Robust and Fully-Dynamic Coreset for Continuous-and-Bounded Learning (With Outliers) Problems

Zixiu Wang[1] Yiwen Guo[2] Hu Ding[1]

[1]University of Science and Technology of China
[2]ByteDance AI Lab

Motivation

Some Facts

- It is hard to store and process data in this big data era.
- The training of most machine learning tasks is to minimize an objective function.

Question

Can we reduce the data size by representing the big dataset with a much smaller data summary, by compromising on the approximation of objective function?

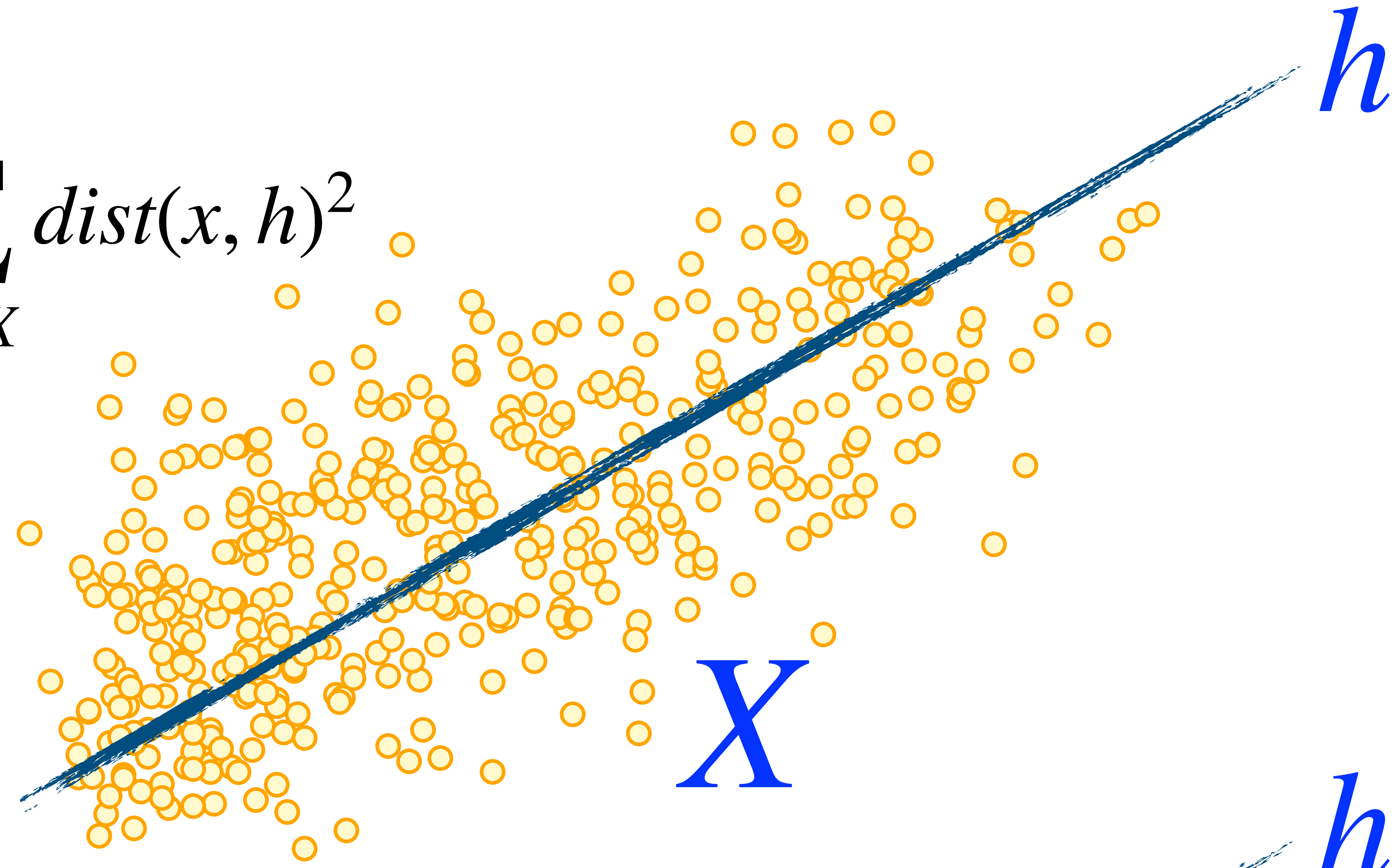
Definition & Illustration

Coreset is designed with the trade-off between size and accuracy

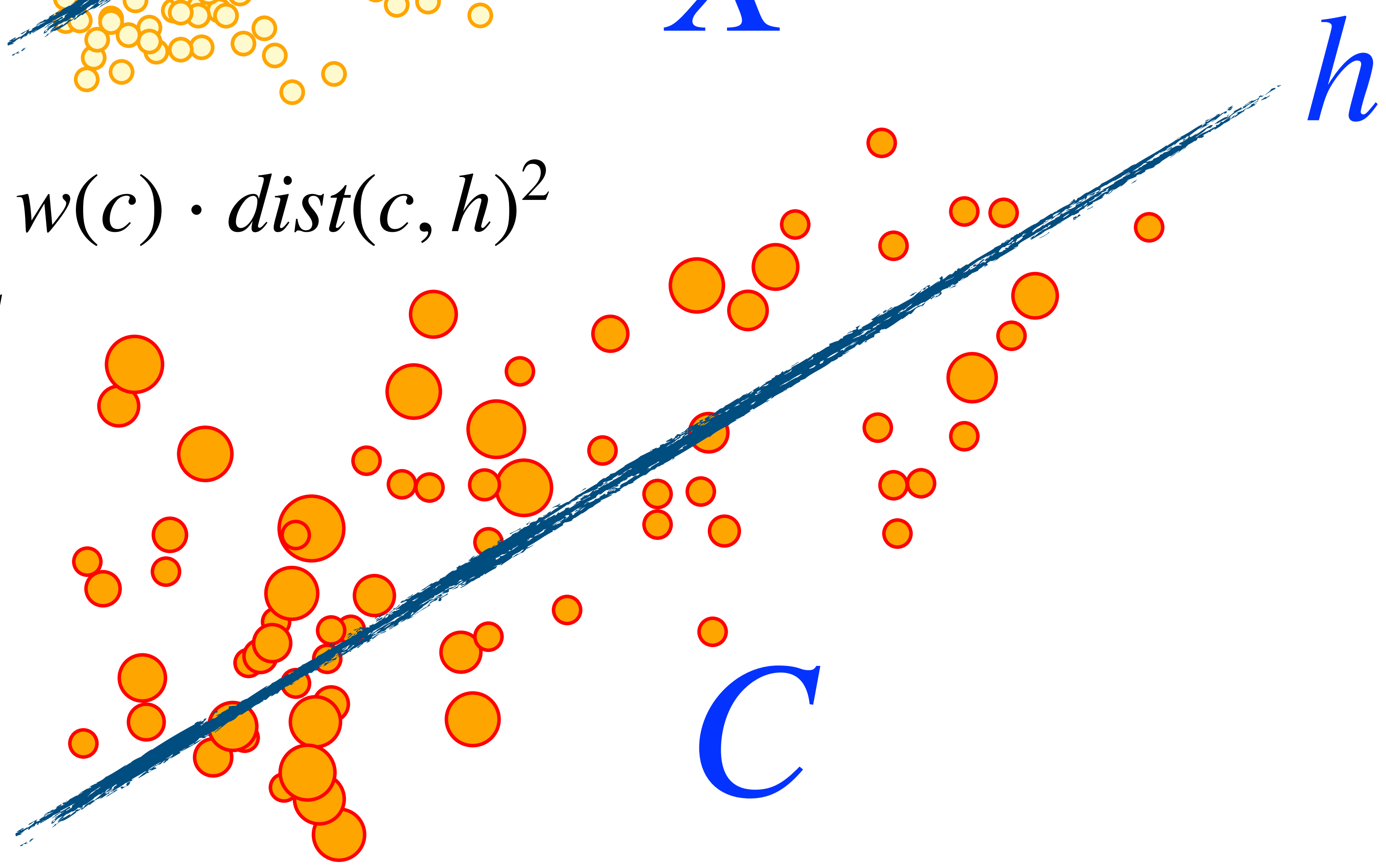
h_C : the solution obtained on C
h_X : the solution obtained on X

We hope that $Loss(h_C; X) \approx Loss(h_X; X)$

$$Loss(h; X) = \sum_{x \in X} dist(x, h)^2$$



$$Loss(h; C) = \sum_{c \in C} w(c) \cdot dist(c, h)^2$$



Definition (Strong Coreset)

For any h on the plane, we have that
$Loss(h; X) \approx Loss(h; C)$

Why Coreset?

