Fast Clustering leads to Fast SVM Training and More

Daniel Boley

University of Minnesota

Supported in part by NSF

NSF grants 0208621 & 053486

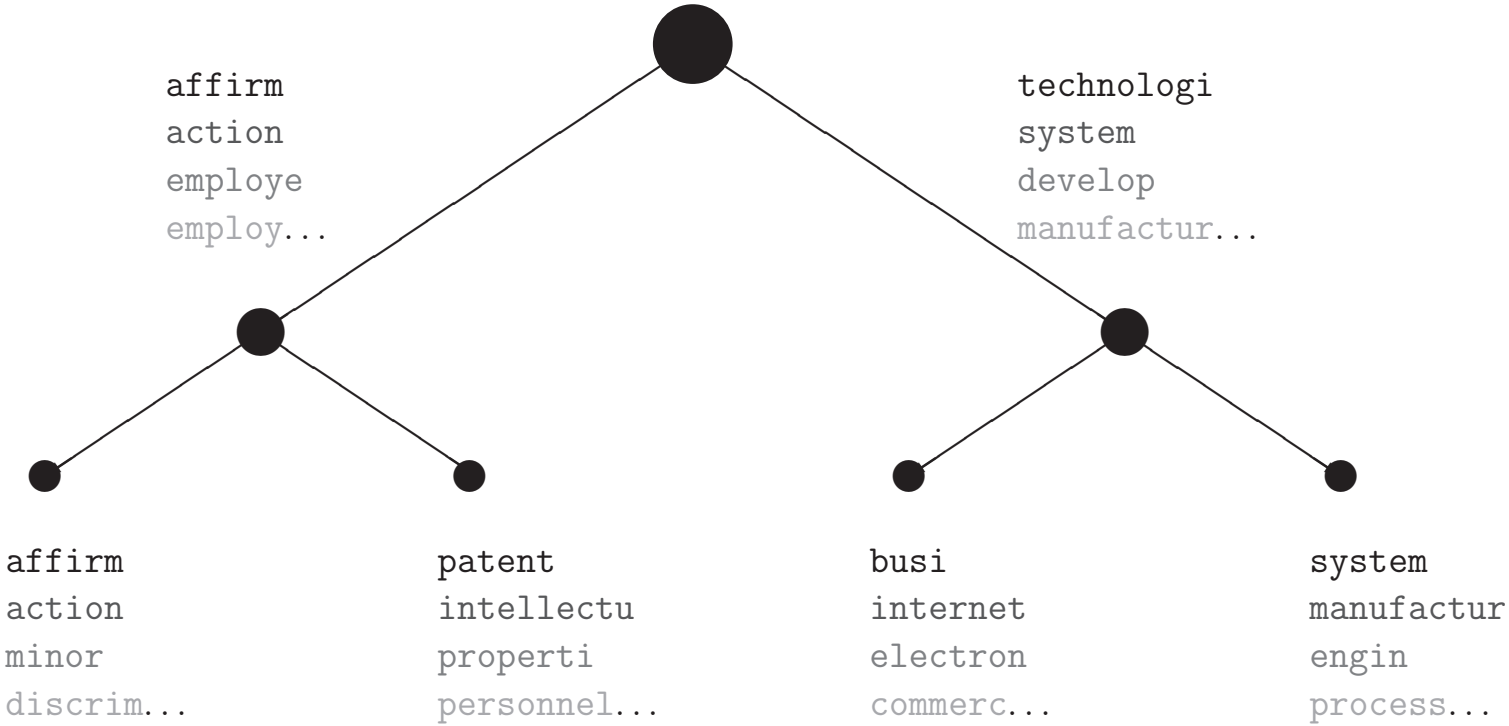
Goals and Outline

- Existence of Fast Clustering methods makes possible several applications.
  - Compare deterministic and non-determ. clusterers.
- Fast training of Support Vector Machines.
- Low Memory Factored Representation,
  for data too big to fit in memory.
  - Fast clustering of datasets too big to fit in memory.
  - Fast generalization of LSI for document retrieval.
  - Representation of Streaming Data.

Hierarchical Clustering

- Clustering at all levels of resolution.
- Bottom-up clustering is $O(n^2)$.
- Top-down clustering can be made $O(n)$.
- Leads to PDDP. [basis of this talk].

Hierarchical Clustering: Get a Tree



K-means: Popular Fast Clustering

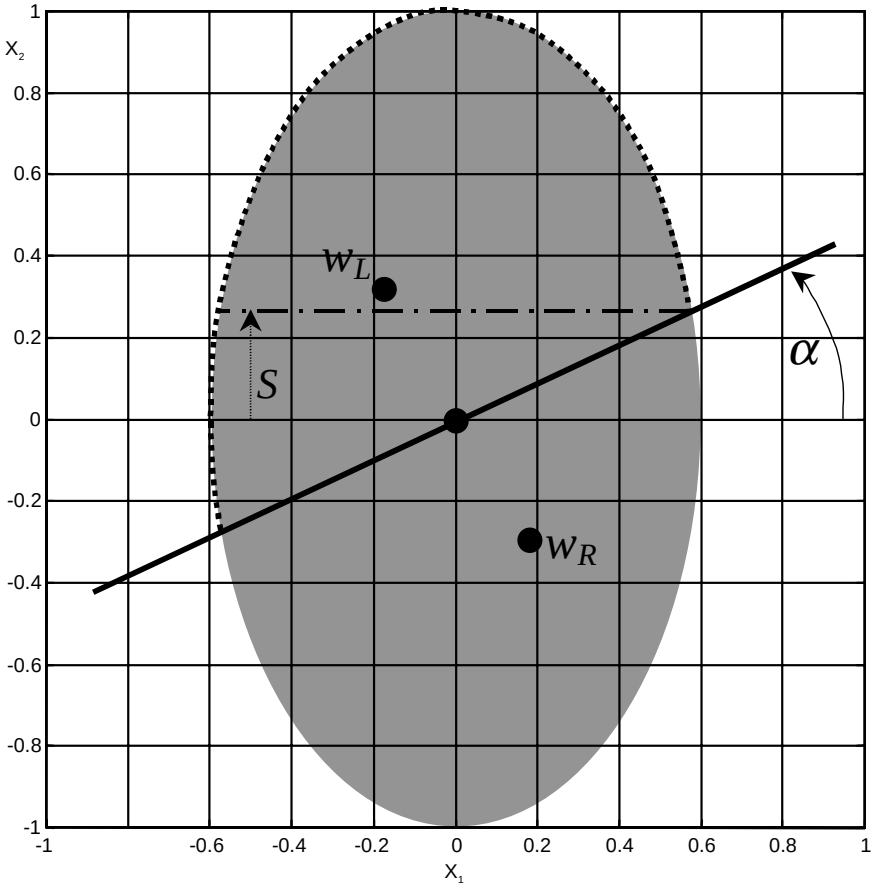
- Quality of final result depends on initialization
- Random initialization $\Rightarrow$ results hard to repeat.
- Deterministic initialization - no universal strategy
- Cost: $O(\#\text{iters} \cdot m \cdot n) \Rightarrow$ linear in n.

  where n = number of data samples

  m = number of attributes per sample.

Modelling K-means Convergence

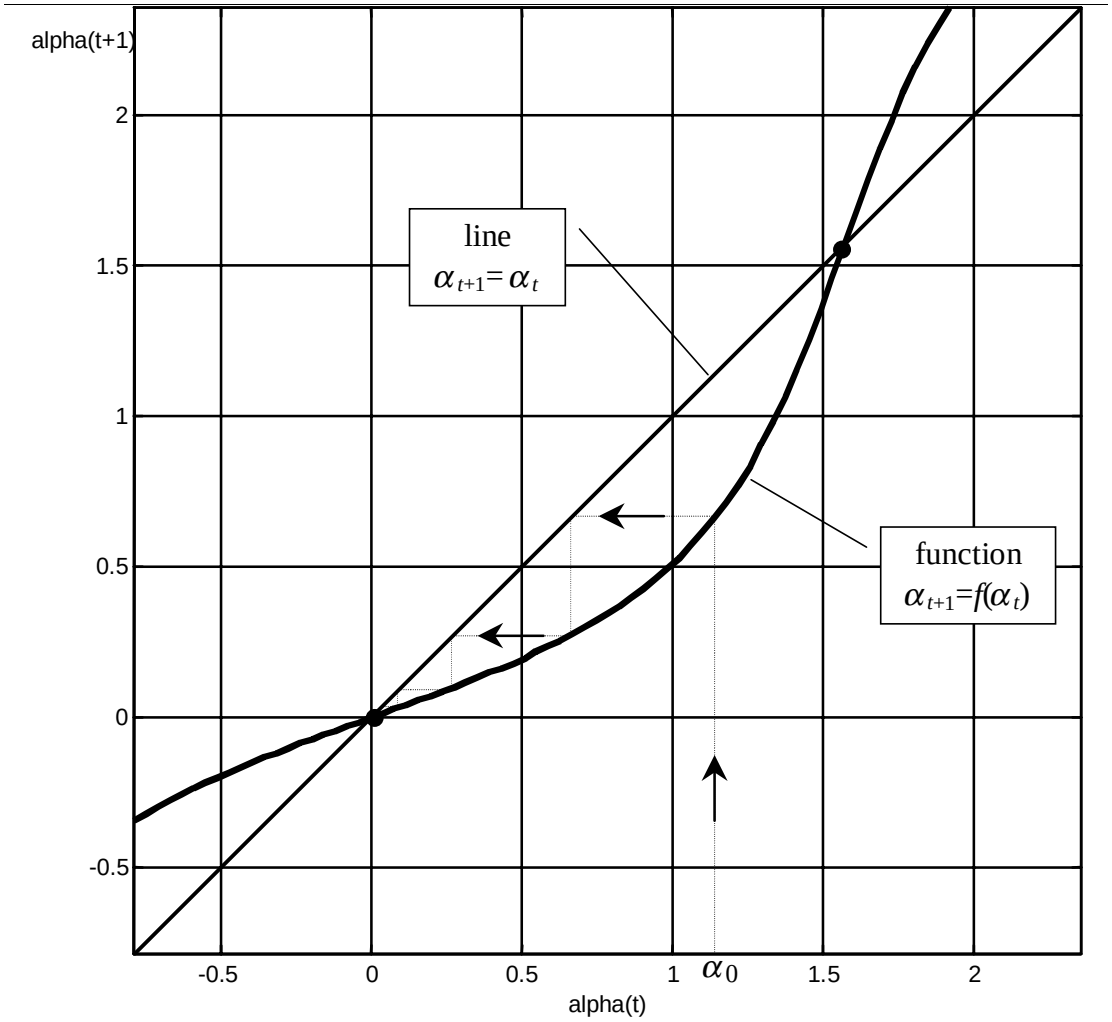
[Savaresi]



