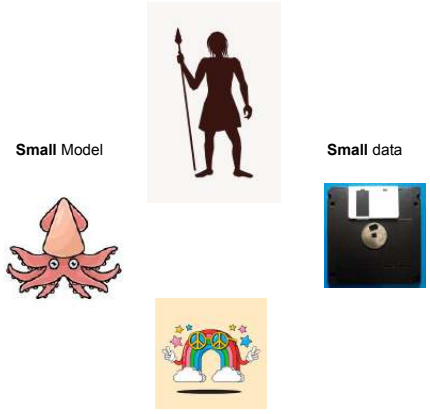
Modern Algorithmic for Modern Data Efficiency

Vincent Cohen-Addad

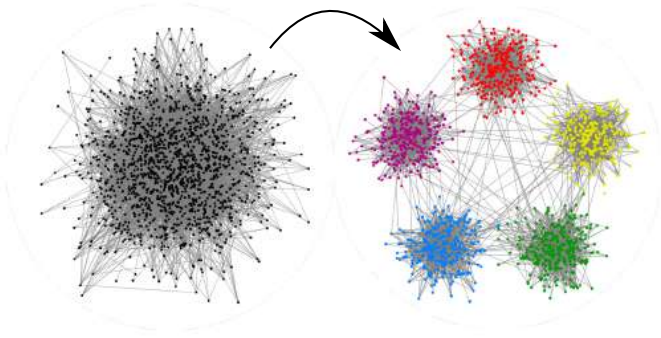
Google Research

Today's Menu

1. Embeddings Data



2. Graph Data



Data Efficiency

Small Model

Small data

Data Efficiency

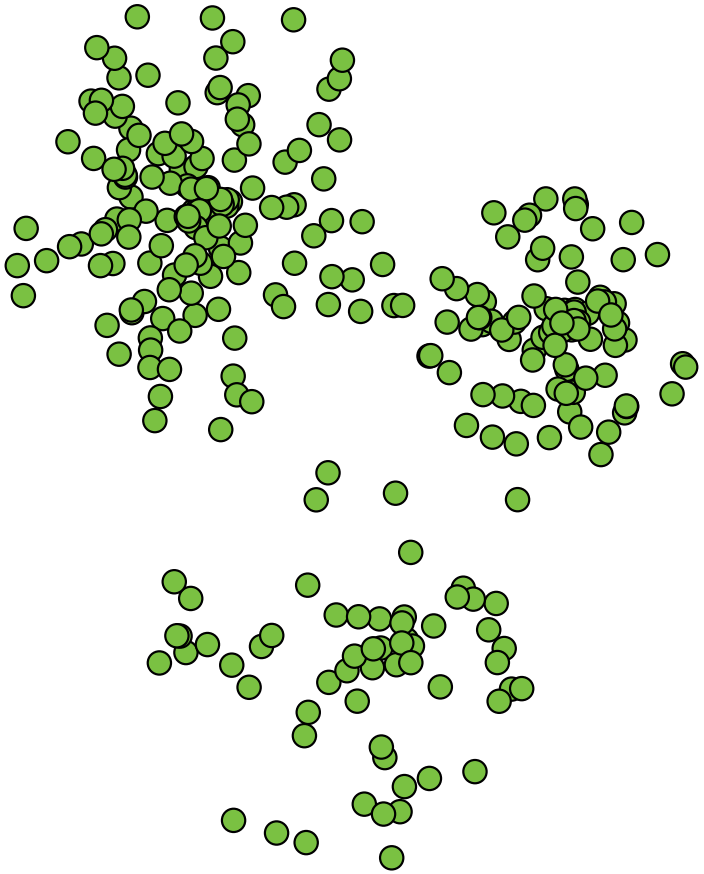
Small Model

Big Data

First Challenge

Reduce data size to speed data mining and model training up.

k-Means Clustering



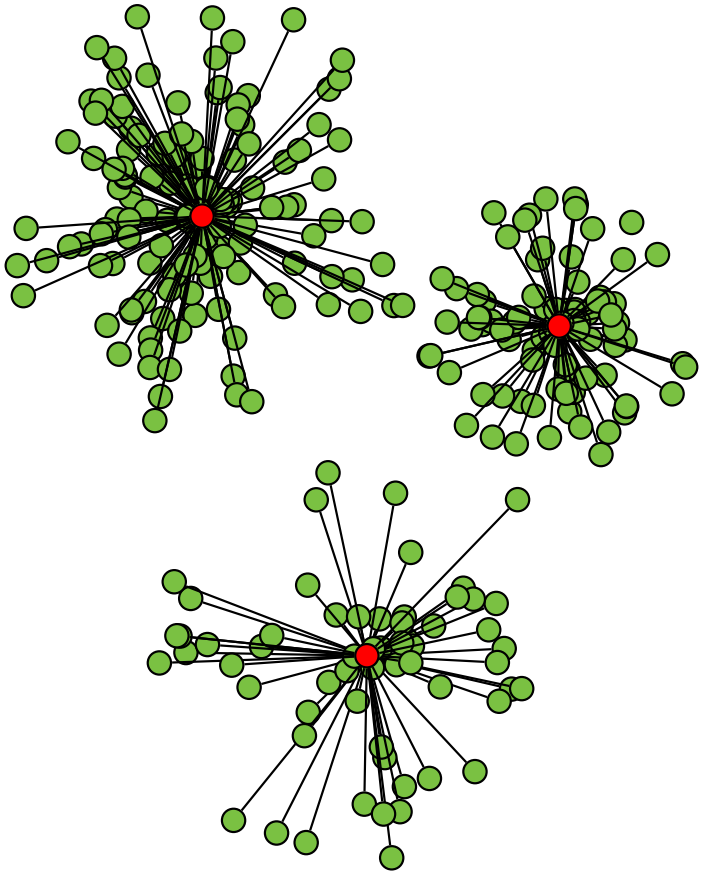
Problem definition

Input: Let A be a set of points in $\mathbb{R}^d$ and $k > 0$.

