Unsupervised Learning

GeoComput & ML

2021-05-11 Tue

Overview

Supervised

Given response variable $\boldsymbol{Y}$ and predictor variables $\boldsymbol{X}^T = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p)$, supervised learning can be formulated as a density estimation problem with the interest of determining the properties of the conditional probability $P(\boldsymbol{Y}|\boldsymbol{X})$, commonly such as the location parameter μ.

$$\mu(\boldsymbol{x}) = \arg\min_{\boldsymbol{\theta}} E_{\boldsymbol{Y}|\boldsymbol{X}} L(\boldsymbol{Y}, \boldsymbol{\theta})$$

Unsupervised

inferring the properties of $P(\boldsymbol{X})$

Sammon Mapping

Aim

mapping a high dimensional space to a space of lower dimensionality
Let $\boldsymbol{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be a set of vectors and in a d space, we seek a new set of representation $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$ being the corresponding vectors in the reduced d^* space with the intention of preserving the distance structure in $\boldsymbol{D}$.

Optimisation

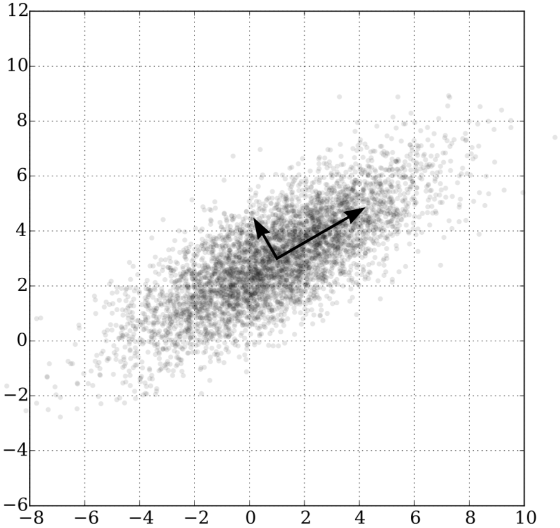
$$E^{(t)} = \frac{1}{\sum_i d_{is}} \sum_i \frac{(d_{is} - (d^*_{is})^{(t)})^2}{d_{is}}$$

where $(d^*_{is})^{(t)} = \sqrt{\sum_{j=1}^{d^*} \left(y_{ij}^{(t)} - y_{sj}^{(t)}\right)^2}$ and $d_{is} = \sqrt{\sum_{j=1}^{d} (x_{ij} - x_{sj})^2}$

new configuration

$$\boldsymbol{y}^{(t+1)} = \boldsymbol{y}^{(t)} - \alpha \nabla^{(t)}$$

PCA



PCA

Statistical View

Let multivariate random variable $\boldsymbol{x} \in \mathbb{R}^D$. We seek a set of orthonormal basis as the linear components of $\boldsymbol{x}$.

$$y_i = \boldsymbol{u}_i^T \boldsymbol{x}$$

such that the variance y_i is maximised subject to

$$\boldsymbol{u}_i^T \boldsymbol{u}_i = 1$$

Theorem

Assume rank $\Sigma_{\boldsymbol{x}} \doteq E(\boldsymbol{x}\boldsymbol{x}^T) \geq d$, then the first d principle components of a zero-mean multivariate random variable $\boldsymbol{x}$ denoted by y_i are given by

$$y_i = \boldsymbol{u}_i^T \boldsymbol{x}$$

where $\{\boldsymbol{u}_{i=1}^d\}$ are d orthonormal eigenvectors of $\Sigma_{\boldsymbol{x}}$ associated with its d largest eigenvalues $\lambda_i = Var(y_i)$.

