Outline

Vector Quantisation

- Originally developed by the signal processing community for data compression (audio, image and video compression), the VQ idea has been picked up the statistics community and extended to tackle a variety of tasks (including clustering and classification).
- VQ is a simple idea for summarising data by use of codewords.
- The algorithm is very closely related to the K-means algorithm, yet works sequentially through the data when updating cluster centers.

Vector Quantisation

- Given p-dimensional data, a finite set of vectors $Y = \{y_1, \ldots, y_K\}$ of the same dimensionality must be found. Vectors y_k are called *codewords* and Y the *codebook*.
- All n observations are mapped to the indices of the code book using the following rule,

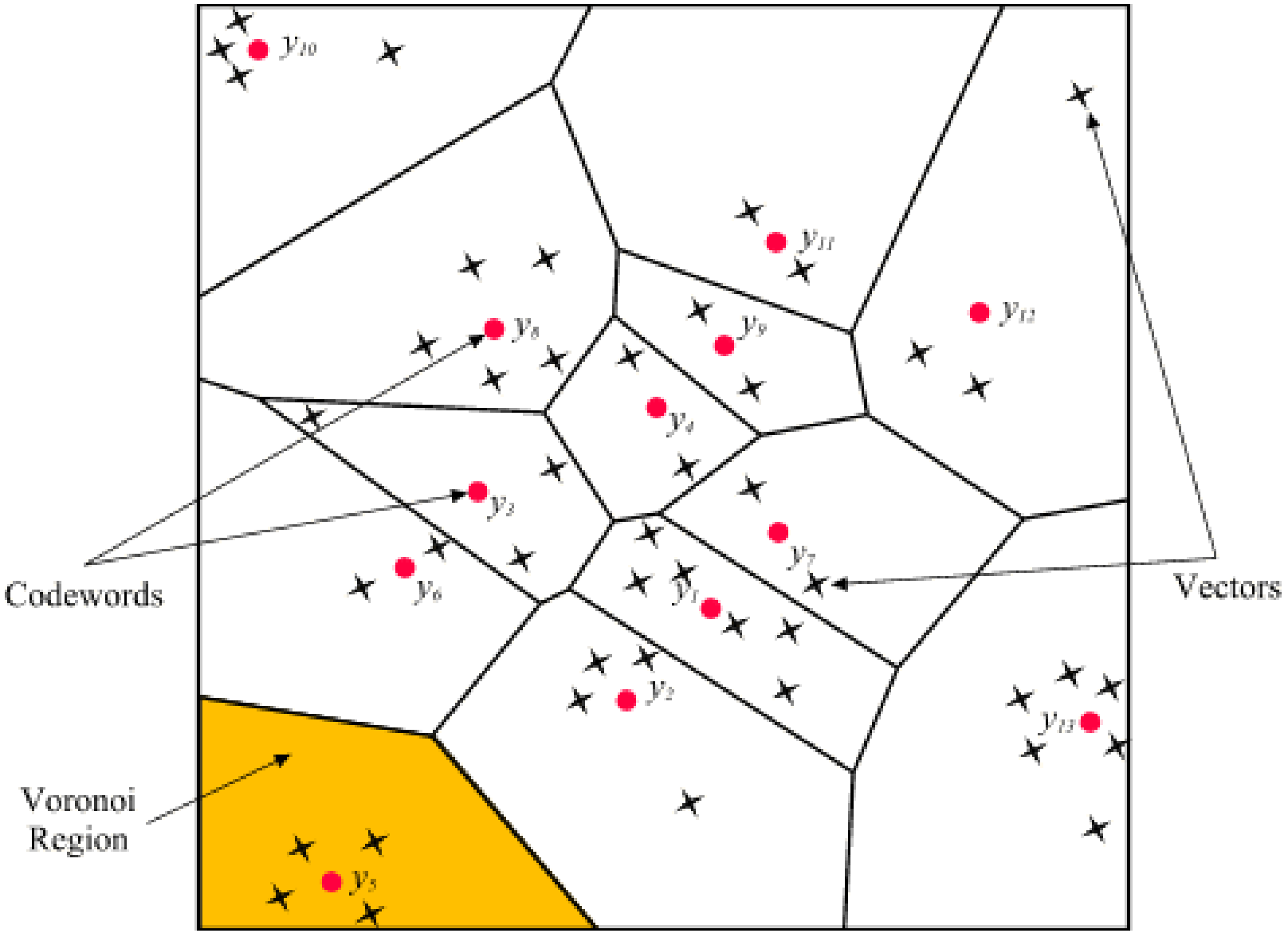
$$x_i \to y_k \quad \Leftrightarrow \quad |x_i - y_k| \leq |x_i - y_{k'}| \; \forall k'.$$

- Such a mapping induces a partition of $\mathbb{R}^p$ into Voronoi regions defined as

$$V_k = \{x \in \mathbb{R}^p : |x - y_k| \leq |x - y_{k'}| \forall k'\}$$

where $\cup_{k=1}^{K} V_k = \mathbb{R}^p$ and V_k's are disjoint except for boundaries.

