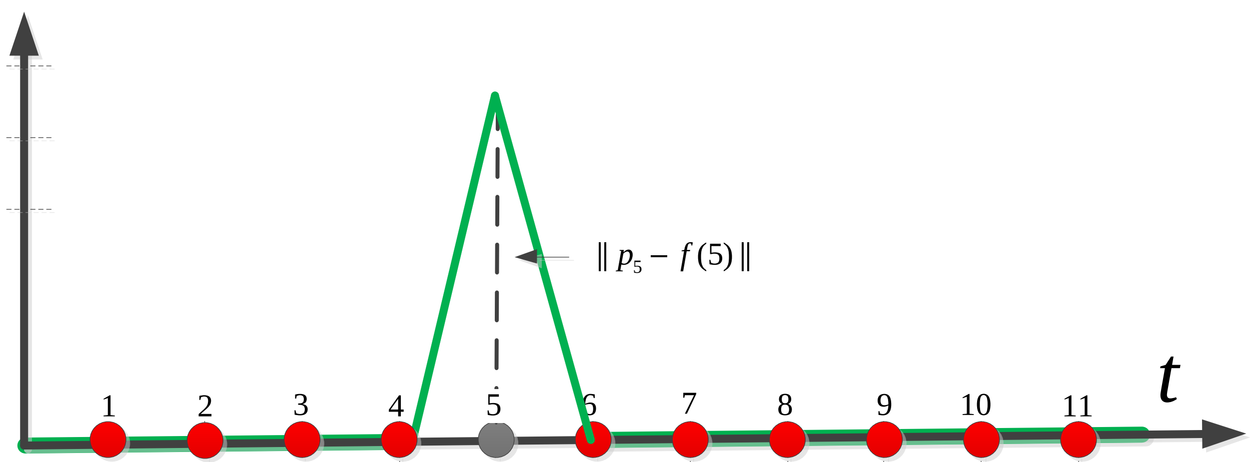
Input P: *n* points on the *x*-axis

Coreset *C*: all points except one

Query *f*: covers all except this one

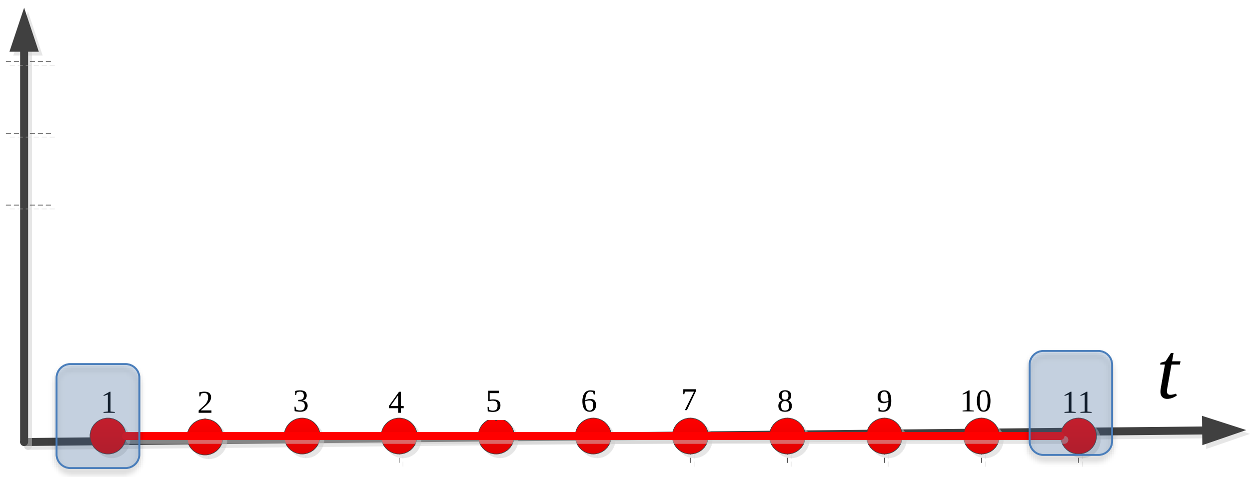
$$\frac{\mathrm{Cost}(P, f) > 0}{\mathrm{Cost}(C, f) = 0}$$

Unbounded factor approximation



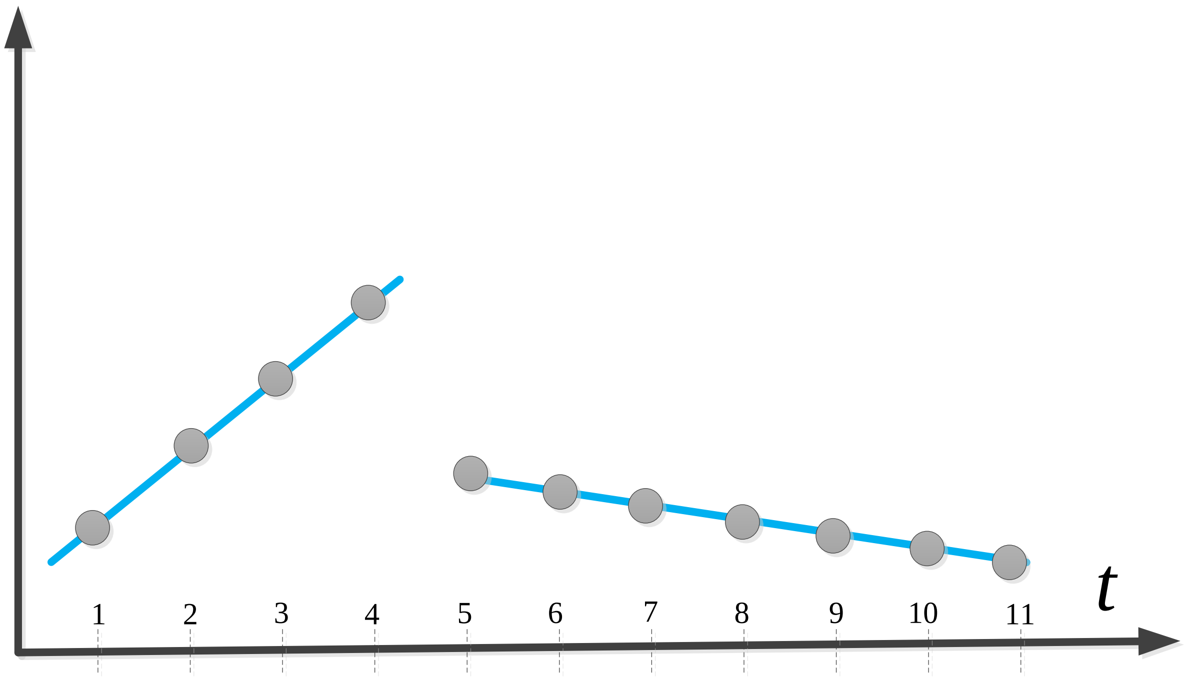
Observation:

Points on a segment can be stored by the two indexes of their end-points



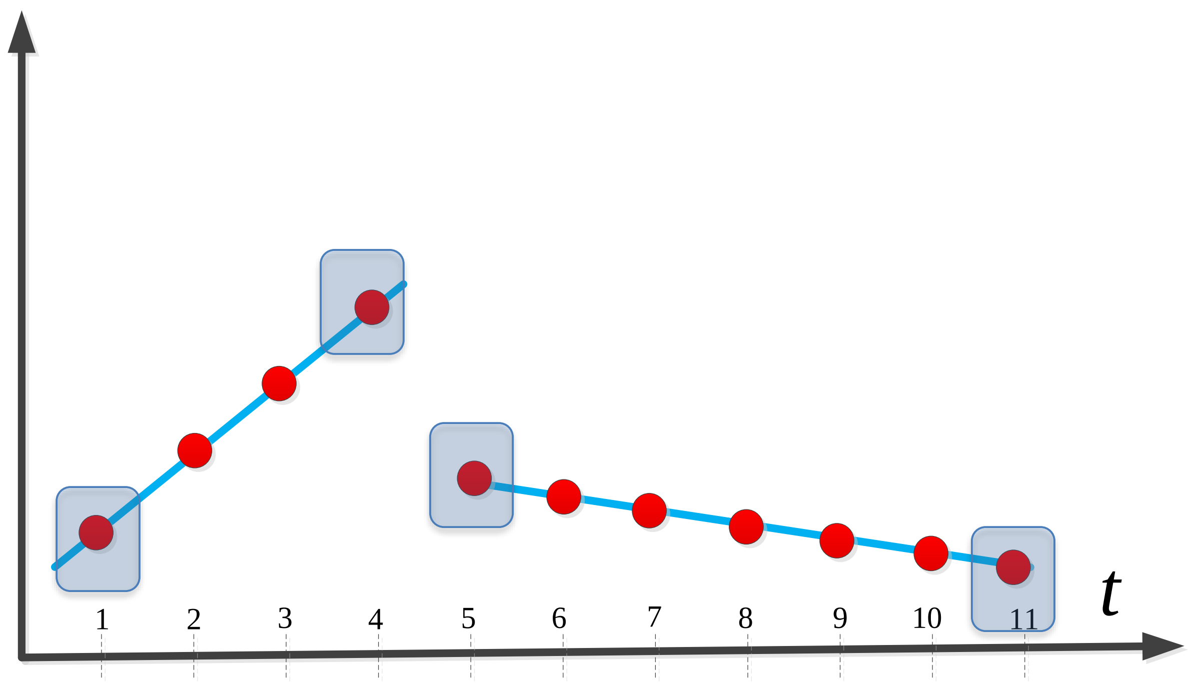
Observation:

Points on a segment can be stored by the two indexes of their end-points and the slope of the segment



Observation:

Points on a segment can be stored by the two indexes of their end-points and the slope of the segment

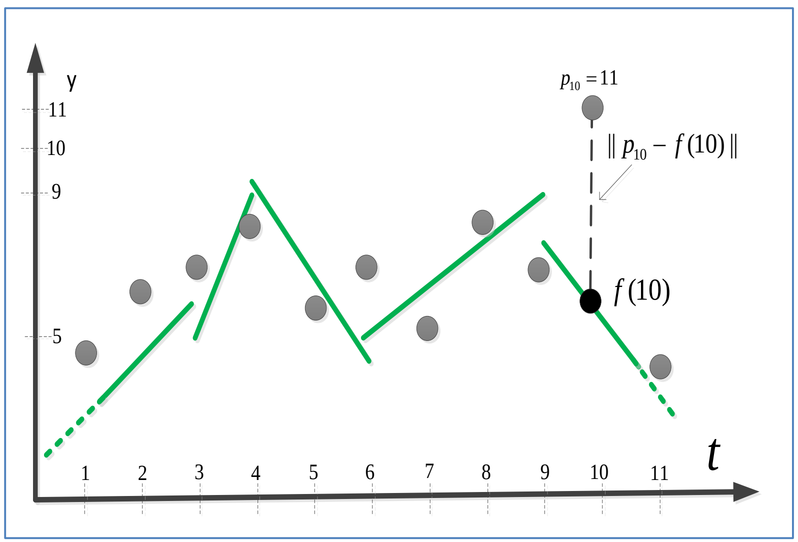


new Coreset definition

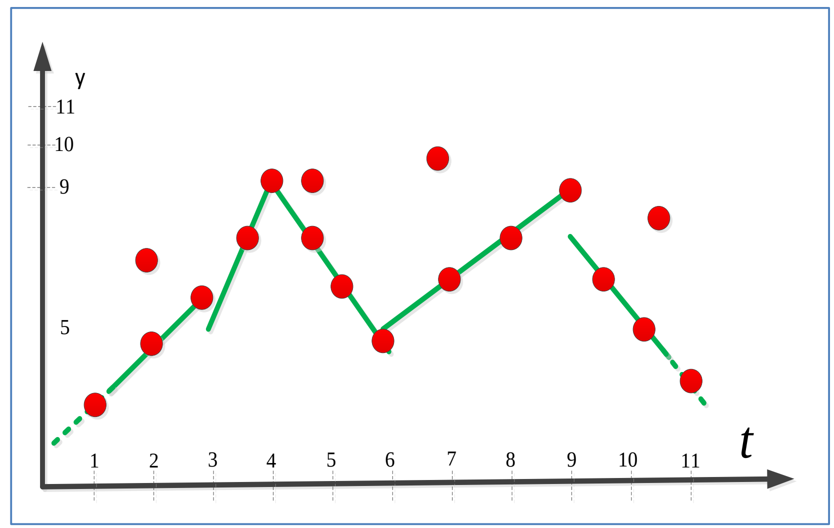
A weighted set $C \cancel{\subseteq} P$ such that

for every k-segment f :