Output: k points (called centers) S minimizing

$$\text{cost}(A, S) := \sum_{p \in A} \min_{c \in S} \|p - c\|^2.$$

k-Means Clustering



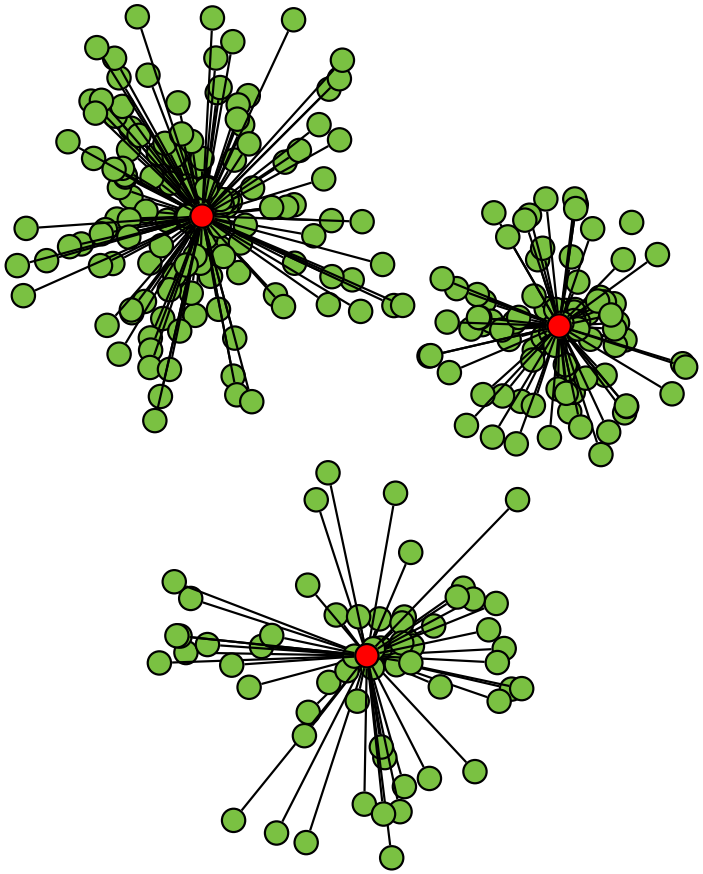
Problem definition

Input: Let A be a set of points in $\mathbb{R}^d$ and $k > 0$.

Output: k points (called centers) S minimizing

$$\text{cost}(A, S) := \sum_{p \in A} \min_{c \in S} \|p - c\|^2.$$

k-Means Clustering



Problem definition

Input: Let A be a set of points in $\mathbb{R}^d$ and $k > 0$.

Output: k points (called centers) S minimizing

$$\text{cost}(A, S) := \sum_{p \in A} \min_{c \in S} \|p - c\|^2.$$

First Challenge

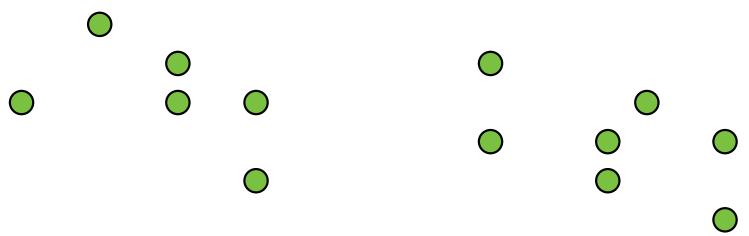
Reduce data size to speed data mining and model training up.

Coreset Definition

Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for all sets S of k centers it holds

$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$

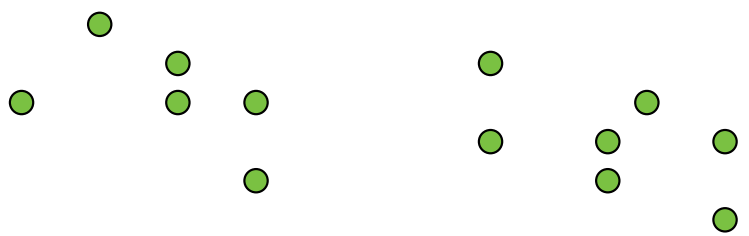


Coreset Definition

Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for all sets S of k centers it holds

$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$

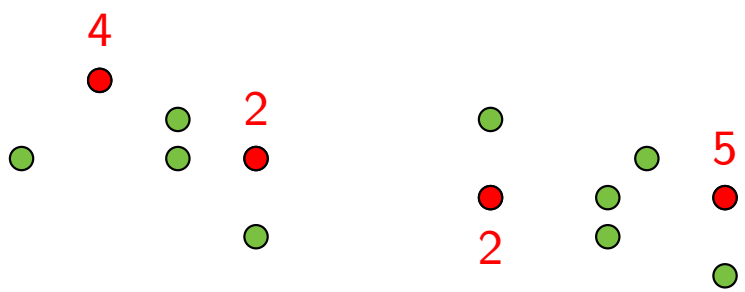


Coreset Definition

Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for all sets S of k centers it holds

$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$

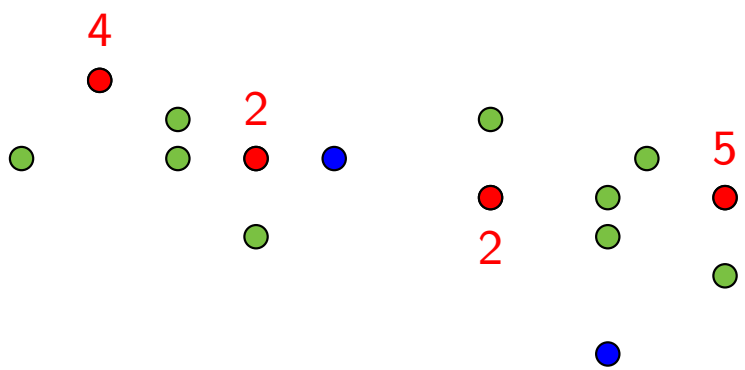


Coreset Definition

Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for all sets S of k centers it holds