Simple Model

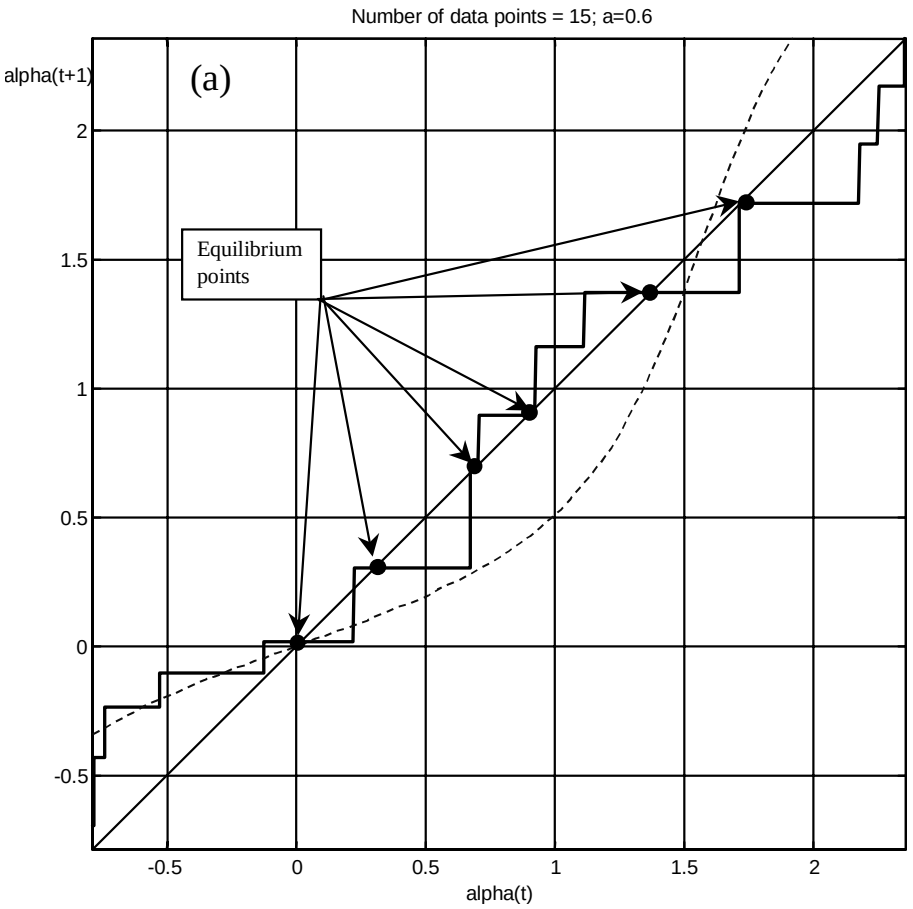
- Reduce to 1 parameter: angle α.
- Major axis $= 1$, Minor axis $= a < 1$.
- Non-linear dynamic system: $\alpha_{t+1} = \text{atan}[a^2 \tan \alpha_t]$.
- # iterations to converge: $\approx \quad -1/\log a^2$.