$$\text{cost}(P, f) \sim \text{cost}_w(C, f)$$

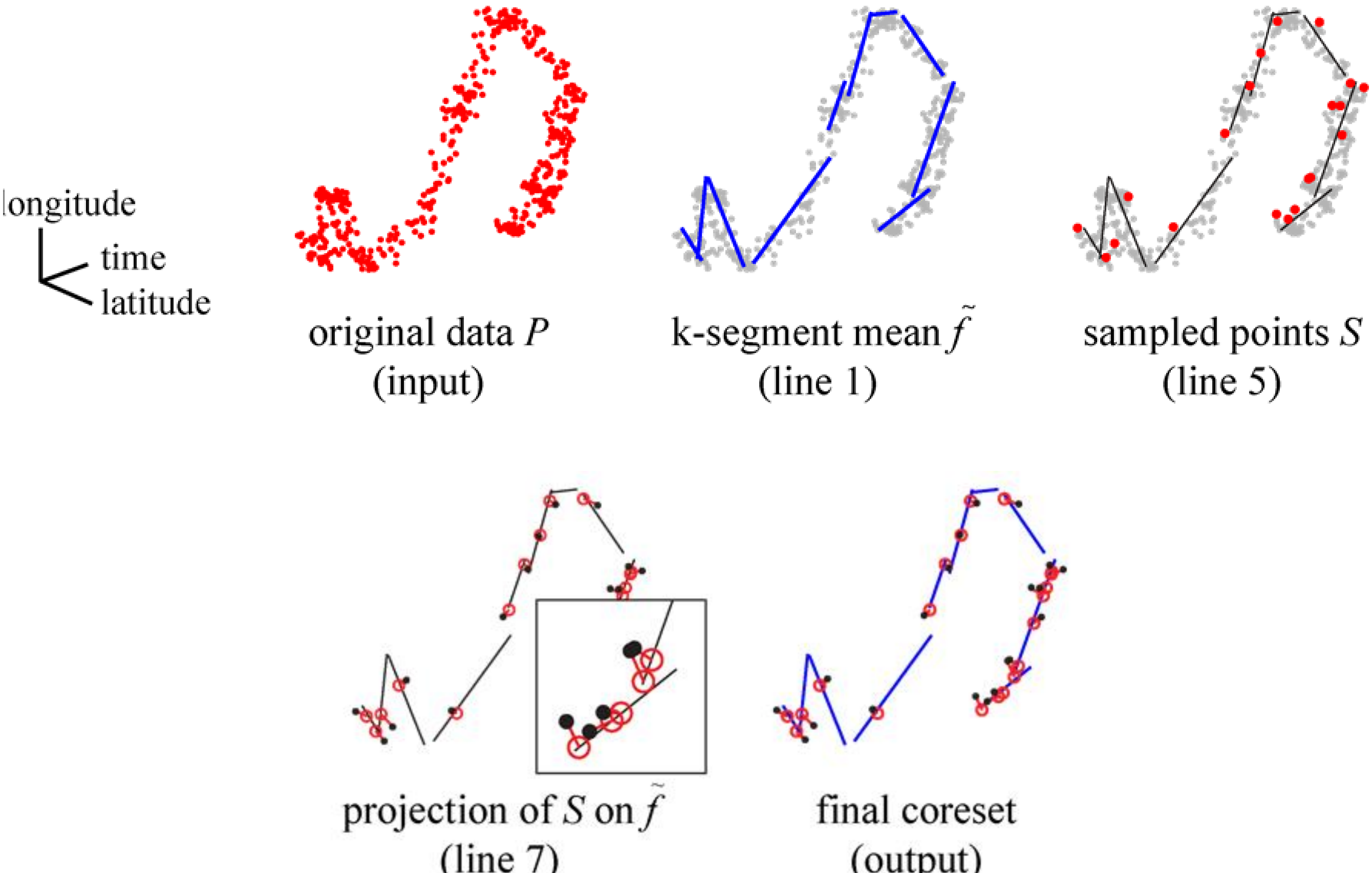


$\sim$



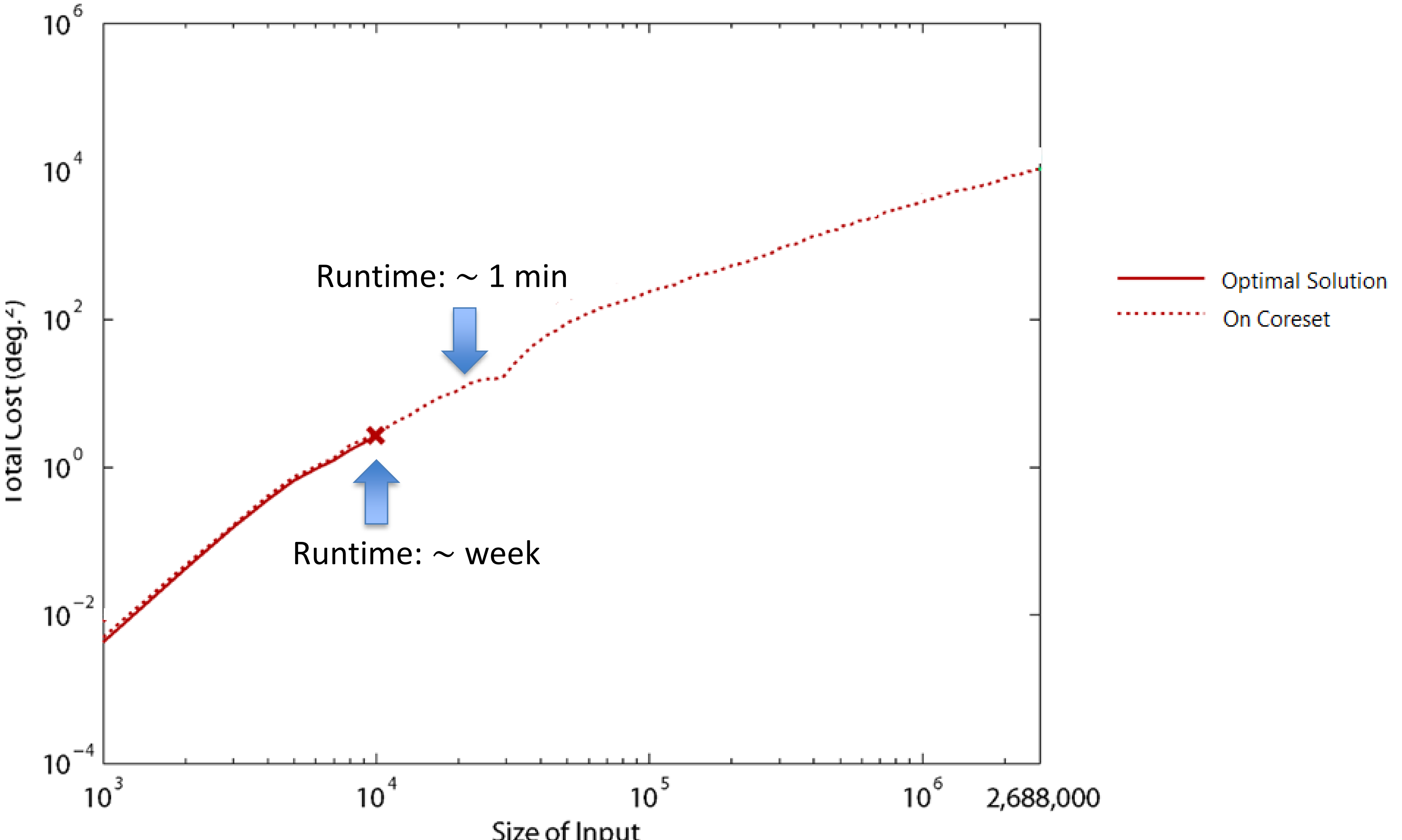