Voronoi Regions



Finding a Useful Codebook

- As with K-means, a predefined number of K codewords must be found. They should be chosen to give the greatest compression in the data with minimal loss in data quality.
- Where we have more codewords than clusters, it is easy to see that we should simply place codewords at the center of areas of high density, i.e. good codebooks find cluster centers.

Vector Quantisation

The following iterative algorithm finds a good approximate solutions to this problem.

1. Randomly choose K observations to initialise the codebook.
2. Sample an observation x and let V_c be the Voronoi region where it falls.
3. Update the codebook

$$\begin{aligned} y_c &= y_c + \alpha(t)\,[x - y_c] \\ y_k &= y_k \;\; \forall k \neq c. \end{aligned}$$

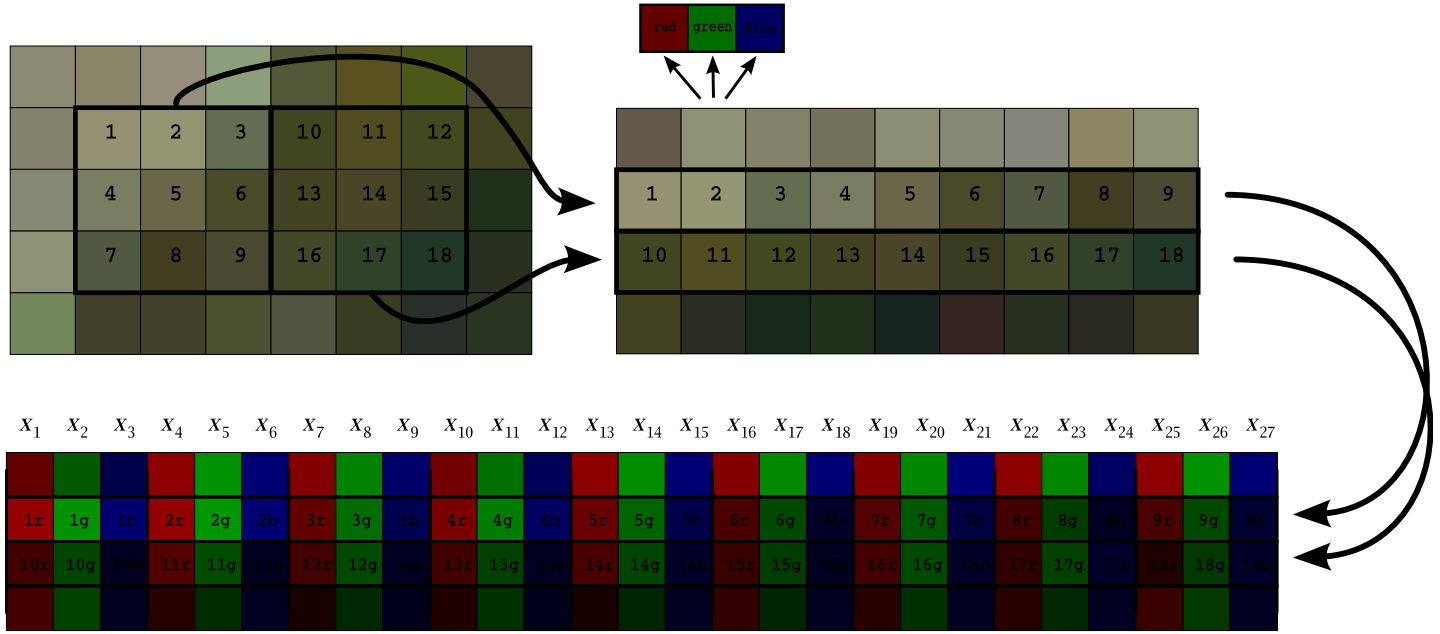
   $\alpha(t)$ quantifies the amount by which y_c moves towards of the x and decays over time to 0.
4. Repeat 2-3 until there is no change.
5. Return the codebook $Y = \{y_1, \ldots, y_K\}$

Compression

- For compression purposes, any observation $x \in \mathbb{R}^p$ is now just mapped to the set $\{1, \ldots, K\}$ of codewords, according to which Voronoi region the observation falls into.
- If a large number of observations $x_1, \ldots, x_n$ needs to be transferred, alternatively the vector of corresponding codewords in $\{1, \ldots, K\}^n$ can be transferred to achieve a compression (with a certain loss of information). Some audio and video codecs use this method.
- As with K-means, K must be specified. Increasing K 'improves the quality of the compressed image' but worsens the 'data compression rate', so there is a clear tradeoff. (For clustering, the choice of K is harder and does not have an entirely satisfactory answer).

Example: Image Compression

3×3 block VQ: View each block of 3×3 pixels as single observation



Example: Image Compression

Original image (24 bits/pixel, uncompressed size 1,402 kB)

Example: Image Compression

Codebook length 1024 (1.11 bits/pixel, total size 88kB)

Example: Image Compression

Codebook length 128 (0.78 bits/pixel, total size 50kB)

Example: Image Compression

Codebook length 16 (0.44 bits/pixel, total size 27kB)