# Analysis and Design of Deep Neural Networks

## Chapter 2

## Complexity Indices and Data analysis

Fall 2023

# 2 Complexity Indices and Data analysis

## 2.1. Complexity Indices

2.1.1 Separation index and methods

2.1.2 Smoothness index and methods

2.1.3 Linear Density Index index and methods

## 1.2. Data Analysis

2.2.1 Dataset evaluation and Scoring

2.2.2 Supervised Feature Selection

2.2.4 Data Connectivity Matrix (Smi Table)

2.2.5 Data Clustering

2.2.3 Unsupervised Feature Selection

# 2.1 Complexity Indices

Supervised Indices: 1- Separation Index (Classification Prob.), 2-Smoothness Index(Regression Prob.)

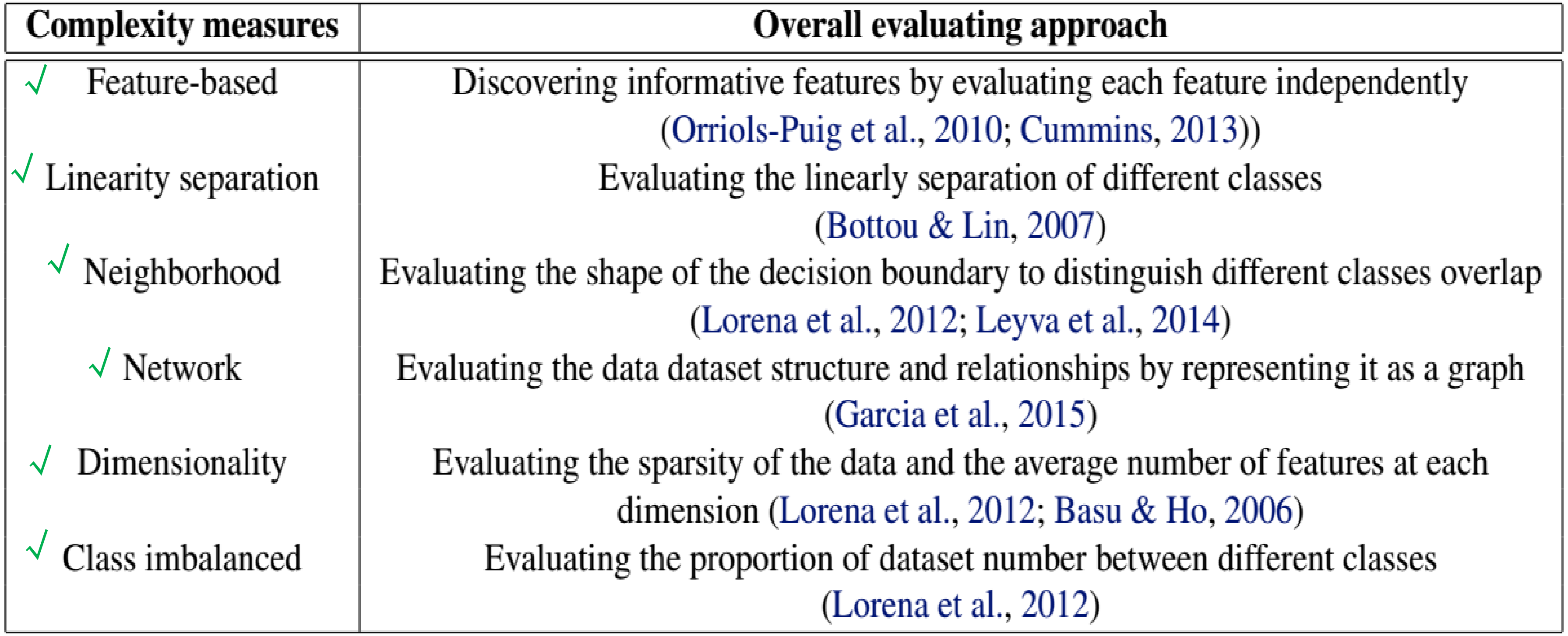
### 2.1.1 Separation index(SI)

- First order SI
- High order SI
- High order soft SI
- Center Based SI
- Cross SI
- Anti SI
- Self Supervised SI

### 2.1.2 Smoothness index(SmI)

- First order SmI
- High order SmI
- High order soft SmI
- Cross SmI
- Global SmI
- Data Connectivity SmI

Complexity measures



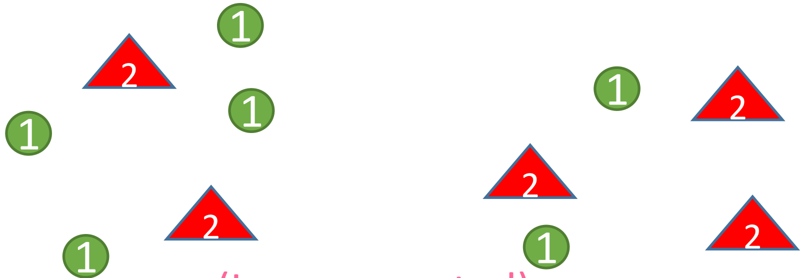
| Complexity measures | Overall evaluating approach |
|---|---|
| ✓ Feature-based | Discovering informative features by evaluating each feature independently (Orriols-Puig et al., 2010; Cummins, 2013)) |
| ✓ Linearity separation | Evaluating the linearly separation of different classes (Bottou & Lin, 2007) |
| ✓ Neighborhood | Evaluating the shape of the decision boundary to distinguish different classes overlap (Lorena et al., 2012; Leyva et al., 2014) |
| ✓ Network | Evaluating the data dataset structure and relationships by representing it as a graph (Garcia et al., 2015) |
| ✓ Dimensionality | Evaluating the sparsity of the data and the average number of features at each dimension (Lorena et al., 2012; Basu & Ho, 2006) |
| ✓ Class imbalanced | Evaluating the proportion of dataset number between different classes (Lorena et al., 2012) |

*Table 1.* Some complexity measures and their evaluating approaches in a classification problem

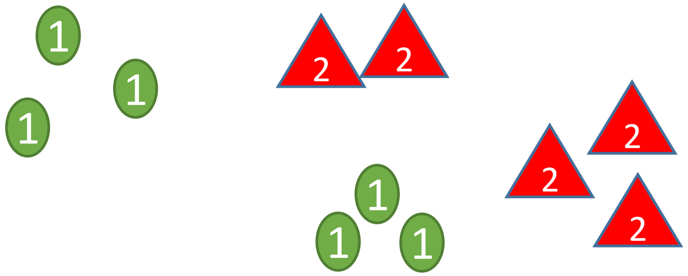
# Two Supervised Complexity measures

1. A separation measure (in classification problems)

   It shows that how much input data points separate the labels from each others.