$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$

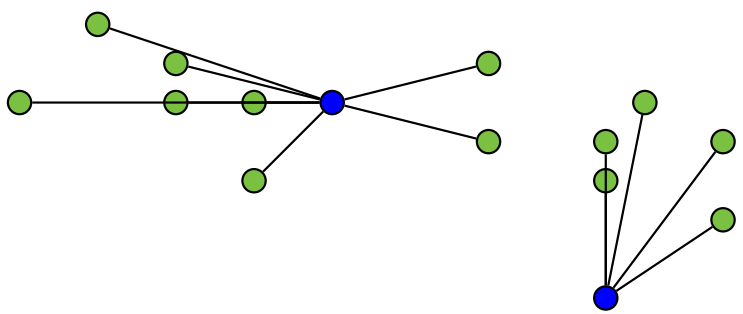


Coreset Definition

Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for all sets S of k centers it holds

$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$

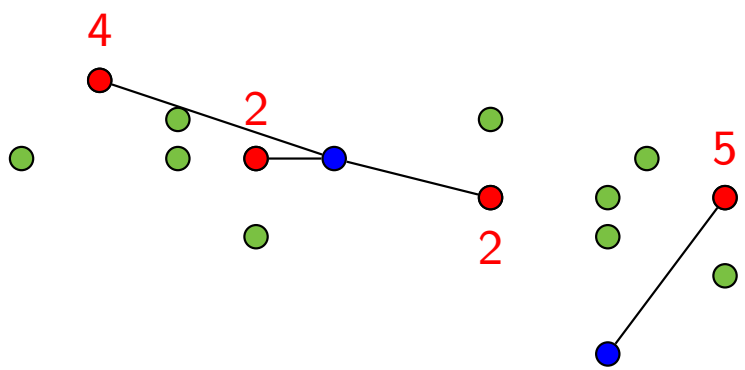


Coreset Definition

Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for all sets S of k centers it holds

$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$

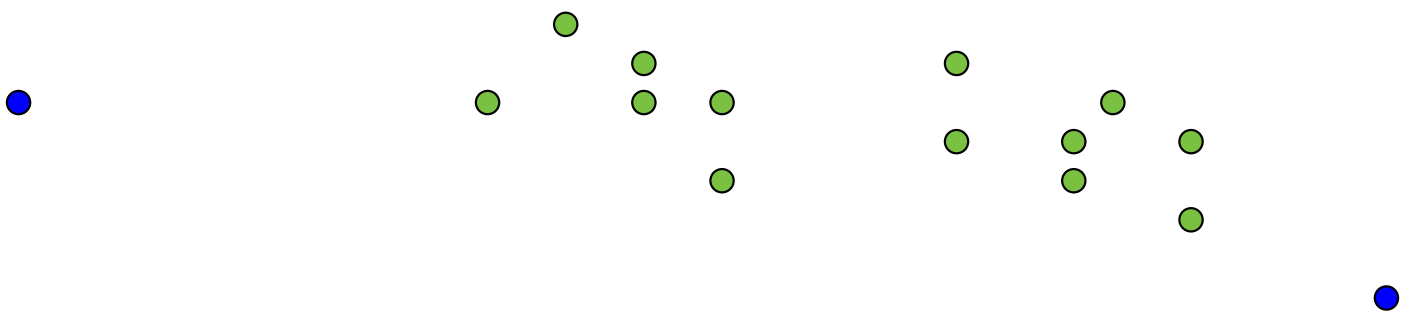


Coreset Definition

Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for all sets S of k centers it holds

$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$

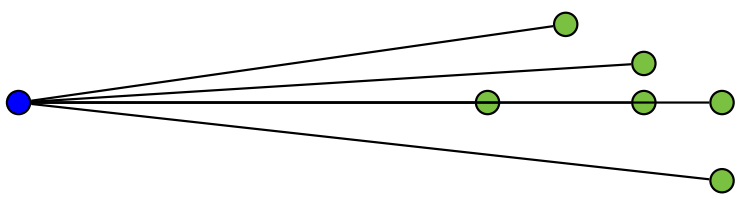


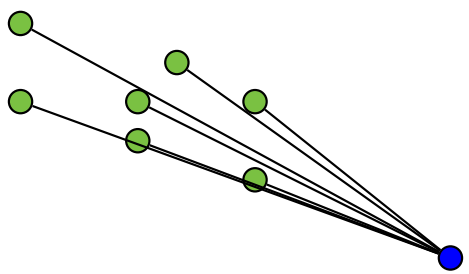
Coreset Definition

Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for all sets S of k centers it holds