$$\sum_t \|f(t) - pt\|$$

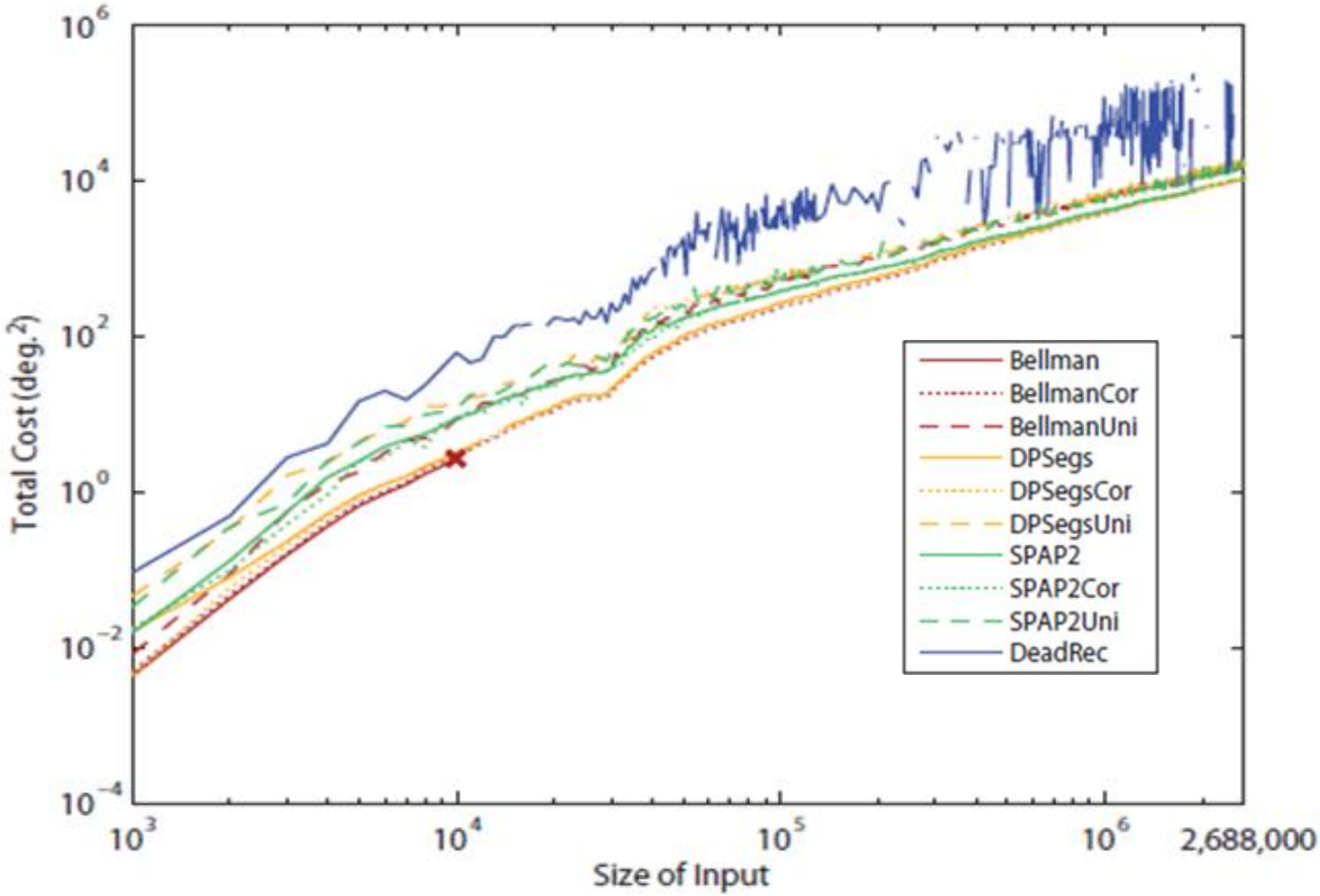
$$\sum_{p_t \in C} w(p_t) \cdot \|f(t) - pt\|$$



