What is producct quantization ?

Was is product quantization: Quantization based methods. These techniques are closely related to clustering. In these methods, a binary code of a data point encodes the index of a cluster containing the data point. Product Quantization (PQ) [13] decomposes the original data space into lower-dimensional subspaces and quantizes each subspace separately using kmeans clustering. It then computes a binary code as a concatenation of cluster indices, encoded in subspaces [Distance Encoded Product Quantization]

\* was ist product quantization

Das ist eine Methode , um die große Datenmenge in höhe Dimension zu kleinen Dimension zu konvertieren, damit die Algorithmus schneller mit Datenmenge arbeiten kann. Wir kÜnnen auch die groºe Dimension einfach mit product quantizaion aufteilen.

[Online Product Quantization Book], [Product Quantization for Nearest Neighbor Search ]

Beispiel: D\* = D / m : m ist die Nummer zu dividieren. B.z: 100 Dimension = 20 Dimension \* 5.

<=> 100 dimension von 1000 Element = 5 Mal \* 1000 Element von 20 Dimension

Neu Vector wurde summariert wie das unter Bild:

codebook C = C1 ×···×CM

K of sub-codewords : 256 defined by user

\* Komplexitèt:

O(DK/M ) : D dimension, k ist 256b, M subvector.

M bigger => O reduce

Encode in PQ Kmeans: pqkmeans\_cluster = pqkmeans.clustering.PQKMeans(encoder=encoder, k=100)

- The product quantizer is clearly the the only one that can be indexed in memory for large values of k

- Binary-based methods convert an input vec-tor to a binary vector

- Jegou: J ╠üegou et al. applied the idea to the ANN problem26) by demonstrating that

(1) PQ can be used to compress high-dimensional fea- ture vectors,

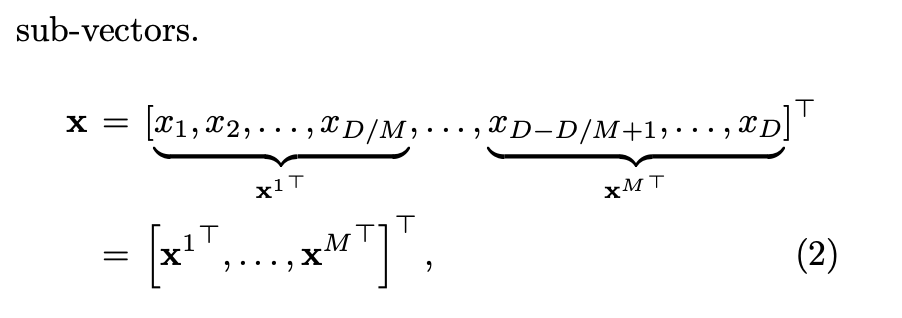
(2) the distance between an original vector and a PQ-encoded code can be efficiently approximated, and

(3) fast search system can be built by combining PQ-encoding and inverted indexing.

where mth sub-vector is denoted as xm Γêê RD/M, for each m Γêê {1,...,M}.

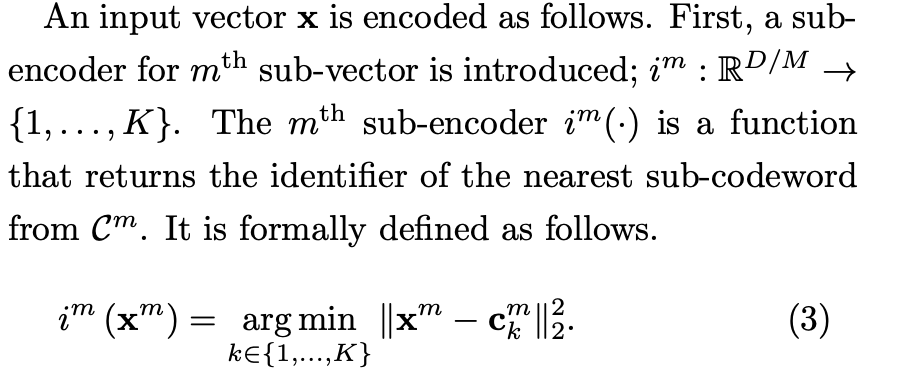
[A Survey of Product Quantization Yusuke Matsui ]