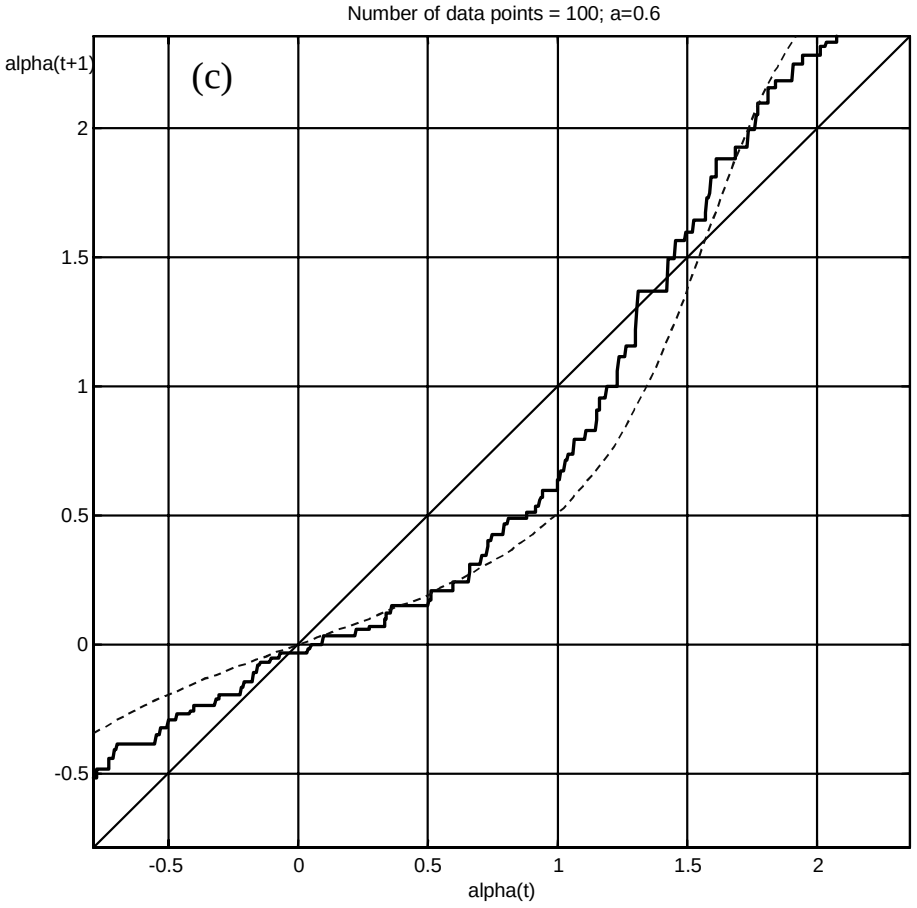
Infinitely Many Points



K-means modelled as a fixed point iteration

Finite Number of Points



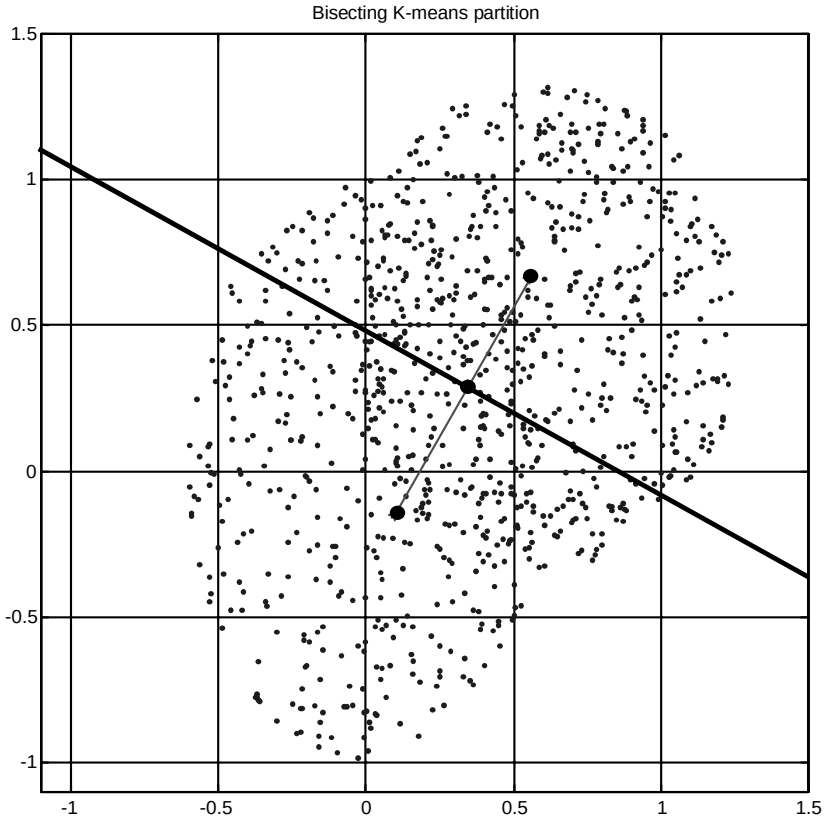


Finite Number of Points

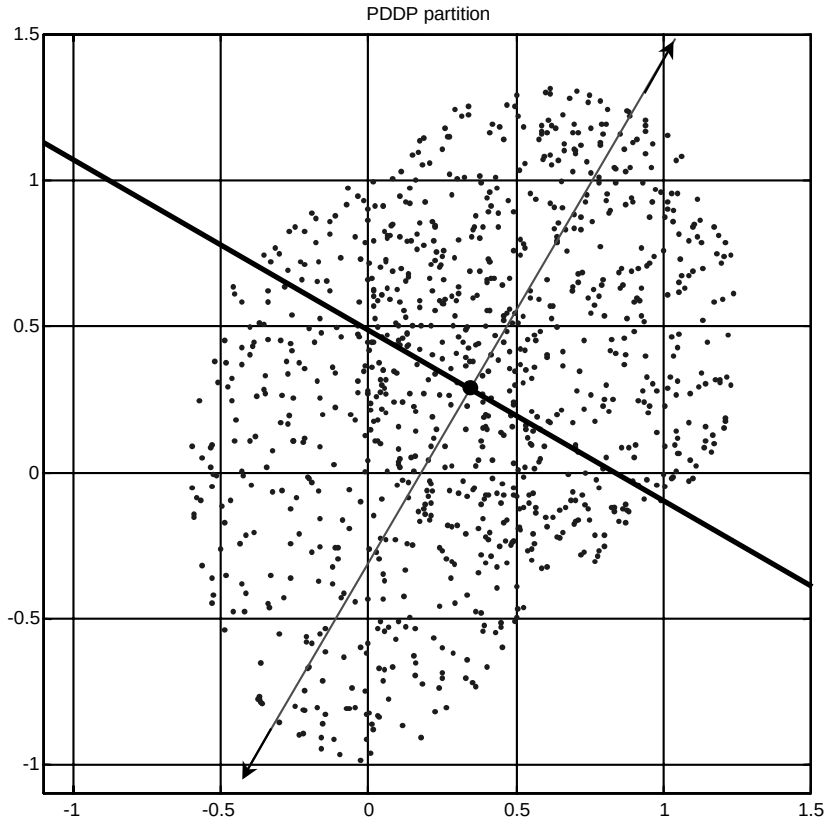
- Many equilibrium points $\Longrightarrow$ many local minima.
- As # points grows, local minima tend to vanish.
- As minor axis $\rightarrow 1$, more local minina tend to appear.

PDDP vs K-means on Model Problem

- In the limit, PDDP & K-means yield same split here. [Savaresi]



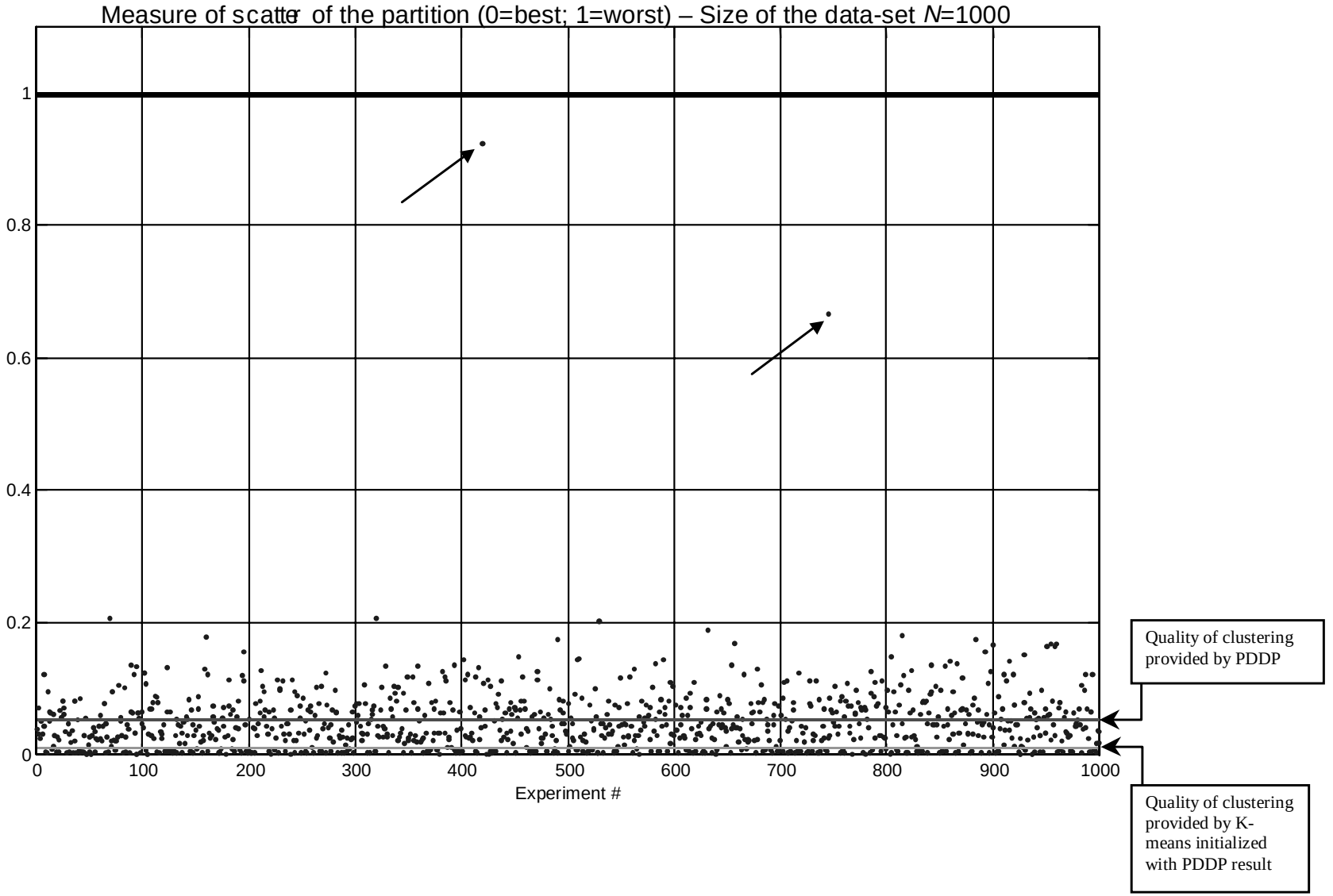
(a)



(b)