(Less separated)

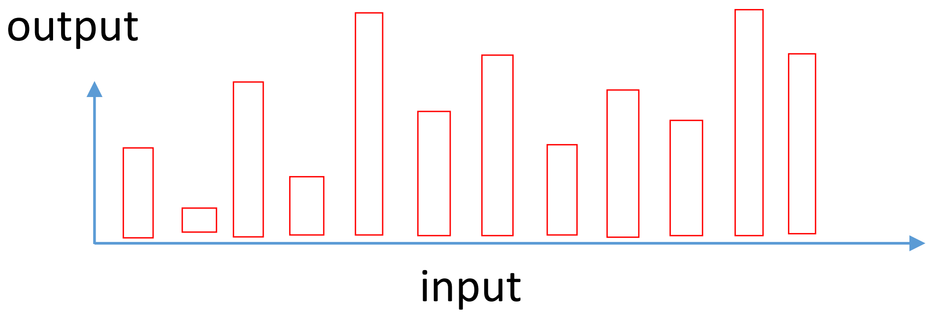


(More separated)

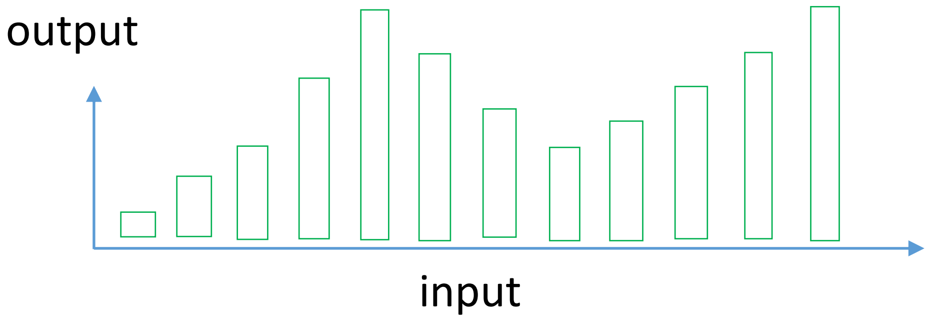
2. An smoothness measure (in regression problems)

   It shows that how much input data points make the output targets smooth

(Less Smooth)



(More Smooth)



# Separation index (SI)

“SI” measures that how much input data points(the feature space) separate different class labels from each others.

“SI” is a variant of similarity measure between feature space(input distribution) and the label space(output distribution)

# 2.1. Separation index (SI)

## 1. First order SI

$Data = \{(\boldsymbol{x}_i, l_i)\}_{i=1}^{m} \forall i: \boldsymbol{x}^i \epsilon R^{n\times 1}$ $l_i \epsilon \{1,2,\ldots,n_C\}$ $n_C$:number of classes

*it is assumed that “Data” is a measured sample from a domain with high enough diversity.

*$\boldsymbol{x}_i$may have any format (video, image, time series, etc.) ; however, to compute SI, it must be reshaped as a vector.

$$\mathrm{SI}(Data) = \frac{1}{m}\sum_{i=1}^{m} \delta(l_i, l_{i^*})$$

$$i^* = \underset{\forall q \neq i}{\arg min} \left\| \boldsymbol{x}_i - \boldsymbol{x}_q \right\| \qquad \delta(l_i, l_{i^*}) = \begin{cases} 1 & if\ l_i = l_{i^*} \\ 0 & else \end{cases}$$

kronecker delta

*$\|\cdot\|$ denotes Euclidian distance ($L_2$ norm) but it may be another distance definition such as $L_p$ norm:

$$\left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|_{L_p} = \sqrt[p]{\sum_{k=1}^{n} \left| \boldsymbol{x}_i(k) - \boldsymbol{x}_j(k) \right|^p}$$

