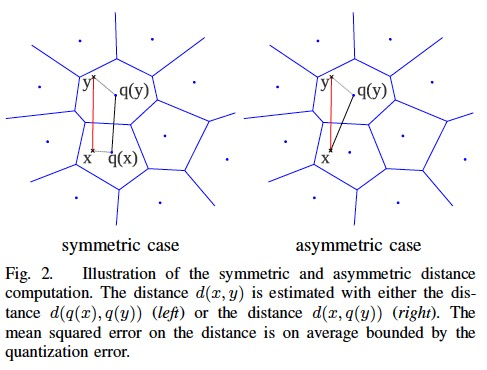
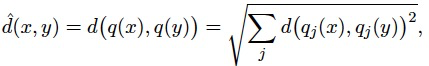
# ANN: 乘积量化与优化乘积量化

乘积量化（Product Quantization, PQ）[1][2][3]可以看做是一种处理高维矢量数据的有损压缩技术。它可以在压缩空间相对正确地重建数据和进行距离计算。

PQ将高维矢量空间分解为低维子空间的笛卡尔积，且独立量化每个子空间。最终的矢量是一个包含子空间量化索引的短码。两个矢量的欧氏距离可以高效地通过它们的短码估算。其精确度可以通过非对称版本增加，搜索效率也可以通过结合反转索引文件提高。



1. PQ的对称与非对称距离计算



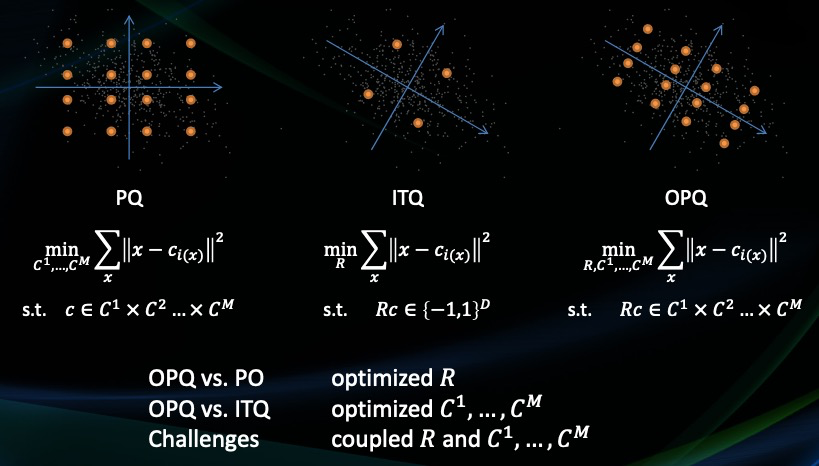
对称场景



非对称场景

Optimized PQ[4]即是寻找最优的centroids和对应的变换函数R（将x映射为对应的centroids）。

具体实现可参考faiss[3]。简单的python版本例子可参见[6]。



1. PQ和OPQ的关系

Polysemous codes将量化结果转化为二进制码，使用训练的方法获取最优的centroids。相似的数据得到相近的Hamming距离。

[7]介绍PQ和HNSW，值得一看。链接[8,9]不错.[11]是LSH

1. <https://hal.inria.fr/inria-00514462v2/document>
2. <https://blog.csdn.net/u010368556/article/details/80960510>
3. <https://github.com/facebookresearch/faiss/wiki>
4. <http://ieeexplore.ieee.org/abstract/document/6678503/>
5. <http://kaiminghe.com/cvpr13/cvpr13opq_ppt.pdf>
6. <https://github.com/matsui528/nanopq>
7. <https://yongyuan.name/blog/opq-and-hnsw.html>
8. <https://github.com/hudengjunai/DeepEmbeding>
9. <https://github.com/caoyue10/DeepHashingBaselines>
10. <https://github.com/FALCONN-LIB/FALCONN/wiki/LSH-Primer>