$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$



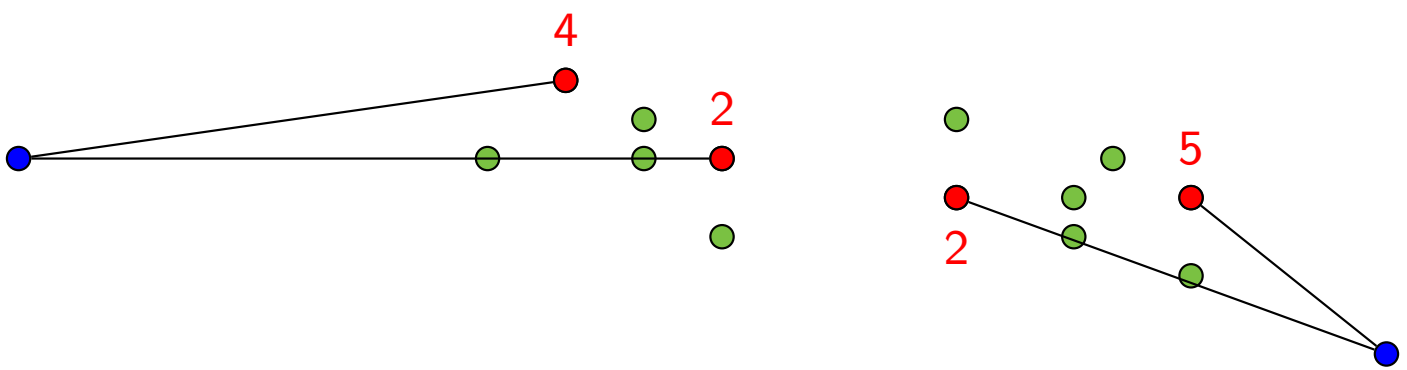


Coreset Definition

Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for all sets S of k centers it holds

$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$



Theoretical Results on Coresets for Euclidean k-Means

Coreset Size

Upper Bounds

Har-Peled, Mazumdar (STOC'04)	$O(\frac{k}{\epsilon^{-d+2}} \log n)$
Chen (Sicomp'09)	$O(d\frac{k}{\varepsilon}^2 \log n)$
Langberg, Schulman (SODA'10)	$O(d^2k^3\epsilon^{-2})$
Feldman, Langberg (STOC'11)	$O(dk\epsilon^{-4})$
Feldman, Schmidt, Sohler (Sicomp'20)	$O(k^3\epsilon^{-4})$
Becchetti, Bury, C.-A., Grandoni, Schwiegelshohn (STOC'19)	$O(k\epsilon^{-8})$
Huang, Vishnoi (STOC'20)	$O(k\epsilon^{-6})$
Braverman, Jiang, Krauthgamer, Wu (SODA'21)	$O(k^2\epsilon^{-4})$
C.-A., Saulpic, Schwiegelshohn (STOC'21)	$O(k\epsilon^{-4})$
C.-A., Larsen, Saulpic, Schwiegelshohn (STOC'22)	$O(k^2\epsilon^{-2})$
C.-A., Larsen, Saulpic, Schwiegelshohn, Sheikh-Omar (NeurIPS'22)	$O(k^{1.5}\epsilon^{-2})$

Theoretical Results on Coresets for Euclidean k-Means

Coreset Size

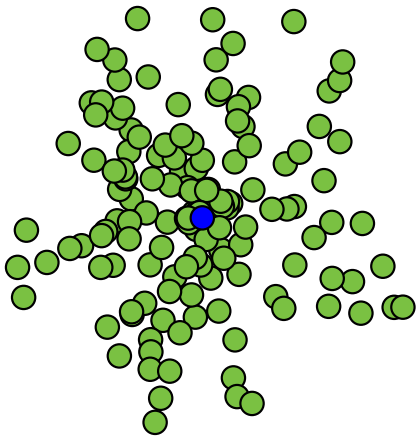
Upper Bounds

Har-Peled, Mazumdar (STOC'04)	$O(\frac{k}{\epsilon^{-d+2}} \log n)$
Chen (Sicomp'09)	$O(d\frac{k}{\varepsilon}^2 \log n)$
Langberg, Schulman (SODA'10)	$O(d^2k^3\epsilon^{-2})$
Feldman, Langberg (STOC'11)	$O(dk\epsilon^{-4})$
Feldman, Schmidt, Sohler (Sicomp'20)	$O(k^3\epsilon^{-4})$
Becchetti, Bury, C.-A., Grandoni, Schwiegelshohn (STOC'19)	$O(k\epsilon^{-8})$
Huang, Vishnoi (STOC'20)	$O(k\epsilon^{-6})$
Braverman, Jiang, Krauthgamer, Wu (SODA'21)	$O(k^2\epsilon^{-4})$
C.-A., Saulpic, Schwiegelshohn (STOC'21)	$O(k\epsilon^{-4})$
C.-A., Larsen, Saulpic, Schwiegelshohn (STOC'22)	$O(k^2\epsilon^{-2})$
C.-A., Larsen, Saulpic, Schwiegelshohn, Sheikh-Omar (NeurIPS'22)	$O(k^{1.5}\epsilon^{-2})$

Huang, Jian, and Wu (2023) showed that $O(k\epsilon^{-2}\min(\varepsilon^{-2}, \sqrt{k}))$ is optimal.

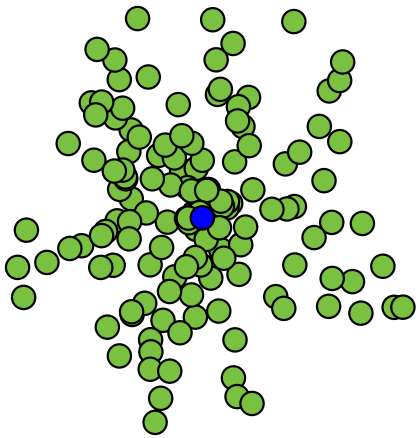
Uniform Sampling

We sample S points uniformly at random and hope for the best.



Uniform Sampling

We sample S points uniformly at random and hope for the best.



We will never sample from the lower right set of points unless we pick the entire point set.

Sensitivity Sampling

The lower right points were, individually, more important than the upper left ones.

Sensitivity Sampling

The lower right points were, individually, more important than the upper left ones.
Can we formalize and use the importance of points in the sampling distribution?

Sensitivity Sampling

The lower right points were, individually, more important than the upper left ones.
Can we formalize and use the importance of points in the sampling distribution?