PCA

$\forall \boldsymbol{u} \in \mathbb{R}^D, \quad Var(\boldsymbol{u}^T \boldsymbol{x}) = E[(\boldsymbol{u}^T \boldsymbol{x})^2] = \boldsymbol{u}^T \Sigma_{\boldsymbol{x}} \boldsymbol{u}$

1st principle component

$$\max_{\boldsymbol{u_1} \in \mathbb{R}^D} \boldsymbol{u}_1^T \Sigma_{\boldsymbol{x}} \boldsymbol{u}_1 \quad \text{s.t. } \boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$$

$$\mathcal{L} = \boldsymbol{u}_1^T \Sigma_{\boldsymbol{x}} \boldsymbol{u}_1 + \lambda_1 (1 - \boldsymbol{u}_1^T \boldsymbol{u}_1)$$

$$\Sigma_{\boldsymbol{x}} \boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1, \quad \boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$$

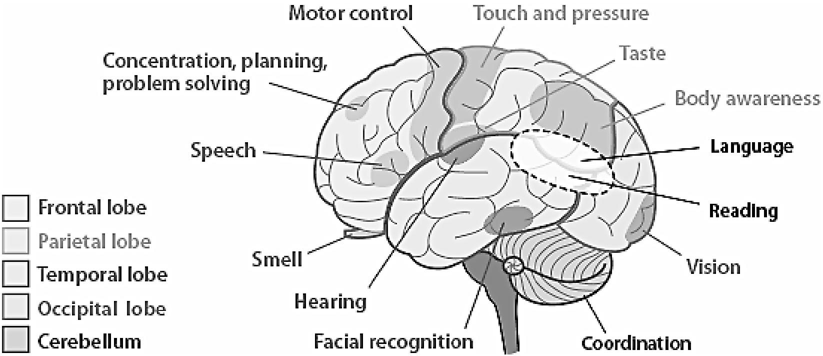
2nd principle component

$$\max_{\boldsymbol{u_2} \in \mathbb{R}^D} \boldsymbol{u}_2^T \Sigma_{\boldsymbol{x}} \boldsymbol{u}_2 \quad \text{s.t. } \boldsymbol{u}_2^T \boldsymbol{u}_2 = 1, \ \ \boldsymbol{u}_1^T \boldsymbol{u}_2 = 0$$

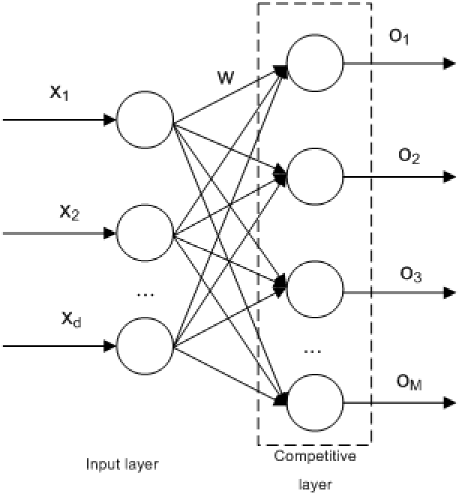
$$\mathcal{L} = \boldsymbol{u}_2^T \Sigma_{\boldsymbol{x}} \boldsymbol{u}_2 + \lambda_2 (1 - \boldsymbol{u}_2^T \boldsymbol{u}_2) + \gamma \boldsymbol{u}_1^T \boldsymbol{u}_2$$

$$\Sigma_{\boldsymbol{x}} \boldsymbol{u}_2 + \gamma \boldsymbol{u}_1 / 2 = \lambda_2 \boldsymbol{u}_2, \quad \boldsymbol{u}_2^T \boldsymbol{u}_2 = 1, \quad \boldsymbol{u}_1^T \boldsymbol{u}_2 = 0$$

SOM


Motor control
Touch and pressure
Concentration, planning, problem solving
Taste
Body awareness
Speech
Language
Reading
Frontal lobe
Parietal lobe
Temporal lobe
Occipital lobe
Cerebellum
Smell
Vision
Hearing
Facial recognition
Coordination