** It is assumed that the input data is normalized at each dimension just before computing separation index.

# Some notes

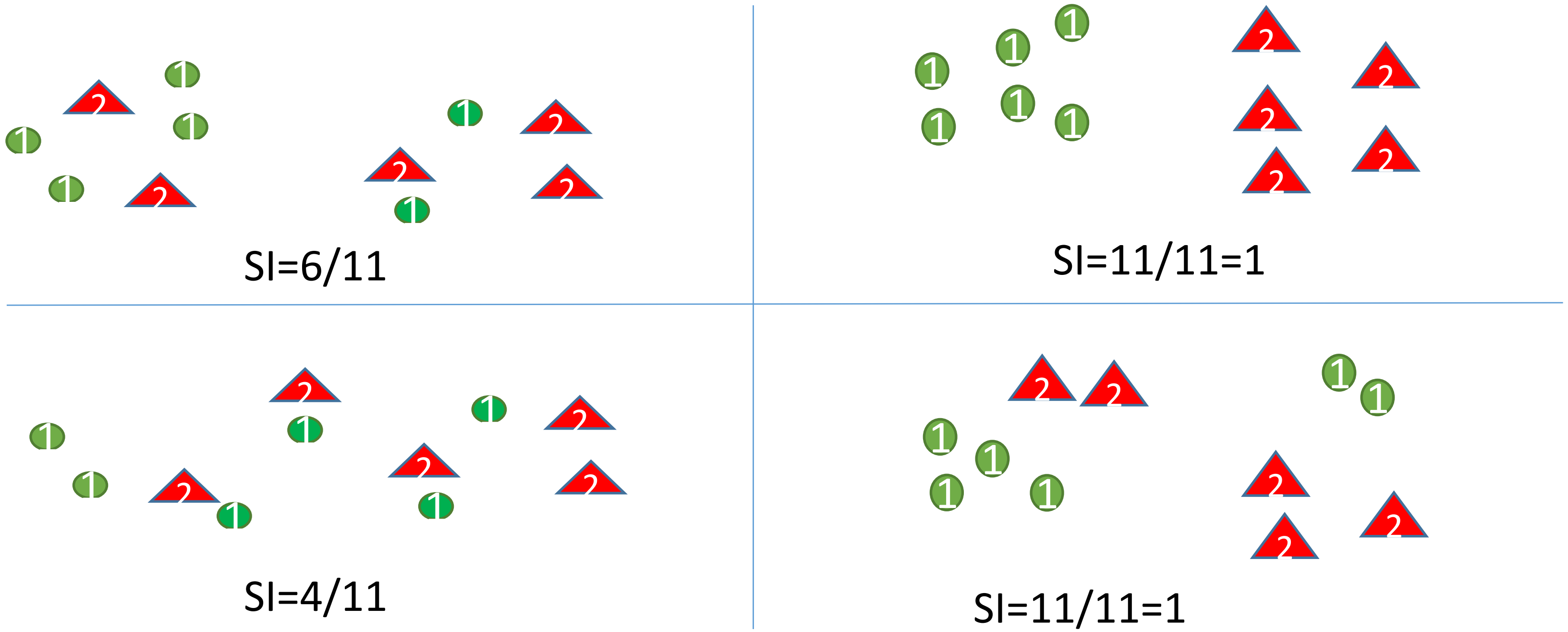
1. “SI” is a normalized index between zero and one: $SI \in [0,1]$
2. $SI \to 1$ (*Sepration is maximmum*) and $SI \to 0$(*Sepration is minimmum*)
3. “SI” counts (average of) all data points whose nearest neighbors have the same label
4. “SI” is equal to the accuracy of the nearest neighbor classifier as a non-parametric model. Hence, SI is an informative index having strong correlation with the best accuracy one can access by a model without filter process.
5. SI does not change against shift and scales of data points.

   $\forall \beta \neq 0, \forall \alpha \neq 0, \forall x_0, \forall l_0 \qquad SI\left(\left\{\left(\boldsymbol{x}^i, l^i\right)\right\}_{i=1}^m\right) = SI\left(\left\{\left(\beta x_i + \boldsymbol{x}_0, \alpha l_i + l_0\right)\right\}_{i=1}^m\right)$

6. Separatin index of *the target labels with themselves is maximum*: $SI(\{(l_i, l_i)\}_{i=1}^m)$=1;

it means that how input data become more similar to labels the separation index will increase.

# Two dimensional examples (binary classification)



SI=6/11

SI=11/11=1

SI=4/11

SI=11/11=1

Some notes

- To have a high SI, It is enough that examples of each class become near and near together in some regions
- The number of regions is not important but each region must have at least two members.
- The shape of each region is not important.

# The distance matrix

- To achieve SI, matrix distance of all data points must be computed (to get nearest neighbor for each data point)

$$Data = \{(\boldsymbol{x}_i, l_i)\}_{i=1}^{m} \qquad \boldsymbol{x}^i \epsilon R^{n\times 1}$$

Distance matrix: $D = [d_{ij}] \quad d_{ij} = \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2$

Steps

1- Provide data Matrix:$\mathrm{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]^T, \quad \mathrm{X}\epsilon R^{m\times n}$

2- M= $\mathrm{XX^T}, \quad \mathrm{M}\epsilon R^{m\times m}$

3- $\mathrm{d} = \mathrm{diag(M)}, \quad \mathrm{d}\epsilon R^{m\times 1}$

4- W=[d,d,…,d], $\mathrm{W}\epsilon R^{m\times m}$

5- Distance matrix is computed as follows:

$$\mathrm{D} = \mathrm{W} + \mathrm{W^T} - 2\mathrm{M}$$

# Separation index of Each data point

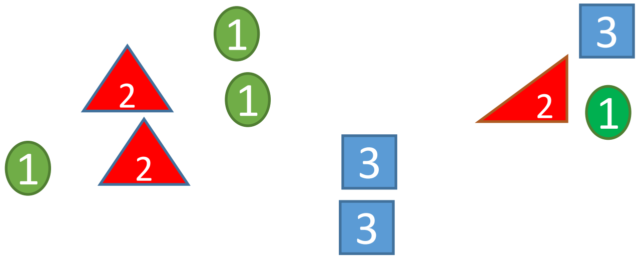
* $SI(Data) = \frac{1}{m}\sum_i si(x_i, l_i)$ ,$si(x_i, l_i) = \delta(l_i, l_{i^*})$

SI definition with data distribution: $SI(Data) = Exp_{p(x,y)}(si(x, l))$

Challenge: to compute $si(x_i, l_i)$ it is required to have $x^*$ as the nearest neighbor of $x$.

❖For a sample of data with high enough diversity SI can be approximated by equation *

A two dimensional illustrative example



i=1,2,….,10

$si(x_i, l_i) = 0, \quad i$=1,8,9,10

$si(x_i, l_i) = 1, \quad i = 2,3,4,5,6,7$

$SI(Data)$ =0.6

# Separation index of Each Class

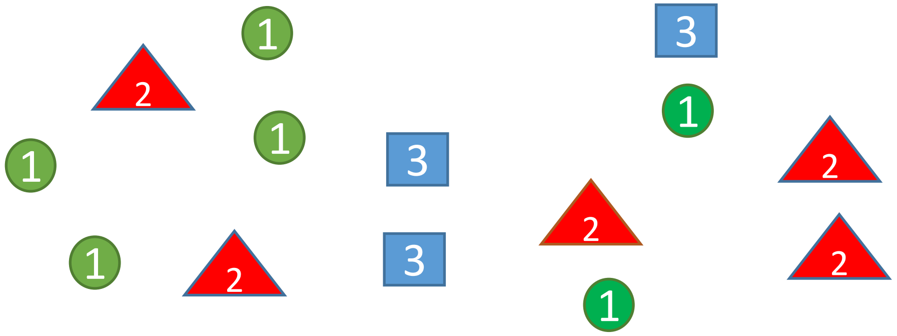
$\mathrm{SI}_c^{class}(Data) = \frac{1}{m_c}\sum_i \delta(l_i, c)\delta(l_i, l_{i^*})$ $\quad c$=1,2,..,$n_C$

