# Lecture 15: Coreset

2024.5.10

*Lecturer:* 丁虎 *Scribe:* 王向禄

Coreset（核心集）是一种高效的数据压缩技术，旨在从大规模数据中选取一个小而具有代表性的子集，以在近似保持原始数据统计性质的同时显著降低计算成本。Feldman [2] 和 Chen [1] 在核心集理论方面作出了重要贡献，分别提出了基于敏感度的采样方法和分层采样方法，为多种机器学习任务构造近似最优的核心集奠定了理论基础 [3]。近年来，coreset 技术被广泛应用于深度学习任务中：在持续学习（Continual Learning）中，coreset 被用于缓解灾难性遗忘，通过保留具有代表性的样本来巩固旧知识 [6]；在主动学习（Active Learning）中，coreset 帮助选取最具信息量的样本，从而提升标注效率 [4]；此外，在生成模型（Deep Generative Models）的训练中，coreset 的引入也有助于加速训练过程，同时保持模型的生成性能 [5]

## 1 基本知识

**Definition 1.1** (Coreset)**.** 设有目标函数 $f(P, c)$，其中 $c \in \mathcal{F}$ 为解空间中的任意解。一个集合 $S \subset \mathbb{R}^d$ 被称为 $P$ 的一个 $\varepsilon$-coreset，若满足：

$$(1-\varepsilon)f(P,c) \leq f(S,c) \leq (1+\varepsilon)f(P,c); \quad \forall c \in \mathcal{F}$$

其中 $\varepsilon \in (0,1)$ 是误差容忍度。

**结论：** 如果 $C^*$ 是 $S$ 上的一个 $\alpha$-approx solution，则 $C^*$ 是 $P$ 上一个 $\alpha \cdot \frac{1+\varepsilon}{1-\varepsilon}$-approx solution。

*Proof.* 我们希望证明：如果 $C_*$ 是在核心集 $S$ 上的一个 $\alpha$-approx 解，则它在原始数据集 $P$ 上是一个 $\alpha \cdot \dfrac{1+\varepsilon}{1-\varepsilon}$-approx 解。

1. 在核心集上是近似解

$$f(S, C_*) \leq \alpha \cdot f(S, C_{\text{opt}})$$

这是因为 $C_*$ 是在 $S$ 上的一个 $\alpha$-approx 解。

2. 利用 coreset 的性质估计上下界

根据 coreset 定义，对任意 $c$ 有：

$$(1-\varepsilon)f(P,c) \leq f(S,c) \leq (1+\varepsilon)f(P,c)$$

对 $C_{\text{opt}}$ 应用上界：

$$f(S,C_{\text{opt}}) \leq (1+\varepsilon)\cdot f(P,C_{\text{opt}})$$

对 $C_*$ 应用下界：

$$f(S,C_*) \geq (1-\varepsilon)\cdot f(P,C_*) \Rightarrow f(P,C_*) \leq \frac{1}{1-\varepsilon}\cdot f(S,C_*)$$

3. 合并不等式将以上不等式联立可得：

$$f(P,C_*) \leq \frac{1}{1-\varepsilon}\cdot f(S,C_*) \leq \frac{1}{1-\varepsilon}\cdot\alpha\cdot f(S,C_{\text{opt}}) \leq \frac{1}{1-\varepsilon}\cdot\alpha\cdot(1+\varepsilon)\cdot f(P,C_{\text{opt}})$$

最终推出：

$$f(P,C_*) \leq \alpha\cdot\frac{1+\varepsilon}{1-\varepsilon}\cdot f(P,C_{\text{opt}})$$

