Introduction to Audio Content Analysis

Module 4.2: Regression & Clustering

alexander lerch

Georgia Tech Center for Music Technology
College of Design

corresponding textbook section

Sections 4.2 & 4.3

- **lecture content**
  - regression: non-categorical data analysis
  - clustering: unsupervised data analysis

- **learning objectives**
  - describe the basic principles of data-driven machine learning approaches
  - implement linear regression in Python
  - implement kMeans clustering in Python

corresponding textbook section

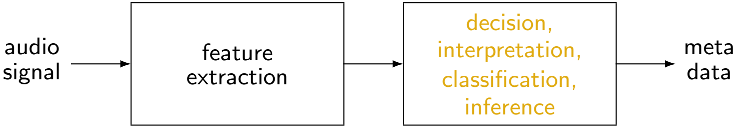
Sections 4.2 & 4.3

- **lecture content**
  - regression: non-categorical data analysis
  - clustering: unsupervised data analysis

- **learning objectives**
  - describe the basic principles of data-driven machine learning approaches
  - implement linear regression in Python
  - implement kMeans clustering in Python

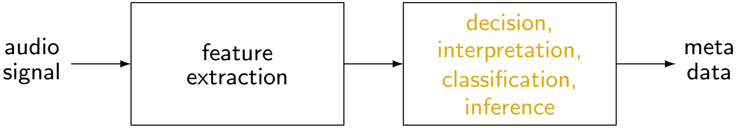
regression
introduction

remember the flow chart of a general ACA system:



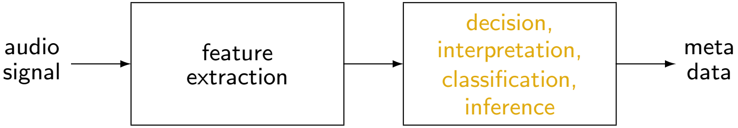
- *classification*:
  - assign class labels to data
- *regression*:
  - estimate numerical labels for data
- *clustering*:
  - find grouping patterns in data

remember the flow chart of a general ACA system:



- *classification*:
    - assign class labels to data
- *regression*:
    - estimate numerical labels for data
- *clustering*:
    - find grouping patterns in data

remember the flow chart of a general ACA system:



- *classification*:
    - assign class labels to data
- *regression*:
    - estimate numerical labels for data
- *clustering*:
    - find grouping patterns in data

regression
introduction

- given a set of pairs of data and corresponding output observations
- find model that maps input to output

- model can then be used to predict (continuous value) output for an unknown new input

regression
introduction

- given a set of pairs of data and corresponding output observations
- find model that maps input to output

- model can then be used to predict (continuous value) output for an unknown new input

regression
linear regression

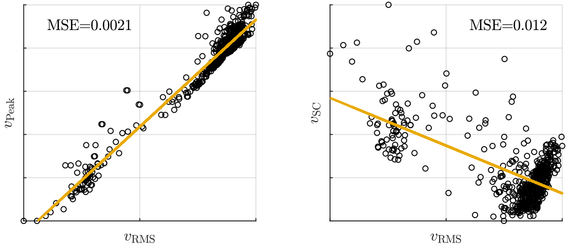
- estimate the slope m and offset b of a straight line that fits the data best:

$$\hat{y}(r) = m \cdot v(r) + b$$

- minimizing the mean squared error leads to:

$$b = \mu_y - m \cdot \mu_v$$

$$m = \frac{\sum\limits_{r=0}^{\mathcal{R}-1} (y(r) - \mu_y) \cdot (v(r) - \mu_v)}{\sum\limits_{r=0}^{\mathcal{R}-1} (v(r) - \mu_v)^2}$$



matlab source: plotLinearRegression.m

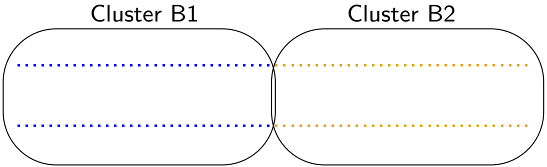
- clustering is usually unsupervised and exploratory
- group observations
  - 'similar' observations are grouped together
  - 'dissimilar' observations are in different groups
- depends on definition of 'similarity'/ distance

clustering

introduction

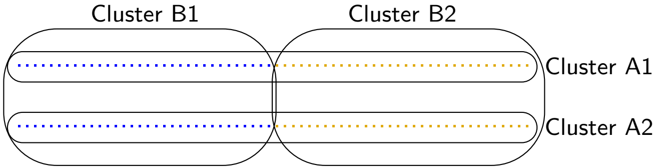
- clustering is usually unsupervised and exploratory
- group observations
  - 'similar' observations are grouped together
  - 'dissimilar' observations are in different groups
- depends on definition of 'similarity'/ distance

- clustering is usually unsupervised and exploratory
- group observations
  - 'similar' observations are grouped together
  - 'dissimilar' observations are in different groups
- depends on definition of 'similarity'/ distance



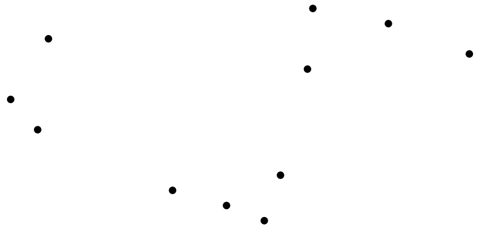
clustering
introduction

- clustering is usually unsupervised and exploratory
- group observations
  - 'similar' observations are grouped together
  - 'dissimilar' observations are in different groups
- depends on definition of 'similarity'/ distance