$m_c$=$\sum_i \delta(l_i, c)$ $\quad m_c$: number of all data points $x^i$ which $l^i = c$

*Relation between "total SI" and "SI of classes"*

$$SI(Data) = \frac{1}{m}\sum_{c=1}^{n_C} m_c SI_c^{class}(Data) \qquad \sum_{c=1}^{n_C} m_c = m$$

A two dimensional illustrative example



$n_C = 3, \qquad c = 1,2,3$

$SI_1^{class}(Data) = \frac{4}{6}$

$SI_2^{class}(Data) = 2/5$

$SI_3^{class}(Data) = 2/3$

SI=(4+2+2)/(6+5+3)=8/14

* For when for each class c: $m_c = \frac{m}{n_C}$ and a sufficient high number of data points are distributed with a *uniformly distributed random* variable then it is expected that $SI \rightarrow 1/n_C$

## 2. High order SI

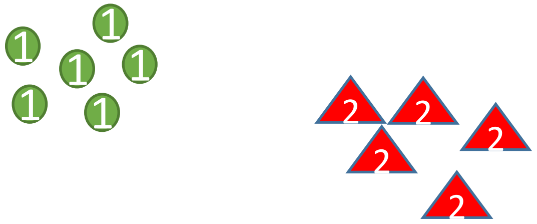
$Data = \{(\boldsymbol{x}_i, l_i)\}_{i=1}^{m}$ $\forall i$: $\boldsymbol{x}_i \epsilon R^{n\times 1}$ $l_i \epsilon \{1,2,\ldots,n_C\}$ $n_C$:number of classes

$$\mathrm{SI}^{\mathrm{r}}(Data) = \frac{1}{m}\sum_{i=1}^{m}\prod_{j=1}^{r}\delta\left(l_i, l_{i_j^*}\right)$$ r: the order of "SI"

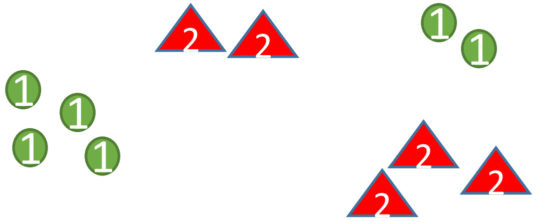
$$i_j^* = \underset{\forall q \neq i, i_1^*, \cdots, i_{j-1}^*}{\arg} \quad min\left\|\boldsymbol{x}_i - \boldsymbol{x}_q\right\| \qquad \mathrm{SI}^{\mathrm{r}} \in [0,1]$$

- “$\mathrm{SI}^{\mathrm{r}}$” counts (average of) all data points whose all “r” nearest neighbors have the same label
- $\mathrm{SI}^{\mathrm{r}}$ considers more restricted condition of separation than $\mathrm{SI}^{\mathrm{j}}$ $(j < r)$.
- For each “Data” we have: $\mathrm{SI}^{\mathrm{r}} \le \mathrm{SI}^{\mathrm{r}-1} \le \cdots \le \mathrm{SI}^{1}$ $\qquad \mathrm{SI}^{1} = \mathrm{SI}$

# Two illustrative Examples



$SI^1$ =11/11
$SI^2$ =11/11
$SI^3$ =11/11
$SI^4$ =11/11



$SI^1$ =11/11
$SI^2$ =7/11
$SI^3$ =4/11
$SI^4$ =0

Some notes

1. To increase high order SI, different regions of data points with the same label should merge together and make a hyper-circle shape distribution. In a such case, we will have $n_C$ hyper-circle shape which can separated, linearly from each other.
2. If in a classification problem, the high order SI $\mathrm{SI}^{\mathrm{r}}(r \to \infty) \to 1$ , the data points of any pair of classes become more linearly separable.
3. If in a classification problem, the high order SI $\mathrm{SI}^{\mathrm{r}}(r \to \infty) \to 1$ , then there is a global separation index (gsi).

# 3. High order soft SI

$Data = \{(\boldsymbol{x}_i, l_i)\}_{i=1}^{m} \forall i: \boldsymbol{x}_i \epsilon R^{n\times 1}$ $l_i \epsilon \{1,2, \dots, n_C\}$ $n_C$:number of classes

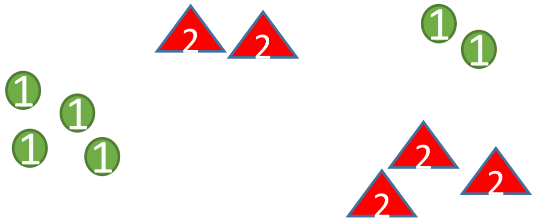
$$\mathrm{SI}_{\mathrm{soft}}^{\mathrm{r}}(Data) = \frac{1}{m\times r} \sum_{i=1}^{m} \sum_{j=1}^{r} \delta(l_i,, l_{i_j^*})$$

r: the order of SI $i_j^* = \underset{\forall q \neq i, i_1^*, \cdots, i_{j-1}^*}{\arg} \quad min\|\boldsymbol{x}_i - \boldsymbol{x}_q\|$ $\mathrm{SI}_{\mathrm{soft}}^{\mathrm{r}} \in [0,1]$

