Intelligent Data Analysis: Clustering

Martin Russell

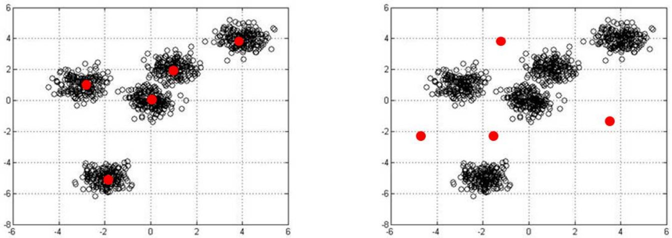
School of Computer Science, University of Birmingham

February 20, 2020

Overview

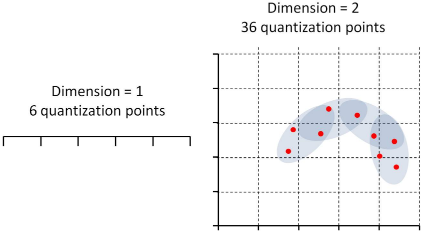
Representing clusters as centroids

Figure: Good (L) and poor (R) representation of clusters with centroids



- *Centroids* are data points located to represent a set of clusters
- Requires correct number of centroids in correct locations
- In general, the number and location of the clusters is unknown

Vector Quantization



- 1-dimensional space requires N quantization points
- M-dimensional space requires N^M quantization points
- **Curse of dimensionality**
- **Uniform** distribution of quantization points may not be optimal   non-uniform
- **Vector quantization**

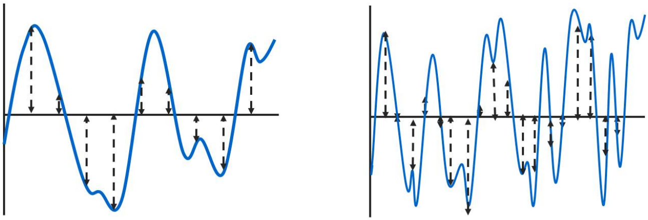
VQ for low bit-rate coding

- Suppose we want to **transmit** high-dimensional vectors across a communication channel in real-time
- Depending on **frequency of transmission** and **dimension of vectors** this may exceed the capacity of the network
- One solution is Vector Quantization:
- Using a large set of example vectors, construct a set of centroids $\{c_1, \ldots c_K\}$ - the **codebook**
- Transmit the codebook at the start of transmission
- Then, for each vector $\vec{v}$ find the closest centroid $c_{i(v)}$.
- Transmit the **index** $i(v)$

VQ for low bit-rate coding (continued)

- Suppose we need to transmit 20 dimensional real vectors at 100Hz
- Suppose each vector coordinate requires 16 bits
- 'Raw' bit rate $= 16 \times 20 \times 100 = 32,000$ bits pe second
- Now suppose we have a Vector Quantizer with 256 centroids
- 8 bits required to encode centroid identity
- VQ bit rate $= 8 \times 100 = 800$ bits per second
- **40 times reduction in bit rate**
- Example - **Speech Coding**

Conventional speech coding



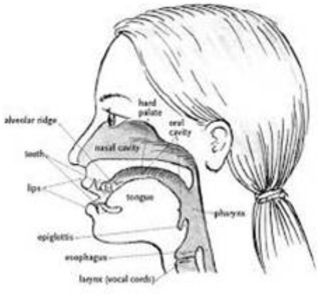
- Pulse Code Modulation (PCM) measures signal amplitude at regular intervals
- Higher frequency, faster changing signals require higher frequency sampling

Nyquists Theorem (Sampling Theorem)

Nyquist

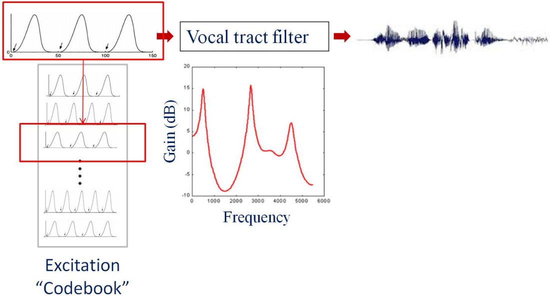
- If highest frequency component in a signal is F_{max} Hertz, then to properly encode the signal PCM must sample the signal at $2 \times F_{max}$ measurements per second
- Humans hear frequencies up to approximately $20,000$ Hertz. Audio CDs sample at $44,000$ measurements per second
- Natural, intelligent speech needs frequencies up to $4,000$ Hertz. A phone system based on PCM would need $8,000$ measurements per second ($64,000$ bits per second)
- How can this be reduced?

How do we produce speech?