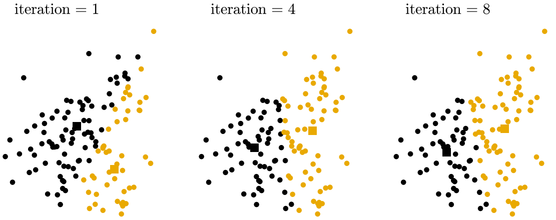
clustering
kMeans clustering

1. *Initialization*: randomly select K observations from the data set as initialization.
2. *Update*: compute the mean for each cluster.
3. *Assignment*: assign each observation to the cluster with the mean of the closest cluster.
4. *Iteration*: go to step 2 until the clusters converge.



matlab source: mat

clustering
kMeans clustering



matlab source: plotKmeans.m

distances
overview

- *Euclidean Distance* (L2 Distance)
- *Manhattan Distance* (L1 Distance)
- *Cosine Similarity*
  - range is from $[-1; 1]$ ($[0; 1]$ for non-negative input),
  - not distance but similarity measure
  - independent of vector length, only on angle
- *Kullback-Leibler Divergence*
  - not symmetric: $d_{\mathrm{KL}}(\boldsymbol{v}_\mathrm{a}, \boldsymbol{v}_\mathrm{b}) \neq d_{\mathrm{KL}}(\boldsymbol{v}_\mathrm{b}, \boldsymbol{v}_\mathrm{a})$,
  - designed to measure distance between probability distributions

$$d_{\mathrm{EU}}(\boldsymbol{v}_\mathrm{a}, \boldsymbol{v}_\mathrm{b}) = \|\boldsymbol{v}_\mathrm{a} - \boldsymbol{v}_\mathrm{b}\|_2 = \sqrt{\sum_{j=0}^{\mathcal{J}-1} \big(v_\mathrm{a}(j) - v_\mathrm{b}(j)\big)^2}.$$

distances
overview

- *Euclidean Distance* (L2 Distance)
- *Manhattan Distance* (L1 Distance)
- *Cosine Similarity*
    - range is from $[-1;1]$ ($[0;1]$ for non-negative input),
    - not distance but similarity measure
    - independent of vector length, only on angle
- *Kullback-Leibler Divergence*
    - not symmetric: $d_\mathrm{KL}(\boldsymbol{v}_\mathrm{a}, \boldsymbol{v}_\mathrm{b}) \neq d_\mathrm{KL}(\boldsymbol{v}_\mathrm{b}, \boldsymbol{v}_\mathrm{a})$,
    - designed to measure distance between probability distributions

$$d_\mathrm{M}(\boldsymbol{v}_\mathrm{a}, \boldsymbol{v}_\mathrm{b}) = \|\boldsymbol{v}_\mathrm{a} - \boldsymbol{v}_\mathrm{b}\|_1 = \sum_{j=0}^{\mathcal{J}-1} \left|v_\mathrm{a}(j) - v_\mathrm{b}(j)\right|.$$

- *Euclidean Distance* (L2 Distance)
- *Manhattan Distance* (L1 Distance)
- *Cosine Similarity*
  - range is from $[-1; 1]$ ($[0; 1]$ for non-negative input),
  - not distance but similarity measure
  - independent of vector length, only on angle
- *Kullback-Leibler Divergence*
  - not symmetric: $d_{\mathrm{KL}}(\boldsymbol{v}_{\mathrm{a}}, \boldsymbol{v}_{\mathrm{b}}) \neq d_{\mathrm{KL}}(\boldsymbol{v}_{\mathrm{b}}, \boldsymbol{v}_{\mathrm{a}})$,
  - designed to measure distance between probability distributions

$$s_{\mathrm{C}}(\boldsymbol{v}_{\mathrm{a}}, \boldsymbol{v}_{\mathrm{b}}) = \frac{\sum\limits_{j=0}^{\mathcal{J}-1} v_{\mathrm{a}}(j) \cdot v_{\mathrm{b}}(j)}{\sqrt{\sum\limits_{j=0}^{\mathcal{J}-1} v_{\mathrm{a}}(j)^2} \cdot \sqrt{\sum\limits_{j=0}^{\mathcal{J}-1} v_{\mathrm{b}}(j)^2}}.$$

$d_{\mathrm{C}}(\boldsymbol{v}_{\mathrm{a}}, \boldsymbol{v}_{\mathrm{b}}) = 1 - s_{\mathrm{C}}(\boldsymbol{v}_{\mathrm{a}}, \boldsymbol{v}_{\mathrm{b}})$, and

- *Euclidean Distance* (L2 Distance)
- *Manhattan Distance* (L1 Distance)
- *Cosine Similarity*
  - range is from $[-1; 1]$ ($[0; 1]$ for non-negative input),
  - not distance but similarity measure
  - independent of vector length, only on angle
- *Kullback-Leibler Divergence*
  - not symmetric: $d_{\mathrm{KL}}(\boldsymbol{v}_{\mathrm{a}}, \boldsymbol{v}_{\mathrm{b}}) \neq d_{\mathrm{KL}}(\boldsymbol{v}_{\mathrm{b}}, \boldsymbol{v}_{\mathrm{a}})$,
  - designed to measure distance between probability distributions

$$d_{\mathrm{KL}}(\boldsymbol{v}_{\mathrm{a}}, \boldsymbol{v}_{\mathrm{b}}) = \sum_{j=0}^{\mathcal{J}-1} v_{\mathrm{a}}(j) \cdot \log\left(\frac{v_{\mathrm{a}}(j)}{v_{\mathrm{b}}(j)}\right).$$

summary
lecture content

- **regression**
  - model to estimate numeric labels from features
  - linear regression assumes model is straight line
- **clustering**
  1. unsupervised grouping
  2. feature space and distance measure determine result
  3. number of clusters usually has to be known
  4. kMeans is simple way of clustering