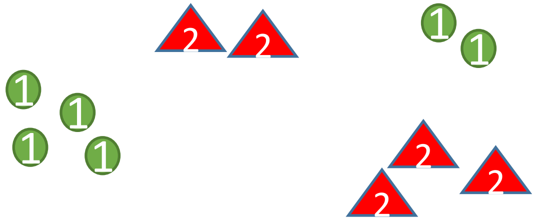
- $\mathrm{SI}_{\mathrm{soft}}^{\mathrm{r}}$ considers less restricted condition of separation than $\mathrm{SI}^{\mathrm{r}}$

$$\mathrm{SI}_{\mathrm{soft}}^{\mathrm{r}} \geq \mathrm{SI}^{\mathrm{r}} \quad \text{and} \quad \mathrm{SI}_{\mathrm{soft}}^{1} = \mathrm{SI}^{1}$$

# Two illustrative Examples



$SI^1$ =11/11
$SI^2$ =7/11
$SI^3$ =4/11
$SI^4$ =0



$SI_{soft}^1$ =11/11
$SI_{soft}^2$=(4+3+0.5+0.5)/11=8/11
$SI_{soft}^3$=(4+3(2/3)+4*(1/3)/11=8.33/11
$SI_{soft}^4$=(4*(3/4)+2*(1/4)+2*(1/4)+3*(2/4))/11
=6.5/11

## 4. Center based Separation Index (CSI)

$Data = \{(\boldsymbol{x}_i, l_i)\}_{i=1}^{m} \forall i: \boldsymbol{x}^i \epsilon R^{n\times 1}$ $\quad l_i \epsilon \{1,2,\ldots,n_C\}$ $\quad n_C$:number of classes

Center of each class is the mean of all input data points having the label of that class:

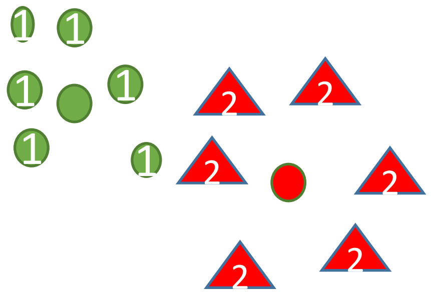
$$\boldsymbol{\mu}_c = \frac{1}{m_c}\sum_{i=1}^{m} \boldsymbol{x}_i \delta(\,l_i, c),\quad c = 1,2,..,n_C \qquad m_c{=}\sum_{i=1}^{m} \delta(\,l_i, c)$$

$$\mathrm{CSI}(Data) = \frac{1}{m}\sum_{i=1}^{m} \delta(l_i, c^*)$$

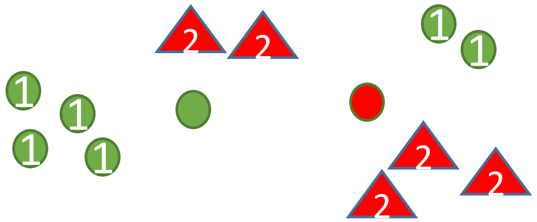
$$c^* = \underset{\forall c}{\arg min}\|\boldsymbol{x}_i - \boldsymbol{\mu}_c\|$$

- CSI is computed much faster than SI because $n_C \ll m$ and you only need to compute the distance matrix of input data points to center of classes.
- It is suggested to compute CSI instead of SI in cases where examples of each class has an indpendent unimodal distribution over a focal point.
- In such cases, each example of a class has most of the exclusive features of that class and has less common features with examples of other classes.

# An illustrative Examples



$SI$ =9/11
CSI=11/11



$SI = 1$
CSI=7/11

# 5. **Self** supervised SI (SSSI)

- $Data = \{(\boldsymbol{x}_i, ?)\}_{i=1}^{m} \forall i: \boldsymbol{x}_i \epsilon R^{n\times 1}$ *Labels are unknown*
- For each $\boldsymbol{x}_i$ we generate some augmented data points: $x_{i_h}, h \in \{1,2,..,n_i\}$
- It is assumed that each $x_{i_h}$ inherits at least an exclusive feature of $\boldsymbol{x}_i$
- An exclusive feature of $x_{i_h}$ is a feature that is sufficient to reveal the label of $\boldsymbol{x}_i$.

$$Data_{\text{aug}} = \{\overbrace{\{(\boldsymbol{x}_{i_h}, \boldsymbol{i})\}_{h=1}^{n_i}}^{i\text{th (self) class}}\}_{i=1}^{m}$$

$$\text{SSSI}^{\text{r}}(Data) = \text{SI}^{\text{r}}\left(Data_{\text{aug}}\right),\ n_C = m,\ m_{aug} = \sum_{i=1}^{m} n_i$$

# Cross SI

$Data = \{(\boldsymbol{x}_i, l_i)\}_{i=1}^{m}$ $\quad D_{test} = \{(\breve{\boldsymbol{x}}_i, \check{l}_i)\}_{i=1}^{m_{test}}$

$$SI_{cross}(D_{test}, Data) = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} \delta(\check{l}_i, l_{i^{\#}})$$

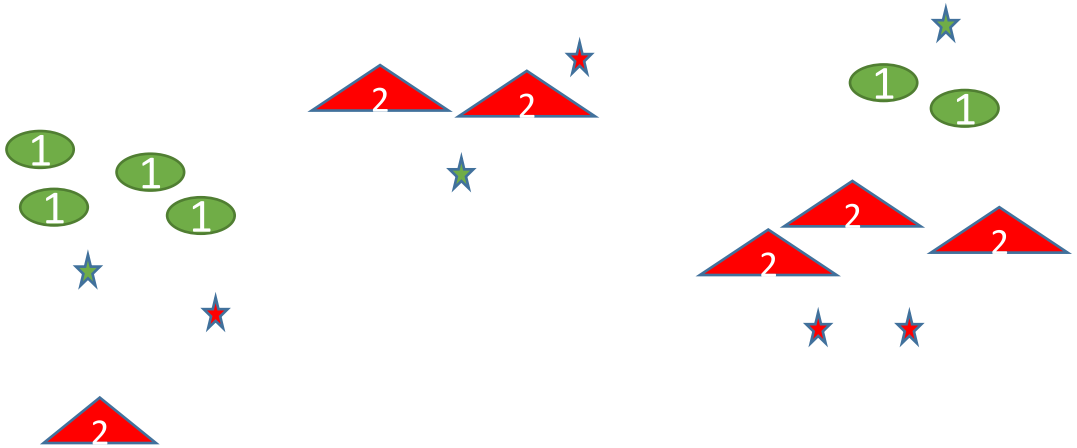
$i^{\#} = \underset{\forall q}{\arg min} \|\breve{\boldsymbol{x}}_i - \boldsymbol{x}_q\|$

Cross SI measures the separation index of a test domain of dataset $D_{test}$ based on the main domain of dataset $Data$.

