Документация GradientKMeans
Описание модели

1 Общая формулировка задачи

Пусть $X = \{x_1, x_2, ..., x_N\}$ - множество данных, где $x_i \in \mathbb{R}^D$, и K - количество кластеров. Цель - найти центроиды $C = \{c_1, c_2, ..., c_K\}$, минимизирующие целевую функцию:

$$\mathcal{L}(X, C) = \sum_{i=1}^{N} \min_{k=1}^{K} d(x_i, c_k) \tag{1}$$

где $d(x_i, c_k)$ - функция расстояния между точкой x_i и центроидом c_k.

2 Метрики расстояния

2.1 Евклидово расстояние

$$d_{\text{eucl}}(x, y) = \|x - y\|_2 = \sqrt{\sum_{j=1}^{D} (x_j - y_j)^2} \tag{2}$$

2.2 Манхэттенское расстояние

$$d_{\text{manhattan}}(x, y) = \|x - y\|_1 = \sum_{j=1}^{D} |x_j - y_j| \tag{3}$$

2.3 Косинусное расстояние

$$d_{\text{cosine}}(x, y) = 1 - \frac{x \cdot y}{\|x\|_2 \|y\|_2} = 1 - \frac{\sum_{j=1}^{D} x_j y_j}{\sqrt{\sum_{j=1}^{D} x_j^2} \sqrt{\sum_{j=1}^{D} y_j^2}} \tag{4}$$

2.4 Расстояние Хэмминга

$$d_{\text{hamming}}(x, y) = \frac{1}{D} \sum_{j=1}^{D} \mathbb{1}_{x_j \neq y_j} \tag{5}$$

3 Функции потерь

3.1 Стандартная функция K-средних

$$\mathcal{L}_{\text{standard}} = \frac{1}{N} \sum_{i=1}^{N} d(x_i, c_{a_i}) \tag{6}$$

где $a_i = \arg\min_k d(x_i, c_k)$ - назначение кластера для точки x_i.

3.2 Внутрикластерная дисперсия

$$\mathcal{L}_{\text{variance}} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|S_k|} \sum_{x_i \in S_k} d(x_i, c_k) \tag{7}$$

где $S_k = \{x_i : a_i = k\}$ - множество точек в кластере k.

3.3 Контрастные потери

$$\mathcal{L}_{\text{contrastive}} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\sum_{j \in P_i} \exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{j \neq i} \exp(\text{sim}(z_i, z_j)/\tau)} \tag{8}$$

где:

- $z_i = \frac{x_i}{\|x_i\|_2}$ - нормализованные эмбеддинги
- $\text{sim}(u, v) = u^T v$ - косинусная схожесть
- $P_i = \{j : a_j = a_i, j \neq i\}$ - положительные пары
- τ - температурный параметр

3.4 Потери с регуляризацией энтропией

$$\mathcal{L}_{\text{entropy}} = \mathcal{L}_{\text{standard}} - \alpha \sum_{k=1}^{K} p_k \log p_k \tag{9}$$

где $p_k = \frac{|S_k|}{N}$ - доля точек в кластере k, α - коэффициент регуляризации.

3.5 Потери с регуляризацией центроидов

$$\mathcal{L}_{\text{centroid}} = \mathcal{L}_{\text{standard}} + \lambda \sum_{i=1}^{K} \sum_{j=i+1}^{K} \exp\left(-\frac{\|c_i - c_j\|_2}{d_{\min}}\right) \tag{10}$$

где λ - коэффициент регуляризации, $d_{\min}$ - минимальное желаемое расстояние.