The coreset has several advantages:

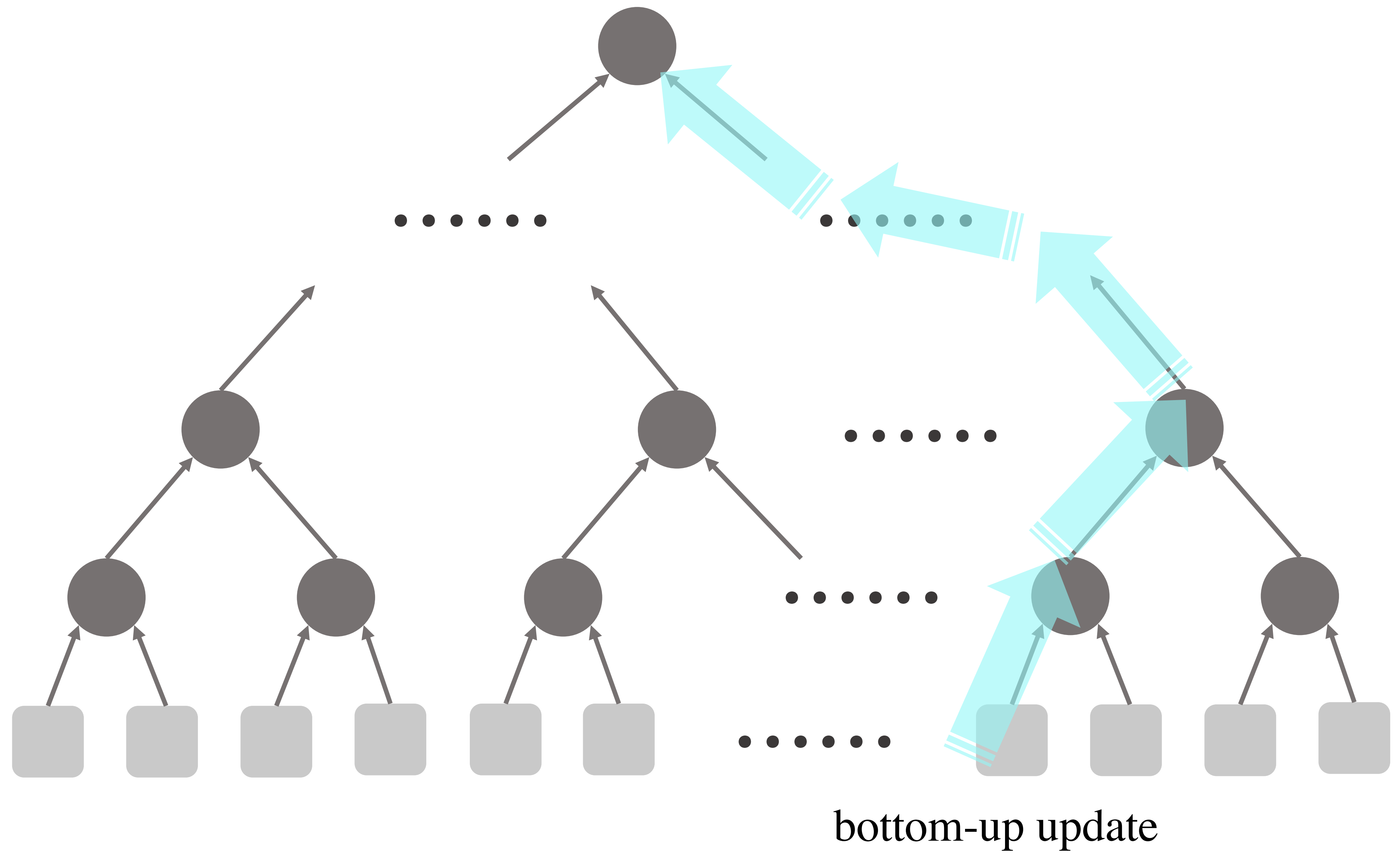
- Has small size thus it can be processed efficiently
- The approximation is guaranteed by the careful construction
- Supports distributed computing naturally
- Can be used to design streaming algorithms
- Can be used to design dynamic algorithms

Based on Merge-and-Reduce

Problems

1. Most existing works concentrate on clustering-style problems.
2. There are some negative results of the existence of small strong Coreset. [Munteanu et al, NeurIPS 2019]
3. Current coreset methods are not robust for handling outliers.

Merge-and-Reduce



- Streaming algorithms from Merge-and-Reduce [Har-Peled and Mazumdar, STOC 2004]
- Fully-dynamic algorithms from Merge-and-Reduce [Henzinger and Kale, ESA 2020]