It can be shown that:

$$SI_{cross}(D_{test}, Data) = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} SI_{cross}\left((\breve{\boldsymbol{x}}_i, \check{l}_i), Data\right)$$

# An illustrative Examples



$$SI = \frac{12}{13} \qquad Corss\ SI = \frac{5}{7}$$

# Anti SI

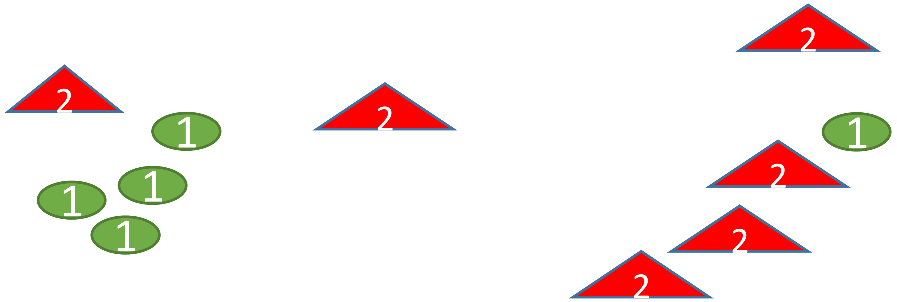
$Data = \{(\boldsymbol{x}_i, l_i)\}_{i=1}^{m}$ $\forall i$: $\boldsymbol{x}_i \epsilon R^{n\times1}$ $l_i \epsilon \{1,2,\dots,n_C\}$ $n_C$:number of classes

$$\mathrm{anti\_SI^r}(Data) = \frac{1}{m}\sum_{i=1}^{m}\prod_{j=1}^{r}(1-\delta\left(l_i, l_{i_j^*}\right))$$ r: the order of "anti_SI"

$$i_j^* = \underset{\forall q \neq i, i_1^*, \cdots, i_{j-1}^*}{\arg} min\left\|\boldsymbol{x}_i - \boldsymbol{x}_q\right\|$$ $\mathrm{anti\_SI^r} \in [0,1]$

- "anti SI$^{\mathrm{r}}$" counts (average of) all data points whose all "r" nearest neighbors have different labels with those data points
- Actually data points having higher anti si make hard examples in  a data set
- They may be risky examples that experts have labeled them incorrectly. In such a case they should be removed in a "data cleaning process"
- For when data points are images and other spatial or temporal formats, before to score them by "SI" or "anti SI", one must encode them.

# An illustrative Examples



$SI^1$=7/11
$anti\_SI^1 = 1 - SI^1$=4/11

$SI^2$=5/11
$anti\_SI^2 = 2/11$

# Smoothens index (SmI)

SmI  measures how much input data points make the output targets smooth

“SmI” is a variant of similarity measure between feature space(input distribution) and the target space(output distribution)

# 2.2 Smoothness index (SmI)

A (linear) smoothness measure for regression problem

1. First order SmI

$Data = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{m}$ $\forall i: \boldsymbol{x}_i \epsilon R^{n\times 1}, \boldsymbol{y}_i \epsilon R^{o\times 1}$ $o$ :number of outputs

*it is assumed that Data is a measured sample with high enough diversity.

*$\boldsymbol{x}_i$ and$\boldsymbol{y}_i$may have any format (video, image, time series, etc.) ; however, to compute SmI, it must be reshaped as a vector.

$$\mathrm{SmI}(Data) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\boldsymbol{d}_{imax} - \boldsymbol{d}_{i^*}}{\boldsymbol{d}_{imax} - \boldsymbol{d}_{imin}}\right)$$

$$i^* = \underset{\forall q\neq i}{\arg\, min}\left\|\boldsymbol{x}_i - \boldsymbol{x}_q\right\| \qquad \boldsymbol{d}_{imax} = \max_{\forall q}\left\|\boldsymbol{y}_i - \boldsymbol{y}_q\right\| \qquad \boldsymbol{d}_{imin} = \min_{\forall q\neq i}\left\|\boldsymbol{y}_i - \boldsymbol{y}_q\right\|$$

$$\boldsymbol{d}_{i^*} = \left\|\boldsymbol{y}_i - \boldsymbol{y}_{i^*}\right\|$$

*$\|\cdot\|$ denotes Euclidian distance ($L_2$ norm) but it may be another distance definition such as $L_p$ norm.

** It is assumed that the input and target output data are normalized at each dimension just before computing the smoothness index.

*** the above definition of SmI can be biased by outliers.

# A modified “linear SmI”

$$\mathrm{SmI}(Data) = \frac{1}{m}\sum_{i=1}^{m} relu\left(1 - \frac{\boldsymbol{d}_{i^*} - \boldsymbol{d}_{imin}}{\boldsymbol{d}_{imean}}\right)$$

$$i^* = \underset{\forall q \neq i}{\arg\, min} \left\| \boldsymbol{x}_i - \boldsymbol{x}_q \right\| \qquad \boldsymbol{d}_{imean} = \frac{1}{m}\sum_{q=1}^{m} \left\| \boldsymbol{y}_i - \boldsymbol{y}_q \right\| \qquad \boldsymbol{d}_{imin} = \min_{\forall q \neq i} \left\| \boldsymbol{y}_i - \boldsymbol{y}_q \right\|$$

$$\boldsymbol{d}_{i^*} = \left\| \boldsymbol{y}_i - \boldsymbol{y}_{i^*} \right\|$$

Some notes:

1. the above definition of SmI is not affected by outliers due to using mean of syance instead of maximum distance.
2. The “relu” function actually assign zero smoothness index to all data points whose their nearest neighbors have far enough targets with them.