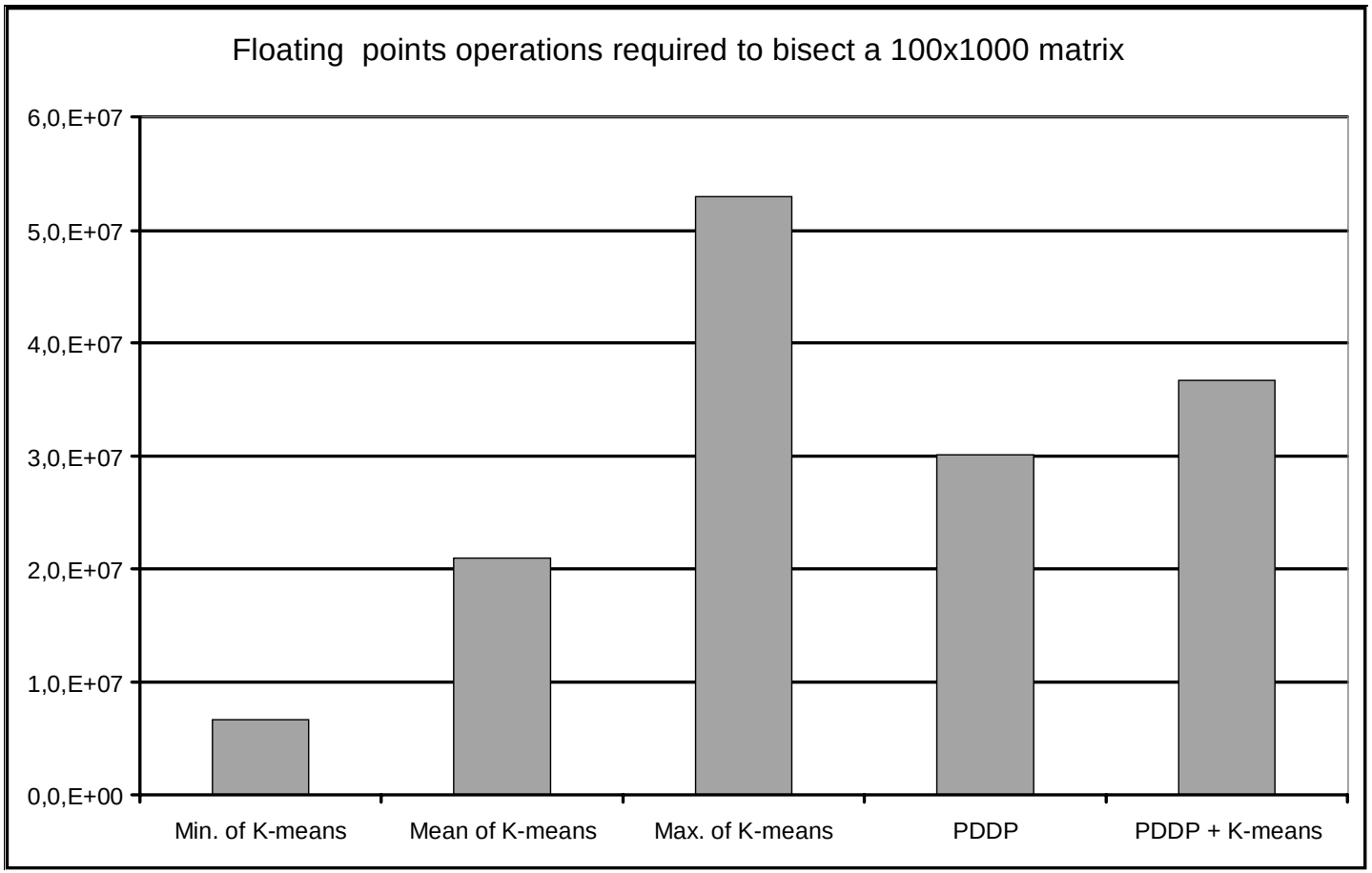
Starting K-means

- Empirically, PDDP is a good seed for K-means.



Cost of K-means vs PDDP

- Both are linear in the number of samples.
- K-means often cheapest, but cost can vary a lot.

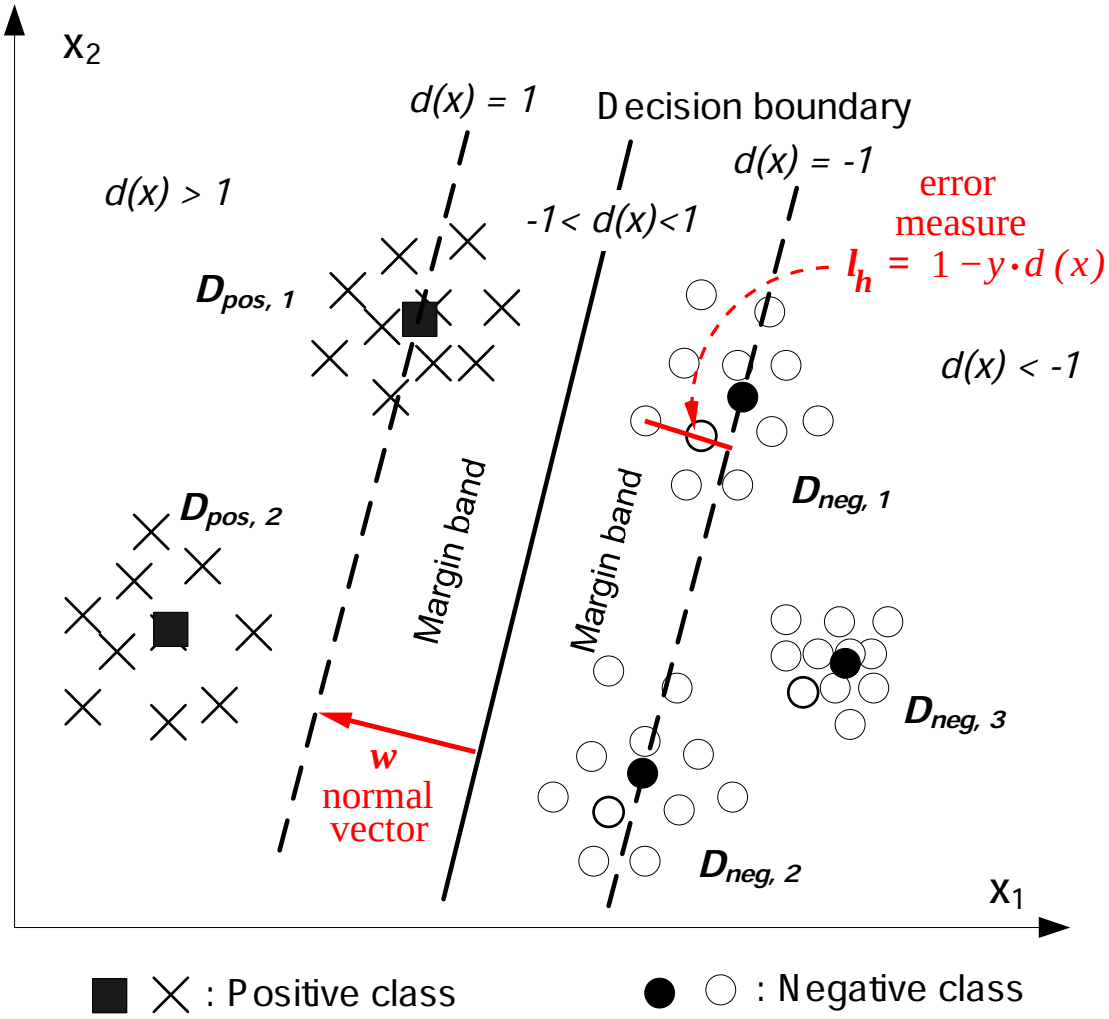


SVM via Clustering

- Motivation: Reduce trainging cost by clustering and use one representative per cluster instead of all the original data.
- Empirically, the resulting approximate SVM has comparable error rates over test sets as the exact SVM.
- Theoretically, A PAC-style generalization bound is proved for the approximate SVM. obtained using all the data.

SVM via Clustering

- Cluster Training Set into partitions
- Train SVM using 1 representative per partition.



Support Vector Machine

- Minimize $R\left(d;\mathcal{D},\lambda\right) = \underbrace{R_{\text{emp}}(d;\mathcal{D})}_{\substack{\text{Empirical}\\ \text{Error}}} + \underbrace{\lambda\cdot\Omega(d)}_{\substack{\text{Regularization/}\\ \text{Complexity Term}}}$

- $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$: training set.
- $\mathbf{x}_i$: datum w/ label $y_i = \pm 1$.
- $\phi(\mathbf{x})$: non-linear lifting.
- $d(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x})\rangle$: discrimin. fcn.
- λ: regularization coefficient
- $\Omega(d) = \|\mathbf{w}\|^2$

- $$\begin{aligned} R_{\text{emp}}(d;\mathcal{D}) &= \frac{1}{n}\sum_{(\mathbf{x},y)\in\mathcal{D}} \ell_{\text{hinge}}(d,(\mathbf{x},y)) \\ &= \frac{1}{n}\sum_{(\mathbf{x},y)\in\mathcal{D}} \max\{0, 1 - y\cdot d(\mathbf{x})\} \end{aligned}$$

Questions to be Resolved

- Assumptions
  - All clusters are pairwise disjoint;
  - All data in the same cluster belong to the same class;
  - There is one representative for each cluster;
- Questions to be resolved
  - How to choose good representatives quickly?
  - What is the generalization performance of the approximate SVM?

Approximate SVM Methods

Ideal Choice of Clustering Method

- Intuition: Good representatives should minimize the difference between the exact and approximate SVMs.
- $\Longrightarrow$ The representatives are the feature-space center given by applying kernel K-means to data of each class 1, -1 separately.
- Problem with kernel K-means
  - Expensive.
  - Local minimum.

Practical Approx SVM Methods

- Solution: Define the representative of a cluster as its feature-space center, but partition $\mathcal{D}$ using a fast clustering method, such as
  - Data K-means (natural choice)
  - Data PDDP (to make deterministic or to init K-means)

Quality of Approx SVMs – Theory

- Could apply VC dimension bounds,
  but we want something tighter.
- Extend Algorithmic-Stability bounds to this case.
  These apply specifically to learning algorithms minimizing some convex functional, whose change is bounded when the training data set is purturbed by substituting one training datum.

Approximate SVMs – Assumptions

- The proved PAC-style bound assumes that all data of a cluster belong to the same class, and the representative is its feature-space center
- There is no assumptions on how to partition the training dataset, so the result applies even when using data K-means, data space PDDP, random partitioning, or even a sub-optimal sol'n from kernel K-means.

