Introduction to Coresets

Stroia Dacian - ML Researcher

About me

Stroia Dacian -- Experience

- Finished my 3rd year at Politehnica University of Timisoara
- At CoreAI / Ethergate for 1 year -- working on coresets for 5 months
- Started learning ML from my 1st uni year

Other fields

- I like cryptography
- I like maths

Contacts

- https://www.linkedin.com/in/dacian-stroia/
- https://github.com/zademn
- Or search my name on other platforms

What are we discussing today?

0. What are we trying to solve?

1. Introduction to Coresets

1. An application -- Streaming data

1. What are coresets?

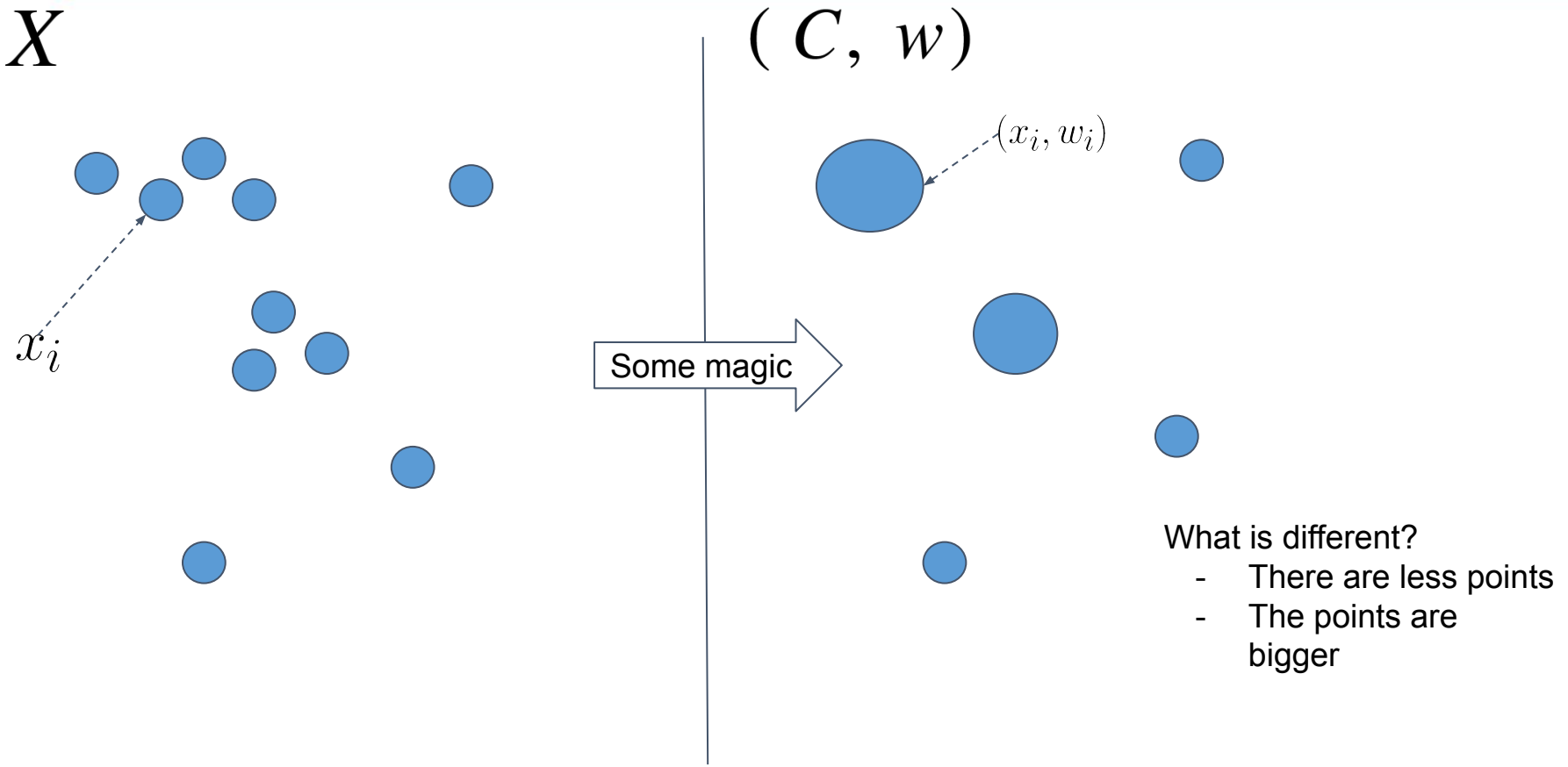
Intuition:

Given a dataset X we want to construct a smaller dataset C that is a good representation of X.