4 Алгоритм инициализации K-means++

Algorithm 1 K-means++ инициализация

Require: Данные X, количество кластеров K
Ensure: Инициализированные центроиды $C = \{c_1, ..., c_K\}$

1: Выбрать первый центроид c_1 случайно из X
2: **for** $k = 2$ to K **do**
3: Вычислить расстояния: $D(x_i) = \min_{j=1}^{k-1} \|x_i - c_j\|^2$
4: Выбрать $c_k = x_i$ с вероятностью $p_i = \frac{D(x_i)}{\sum_{j=1}^{N} D(x_j)}$
5: **end for**
6: **return** C

5 Градиентная оптимизация

5.1 Обновление центроидов

Центроиды обновляются через градиентный спуск:

$$c_k^{(t+1)} = c_k^{(t)} - \eta \nabla_{c_k} \mathcal{L} \tag{11}$$

где η - скорость обучения.

5.2 Вычисление градиентов

Для стандартной функции потерь:

$$\nabla_{c_k} \mathcal{L}_{\text{standard}} = \frac{1}{N} \sum_{i \in S_k} \nabla_{c_k} d(x_i, c_k) \tag{12}$$

Конкретные градиенты для метрик расстояния:
Евклидово расстояние:

$$\nabla_{c_k} d_{\text{eucl}}(x_i, c_k) = \frac{c_k - x_i}{\|x_i - c_k\|_2 + \epsilon} \tag{13}$$

Манхэттенское расстояние:

$$\nabla_{c_k} d_{\text{manhattan}}(x_i, c_k) = \text{sign}(c_k - x_i) \tag{14}$$

6 Инкрементное обучение

При инкрементном обучении на чанке X_{chunk}:

$$\mathcal{L}_{\text{incremental}} = \frac{1}{|X_{\text{chunk}}|} \sum_{x_i \in X_{\text{chunk}}} d(x_i, c_{a_i}) \tag{15}$$

С адаптивной скоростью обучения:

$$\eta_{\text{inc}} = \eta \cdot \gamma \tag{16}$$

где $\gamma \in (0, 1]$ - коэффициент уменьшения.

7 Метрики качества кластеризации

7.1 Инерция (Within-Cluster Sum of Squares)

$$\text{Inertia} = \sum_{i=1}^{N} \min_k \|x_i - c_k\|^2 \tag{17}$$

7.2 Silhouette Score

Для каждой точки x_i:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{18}$$

где:

- $a(i) = \frac{1}{|S_{a_i}|-1} \sum_{j \in S_{a_i}, j \neq i} d(x_i, x_j)$ - среднее расстояние внутри кластера
- $b(i) = \min_{k \neq a_i} \frac{1}{|S_k|} \sum_{j \in S_k} d(x_i, x_j)$ - среднее расстояние до ближайшего кластера

Общий score:

$$\text{Silhouette} = \frac{1}{N} \sum_{i=1}^{N} s(i) \tag{19}$$

7.3 Баланс кластеров

$$\text{Balance} = \frac{\min_k |S_k|}{\max_k |S_k|} \tag{20}$$

8 Условия остановки

8.1 Сходимость по центроидам

$$\|C^{(t)} - C^{(t-1)}\|_F < \epsilon \tag{21}$$

где $\|\cdot\|_F$ - норма Фробениуса.

8.2 Ранняя остановка

Остановка при отсутствии улучшения потерь в течение P итераций:

$$\mathcal{L}^{(t)} > \mathcal{L}_{\text{best}} - \delta \quad \text{для } t = t_{\text{best}} + 1, ..., t_{\text{best}} + P \tag{22}$$

9 Обработка пустых кластеров

При обнаружении пустого кластера $S_k = \emptyset$:

- Находим точку с максимальным расстоянием до ближайшего центроида:
$$x_{\text{far}} = \arg\max_{x_i} \min_j d(x_i, c_j) \tag{23}$$
- Переинициализируем: $c_k = x_{\text{far}} + \mathcal{N}(0, \sigma^2 I)$

10 Заключение

Модель GradientKMeans представляет собой градиентную версию алгоритма K-средних с расширенными возможностями регуляризации, различными метриками расстояния и функциями потерь. Обеспечивает гибкость и устойчивость алгоритма при работе с разнообразными типами данных и сценариями использования.