Stability Bound Theorem

Get theorem much like one for Exact SVM:

- For any $n \geq 1$, $\delta \in (0,1)$, and randomly drawn training set $\mathcal{D}$ of size n, we have with confidence at least $1-\delta$:

$$\underbrace{\mathbb{E}(\mathbb{I}_{\widetilde{h}(\mathbf{x}) \neq y})}_{\text{expected error}} \leq \underbrace{\frac{1}{n} \sum_{(\mathbf{x},y) \in \mathcal{D}} \ell_{\text{hinge}}(\widetilde{h}, \mathbf{x}, y)}_{\text{empirical error}} + \underbrace{\frac{\chi^2}{\lambda n} + \left(\frac{2\chi^2}{\lambda} + 1 \right) \sqrt{\frac{\ln 1/\delta}{2n}}}_{\text{complexity/sensitivity term}} .$$

  - $\widetilde{h}(\mathbf{x}) \stackrel{\text{def}}{=} \text{sign}\,\{\widetilde{d}(\mathbf{x})\}$ is the approximate SVM.
  - $\chi^2 = \max_i K(\mathbf{x}_i, \mathbf{x}_i) = \max_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_i) \rangle$ (1 for RBF kernel).
  - λ is the regularization coefficient.
    trade-off of training error $\longleftrightarrow$ sensitivity.

Experimental Results

- Illustrate performance of approximate SVMs over some datasets.
- We compare the proposed algorithm against the standard training algorithm SMO [Platt, 1999], implemented in LibSVM [Chang+Lin 2001] [Fan 2005];

Experimental Setup

- We partition the training dataset in three steps
  1. Partition the training dataset using PDDP;
  2. Train an approximate SVM h' using the data-space center of resulting clusters;
  3. Refine partition in step 1 using h' by ensuring that a cluster contains support vectors only, or non-support vectors only.
  4. Train a new SVM $\tilde{h}$ using feature space centers of subclusters from step 3.

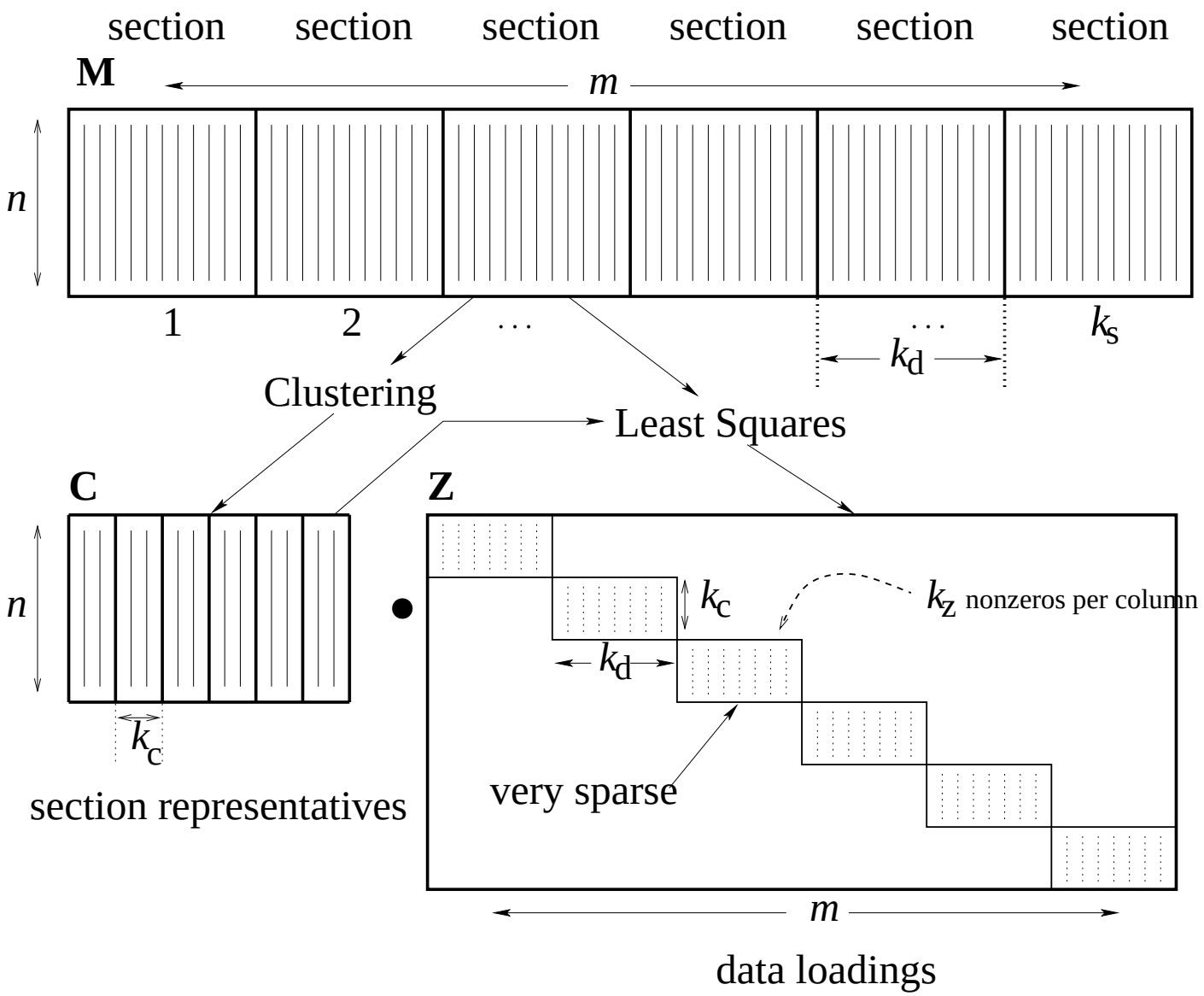
Experimental Performance

Data set (Size)	Exact SVM		Approximate SVM	
	T_{train} (sec.)	Accuracy	T_{train} (sec.)	Accuracy
UCI-Adult (32,561)	$1,865$	95.7%	281	94.2%
UCI-Web (49,749)	$2,894$	99.8%	508	99.5%
MNIST (60,000)	$6,534$	98.8%	$2,825$	95.6%
Yahoo (100,000)	$18,161$	83.8%	$2,853$	82.0%

Low Memory Factored Representation

- Use clustering to contruct a representation of a full massively large data sets in much less space.
- Representation is not exact, but every individual sample has its own unique representative in the approximate representation.
- In principle, would still allow detection and analysis of outliers and other unusual individual samples.
- Next slide has basic idea.

Low Memory Factored Representation



Fast factored representation: LMFR

[Littau]

- $\mathbf{M} = \mathbf{CZ}$ by fast clustering of each section
- $\mathbf{C}$ = matrix of representatives
- Still have $\mathbf{Z}$ to individualize representation of each sample
- Make $\mathbf{Z}$ sparse to save space.
- linear clustering cost $\Rightarrow$ linear cost to construct LMFR
- In principle, could use any fast clusterer.
- We use PDDP to make it more deterministic.

LMFR $\Rightarrow$ Clustering $\Rightarrow$ PMPDDP

Using PDDP on an LMFR yields Piece-Meal PDDP.

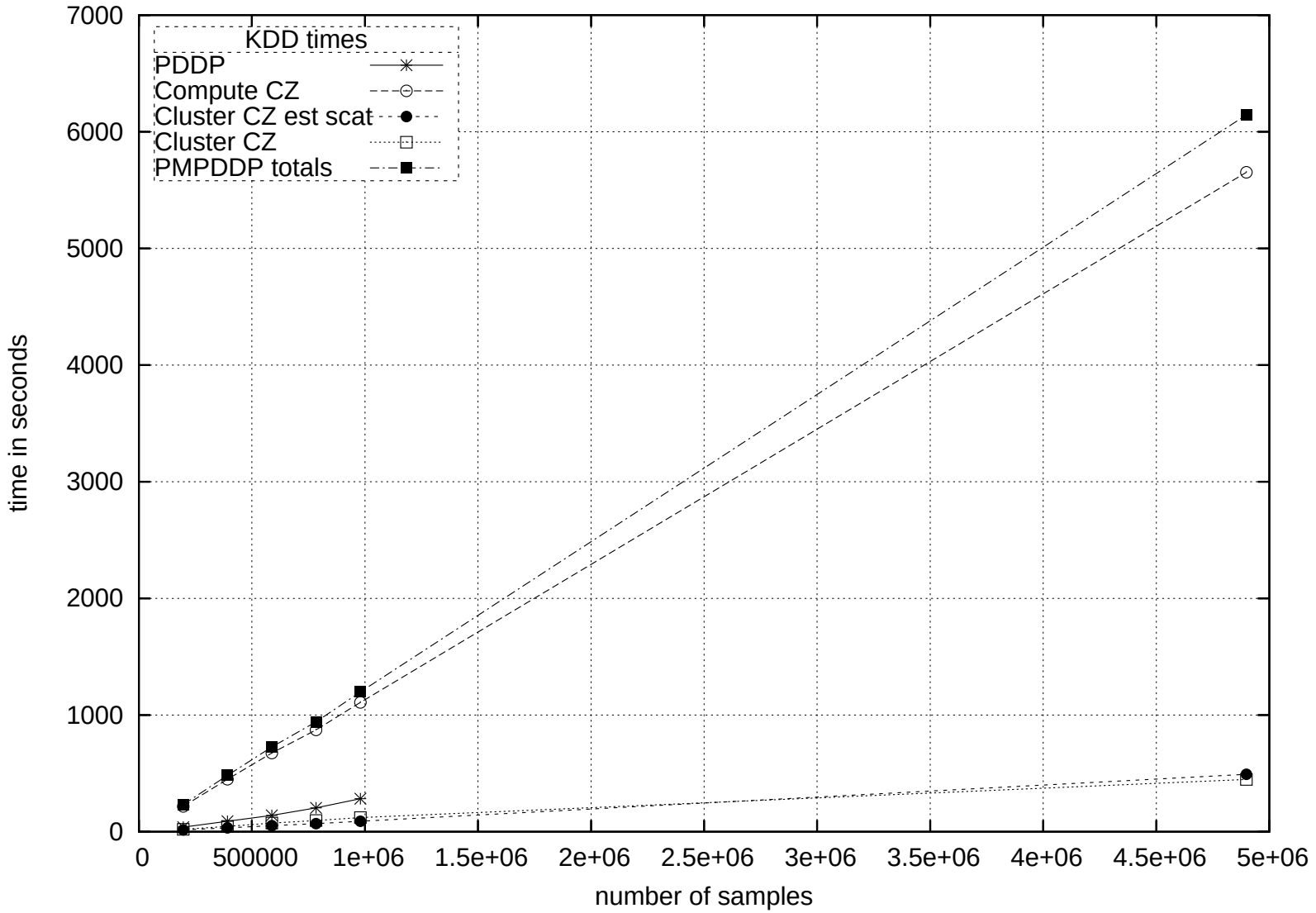
- Factored Representation $\Rightarrow$ to reconstruct data
- Expensive to compute similarities between individual data.
- Want to avoid accessing individual data.
- Ideal for clusterer that depends on $\mathbf{M} \times \mathbf{v}$'s
- A spectral clustering method like PDDP is a good fit.
- Experimentally, cluster quality $\approx$ plain PDDP.