SOM



$$y_i = \sum_{i=1}^{d} w_{ji}\, x_i$$

SOM

Process

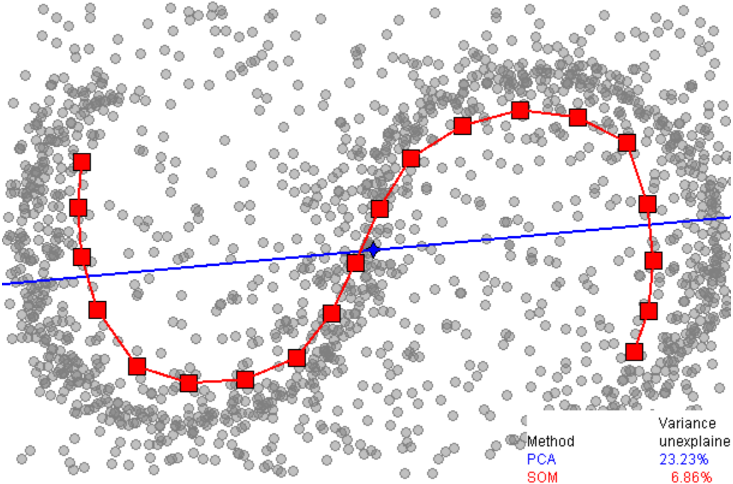
- competition
  input data : $\boldsymbol{x} = (x_1, \ldots, x_m)^T$
  synaptic weight : $\boldsymbol{w} = (w_{j1}, \ldots, w_{jm})^T$
  winning neuron : $\boldsymbol{w}_j^T \boldsymbol{x} \Rightarrow i(\boldsymbol{x}) = arg\, \underset{j}{min} \|\boldsymbol{x} - \boldsymbol{w}_j\|$
- cooperation
  $h_{j,i}$: top neighborhood centred on the winning neuron i
  $$h_{j,i}(s) = \exp\left(-d_{j,i}^2 \middle/ (2\sigma_s^2)\right)$$
  $\sigma_s = \sigma_0 \exp\left(-s/\tau_1\right)$
- adaptation
  $\boldsymbol{w}_j^{s+1} = \boldsymbol{w}_j^s + \eta_s h_{j,i}(s)(\boldsymbol{x} - \boldsymbol{w}_j^w)$
  $\eta_s = \eta_0 \exp\left(-s/\tau_2\right)$

SOM

- Algorithm

```
Initialiation : weight vector
Repeat
  Sampling
  Matching
  Updating
Until convergence
```

SOM vs PCA



k-mean Clustering

Algorithm

```
initialise with K centroids
repeat
  form K clusters using the centroids
  update the centroids using SSE cost
until convergence
```

Minimisation

Given a dataset $D = \{x_1, x_2 \dots x_n\}$, let us denote the clusters as $C = \{c_1, c_2, \dots c_k\}$.

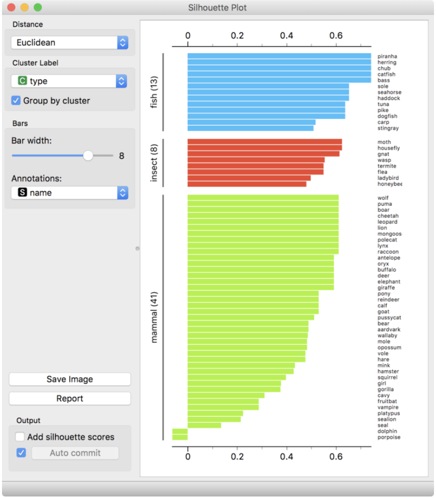
$$\begin{cases} SSE(C) = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - c_k\|^2 \\ c_k = \sum_{x_i \in C_k} x_i \Big/ |C_k| \end{cases}$$

Silhouette Coefficient

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j)$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



k-Mediods

cost function S : absolute error

Algorithm

```
Initialisation :
Repeat
  cluster formation
  randomly select non-representative x(i)
  evalution cost of swapping x(i) and representative object m
  if S < 0, swap and update
Until convergence
```

k-Median

$$S = \sum_{k=1}^{K} \sum_{x_i \in C_k} |x_{ij} - {}_{kj}\ |$$

where med_{kj} represents the median of the j th attribute in the k th cluster.