We will discuss:

- What it means to be a good representation?
- How do we construct coresets

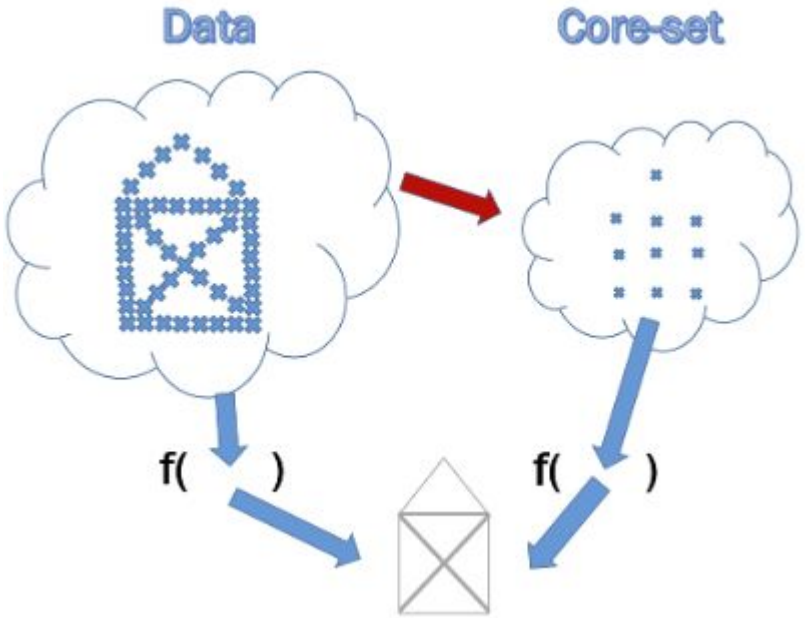
1.1 Coresets - A visual approach



What is different?

- There are less points
- The points are bigger

1.1 Coresets - A visual approach



$$f(X) \approx f(C)$$

“Compression”

1.2 Coresets - A mathematical approach

Notation

X – Dataset of dimension $n \times d$

C – Coreset

$cost$–some additive cost function, usually based on distance functions f

Q–queries - models, classifiers, hypothesis

$w : X \to \mathbb{R}$ – a weight function

1.2 Coresets - A mathematical approach

Example - for kmeans. - Here Q represents the set of cluster centers:

$$cost(X, Q) = \sum_{x \in X} w(x) \cdot f_Q(x) = \sum_{x \in X} w(x) \cdot \min_{q \in Q} \|x - q\|_2^2$$

1.2 Coresets - A mathematical approach

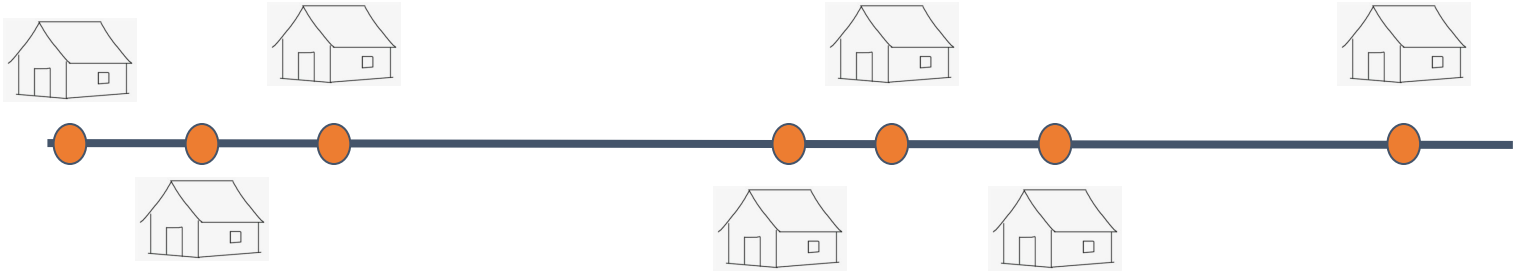
Let's define the coreset: A coreset is a set of points C that for some epsilon it satisfies:

$$|cost(X, Q) - cost(C, Q)| \leq \epsilon \cdot cost(X, Q)$$

If the equation holds for all queries then we call it a **strong coreset**
If the equation holds for at least 1 query then we call it a **weak coreset**

1.3 Simple Example

cost = distance to the farthest resident



Suppose all weights are 1. What is the best coreset C?

