内积量化（分解）

Product Quantization

方佳艳

1. 引言

内积量化，顾名思义，就是对向量进行分解，对高维空间的数据点进行划分。类似于 LSH，PQ 也是为了降低线性复杂度，即，将$\Theta(nd)$降低至$\Theta(n+d)$。

注：n 和 d 都是非常大的值，甚至 $d>n$。

PQ 的缺点：和 LSH 相比，没有理论上的质量保证，不能像 LSH 可以证明成功的概率多大，近似的程度多高。

但 PQ 的实用性非常好，常用于搜索引擎。例如，Yahoo！的图片搜索引擎。

2. K-means 聚类

2.1. 定义

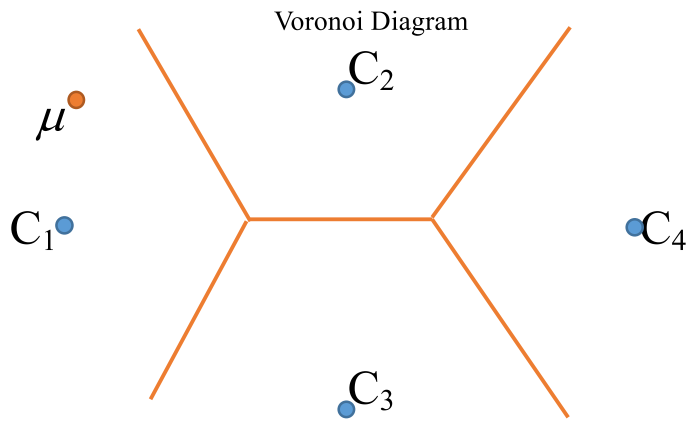
给定某个数据点集 $P \subseteq R^d$，其中$|P|=n$，$k \geq 1$（一般 $k<<n$）

目标：找到类中心$\{C_1,\cdots,C_k\} \subseteq R^d$，满足：

$$\sum_{u \in P} \min_{1 \leq j \leq k} \|u-C_j\|^2 \text{ 最小}$$

即，P 中的每个点被划分到离它最近的类中心。

2.2. 与 Voronoi Diagram 的联系



$$k=1 \Rightarrow \text{0-维 PCA}$$

$$\Rightarrow \forall C_j \text{一定是第 } j \text{ 个类的重心}$$

我们可以通过 Voronoi Diagram 来理解 k-means 聚类：作每个类中心的垂直平分线，将空间分成 k 份。因此，k-means 聚类找到这 k 个类中心之后也就相当于对点集 P 作了 k 个类

划分。

2.3. 一种简单的 K-means 聚类算法-Lloyd 算法

（1）随机初始化 $\{C_1,\cdots C_k\}$

（2）迭代直到目标函数值收敛到稳定区间

①根据 $\{C_1,\cdots C_k\}$ 将 P 划分成 k 类

②对每一类，将类中心更新成该类的重心。

以上即为 Lloyd 算法：

①时间复杂度：

假设迭代次数为 t，则复杂度为 $\Theta(t(knd+nd))=\Theta(tknd)$

②质量保证：只能保证局部最优。

2.4. 结论

（1）即使 $d=2$，k-means 是 NP-hard。

（2）即使 $k=2$，k-means 也是 NP-hard。

注：已被证明：k-means 最优解是 NP-hard，但不代表它的近似解是 NP-hard。

3. 初阶版-向量量化（Vector Quantization，VQ）

3.1. 定义

给定某个数据点集 $P\subseteq R^d$，其中 $|P|=n$，$k\geq 1$（一般 $k<<n$）

目标：找到类中心 $\{C_1,\cdots,C_k\}\subseteq R^d$，满足：

$$\sum_{u\in P}\min_{1\leq j\leq k}\|u-C_j\|^2 \text{ 最小}$$

$\Rightarrow$ VQ 即为 k-means 聚类：

给定一个最近点查询问题，VQ 的目标是用 k 个类中心来近似表达点集 P，

对某个查询点 $q\in R^d$，从 $\{C_1,\cdots C_k\}$ 中返回离 q 最近的类中心。

3.2. 分析：

查询时间：$\Theta(kd),k<<n$

极限情况：$k=n\Rightarrow\Theta(nd)$，误差为 0。

优点：简单易行；

缺点：误差大：k 越大，误差越小，当 k=n 时，误差为 0。

因此，为了降低 VQ 的查询误差，引入 Product Quantization (PQ)。

Quantization：对 d维空间作分解。

4. VQ 的升级版-内积量化（PQ）[1]

4.1. PQ 的构造

①将空间 R^d 划分成 m 个子空间： $R^d \to R^{d/m} \times R^{d/m} \times \cdots \times R^{d/m}$（笛卡尔积）