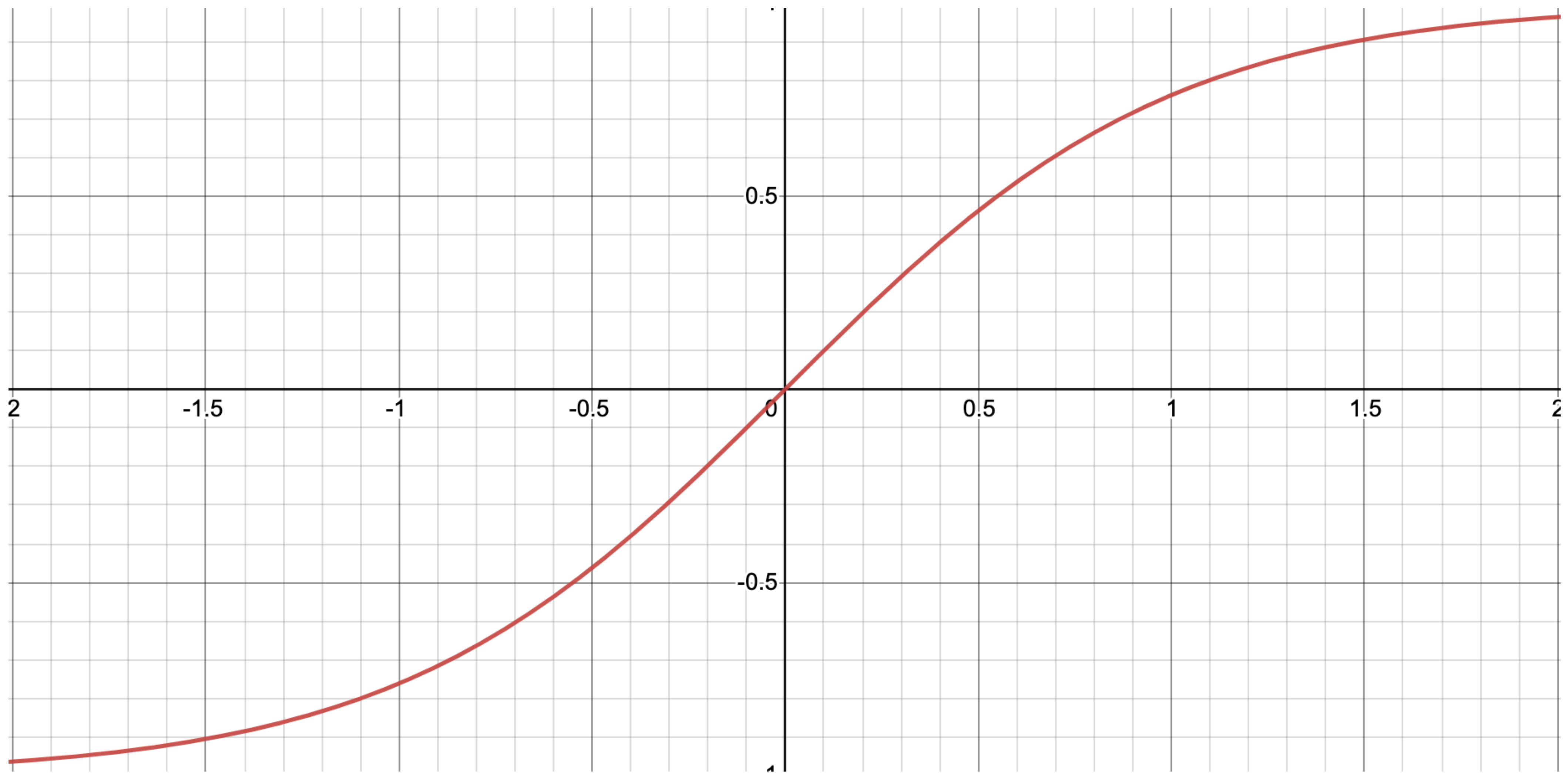
Handling outliers

- The number of outliers is given as z.
- There are robust Coreset constructions that need to know z. [Jiecao Chen et al, NeurIPS 2018]
- The reduce process would accumulate the error on z.

Continuous-and-Bounded Learning Problems

- We consider the learning problems that have smooth objective functions

- To capture the smoothness, we use the notations of L-Lipschitz, L-smooth, etc.