因此，$C_*$ 是在 $P$ 上的 $\alpha\cdot\dfrac{1+\varepsilon}{1-\varepsilon}$-approx 解。

□

# 2 构造方法

**Definition 2.1** (k-median)**.** 给定一个点集 $P=\{p_1,p_2,\ldots,p_n\}\subset\mathbb{R}^d$ 和一个正整数 $k\geq 1$，$k$-median 问题的目标是选择 $k$ 个中心点 $C=\{c_1,c_2,\ldots,c_k\}\subset\mathbb{R}^d$，并将 $P$ 分配到这些中心，使得以下目标函数被最小化：

$$\frac{1}{n}\sum_{i=1}^{n}\min_{1\leq j\leq k}\|p_i-c_j\|$$

其中 $\|\cdot\|$ 表示欧几里得距离（即 $L_2$ 范数），也可根据应用选择其他距离度量。

**Coreset 构造方法** [1]：该算法包括以下两个主要步骤：

1. 将输入点集 $P$ 划分为若干互不重叠的子集；
2. 从每个子集中进行随机抽样。

这些子集样本的并集即构成所需的核集（coreset）。

**步骤一：划分点集 $P$**

为简化表述，我们假设输入数据集 $P$ 为非加权点集。对于加权情况，虽然分析结果仍然成立，但误差界会略有恶化。

设 $\mathcal{A} \subseteq P$ 是一个满足 $[\alpha, \beta]$-bicriteria 近似的中心集合，用于逼近 $P$ 的最优 $k$-median 聚类。形式上表示为：

$$\mathcal{A} = \{a_1, \ldots, a_m\}, \quad \text{满足} \quad \nu(\mathcal{A}, P) \leq \beta \cdot \nu_{\text{opt}}(k, P),$$

其中 $m \leq \alpha k$，且 $\alpha, \beta \geq 1$ 为常数。

令 $P_i \subseteq P$ 表示由中心 $a_i$ 所服务的点集（即属于该中心的聚类），其中 $i = 1, \ldots, m$。定义：

$$R = \frac{\nu(\mathcal{A}, P)}{\beta n}$$

作为最优 $k$-median 聚类平均半径的下界。

进一步设：

$$\phi = \lceil \log(\beta n) \rceil$$

对于每个 $i = 1, \ldots, m$ 和 $j = 0, \ldots, \phi$，定义如下分区：

$$P_{i,j} = \begin{cases} P_i \cap \text{ball}(a_i, R), & j = 0 \\ P_i \cap \left[\text{ball}(a_i, 2^j R) \setminus \text{ball}(a_i, 2^{j-1} R)\right], & j \geq 1 \end{cases}$$

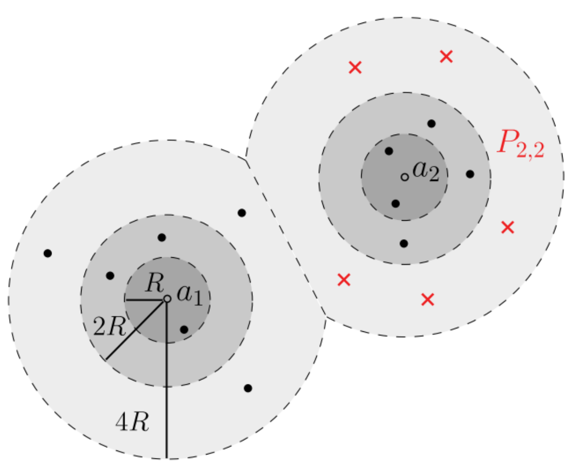


Figure 1: 环状分区示意图（Ring Partitioning of $P$ with respect to center $a_i$）

我们称 $P_{i,j}$ 为中心 $a_i$ 的第 $j$ 个环状集合（ring set），如图 1 所示。易见，对于任意点 $p \in P$，其恰好属于某一个 ring 集合，因为所有点距离 $\mathcal{A}$ 中所有中心的最大距离不超过 $\beta nR$。因此，这些 ring 集合对点集 $P$ 构成了一个不重叠的划分。为计算中心集合 $\mathcal{A}$，我们采用 Indyk 提出的算法，时间复杂度为 $O(nk)$。

**Remark**: 集合 $P_{i,j}$ 的构造过程：

对于每个点 $p \in P$，首先计算其到所有中心的距离：

$$\mathrm{d}(p, a_1),\ \mathrm{d}(p, a_2),\ \ldots,\ \mathrm{d}(p, a_m)$$

由此可确定该点属于的最近中心 $a_i$ 及其对应的簇 $P_i$。接着，根据 $d(p, a_i)$ 与半径阈值 $R$ 的关系，可立即确定其所在的 ring 集合编号 $j$：

$$j = \begin{cases} 0, & 若 d(p, a_i) \leq R \\ \left\lceil \log\left(\dfrac{d(p, a_i)}{R}\right) \right\rceil, & 若 d(p, a_i) > R \end{cases}$$