在每个子空间 $R^{d/m}$，点集 P 都存在一个投影 $P_j, j=1,\cdots,m$

②对 $P_j, j=1,\cdots,m$ 作 k-means 聚类，从而得到第 j 个子空间的 k 个类中心 $\{C_1^j, C_2^j, \cdots, C_k^j\}$

③建立一个“codebook”字典：（由 $k \times m$ 个类中心组成）

$$\begin{array}{c}\{C_1^1, C_2^1, \cdots, C_k^1\} \\ \vdots \quad \vdots \quad \quad \vdots \\ \{C_1^m, C_2^m, \cdots, C_k^m\}\end{array}$$

④建立表 A（m 行 n 列）：

每一列对应着点集 P 中的一个数据点 μ，该列的第 j 个 component 对应的是 μ 在第 j 个子空间上，离它最近的类中心的序号。e.g.

$$\forall \mu \in P, \mu \begin{array}{l} \nearrow C_{t_1}^1 \\ \rightarrow C_{t_2}^2 \\ \quad \vdots \\ \searrow C_{t_m}^m \end{array} \Rightarrow \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_m \end{pmatrix}, \text{ 用 } C_t^1 \text{ 来近似 } \mu \text{ 在 } R^{d/m} \text{ 上的投影。}$$

⑤建立表-B（m 行 C_k^2 列）

每一行分别对应一个子空间，e.g. 第 j 行记录的是第 j 个子空间上，所有类中心 $\{C_1^j, C_2^j, \cdots, C_k^j\}$ 两两之间的距离。

4.2. PQ 的查询

对于任一查询点 $q \in R^d$：

①计算 q 与字典中类中心的距离：

$$q \begin{array}{l} \nearrow C_{s_1}^1 \\ \rightarrow C_{s_2}^2 \\ \quad \vdots \\ \searrow C_{s_m}^m \end{array} \Rightarrow \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{pmatrix}$$

②和表-A 中每一列作比较，比较的差值可以在表-B 中查找，即，用表-B 中的距离来近似 $\|q-\mu\|$。（用 B 中的距离作近似，复杂度为 $\Theta(m)$，原始距离的复杂度为 $\Theta(d)$，而 $m << d$，很大程度上降低了计算复杂度）

③输出 q 的最近邻。

4.3. PQ 的思路总结

空间分解：$R^d \to R^{d/m} \times R^{d/m} \times \cdots \times R^{d/m}$

$\forall \mu \in P \subseteq R^d, \mu = [\mu_1, \mu_2, \cdots, \mu_m]$，其中 $\mu_j \in R^{d/m}, j = 1, \cdots, m$

对任一查询点，$q \in R^d$，$q = [q_1, q_2, \cdots, q_m]$，其中 $q_j \in R^{d/m}, j = 1, \cdots, m$

$$\begin{aligned}\|\mu - q\|^2 &= \|\mu_1 - q_1\|^2 + \|\mu_2 - q_2\|^2 + \cdots + \|\mu_m - q_m\|^2 \\ &\approx \left\|C_{t_1}^1 - C_{s_1}^1\right\|^2 + \left\|C_{t_2}^2 - C_{s_2}^2\right\|^2 + \cdots + \left\|C_{t_m}^m - C_{s_m}^m\right\|^2\end{aligned}$$

上述差值均存在表-B 中。

4.4. 复杂度分析

①构造 PQ 的时间复杂度：

k-means 聚类的时间+建立表-A 的时间+建立表-B 的时间：

$T(k-means) + \Theta(mn) + \Theta(k^2 m)$，其中，$k, m << n, d$

当使用 Lloyd 算法时，$T(k-means) = m(t \cdot k \cdot n \cdot (d/m)) = \Theta(knd) = \Theta(nd)$，其中，$t$ 为迭代次数，t, k 均可以看成常值。

②空间复杂度：

存储各个子空间的所有类中心+表-A 的空间+表-B 的空间：

$\Theta(m \cdot k \cdot (d/m)) + \Theta(mn) + \Theta(k^2 m) = \Theta(n + d)$

③查询时间复杂度：

$\Theta(m \cdot k \cdot (d/m)) + \Theta(mn) = \Theta(n + d) << \Theta(nd)$

5. PQ 的缺陷

PQ 可能误差很大，因为 PQ 的效果取决于子算法 k-means 的效果，而 k-means 的聚类效果本身是不确定的，没有理论上的保证。因此，在做最近点查询的时候，PQ 的误差可能很大。

6. 参考文献

[1] Jegou, Herve, Matthijs Douze, and Cordelia Schmid. "Product quantization for nearest neighbor search." IEEE transactions on pattern analysis and machine intelligence 33.1 (2010): 117-128.