Examples

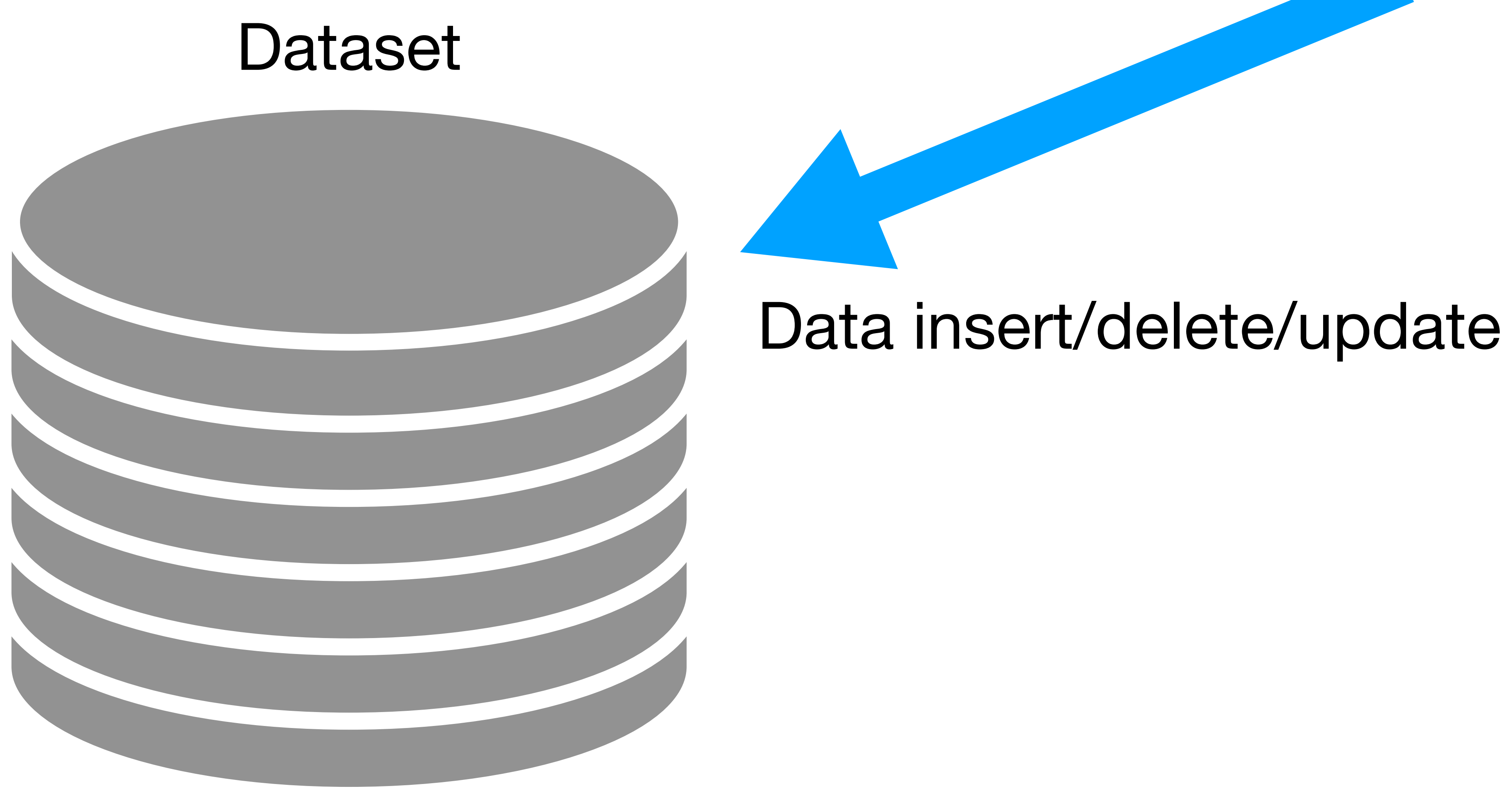
- Logistic Regression: D-Lipschitz where D is the diameter of the dataset

- Bregman Divergence: $2L$-Lipschitz if ϕ is L-smooth

- Truth Discovery [Shi Li et al, Algorithmica]: 2-Lipschitz

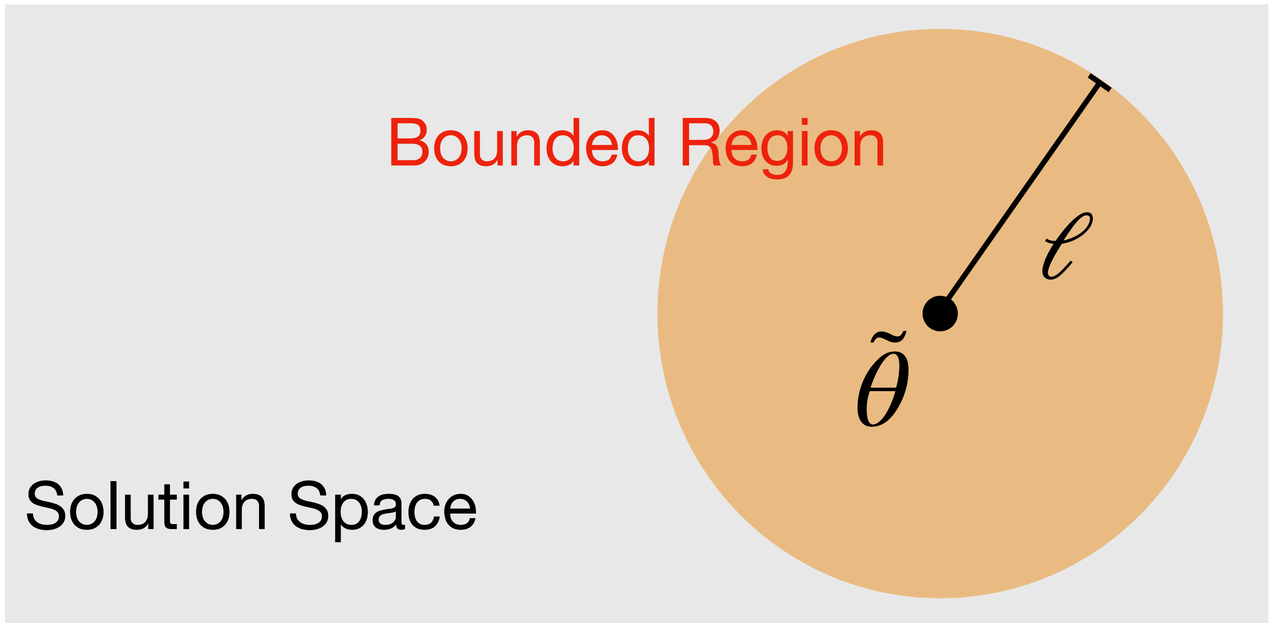
Continuous-and-Bounded Learning Problems