Sensitivity

The sensitivity of a point p is defined as

$$sens(p) := \sup_{\text{set of } k \text{ points } C} \frac{cost(p, C)}{cost(A, C)}.$$

Sensitivity Sampling

The lower right points were, individually, more important than the upper left ones.
Can we formalize and use the importance of points in the sampling distribution?

Sensitivity

The sensitivity of a point p is defined as

$$sens(p) := \sup_{\text{set of } k \text{ points } C} \frac{cost(p, C)}{cost(A, C)}.$$

The higher the sensitivity, the more important p is for some clustering.

Sensitivity Sampling

The lower right points were, individually, more important than the upper left ones.
Can we formalize and use the importance of points in the sampling distribution?

Sensitivity

The sensitivity of a point p is defined as

$$sens(p) := \sup_{\text{set of } k \text{ points } C} \frac{cost(p, C)}{cost(A, C)}.$$

The higher the sensitivity, the more important p is for some clustering.

Sensitivity Sampling (2)

Idealized Algorithm

1. Sample each point p proportionate to $sens(p)$.
2. Weight each sampled point inversely proportionate to the sampling probability.

Sensitivity Sampling (2)

Idealized Algorithm

1. Sample each point p proportionate to $sens(p)$.
2. Weight each sampled point inversely proportionate to the sampling probability.

Bansal, C.-A., Prabhu, Saulpic, Schwiegelshohn '24

Theorem

Sensitivity sampling yields a coreset of nearly-optimal size.

Data Efficiency

Large Model

Big Data

New Challenge

Reduce data size efficiently.

Sublinear query time algorithm: How to identify the relevant elements in the data with few model queries?

High Level Goal

Find a subset S in the data s.t. average gradient (or loss) of the model on $S \sim$ average gradient (or loss) of the model on whole data.

High Level Goal

Find a subset S in the data s.t. average gradient (or loss) of the model on $S \sim$ average gradient (or loss) of the model on whole data.

Sener & Savarese (2018)

Input: A dataset D
Oracle access to the loss of the model $\ell : D \mapsto \mathbb{R}_+$,
Target sample size k

Output: A sample $S \subseteq D$ of size at most k and a weight function $w : S \mapsto \mathbb{R}_+$ s.t.:

1. The number of queries to ℓ (i.e.: inferences) is at most k.
2. S minimizes $\Delta(S) := |\sum_{e \in D} \ell(e) - \sum_{s \in S} w(s)\ell(s)|$.

Our Modelisation

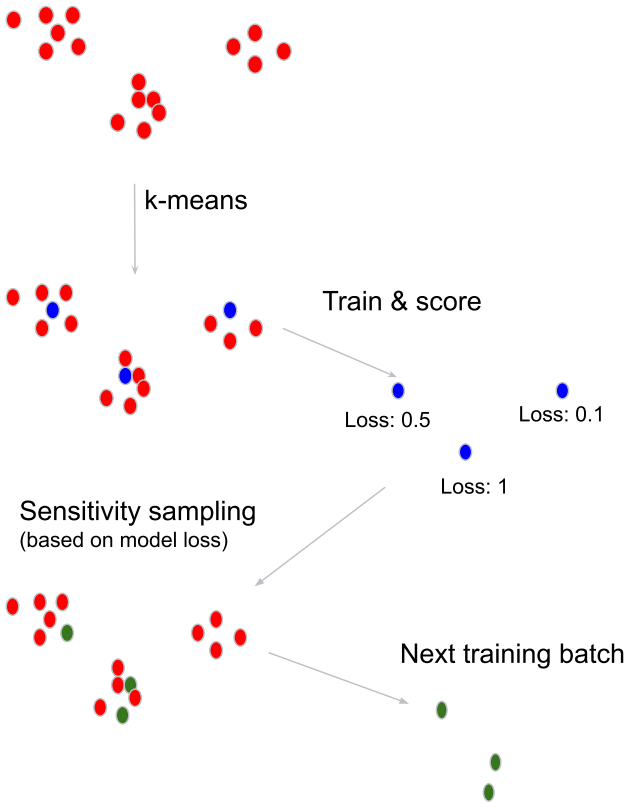
Assumptions

1. Input provides an embedding of the data in $\mathbb{R}^d$.
2. Loss of the model is (z, λ)-Holder continuous w.r.t. the embedding: For two elements e, e', $|\ell(e) - \ell(e')| \leq \lambda ||embedding(e) - embedding(e')||^z$.

Why?

1. In many cases the model is pretrained (fine-tuning, distillation, etc.).
2. Or we want to improve a pre-existing model (for which the last layers may provide a good embedding).
3. There are many generic embeddings that capture the high-level input structure of the input (e.g.: BERT, etc.).

Our Algorithm



- **Step 1**: Run k-means clustering to partition the data into k clusters and pick the most representative element in each cluster
- **Step 2**: Inference on the k representatives.
- **Step 3**: Based on the loss on the k representatives, select another batch of k representatives using **sensitivity sampling.**

Given a solution S for k-means:

$\text{sens}(p) := \lambda d(p, S(p))^2 + \ell(S(p))$, where $S(p)$ is closest center to p