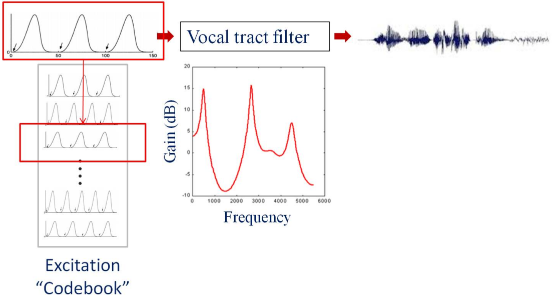
- **IDEA!** Vocal tract moves slowly
- Only need to measure it 100 times per second
- If it can be encoded with N measurements, we only need to transmit $100 \times N$ measurements per second!

The source-filter model



- **Linear Prediction (LP)** encodes VT filter with 10 integers
- Measure filter 100 times per second
- $1,000$ **measurements per second!**

Codebook Excited Linear Prediction (CELP)



- But, good quality speech requires correct excitation signal
- Build a VQ to encode possible excitation signals
- Transmit VT filter shape + excitation codebook index
- **Codebook Excited Linear Prediction (CELP)**

Clustering / VQ

- Two key questions for VQ and clustering
  1. What is a **good** VQ codebook / set of centroids?
  2. How can we obtain it?

Distortion

- *Distortion* is a measure of how well a set of centroids $C = \{c_1, ..., c_K\}$ fits a set of data $X = \{x_1, ..., x_N\}$

c_k K x_n N

- Let d be a metric
- Let $c_{i(n)}$ be the closest centroid to x_n $(n = 1, ..., N)$

$c_{i(n)}$

$$d(x_n, c_{i(n)}) = min_{k=1,...,K} d(x_n, c_k) \quad (1)$$

- The *Distortion* for the centroids C relative to the data set X is

$$Dist(C, X) = \frac{1}{N} \sum_{n=1}^{N} d(x_n, c_{i(n)}) \quad (2)$$

$\tilde{C} = \text{argmin} \mid Dist(C, X)$

Finding the 'best' set of centroids

- The best set of K centroids $\hat{C}$ minimizes

$$D(C,X) = \frac{1}{N}\sum_{n=1}^{N} d(x_n, c_{i(n)}) \tag{3}$$

- For each k let $X(k)$ be the set of data points for which c_k is the closest centroid.
- Then (3) can be re-written

$$D(C,X) = \frac{1}{N}\sum_{k=1}^{K}\sum_{x\in X(k)} d(x, c_k) \tag{4}$$

Finding the 'best' set of centroids

- Suppose that the dimension of the vector space is D. Write

$$c_k = \begin{bmatrix} c_{k,1} \\ \vdots \\ c_{k,D} \end{bmatrix} x_n = \begin{bmatrix} x_{n,1} \\ \vdots \\ x_{n,D} \end{bmatrix} \tag{5}$$

- To minimise $D(C,X)$ differentiate it with respect to each $c_{k,d}$, set the result to zero and solve

$$\frac{d}{dc_{k,d}} D(C,X) = \frac{d}{dc_{k,d}} \frac{1}{N} \sum_{j=1}^{K} \sum_{x \in X(j)} \underbrace{d(x, c_j)}_{} \tag{6}$$

$= 0$ except $j = k$

$$= \frac{1}{N} \frac{d}{dc_{k,d}} \sum_{x \in X(k)} d(x, c_k) \tag{7}$$

Finding the 'best' set of centroids

$$= \frac{1}{N} \sum_{x \in X(k)} \frac{d}{dc_{k,d}} d(x, c_k) \quad (8)$$

If d is the squared Euclidean metric (8) becomes

$$\begin{aligned}
\frac{1}{N} \sum_{x \in X(k)} \frac{d}{dc_{k,d}} d(x, c_k) &= \frac{1}{N} \sum_{x \in X(k)} \frac{d}{dc_{k,d}} \sum_{e=1}^{D} (x_e - c_{k,e})^2 && (9) \\
&= \frac{1}{N} \sum_{x \in X(k)} \frac{d}{dc_{k,d}} (x_d - c_{k,d})^2 && (10) \\
&= \frac{1}{N} \sum_{x \in X(k)} 2(x_d - c_{k,d})(-1) && (11)
\end{aligned}$$

Setting this to zero, multiplying by $\frac{-N}{2}$ and solving gives

Finding the 'best' set of centroids

$$0 = \sum_{x \in X(k)} (x_d - c_{k,d}) \tag{12}$$

$$c_{k,d} = \frac{1}{|X(k)|} \sum_{x \in X(k)} x_d \tag{13}$$

where $|X(k)|$ is the number of data points in $X(k)$.