$\Rightarrow$ PMPDDP - Piece-Meal PDDP

- Divide original data $\mathbf{M}$ up into sections
  Extract representatives for each section, fast.
  [can be imperfect]
- Matrix of representatives $\Rightarrow \mathbf{C}$
- Approximate each original sample as a linear combination of k representatives [selected via nearest neighbor].
- Matrix of coefficients $\Rightarrow \mathbf{Z}$
- k is a small number like 3 or 5.
- Apply PDDP to the product $\mathbf{CZ}$ instead of original $\mathbf{M}$.
  [never multiply out $\mathbf{CZ}$ explicitly]

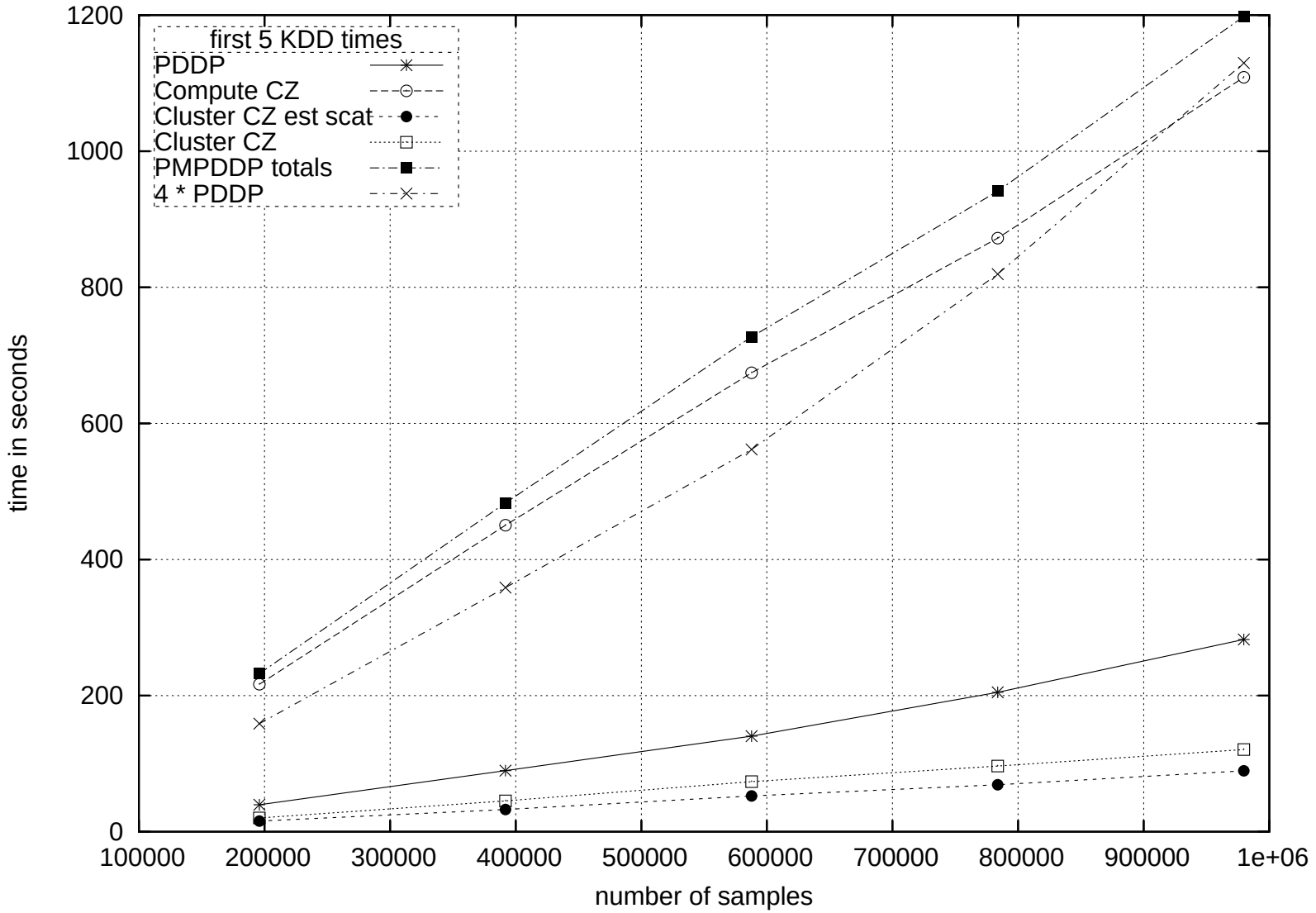
PMPDDP – on KDD dataset

- Still Linear in size of data set.



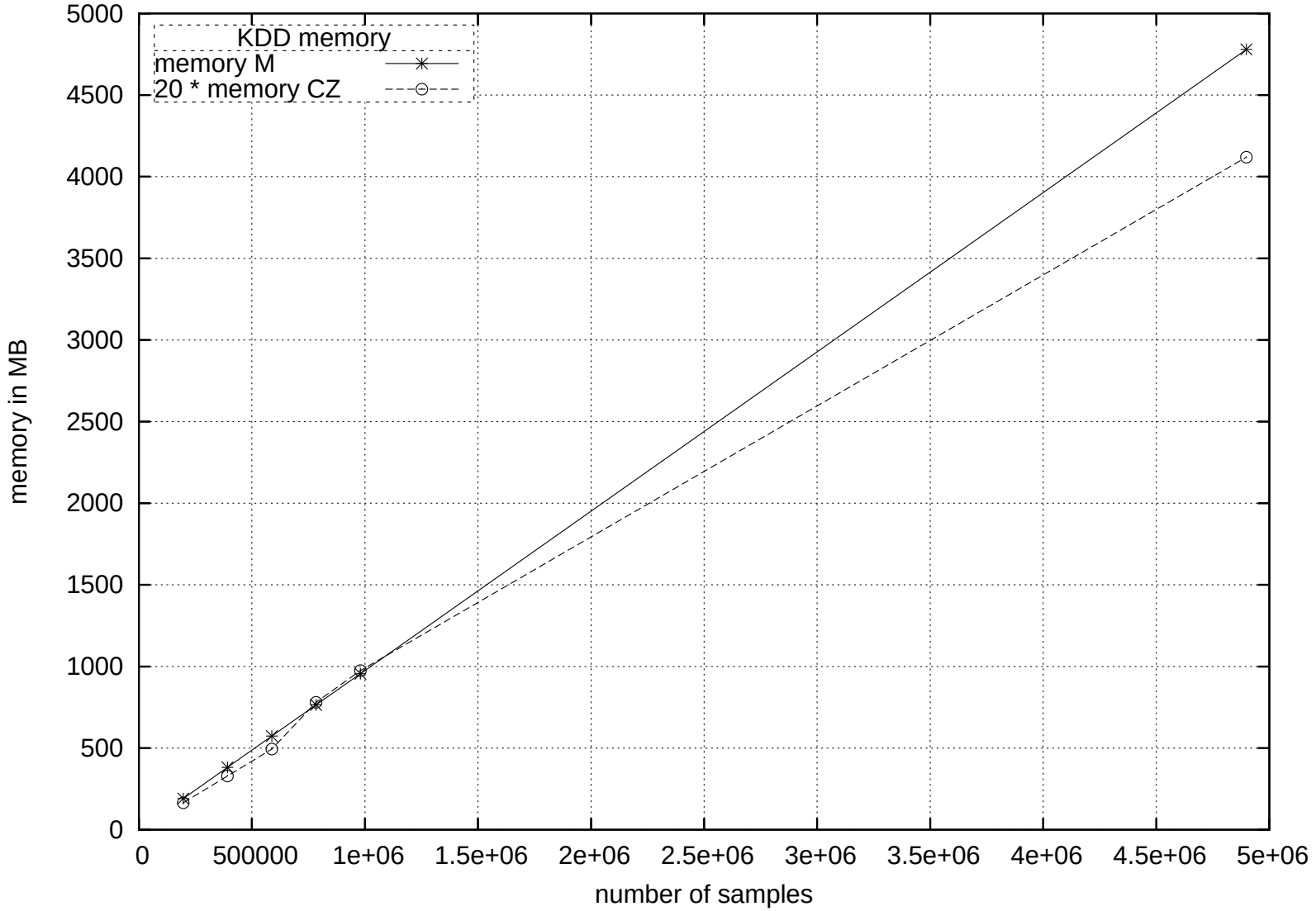
PMPDDP – on KDD dataset

- First 5 samples: PMPDDP cost $\approx 4 \times$ PDDP.