Our Theorem

Axiotis, C.-A., Henzinger, Jerome, Mirrokni, Saulpic, Woodruff, Wunder ICML'24

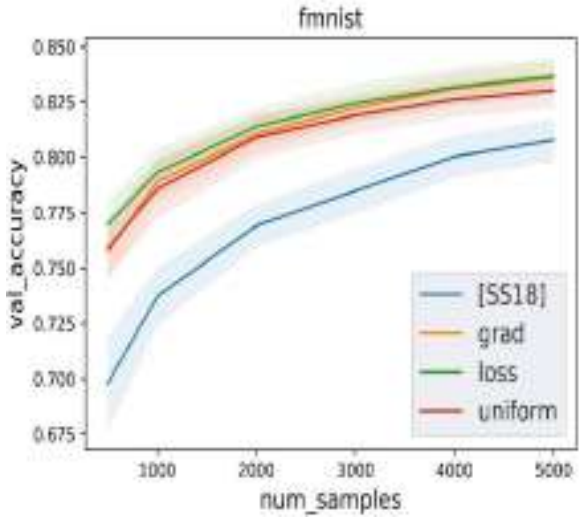
Theorem

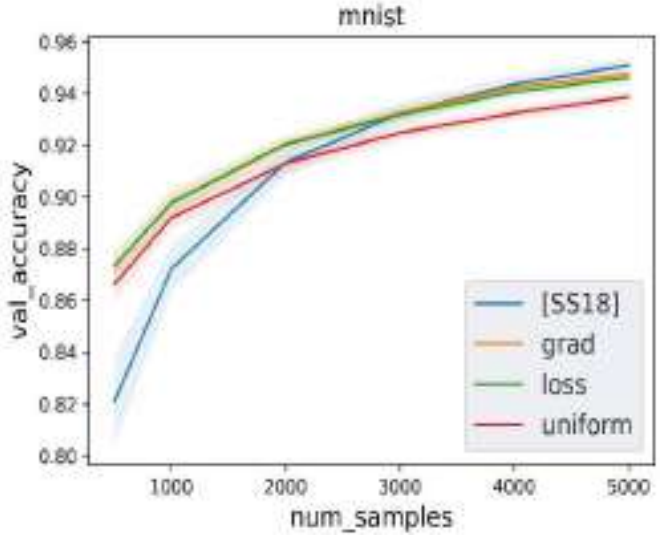
The sample S output by our algorithm satisfies $\Delta(S) := |\sum_{e\in D}\ell(e) - \sum_{s\in S} w(s)\ell(s)| \leq \frac{1}{\sqrt{k}}(\sum_{e\in D}\ell(e) + \lambda k\text{-means cost})$

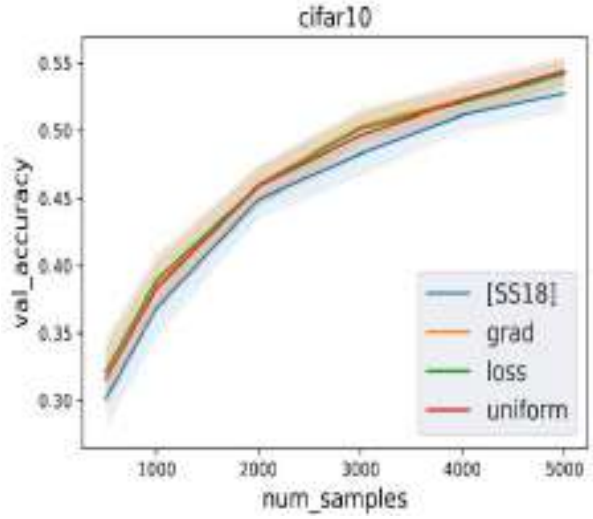
Remark

As k grows, the additive k-means cost term becomes negligible (0 when $k = n$).