- In other words, to minimize the distortion set the d^{th} coordinate of c_k to be the average of the d^{th} coordinates of the data points for which c_k is the closest centroid.
- But $X(k)$ depends on c_k, so c_k appears on *both* sides of (13).
- Equation (13) is not a *closed solution* for c_k

The k-means clustering algorithm

A practical solution is to use (13) for an *iterative* algorithm

1. Estimate initial centroid values $c_1^0, \cdots, c_K^0$
2. Set $i = 0$
3. For $n = 1, \cdots, N$ and $k = 1, \cdots K$ calculate $d(x_n, c_k^i)$
4. For $k = 1, \cdots, K$
5. Let $X^i(k)$ be the set of x_ns that are closest to c_k^i
6. Define c_k^{i+1} to be the average of the data points in $X^i(k)$

$$c_k^{(i+1)} = \frac{1}{|X^i(k)|} \sum_{x \in X^i(k)} x \tag{14}$$

7. $i = i + 1$. Go back to step 3.

Example

Let

$$x_1 = \begin{bmatrix} 0 \\ -5 \end{bmatrix}, x_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, x_3 = \begin{bmatrix} -2 \\ 3 \end{bmatrix}, x_4 = \begin{bmatrix} -4 \\ 7 \end{bmatrix}, x_5 = \begin{bmatrix} 3 \\ 1 \end{bmatrix},$$

$$x_6 = \begin{bmatrix} 4 \\ -2 \end{bmatrix}, x_7 = \begin{bmatrix} -1 \\ 6 \end{bmatrix}, x_8 = \begin{bmatrix} 5 \\ -6 \end{bmatrix}. \quad (15)$$

and suppose that the initial estimates of two centroids are

$$c_1^0 = \begin{bmatrix} -3 \\ 5 \end{bmatrix}, c_2^0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad (16)$$

Find the new values of c_1 and c_2 after one iteration of k-means clustering. Use the "city block" d_1 metric.

Example (continued)

The first step is to calculate the distances. For example

$$
\begin{aligned}
d_1(x_1, c_1^0) &= d_1\left(\begin{bmatrix} 0 \\ -5 \end{bmatrix}, \begin{bmatrix} -3 \\ 5 \end{bmatrix}\right) \\
&= |0-(-3)| + |-5-5| \\
&= 3+10=13
\end{aligned} \tag{17}
$$

Continue in this way to obtain the matrix of distances between data points and centroids

Example (continued)

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
c_1^0	13	7	3	3	10	14	3	19
c_2^0	9	1	5	11	2	6	7	11
c_1^0			1	1			1	
c_2^0	1	1			1	1		1

Table 1: Distances between centroids and data points (rows 2,3) and indicator of closest centroid to each data point (rows 4,5)

- So $X^0(1) = \{x_3, x_4, x_7\}$ and $X^0(2) = \{x_1, x_2, x_5, x_6, x_8\}$, and

$$c_1^1 = \frac{1}{3}(x_3 + x_4 + x_7) = \begin{bmatrix} -2.33 \\ 5.33 \end{bmatrix} \tag{18}$$

$$c_2^1 = \frac{1}{5}(x_1 + x_2 + x_5 + x_6 + x_8) = \begin{bmatrix} 2.6 \\ -2 \end{bmatrix} \tag{19}$$

Optimality

- Is the set of k centroids $\hat{C}$ created by k-means globally optimal? In other words is it true that for any set of k centroids
$$D(C, X) \geq D(\hat{C}, X)? \tag{20}$$
- No, k-means clustering is only guaranteed to find a *local* optimum.
- The solution obtained from k-means clustering depends on the *initial* centroids.

MatLab demonstration

- "Toy" 2-dimensional data set
- $K = 6$ (6 centroids)
- Initial centroids chosen at random in the "box" $-10 \leq x, y \leq 10$
- 20 iterations of k-means clustering
- Repeated 20 times

MatLab distance calculation

- Suppose X is an $N \times D$ matrix whose rows are N D-dimensional vectors
- Suppose c is a D-dimensional vector (a centroid)
- In MatLab, how do I calculate the Euclidean (d_2) distance between c and each of the N vectors in X?

MatLab distance calculation

Editor - C:\Users\Martin\Dropbox\My Courses\Engineering_Maths_2\2017-18\wk5S\MatLab\distcalc.m*

ex1.m | q1.m | solution1.m | distcalc.m*

```matlab
% Calculate distance matrix
for k=1:1:K
    Y=X;
    % Subtract C(k,:) from each row of Y
    Y=Y-C(k,:);
    % Square the result
    Y=Y.^2;
    % Sum contributions from each dimension and take square root
    if k==1
        D=sqrt(sum(Y,2));
    else
        D=[D sqrt(sum(Y,2))];
    end
end
```

Summary

- Centroids and distortion
- Derivation of the k-means clustering algorithm
- Example application of k-means clustering
- MatLab demonstration and example