PMPDDP – on KDD dataset

- Memory usage small.



LMFR for Document Retrieval

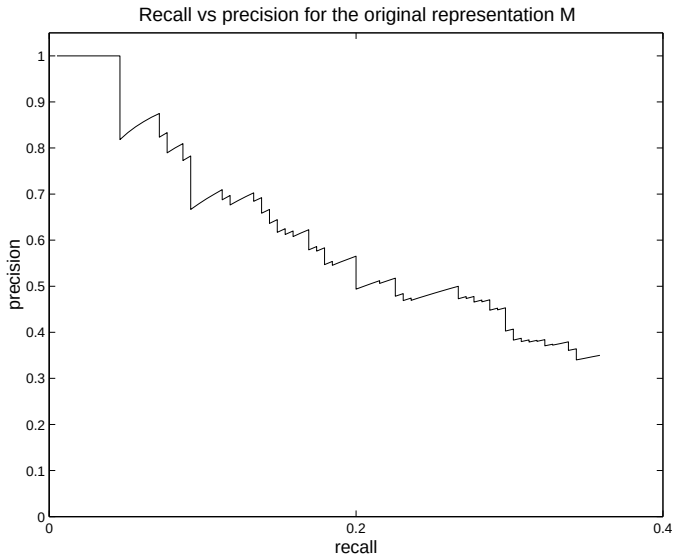
- Mimic LSI, except we use factored representation $\mathbf{CZ}$.
- Different from finding nearest concepts (ignoring $\mathbf{Z}$)
- Can handle much larger datasets than Concept Decomposition [full $\mathbf{Z}$]
- Less time needed to achieve similar retrieval accuracy.

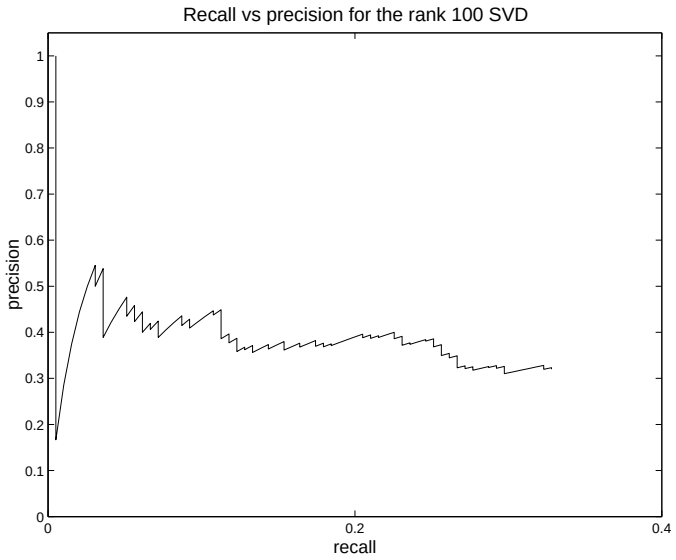
Doc Retrieval Experiments

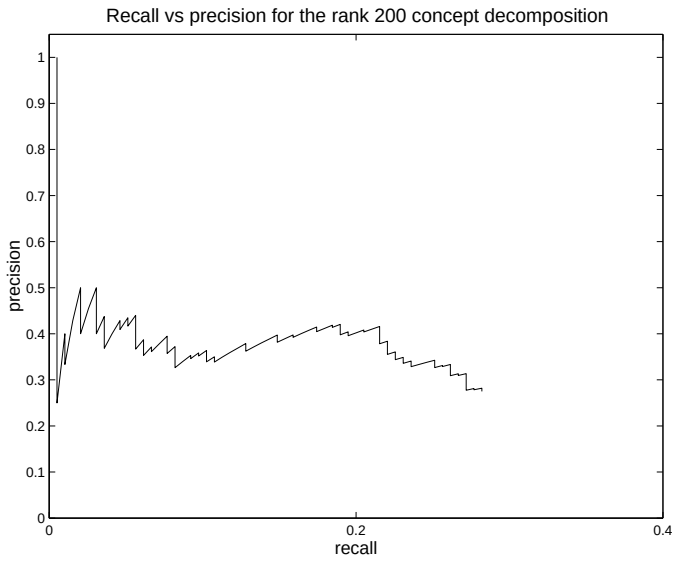
- Compare methods achieving similar retrieval accuracy.

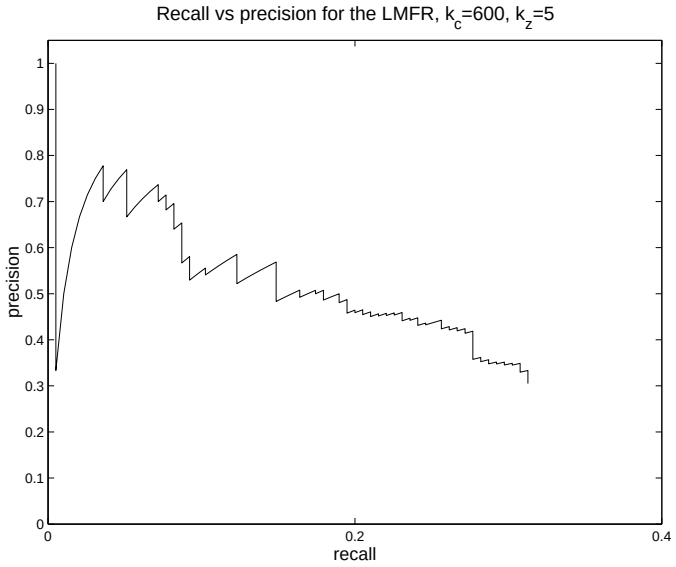
method	k_c	k_z	MB	sec
Data Matrix **M**	−−	−−	18.34	−−
rank 100 SVD	−−	−−	40.12	438
rank 200 concept decomposition	200	200	25.88	10294
LMFR	200	5	8.10	185
LMFR	300	5	9.17	188
LMFR	400	5	10.02	187
LMFR	500	5	10.68	189
LMFR	600	5	11.32	187