# A modified Exponential SmI

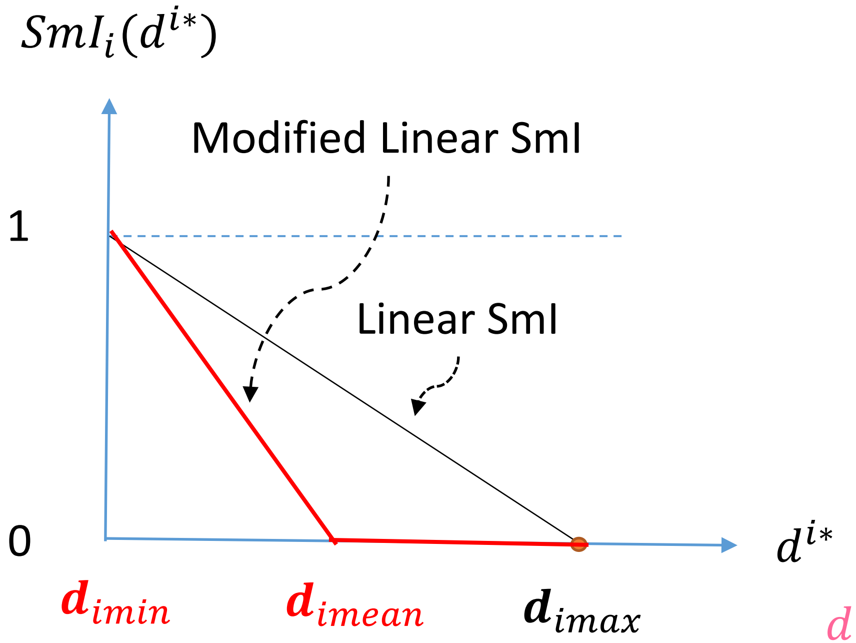
$$\mathrm{SmI}(Data)=\frac{1}{m}\sum_{i=1}^{m} SmI^{i} \quad SmI^{i}=\exp\left(-\gamma\,\frac{\boldsymbol{d}_{i^*}-\boldsymbol{d}_{imin}}{\boldsymbol{d}_{imean}}\right)$$

$$\boldsymbol{d}_{imean}=\frac{1}{m}\sum_{q=1}^{m}\left\|\boldsymbol{y}^{i}-\boldsymbol{y}^{q}\right\|$$ ,Smoothness rate $\gamma$>0

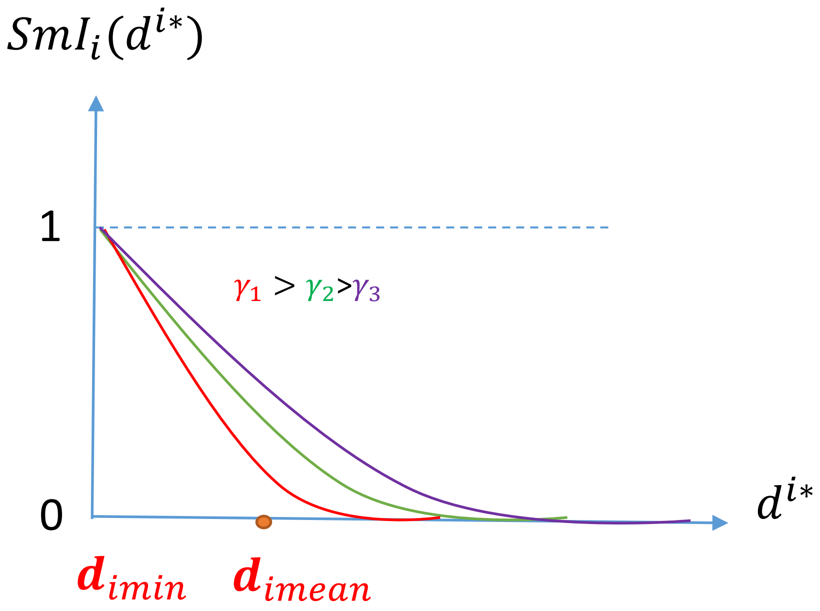
Some Notes:

1. Exponential SmI is not sensitivity to outliers.
2. For when $\gamma\rightarrow\infty$ , any distance variation: $\left(\frac{\boldsymbol{d}_{i^*}-\boldsymbol{d}_{imin}}{\boldsymbol{d}_{imean}}\right)$ drops the SmI, significantly.
3. Here we have an exponential smoothness with an optional smoothness rate.
4. To have more restricted definition of SmI, the smoothness rate must be chosen high or $\gamma$>>1.

# SmI Diagrams versus $d^{i*}$



Linear SmI



Exponential SmI

$$d^{i*} = \|y_i - y_{i^*}\|$$

$$\boldsymbol{d}_{imax} = \max_{\forall q}\|\boldsymbol{y}_i - \boldsymbol{y}_q\|$$

$$\boldsymbol{d}_{imin} = \min_{\forall q \neq i}\|\boldsymbol{y}_i - \boldsymbol{y}_q\|$$

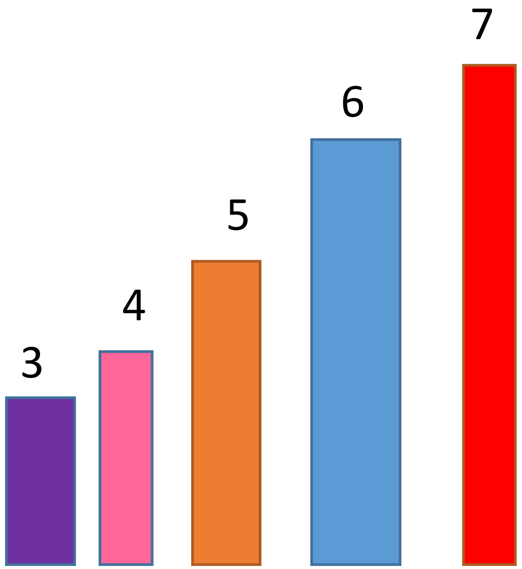
$$\boldsymbol{d}_{imean} = \frac{1}{m}\sum_{q=1}^{m}\|\boldsymbol{y}^i - \boldsymbol{y}^q\|$$