这一过程可在 $O(mn)$ 时间内完成。由于 $m \leq \alpha k$ 且 $\alpha = O(1)$, 故总时间复杂度为 $O(\alpha kn) = O(nk)$。

**步骤二：随机抽样**

设采样大小为：

$$s = \left\lceil \frac{c\beta^2}{\varepsilon^2} \left( k \ln n + \ln \frac{1}{\lambda} \right) \right\rceil, \tag{1}$$

其中 $c$ 是一个充分大的常数。

对于所有 $i = 1, \ldots, m$ 和 $j = 0, \ldots, \phi$，若 $|P_{i,j}| \leq s$，则直接令：

$$\mathcal{S}_{i,j} = P_{i,j}.$$

否则，从 $P_{i,j}$ 中以**有放回**的方式独立、均匀地随机采样 $s$ 个点，并为每个采样点赋予权重 $|P_{i,j}|/s$，从而构成加权集合 $\mathcal{S}_{i,j}$。我们假设 $|P_{i,j}|/s$ 为整数（可通过适当取整或补齐实现）。

最终，构造的核集 $\mathcal{S}$ 定义为：

$$\mathcal{S} = \bigcup_{i,j} \mathcal{S}_{i,j}.$$

我们称该集合 $\mathcal{S}$ 是点集 $P$ 的一个 $(k, \varepsilon)$-coreset。

**分析**

固定 $\forall c = \{c_1, c_2, \ldots, c_k\} \subseteq \mathbb{R}^d$, 定义距离函数 $g(p, c) = \min_{1 \leq i \leq k} \|p - c_i\|$, 对于 $\forall p, p' \in P_{i,j}$ 有：

$$|g(p, c) - g(p', c)| \leq 2^{j+1} \cdot R$$

利用 Hoeffding Bound, 在 $P_{i,j}$ 中取 $x = \Theta(\frac{1}{\epsilon_0^2} log\frac{1}{\lambda})$ 个点，以 $1-\lambda$ 的概率有:

$$\left| \frac{1}{|S_{i,j}|} \sum_{p \in S_{i,j}} g(p,c) - \frac{1}{|P_{i,j}|} \sum_{p \in P_{i,j}} g(p,c) \right| \leq \varepsilon_0 \cdot 2^{j+1} R$$

同时有:

$$\left| \sum_{i=1}^{m} \sum_{j=0}^{\phi} \sum_{p \in S_{i,j}} \frac{|P_{i,i}|}{x} g(p,c) - \sum_{j=1}^{m} \sum_{t=0}^{\phi} \sum_{p \in S_{i,j}} g(p',c) \right| \leq \sum_{i=1}^{m} \sum_{j=0}^{\phi} \varepsilon_0 \cdot 2^{j+1} R \cdot |P_{i,j}| \leq 4\epsilon_0 \cdot \nu(\mathcal{A}, P)$$

for $j \geq 1$.

**Theorem 2.2.** 对于所有大小不超过 $k$ 的集合 $C \subseteq P$，有:

$$|\nu(C,P) - \nu(C,S)| \leq \varepsilon \nu(C,P),$$

成立的概率至少为 $1 - m(\phi+1)\lambda$, $x = O(\frac{\alpha^2}{\epsilon^2} log\frac{klogn}{\lambda})$, 所以 $|\mathcal{S}| = O(m\phi x)$。

**性质:**

1. 如果 $\mathcal{S}_1$ 和 $\mathcal{S}_2$ 分别是两个互不相交的集合 $P_1$ 和 $P_2$ 的 $(k,\varepsilon)$-coreset，则 $\mathcal{S}_1 \cup \mathcal{S}_2$ 是 $P_1 \cup P_2$ 的一个 $(k,\varepsilon)$-coreset。

2. 如果 $\mathcal{S}_1$ 是 $\mathcal{S}_2$ 的一个 $(k,\varepsilon)$-coreset，且 $\mathcal{S}_2$ 是 $\mathcal{S}_3$ 的一个 $(k,\delta)$-coreset，则 $\mathcal{S}_1$ 是 $\mathcal{S}_3$ 的一个 $(k,(1+\varepsilon)(1+\delta)-1)$-coreset。

**Merge-Reduce Tree:**

1. 叶节点为 1 到 12，表示最底层的数据块。

2. 节点 3, 6, 10, 13 分别合并子节点。

3. 节点 7 和 14 进一步合并，最终合并为根节点 15。

主要应用于流数据更新 Coreset.