1.4 Constructing coresets

Data types

1. Subset sampling -- Ex: random sampling, importance sampling
$$X \subset \mathbb{R}^d,\ C \subseteq X$$
2. Subset space -- Ex: constructing new samples in kmeans
$$X \subset \mathbb{R}^d,\ C \subset \mathbb{R}^d$$
3. Linear combination of input points
$$X \subset \mathbb{R}^d,\ C = SX \text{ for some } S$$

1.4 Constructing coresets

Construction techniques

1. *Sampling* -- Find an algorithm to compute probabilities

2. *Deterministic (greedy)* -- Find an algorithm to select samples based on history

3. *Grids* -- Discretize the space and select best representatives

1.5 Constructions via sampling

The sensitivity framework

Intuition: For each sample, we want to assign a number representing the **importance** of the sample in the dataset. We can look at it as **reweighting** the dataset

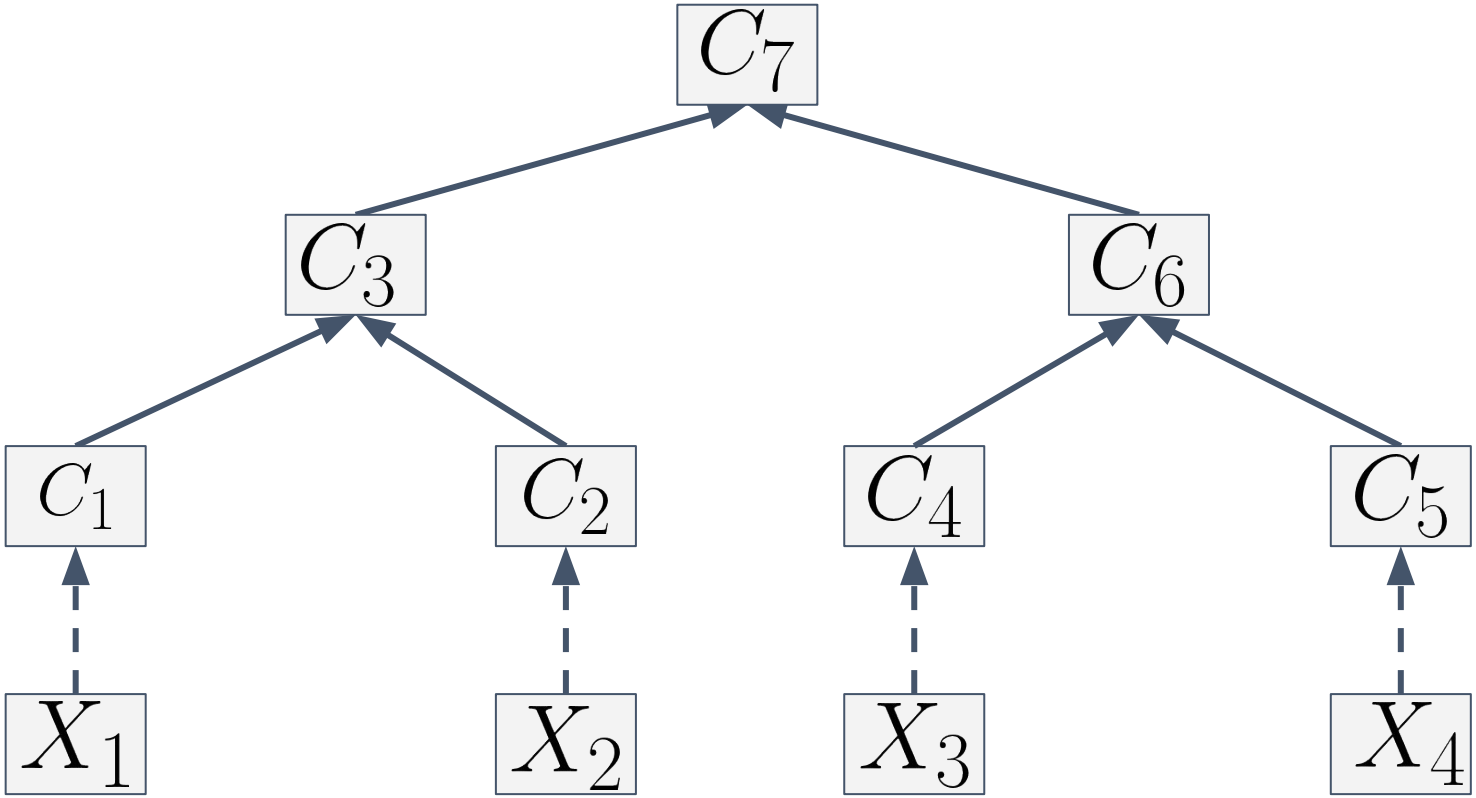
$$\sigma(x) = sup_{Q \in \mathcal{Q}} \frac{f_Q(x)}{\sum_{x' \in X} w_X(x') f_Q(x')}$$