1. Encoding



encoder ist eine Verkehrtung von sub-encoder

PROCESS:



**Next, let us describe how to encode a vector**. The idea is to independently encode each sub-vector to an **identifier**, and to represent the vector as a concatenation of the **identifiers**. In a training phase, a sub-codebook for each m ∈ {1,...,M} is created: Cm = {cmk }Kk=1, where we call each cmk ∈ RD/M as a sub-codeword. The number K of sub-codewords for each sub-codebook is a parameter specified by a user. Cm is trained by run- ning the standard k-means clustering over the mth part of the training vectors.

In a training phase, a sub-codebook for each m Γêê {1,...,M} is created:

Cm = {cmk }K k=1, where we call each cmk Γêê RD/M as a sub-codeword.

- sub-encoder finds the nearest sub-codeword. The formula is : i (x)=argminΓêÑx ΓêÆckΓêÑ2

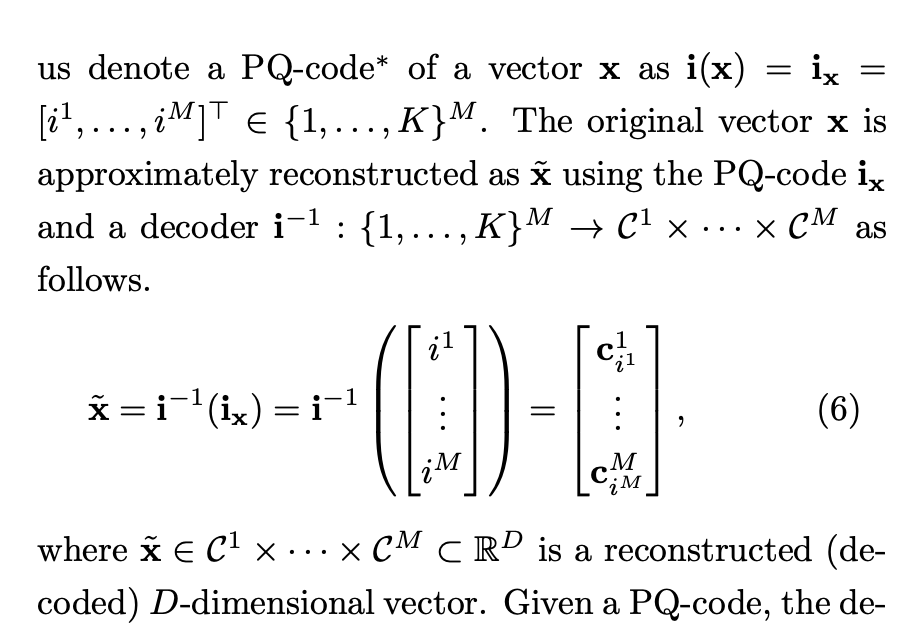
- Komplexity of the process is O(DK), D is dimension , k is

- next encoder will define the concatetion of sub-encoders

- encoder breaks the input vector to sub-vector, then applies sub-encoders for each sub-vector

1.2 Decoding (reconstruction)

An advantage of PQ is that the original vector can be approximately reconstructed from a PQ-code.



1. Memory Comsumption  
   PQ-code is a memory efficient data representation.

The original 100 D vector requies  100 \* 64 = 6400 bit if 64 bit float is used for each element. On the other, a PQ-code requires only 20 Dimension \* log2(256) = 20 \* 8 = 160bit.

1. Problem of product quantization

PQKmeans can not work with big data (e.g: 100 dimension with 6 , 7 milion data)

1. Conslution

Approximation of k-means

Tens to hundreds of times faster than k-means

Tens to hundreds of times more memory efficient than k-means

Compatible with scikit-learn

Portable; one-line installation

PQk-means would be tens to hundreds of times faster than k-means depending on your machine. Then let's see the accuracy. Since the result of PQk-means is the approximation of that of k-means, k-means achieved the lower error:

[PQk-means: Billion-scale Clustering for Product-quantized Codes] :

„ In this paper, we propose PQk-means, which is a billion-scale clustering method, and can be performed on a single computer with only a reasonable memory consumption (less than 32 GB of RAM) within a single day.“.