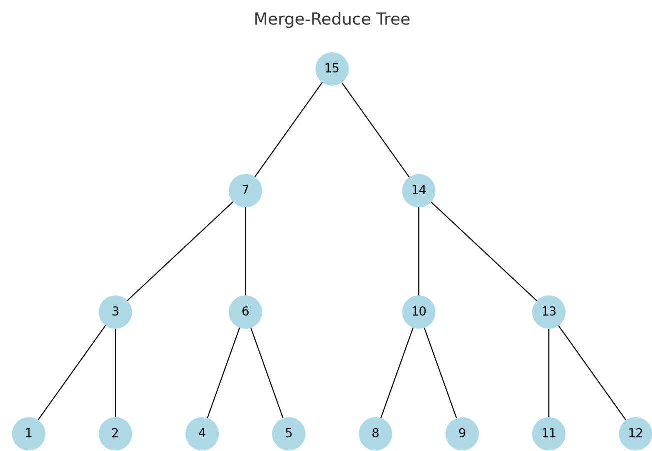


Figure 2: Merge-Reduce Tree

# 3 Importance Sampling

设 $\{q_1, q_2, \ldots, q_n\} \subseteq [0, \Delta]$。

根据 **Hoeffding 不等式**，若均匀采样点数为 $m = O\left(\frac{\Delta^2}{\varepsilon^2} \log \frac{1}{\delta}\right)$，则以 $1-\delta$ 的概率，误差不超过 $\varepsilon$。但是当 $\mu = \frac{1}{n}\sum_{i=1}^{n} q_i \ll \Delta$ 时，均匀采样效果不好。

如果对所有数据点做变换：

$$q_i \to \bar{q}_i = \frac{\phi}{n \cdot \phi_i} \cdot q_i, \quad 0 \leq \bar{q}_i \leq \phi_i, \quad \phi = \frac{1}{n}\sum_{i=1}^{n} \phi_i$$

采样概率为：

$$\mathrm{Prob}(\bar{q}_i) = \frac{\phi_i}{\phi}$$

期望不变：

$$\sum_{i=1}^{n} \mathrm{Prob}(\bar{q}_i) \cdot \bar{q}_i = \sum_{i=1}^{n} \frac{\phi_i}{\phi} \cdot \frac{\phi}{n \cdot \phi_i} \cdot q_i = \frac{1}{n}\sum_{i=1}^{n} q_i$$

$$\forall i, \quad \bar{q}_i \in [0, \frac{\phi}{n}], \quad q_i \in [0, \Delta]$$

所以样本复杂度变为 $m = O\left(\frac{\phi^2}{n^2\varepsilon^2} \log \frac{1}{\delta}\right)$。

# References

[1] K. Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.

[2] D. Feldman. Core-sets: Updated survey. *Sampling techniques for supervised or unsupervised tasks*, pages 23–44, 2020.

[3] J. Huang, R. Huang, W. Liu, N. Freris, and H. Ding. A novel sequential coreset method for gradient descent algorithms. In *International Conference on Machine Learning*, pages 4412–4422. PMLR, 2021.

[4] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

[5] S. Sinha, H. Zhang, A. Goyal, Y. Bengio, H. Larochelle, and A. Odena. Small-gan: Speeding up gan training using core-sets. In *International Conference on Machine Learning*, pages 9005–9015. PMLR, 2020.

[6] J. Yoon, D. Madaan, E. Yang, and S. J. Hwang. Online coreset selection for rehearsal-based continual learning. In *International Conference on Learning Representations*.