IK-Mean

idea : farther points to the centroid, more interesting

Algorithm

```
calculate the centre of gravity of the dataset Cg
Repeat
  create a centroid c farthest from Cg
  create a cluster Si around c if d(x,c) < d(x,Cg)
  update Sg = Si
  set Cg = Sg
  discard small clusters below a threshold
Until convergence
```

Hierarchical clustering

- more deterministic

Hierarchical clustering

- more deterministic
- agglomerative and divisive

Hierarchical clustering

- more deterministic
- agglomerative and divisive
- binary tree, dendrogram

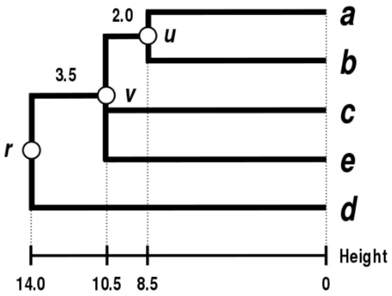
Hierarchical clustering

Names	Formula
Euclidean distance	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Squared Euclidean distance	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan distance	$\|a-b\|_1 = \sum_i \|a_i - b_i\|$
Maximum distance	$\|a-b\|_\infty = \max_i \|a_i - b_i\|$
Mahalanobis distance	$\sqrt{(a-b)^\top S^{-1}(a-b)}$ where S is the Covariance matrix

Names	Formula
Maximum or complete-linkage clustering	$\max\{\, d(a,b) : a \in A,\, b \in B \,\}.$
Minimum or single-linkage clustering	$\min\{\, d(a,b) : a \in A,\, b \in B \,\}.$
Unweighted average linkage clustering (or UPGMA)	$\frac{1}{\|A\| \cdot \|B\|} \sum_{a \in A} \sum_{b \in B} d(a,b).$
Weighted average linkage clustering (or WPGMA)	$d(i \cup j, k) = \frac{d(i,k) + d(j,k)}{2}.$
Centroid linkage clustering, or UPGMC	$\|c_s - c_t\|$ where c_s and c_t are the centroids of clusters s and t, respectively.
Minimum energy clustering	$\frac{2}{nm} \sum_{i,j=1}^{n,m} \|a_i - b_j\|_2 - \frac{1}{n^2} \sum_{i,j=1}^{n} \|a_i - a_j\|_2 - \frac{1}{m^2} \sum_{i,j=1}^{m} \|b_i - b_j\|_2$

Single-link

	a	b	c	d	e
a	0	17	21	31	23
b	17	0	30	34	21
c	21	30	0	28	39
d	31	34	28	0	43
e	23	21	39	43	0



References

- JW. Sammon. IEEE Trans. Computers. (1969) 18, 401
- T. Kohenen. Proc. IEEE (1990) 78, 1464
- T. Kohenen. Self Organisation and Associated Memory (1989)
- M. Heath. Scientific Computing An Introductory Survey (2018)
- T. Hastie *et. al.* The Elements of Statistical Learning (2017)
- G. Gan *et. al.* Data Clustering. Theory, Algorithms and Applications (2007)
- C. Aggarwal *et. al.* Data Clustering : Algorithms and Applications (2014)
- R. Vidal *et. al.* Generalised Principle Components Analysis (2016)
- S. Haykin. Neural networks and Machine Learnings (2008)
- https://en.wikipedia.org/wiki/Self-organizing_map
- https://en.wikipedia.org/wiki/Competitive_learning
- https://en.wikipedia.org/wiki/Silhouette_(clustering)
- https://en.wikipedia.org/wiki/Principal_component_analysis
- https://cybernetist.com/2017/01/13/self-organizing-maps-in-go/
- https://en.wikipedia.org/wiki/Hierarchical_clustering