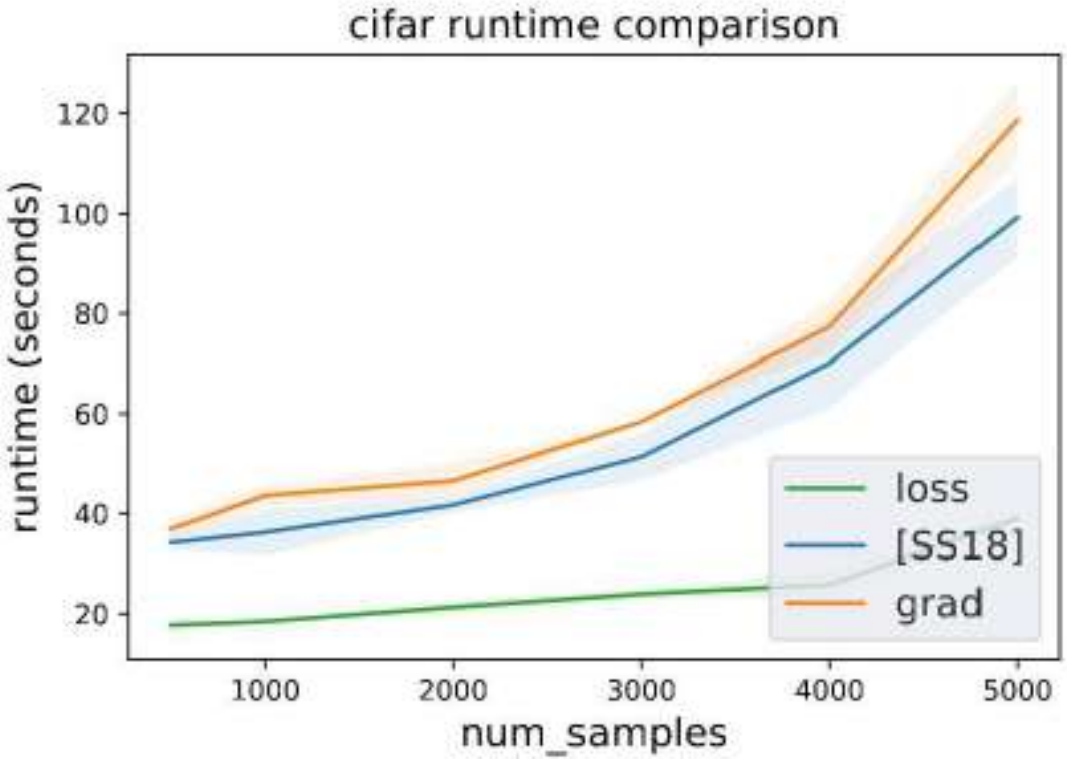
Experimental Results







Experimental Results

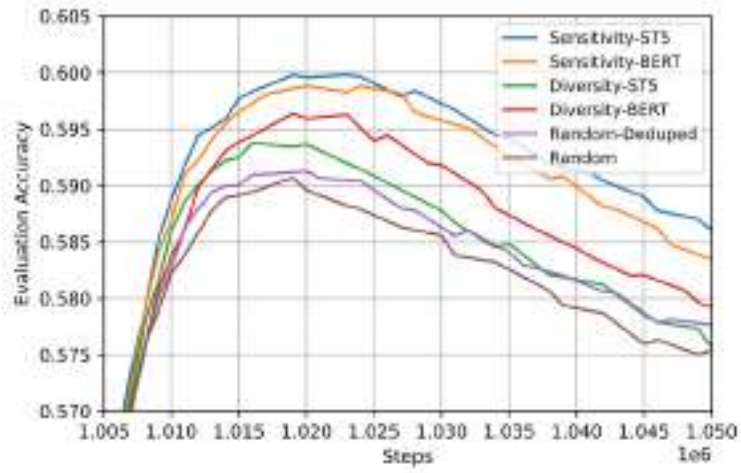


Experimental Results on LLMs

WMT T2T En-De translation task dataset (Bojar et al., 2014)

4,592,289 training examples,
3003 test set
3000 validation set

We fine-tune a T5-Small model
(Raffel et al., 2019) with 77M parameters



Summary and Take Away: Sensitivity is All You Need

- Framework to think about Data Selection / Data Curation.
- Sensitivity sampling + k-means leads to approximately preserving the loss (+- $(1/\sqrt{k})$ k-means cost)
- Generic approach : E.g.: works for linear regression

Questions:

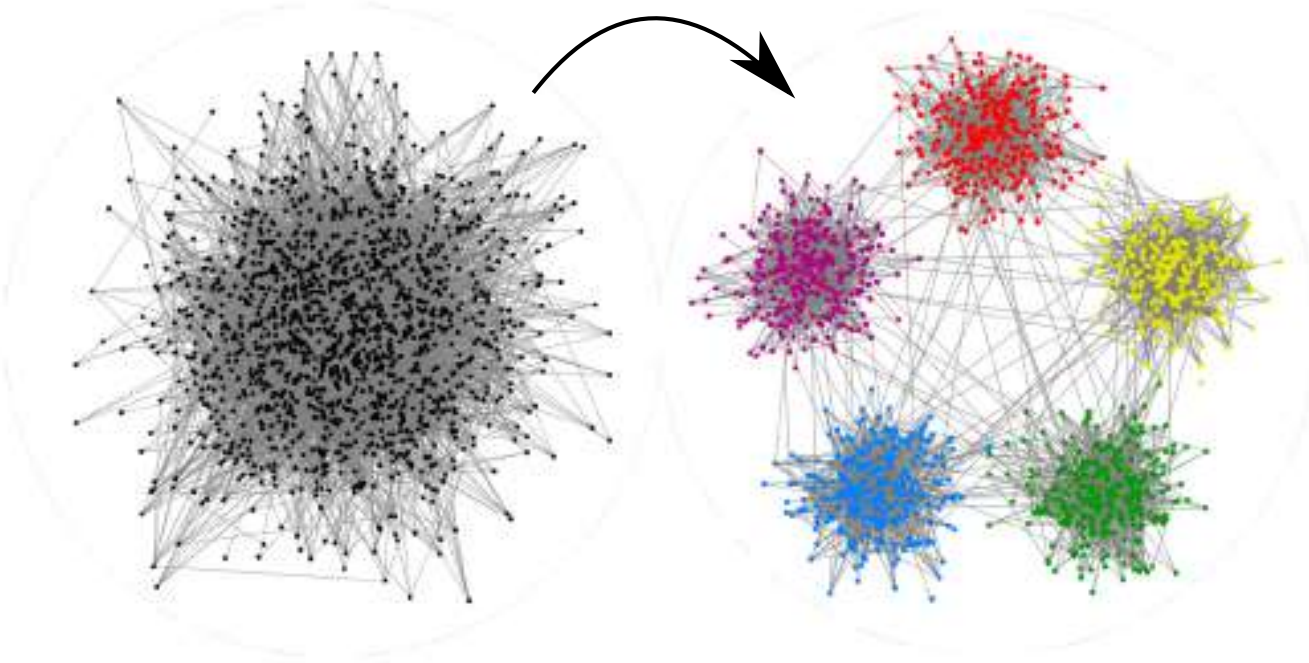
- Privacy?
- Other models?
- Refining Assumptions?
- Other contexts: Distillation, Soft-Prompt?

*Or an **LLM**…*

Graph Clustering: A Classic Data Analysis Task

Graph Clustering: Identify dense subgraphs

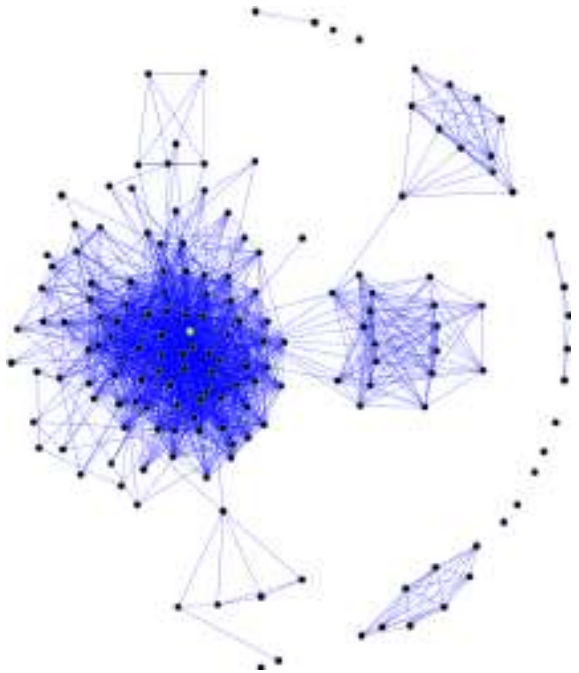
Input: A social network, set of genes of species, the world wide web.