Doc Retrieval Experiments









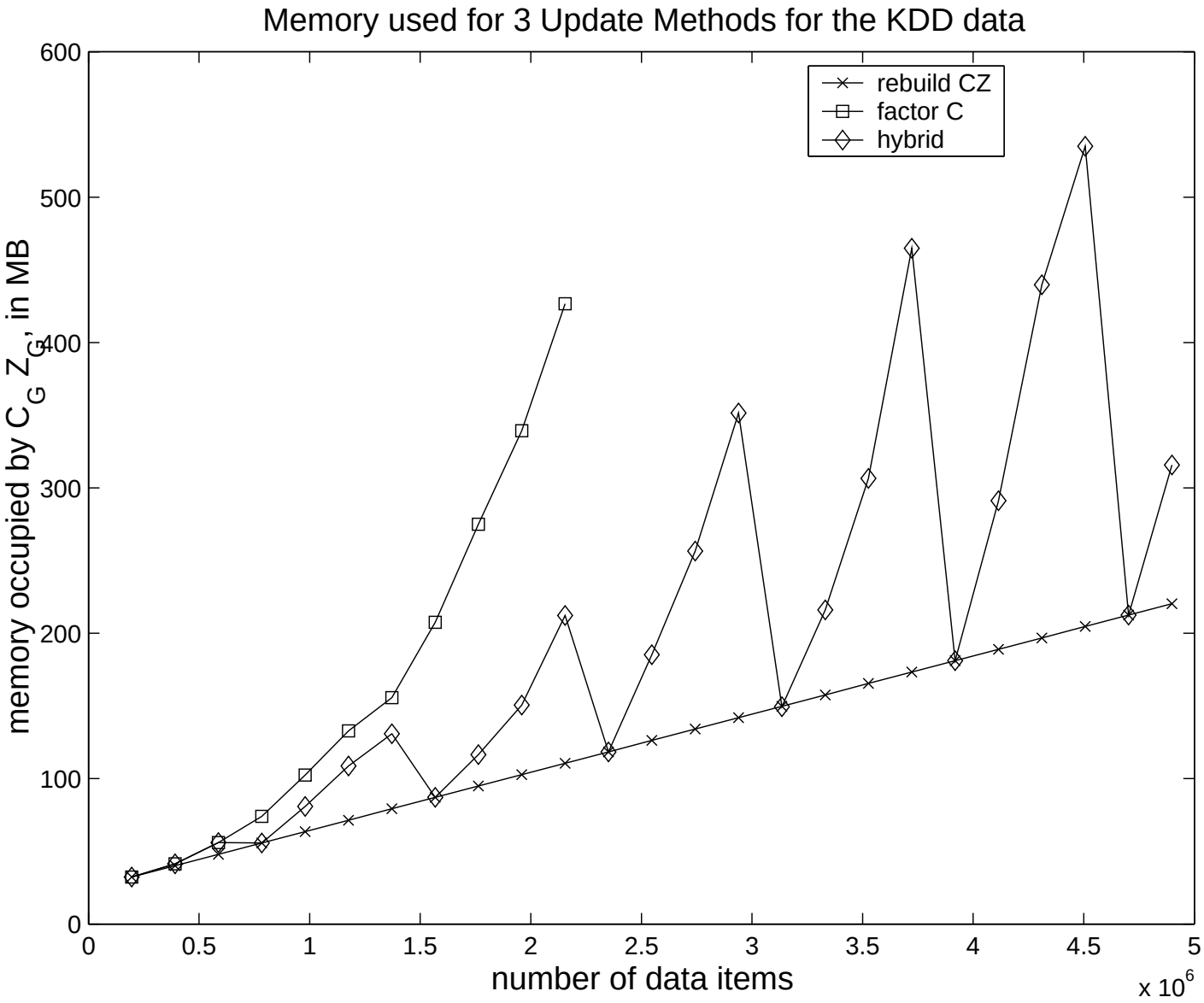
LMFR for Streaming Data

- Simple idea: collect data into sections as they arrive
- Form **CZ** section by section as they fill.
- Get LMFR for data, useful for any application (clustering, IR, aggregate statistics,...]
- No need to decide application in advance

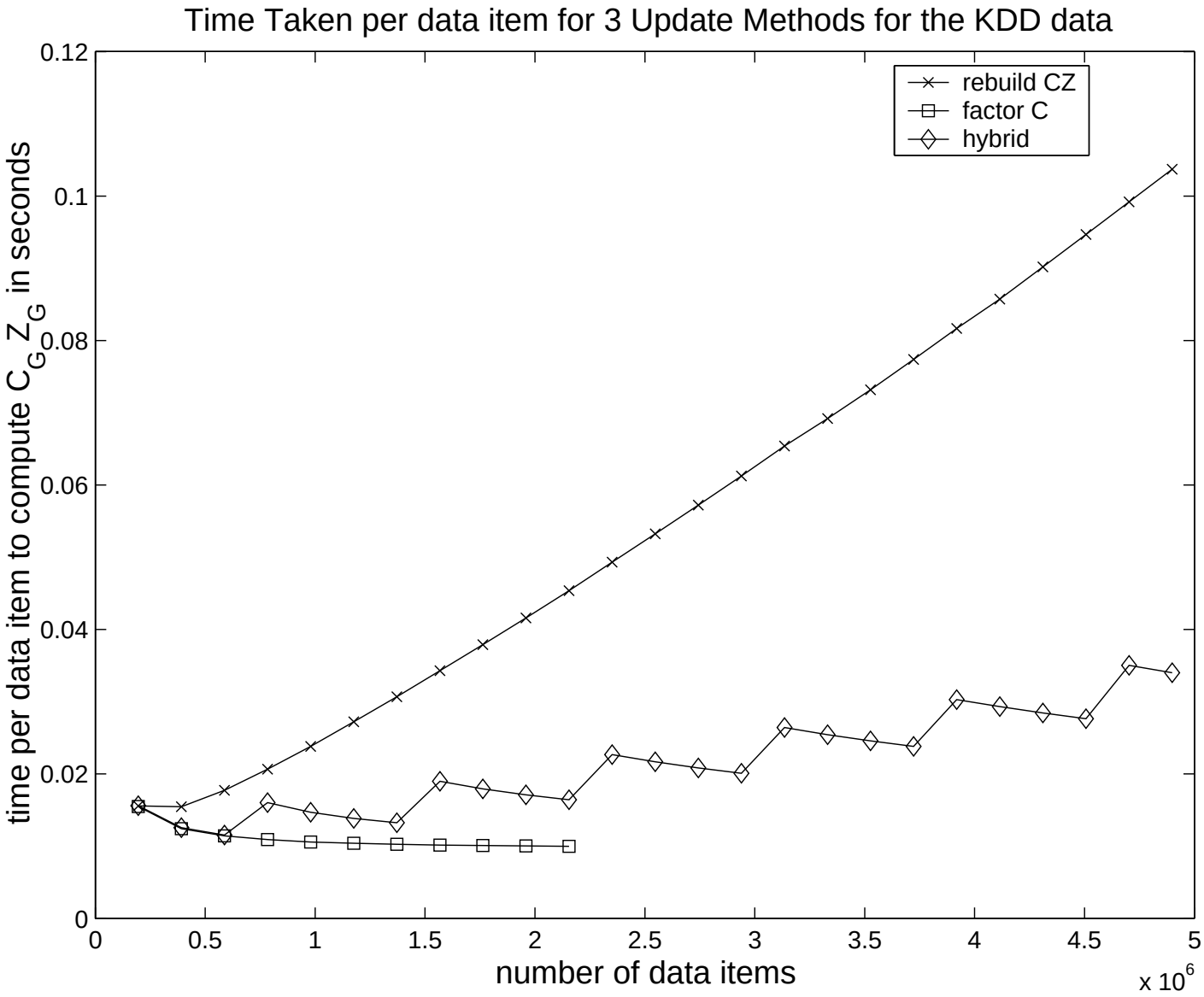
LMFR for Streaming Data

- Memory for $\mathbf{Z}$ grows very slowly
- Memory for $\mathbf{C}$ grows more.
- Recursively factor $\mathbf{C}$ into its own $\widehat{\mathbf{C}}\widehat{\mathbf{Z}} \Rightarrow$ less space.
- Hybrid Approach: once in a while do a completely new LMFR.

Streaming Data Results



Streaming Data Results



References

- Sergio M. Savaresi and Daniel Boley. A comparative analysis on the bisecting K-means and the PDDP clustering algorithms. *Intelligent Data Analysis*, 8(4):345–362, 2004.

- Dongwei Cao and Daniel Boley. A PAC bound for approximate support vector machines. In *SIAM Data Mining Conf. SDM 07*, 2007.
- Dongwei Cao and Daniel Boley. On approximate solutions to support vector machines. In *SIAM Data Mining Conf. SDM 06*, 2006.
- D. Boley and D.W Cao. Training support vector machine using adaptive clustering. SDM'04 Fourth SIAM Conference on Data Mining, April 2004.

- David Littau and Daniel Boley. Streaming data reduction using low-memory factored representations. *Journal of Information Sciences*, 176(14):2016–2041, 2006.
- D. Littau and D. Boley. Using low-memory representations to cluster very large data sets. SDM'03 Third SIAM Conference on Data Mining, May 2003.

Related Work

- SVM via Clustering
  - Chunking (Boser+92, Osuna+97, Kaufman+99, Joachims99)
  - Low Rank Approx (Fine 01, Jordan)
  - Sampling (Williams+Seeger01, Achlioptas+McSherry+Schölkopf 02)
  - Squashing (Pavlov+Chudova+Smith 00)
  - Clustering (Cao+04, Yu+Yang+Han 03)
- Agglomeration on large datasets
  - gather/scatter (Cutting+ 92)
  - CURE(Guha+98)
  - gaussian model (Fraley 99)
  - Heap (Kurita 91)
  - refinement (Karypis 99)

Related Work

- K-means on large datasets
  - Initialization (Bradley-Fayyad 1998)
  - kd-tree (Pelleg-Moore 1999)
  - Sampling (Domingos+01)
  - CLARANS k-medoid, spatial data (Ng+Han 94)
  - Birch (more sampling than k=means) (Ramakrishnan+96)
- Matrix Factorization
  - LSI Berry 95 Deerwester 90
  - Sparse LowRankApprox Zhang+Zha+Simon 2002
  - SDD (Kolda+98) – good for outlier detection (Skillikorn+01)
  - Monte-Carlo sampling (Vempala+98)
  - Concept Decomp (Dhillon+01)

Conclusions

- K-means Clustering
  - Convergence modelled by dynamical system.
  - Helped by seeding w/ deterministic method.
- Performance of fast SVM via clustering.
  - Speeded up in practice
  - Proved theoretical bound.

  See poster for details.
- Low Memory Factored Representation.
  - Cluster w/out computing pairwise distances.
  - Compact representation, easily updatable.
  - Ideally, would like clustering to be faster than linear.
  - Easily used for various applications: clustering, IR, streaming.