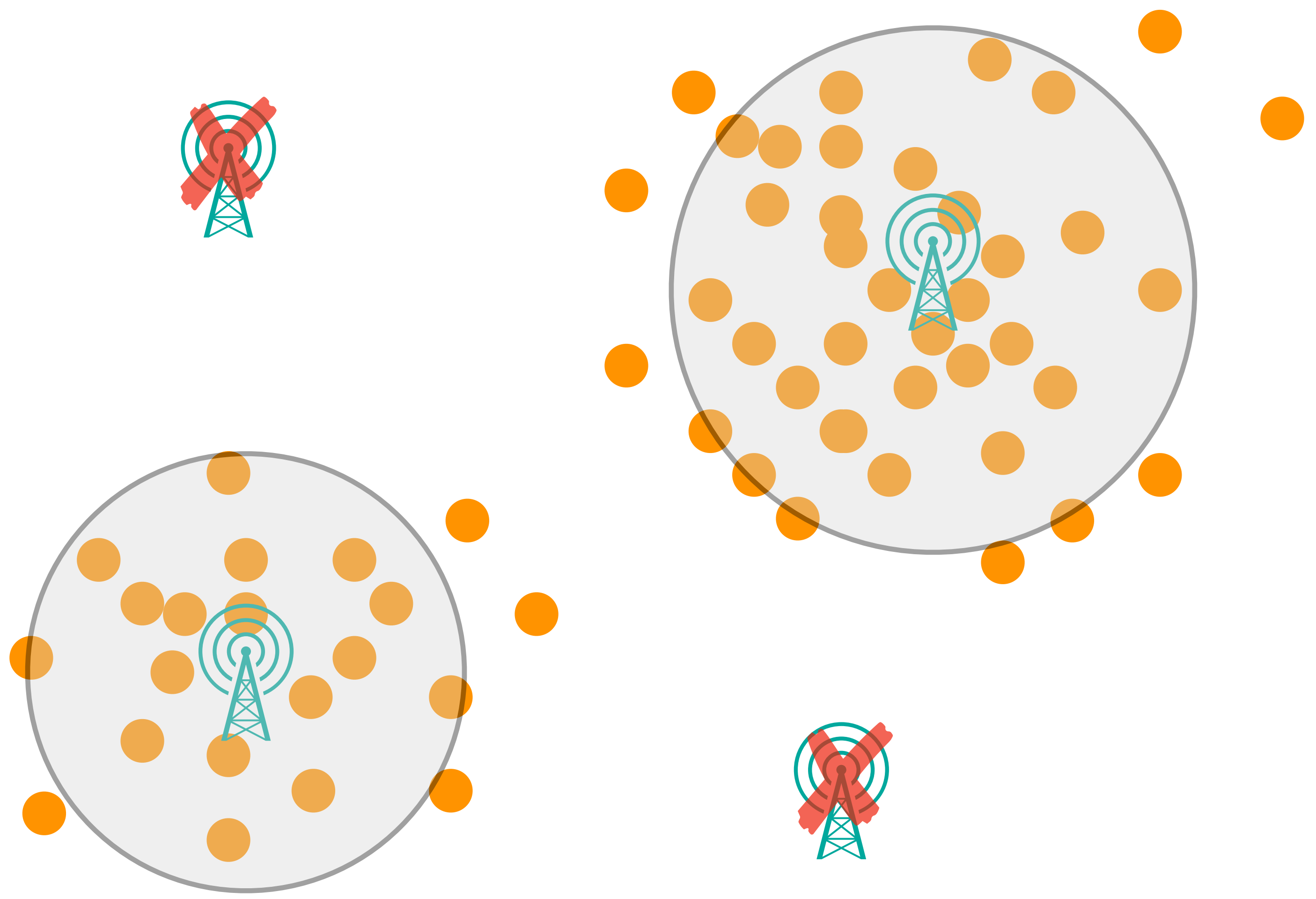
In practice, it is reasonable to consider a bounded region of the solution, rather than the total space