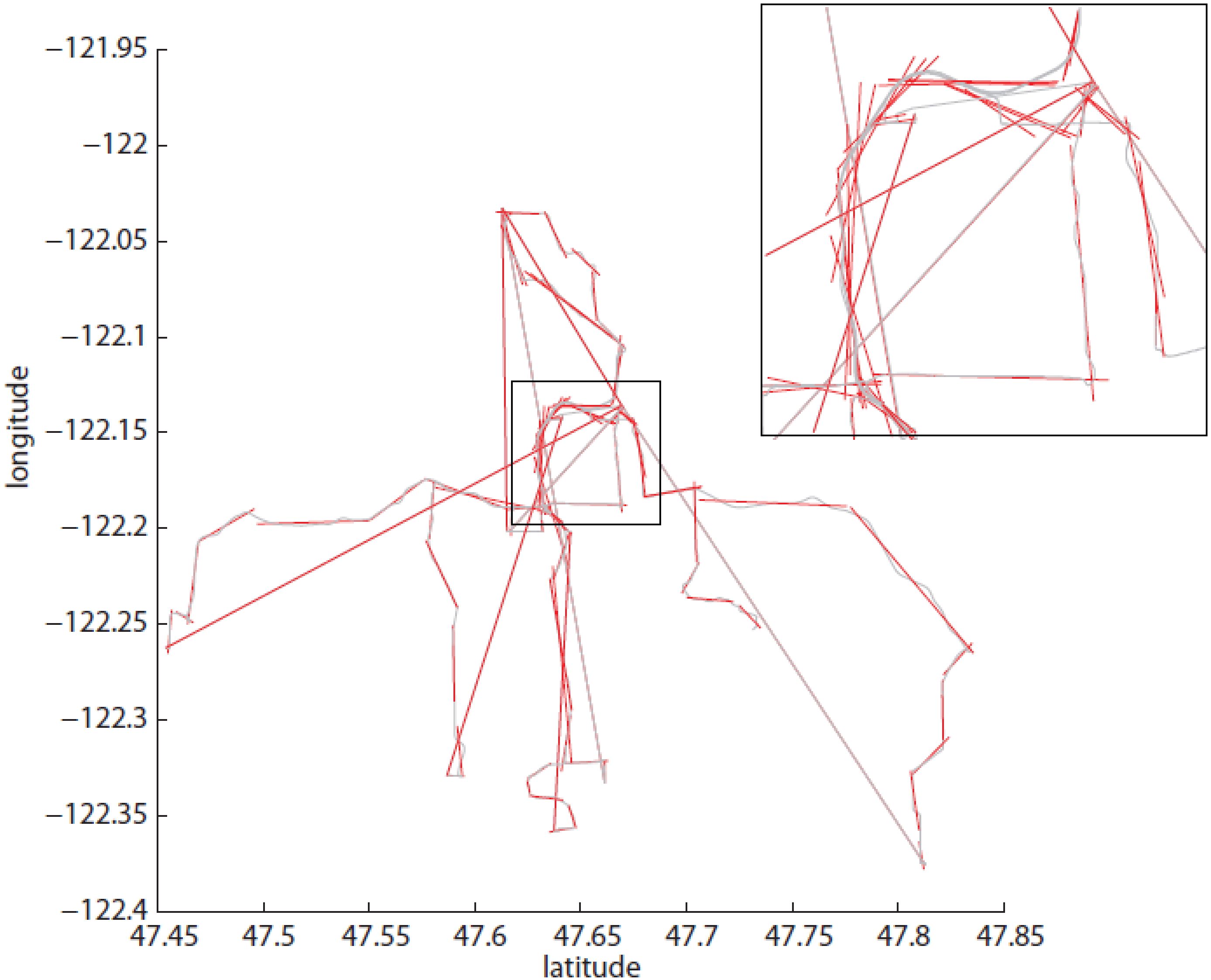
longitude
time
latitude
original data P
(input)
k-segment mean $\tilde{f}$
(line 1)
sampled points S
(line 5)
projection of S on $\tilde{f}$
(line 7)
final coreset
(output)