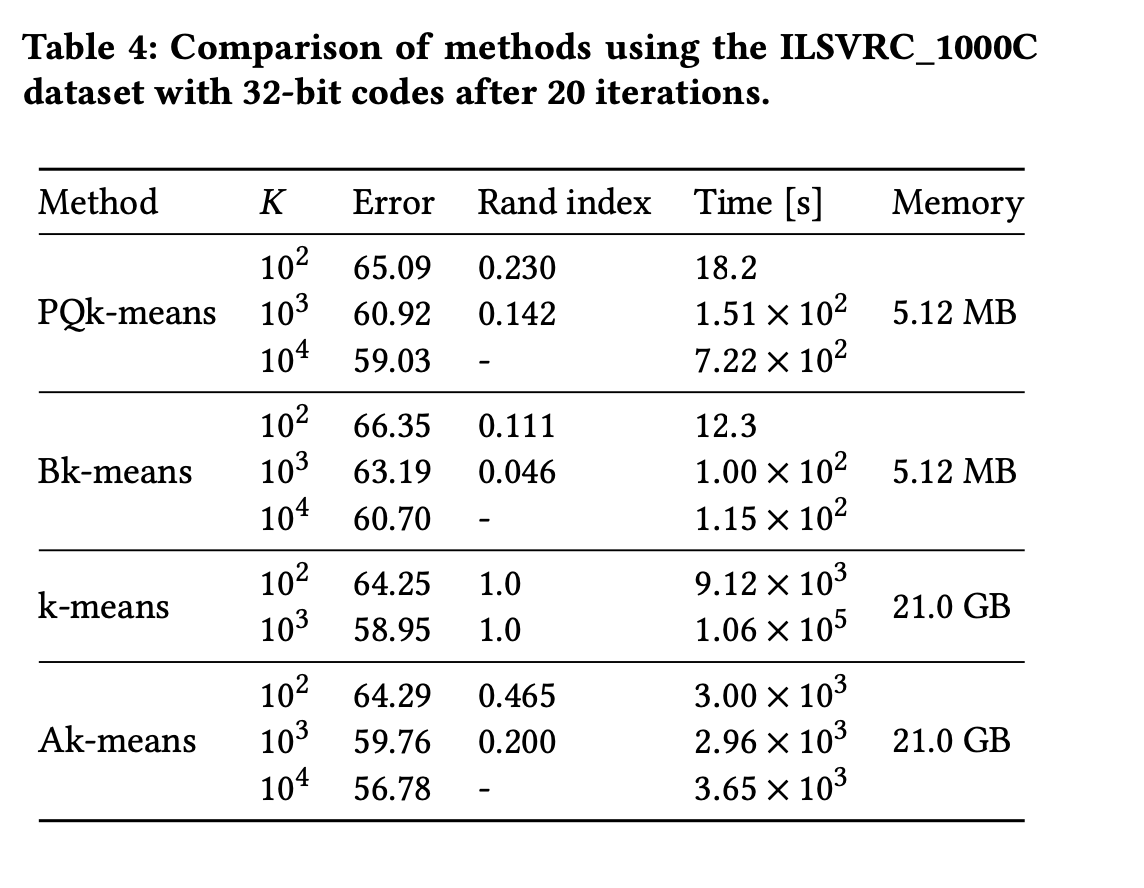
**Fragen**:

Aber PQKmeans ist über 5 milion data gestorben.

Methode 1: SCALAR QUANTIZATION USING kMEANS

Methode 3: Residual Vector Quantization

zeroshot learning,  
vector quantization and  
approximate nearest neighbor search



Why K 10^2, so big ? should try with images big data ?

Clustering for image like [PQk-means: Billion-scale Clustering for Product-quantized Codes ]

Big dataset:

**YFCC100M Downloader**

<https://github.com/stefanbirkner/yfcc100m-downloader>

(1.58 TB, 512 GB, and 384 GB, for YFCC100M, SIFT1B, and Deep1B, respectively),

SIFT1B

Other method:

Hierarchical Methods

<https://github.com/github-pengge/hierarchical_kmeans>

Binary Code Embedding Methods