Bounded Region

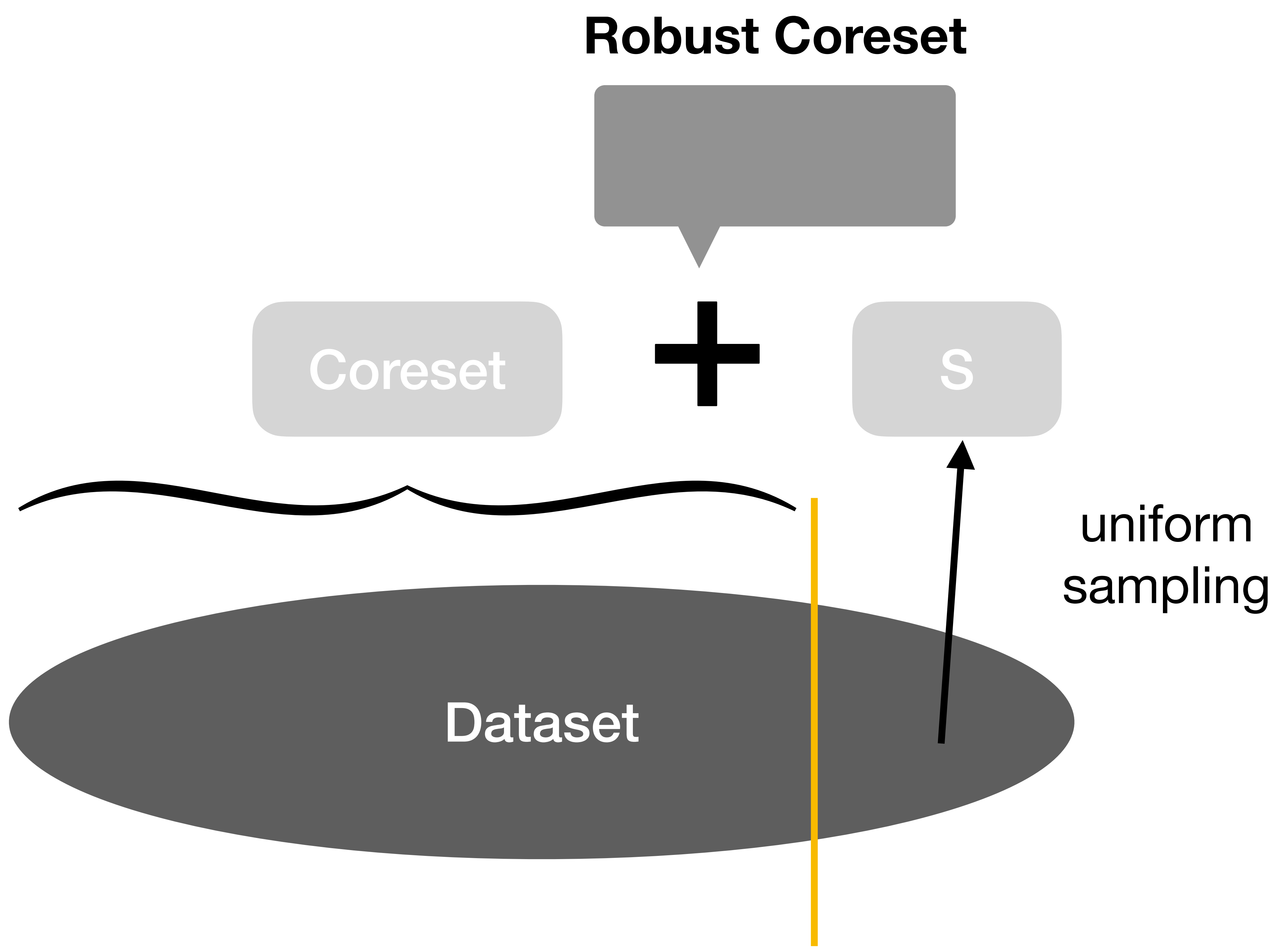
A ball of radius ℓ centered at $\tilde{\theta}$ in the solution space