100-segment mean on GPS traces from taxi-cabs in San-Francisco



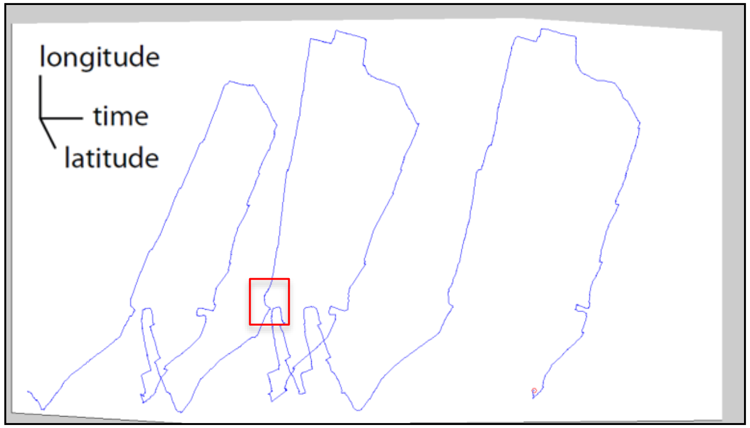


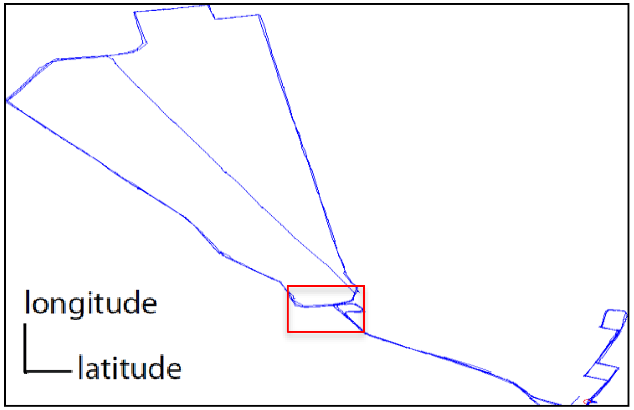
Results on GPS data from 500 Taxi Cabs

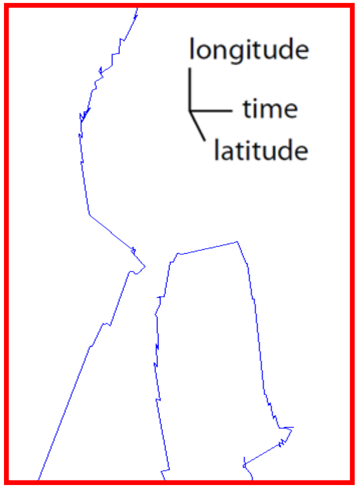


k= 91-segments, SIGSPATIAL

Big Data — Big Noise

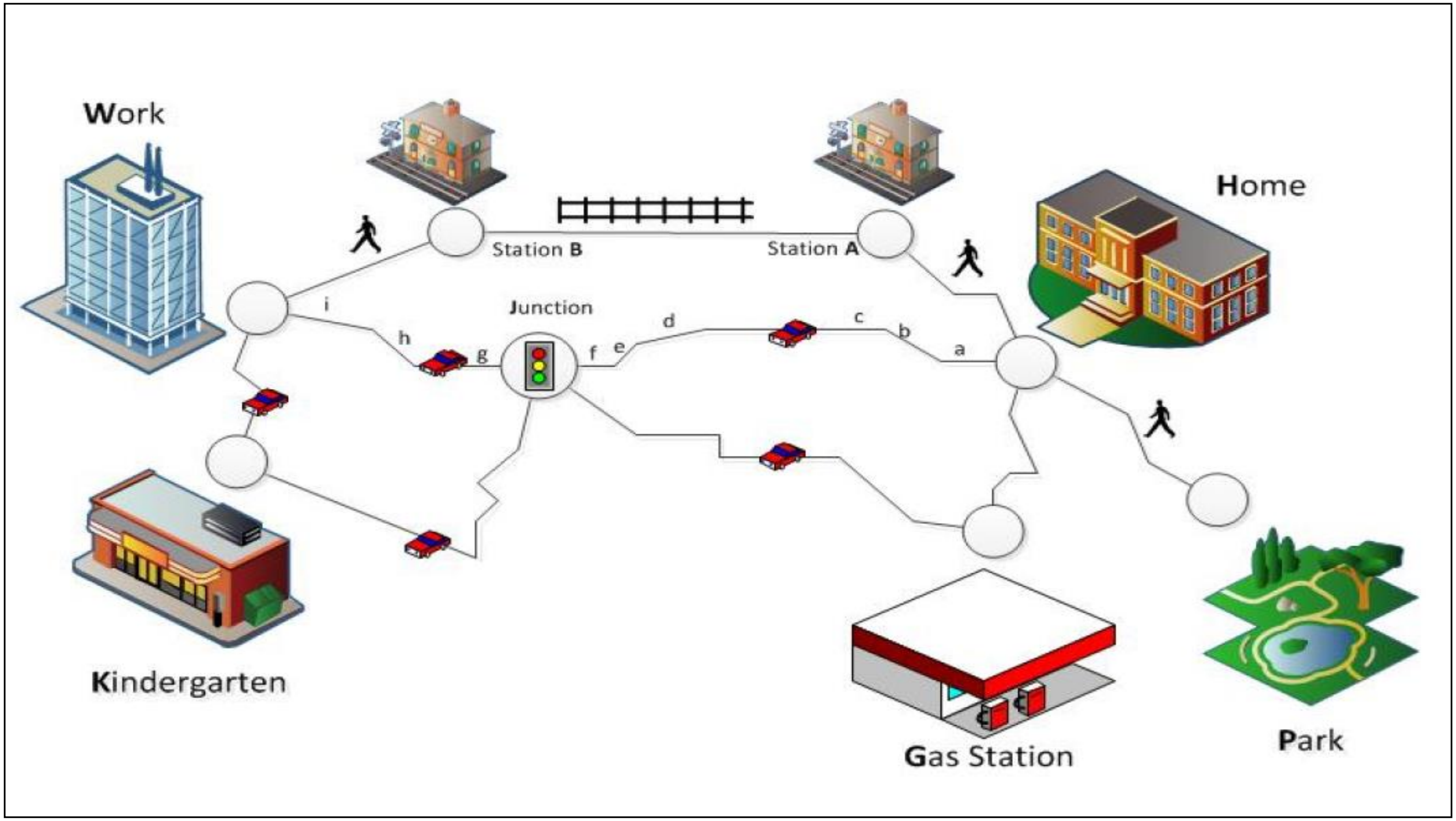






(k,m)- Hidden Markov Model.