Goal: Find communities in social networks, groups of related organisms, designing heuristics or compression schemes...

A slightly more concrete objective for this talk:

Identify very dense subgraphs with small expansion (in sparse graphs).



Identify very dense subgraphs with small expansion (in sparse graphs).

Step 1: Phrase the problem as an optimization problem.

Modularity: An Objective Function from Statistical Physics

Input: A graph $G = (V, E)$, an edge between two vertices u, v if they are similar
Goal: A partition $\{V_1, \ldots, V_k\}$ of V that maximizes

$$\sum_{i=1}^{k} \sum_{u,v \in V_i} \mathbb{1}_{(u,v) \in E} - \frac{\text{degree}(u) \cdot \text{degree}(v)}{2|E|}$$

Intuition:

For a given cluster, the function compares the number of edges within the cluster to the number of edges in a random network with prescribed degree distribution.

Some More Intuition

- All vertices in the same cluster $\implies$ Modularity $= 0$
- G consists of 2 disjoint cliques C_1, C_2 of size $n/2$ $\implies$ Max Modularity Clustering is $\{C_1, C_2\}$.

Interesting Feature

The number of clusters is not imposed.

Question 1

Can we design an algorithm to identify dense subgraphs?

Step 1: Phrase the problem as an optimization problem. ✓

Question 1

Can we design an algorithm to identify dense subgraphs?

Step 1: Phrase the problem as an optimization problem. ✓

Step 2: An algorithm for optimizing this objective function.

The Louvain Algorithm (almost)

A natural approach for maximizing modularity

Input: A graph $G = (V, E)$, an edge between two vertices u, v if they are similar.

Step 1: Start with a partition P where each vertex is in its own cluster.

Step 2: Given a partition $P = \{V_1, \ldots, V_k\}$, consider the set of vertices U that are such that moving a vertex from its current part to another one increases the modularity.
If $|U| > 0$, pick a random vertex in U and move it to a part so as to increase the modularity and repeat Step 2. Otherwise stop.

Step 3: Outputs the partition.

How Good is the Louvain Algorithm?

Awful in theory: No approximation guarantee in the worst-case even for small families of graphs...

Awesome in practice: Method of choice for clustering graphs, more than 8600 citations over the last 10 years...

Our Result: Analysis of Louvain Beyond the Worst-Case

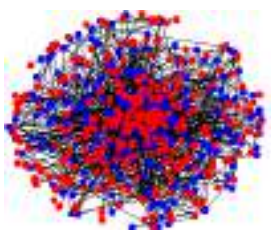
A classic (but naive) model for graphs with cluster structure:

The Stochastic Block Model

The set of vertices consists of two (unknown) equal-size parts A_1, A_2;
An edge between vertices $u \in A_i$ and $v \in A_j$ is generated indep. at random:

- With probability p if $i = j$.
- With probability q if $i \neq j$.

and $p > q$.
Goal: Find A_1, A_2.



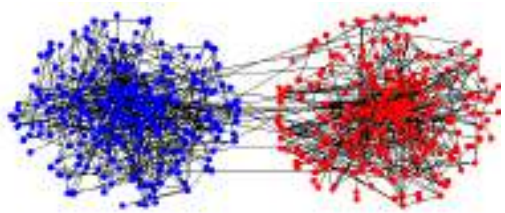
How does Louvain perform on these graphs?

Our Result

C.-A., Kosowski, Mallmann-Trenn, Saulpic Neurips'20

Theorem

If $\frac{p-q}{\sqrt{p}} > n^{-1/6-\varepsilon}$ then Louvain outputs A_1, A_2 with high probability. Moreover, Louvain converges in near-linear time.



Aftermath:

Parallel Setup

Corollary: If all the vertices change partition in parallel, then convergence is done in $O(1)$ rounds.
$\Longrightarrow$ Massively-Parallel-Computation (MPC) algorithm for SBM.

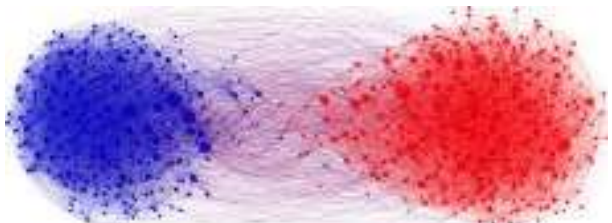
Aftermath:

Parallel Setup

Corollary: If all the vertices change partition in parallel, then convergence is done in $O(1)$ rounds.
$\Longrightarrow$ Massively-Parallel-Computation (MPC) algorithm for SBM.

Consider a social network where everyone follows the opinion of the majority of their friends. Initial opinion is random $\in \{0, 1\}$.

If there are two communities, then it quickly converges to polarized opinions.



Future Work and Open Problems

- Find better heuristics: Robust to inputs containing many clusters of diverse sizes?
- Better model for real world graphs: beyond the stochastic block model. Semi-random? MPC for semi-random graph models?
- MPC Algorithms for exact recovery in the stochastic block model up to O(information theoretic threshold)? Spectral methods?
- Understand local dynamics in networks to favor diversity of opinions, less polarized situations.


DO YOU HAVE ANY
QUESTIONS?


DO YOU HAVE ANY
QUESTIONS?