Robust Coreset Method

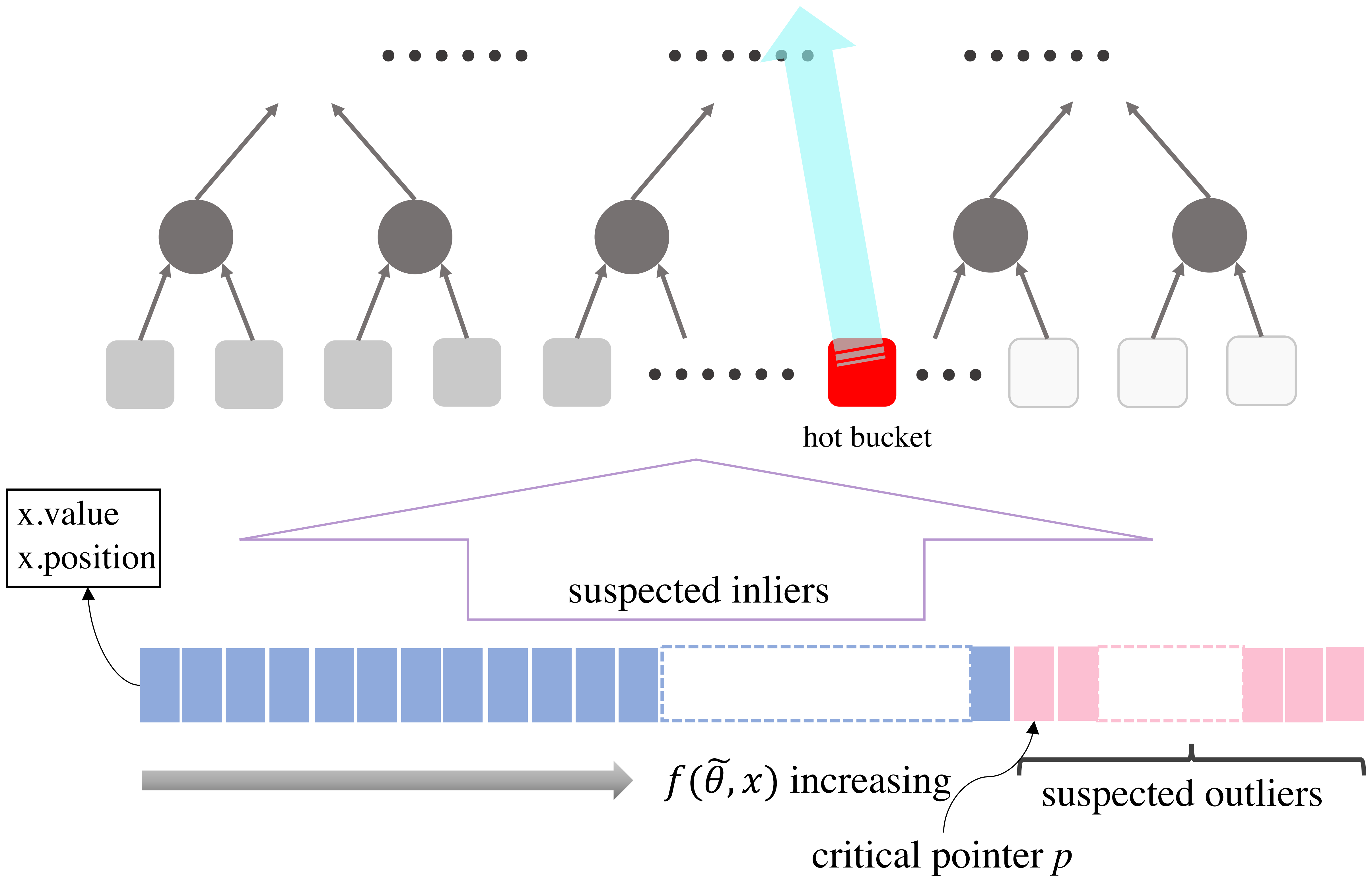
- We assume that the number of outliers z is given.
- As for CnB learning problems with outliers, we can construct robust coreset based on the existing coreset method
- The dataset is partitioned in line with the loss value $f(\tilde{\theta}, x)$ of each data point x



Merge-and-Reduce

- We cannot use the merge-and-reduce method directly in the presence of outliers

- Our robust coreset can induce robust algorithms in the distributed setting and the dynamic data streams



Coreset Method for CnB learning Problems

Importance Sampling based method

- In the importance sampling method, we need to compute an upper bound of the sensitivity
- As for CnB learning problems, we can bound the sensitivity via a quadratic fractional programming

Spatial Partition based method

- Partition dataset X into several parts due to the value of $f(\tilde{\theta}, x)$
- Sample from each part uniformly and take the union
- The size of this coreset depends on the doubling dimension of the solution space