Chain of length *k*, between *m* states

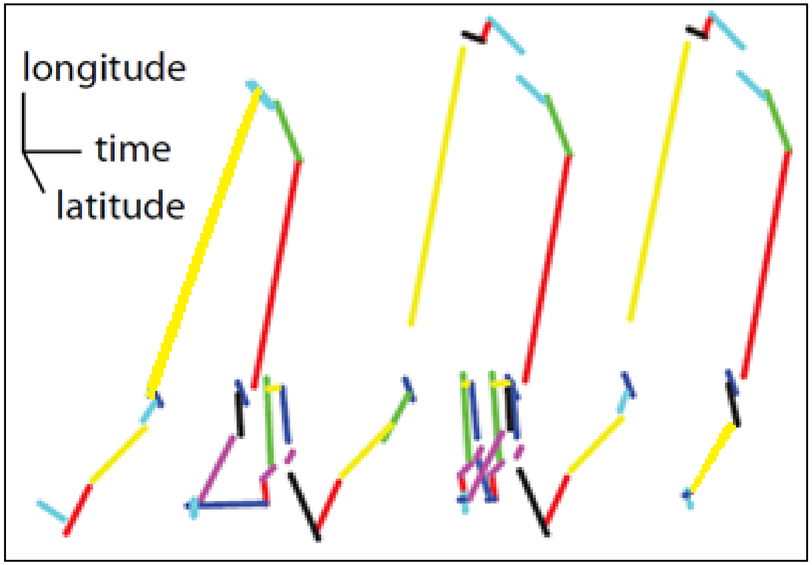


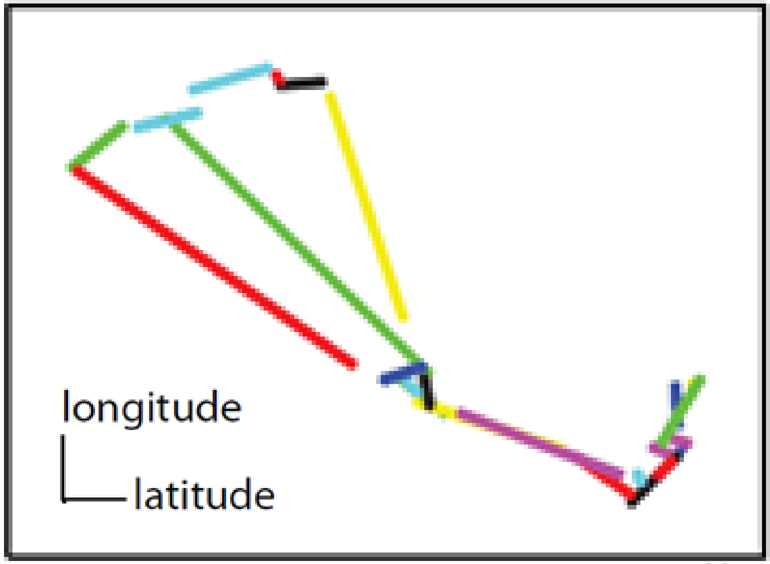
abcdefghihgfedcba…

(k, m)-Hidden Markov Model

Minimizes cost over every k-segments

whose projection is only m segments



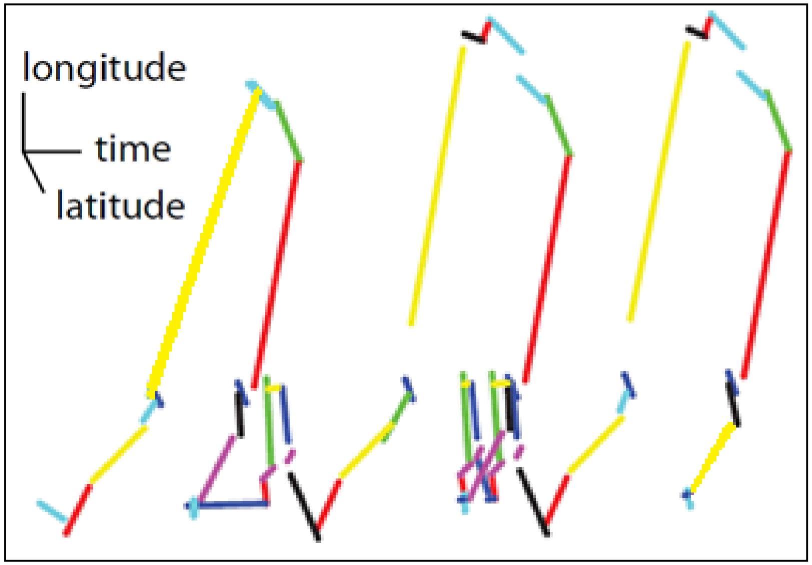


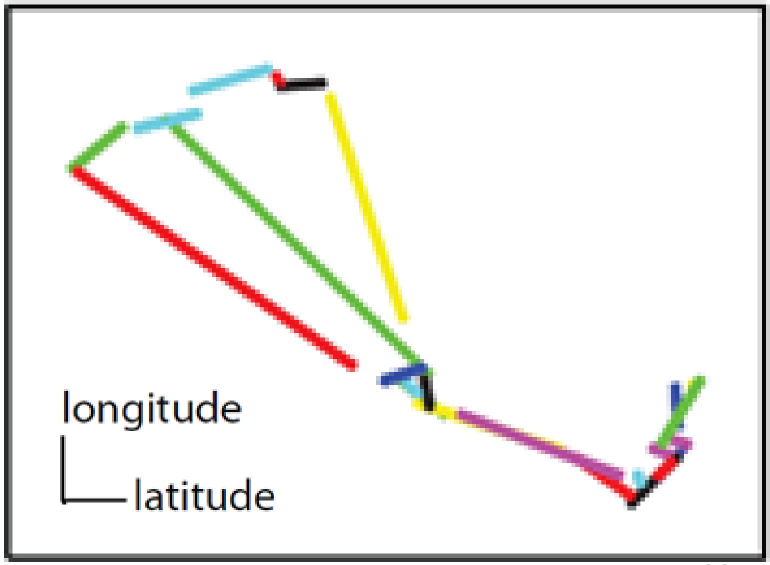
(k, m)-Hidden Markov Model

Minimizes cost over every k-segments

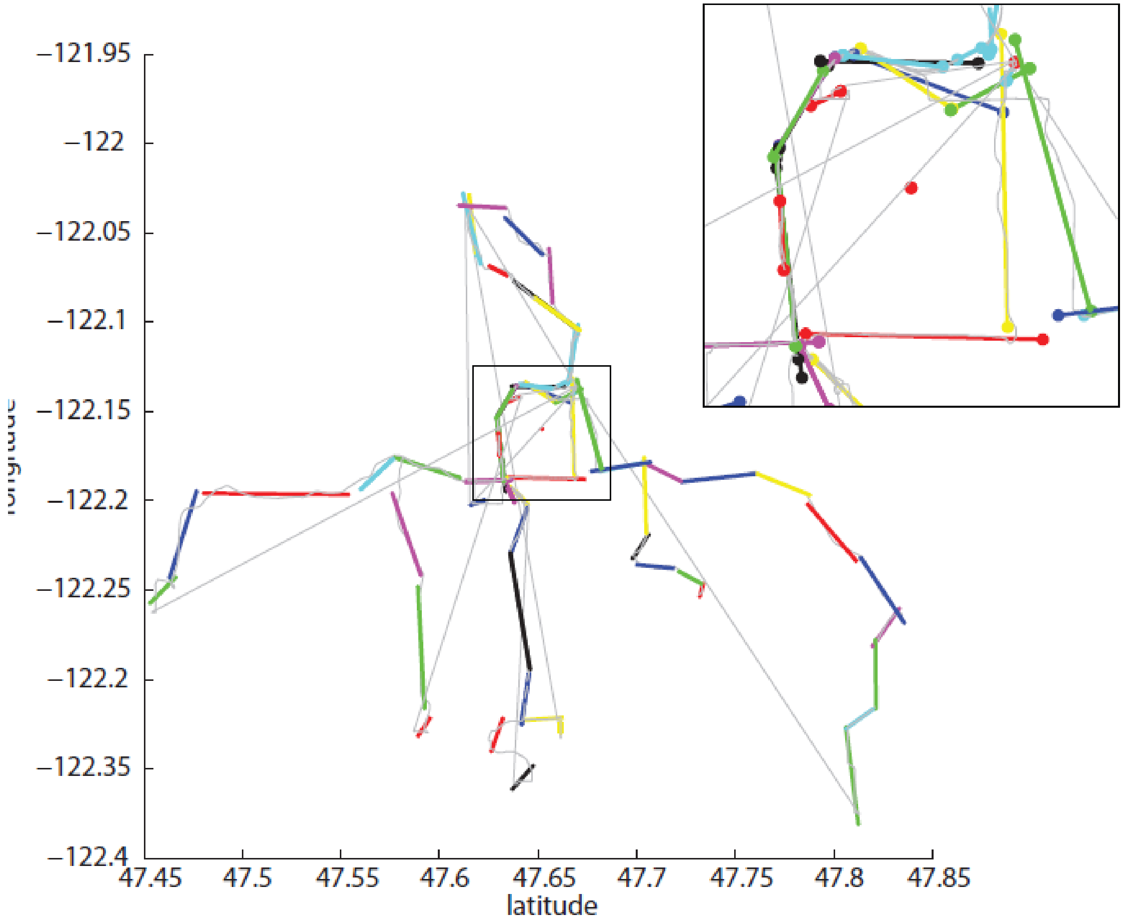
whose projection is only m segments

Observation: We can use the same coreset for k-segments !