Then we construct the probability distribution $\sigma(x) \sim p(x)$ with

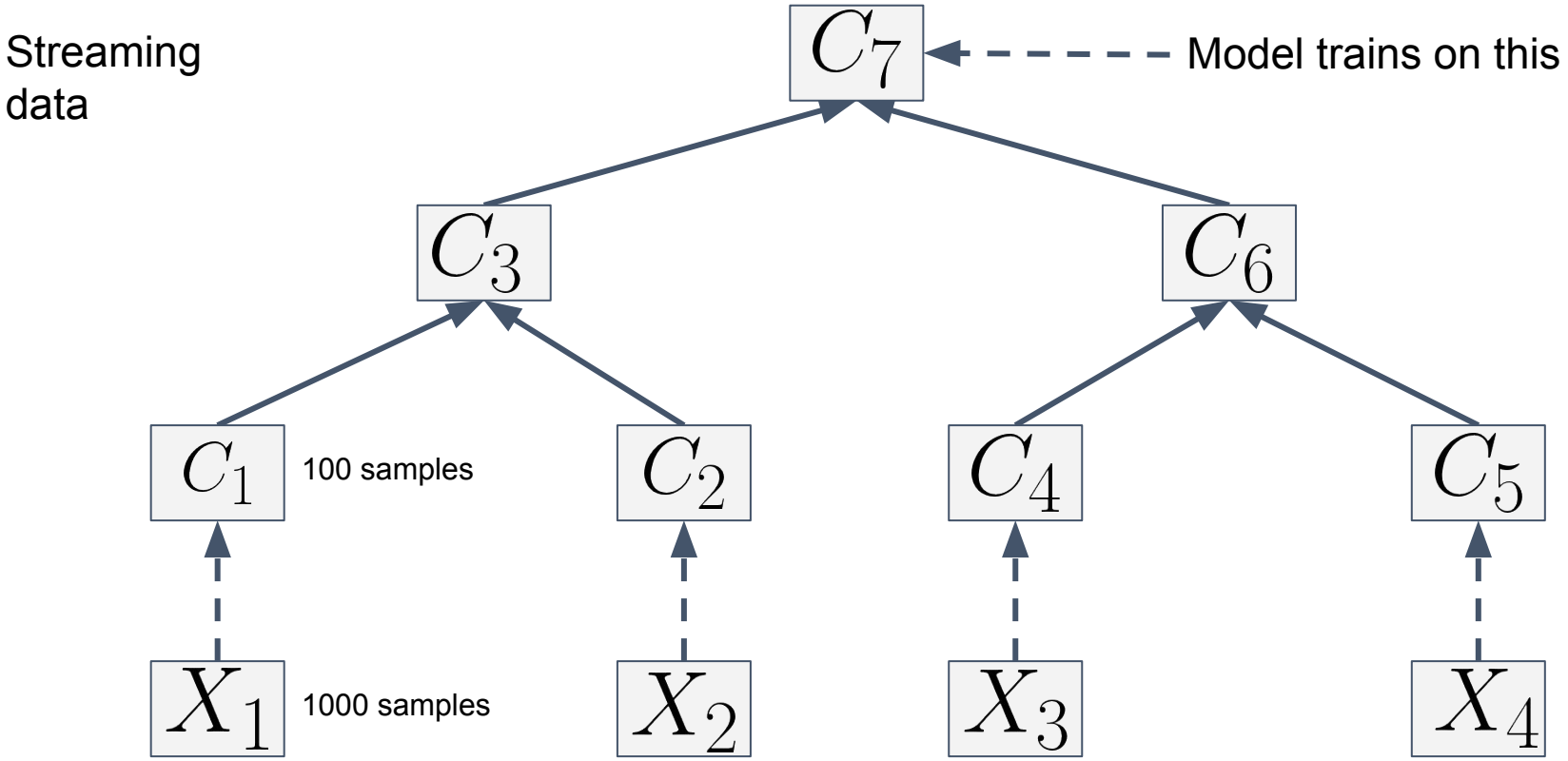
1.6 Drawbacks of coresets

1. Representative coresets might not exist (shortest path in graphs)
2. Hard to design -- like in the optimization field, coresets schemes are hard to construct and sometimes provide little improvement
3. Sometimes the approximation error is too large
4. Coreset constructions might take too long