Apply Heuristics for NP-hard problems on Coresets

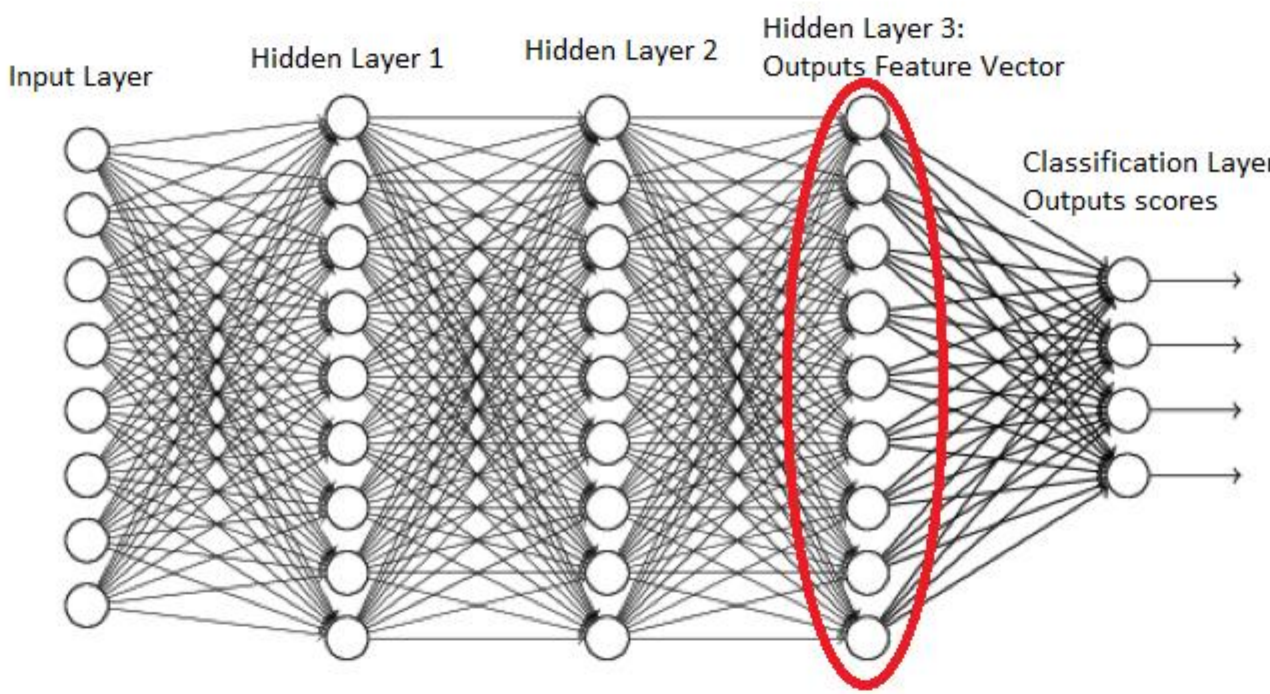


(91,70)-segments, SIGSPATIAL

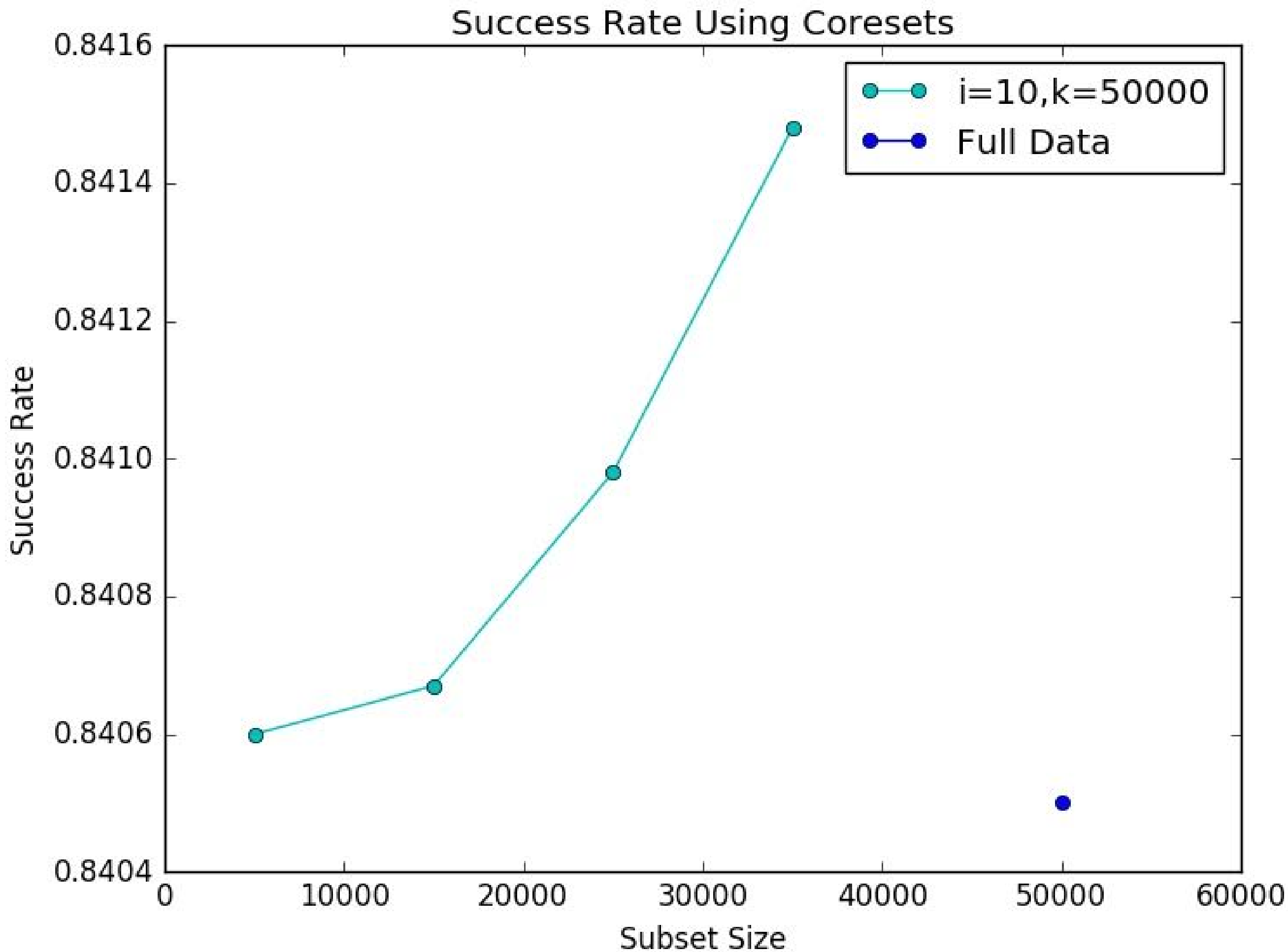
Coreset For Deep Learning

- Use existing coreset for the sigmoid active function: $f(p,x) = \frac{1}{1+e^{px}}$

- Technique: Improve each neuron independently



IMPROVED existing state-of-the-art using core-sets



Summary

Unified Coreset Framework

Hierarchies Class: Problems (functions) | Solutions (techniques) | Coreset Types

Data Models: streaming | distributed | kinematic | dynamic

Computation Models: Privacy | H. Encryption | GPU | Active Learning

Theory: Solve **central open problems** in: TCS, CG, ML, DL, HE, DP, …

Practice/ Industry:

Boost performance of existing systems

Novel practical solutions with provable guarantees

Open Implementation: software | hardware | systems

Applications




Open Problems

- More Coresets
  - Deep learning, Decision trees, Sparse data
- More Applications
  - Signals, Robotics, FFT, Computer Vision, DL
- Private Coresets
  [STOC’11, with Fiat, Nissim and Kaplan]
- Homomorphic Encryption: [F, Akavia, Kaplan]
- Generic software library
  - Coresets on Demand on the cloud
- Sensor Fusion (GPS+Video+Audio+Text+..)