2. Accelerating ML training

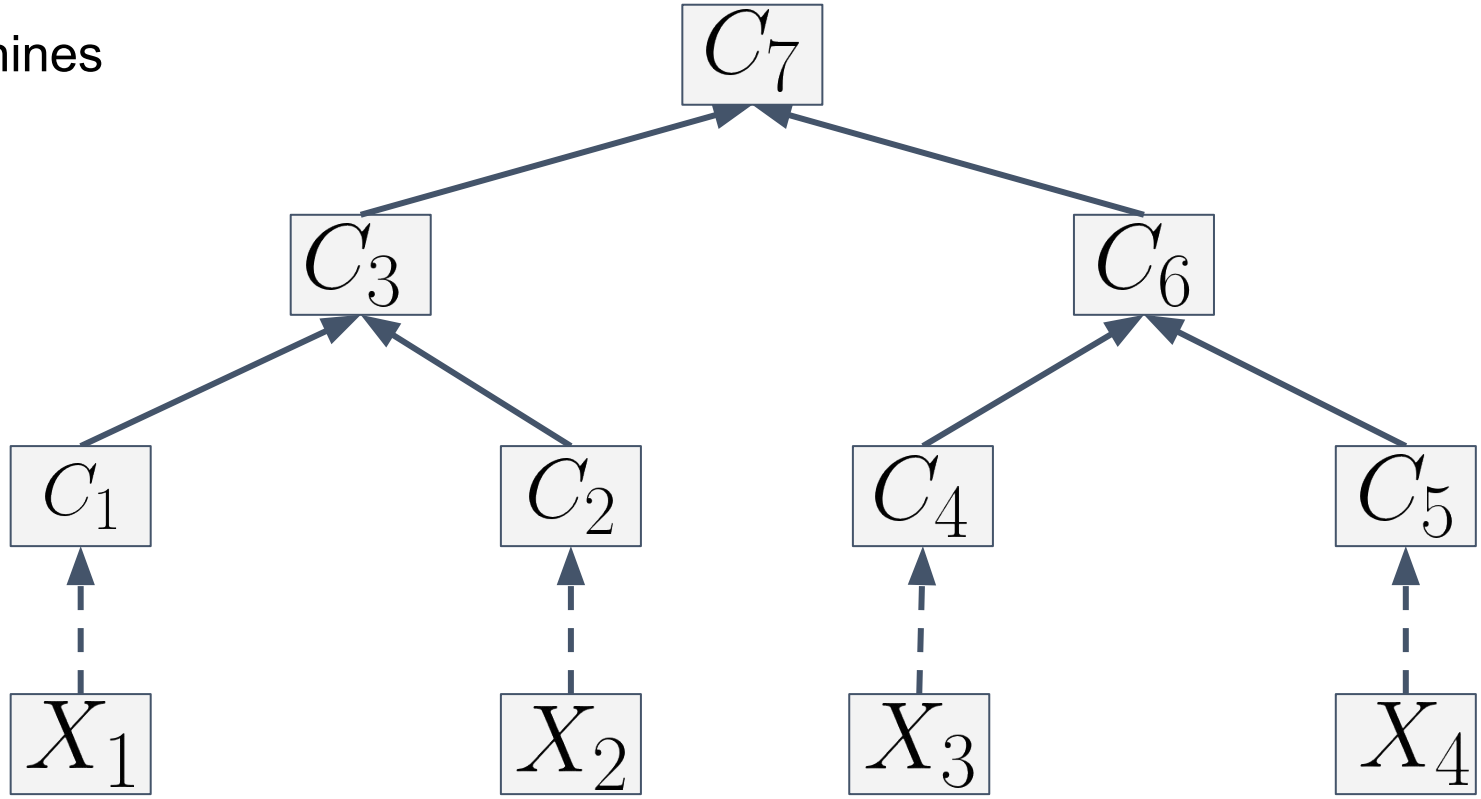


2. Accelerating ML training



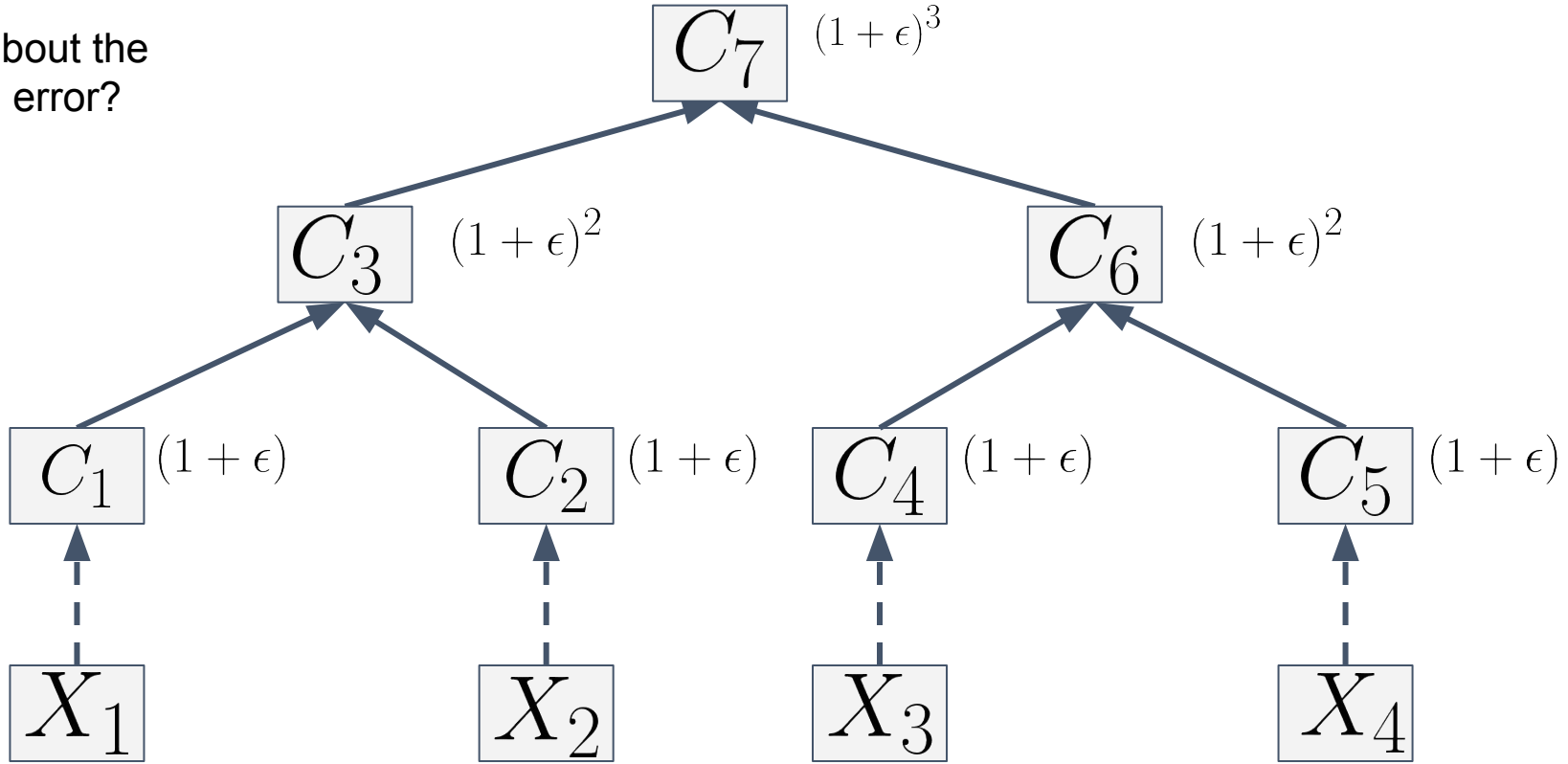
2. Accelerating ML training

n machines



2. Accelerating ML training

What about the epsilon error?



Demo time

Q&A

Resources

- https://arxiv.org/pdf/2011.09384.pdf
- https://arxiv.org/pdf/1910.08707.pdf
- https://arxiv.org/pdf/1703.06476.pdf
- https://arxiv.org/abs/1702.08248