# Some notes

1. “SmI” is a normalized index between zero and one: $\mathrm{SmI} \in [0,1]$
2. $SmI \to 1$ $(Smoothness\ is\ maximmum)$ and $SmI \to 0$ $(Smoothness\ is\ minimmum)$
3. “SmI” measures that how nearness of input data leads to nearness of target data.
4. Assuming, the target outputs are outputs of a classification problem in “one-hot” format, SmI is actually measure the separation index: $\mathrm{SmI} = \mathrm{SI}$
5. Increasing the number of classes and considering a nearness among every two classes, SI is interpreted as a smoothness index. Actually, $\mathrm{SmI}$ shows in average that how neighboring examples in input space have classes with near distances in output.
6. SmI does not change for arbitrary position shift and (scalar) scale of the data
   $\forall \beta \neq 0, \forall \alpha \neq 0, \forall x_0, \forall y_0 \qquad \mathrm{SmI}\left(\left\{\left(x^i, y^i\right)\right\}_{i=1}^{m}\right) = SmI\left(\left\{\left(\beta \boldsymbol{x}_i + x_0, \alpha \boldsymbol{y}_i + y_0\right)\right\}_{i=1}^{m}\right)$

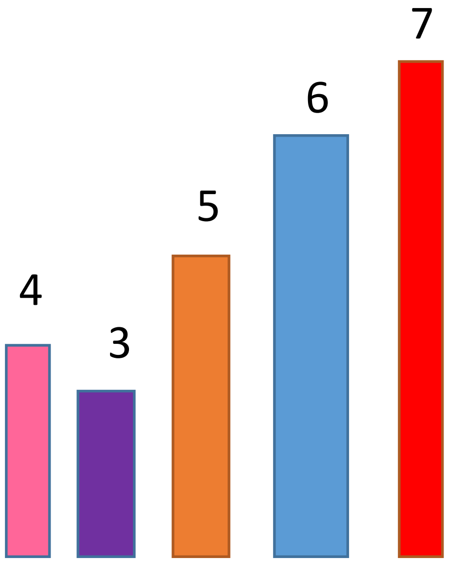
7. Smoothness index of target outputs $with\ themselves\ is$ maximum: $SmI\left(\left\{\left(\boldsymbol{y}_i, \boldsymbol{y}_i\right)\right\}_{i=1}^{m}\right)$=1; it means that how input data become more similar to output the smoothness index will increase.

# One-dimensional illustrative examples



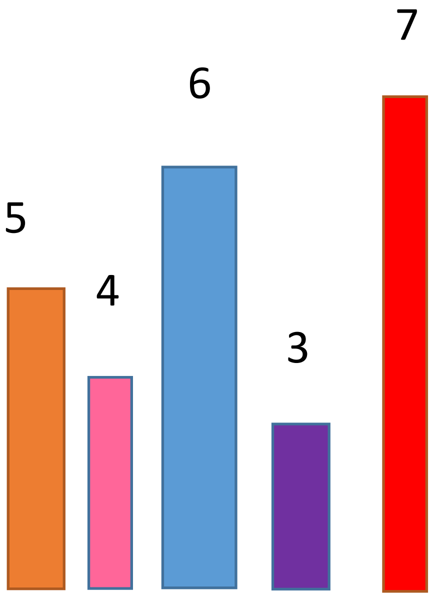
$$SmI = \frac{1}{5}\left(\frac{4-1}{4-1}+\frac{3-1}{3-1}+\frac{2-1}{2-1}+\frac{3-1}{3-1}+\frac{4-1}{4-1}\right)$$

$$SmI = 1$$



$$SmI = \frac{1}{5}\left(\frac{3-1}{3-1}+\frac{4-1}{4-1}+\frac{2-2}{2-1}+\frac{3-2}{3-1}+\frac{4-1}{4-1}\right)$$

SmI=0.7



$$SmI = \frac{1}{5}\left(\frac{2-1}{2-1}+\frac{3-1}{3-1}+\frac{3-2}{3-1}+\frac{4-3}{4-1}+\frac{4-4}{4-1}\right)$$

SmI=0.566

# Smoothness index of Each data point

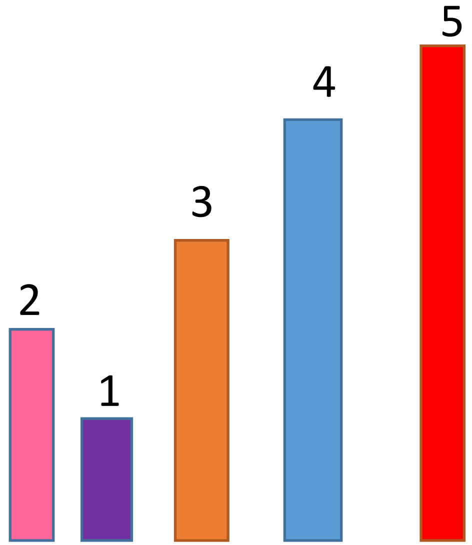
* $SmI(Data) = \frac{1}{m}\sum_i smi(x_i, y_i)$ , $smi(x_i, y_i) = \left(\frac{\boldsymbol{d_{imax}- \ d_{i^*}}}{\boldsymbol{d_{imax}- \ d_{imin}}}\right)$

SmI definition with data distribution: $SmI(Data) = Exp_{p(x,y)}(smi(x,y))$

Challenge: to compute $\mathrm{smi}(x, y)$ it is required to have $x^*$as the nearest neighbor of $x$.

❖ For a sample of data with high enough diversity SI can be approximated by equation *

A one dimensional illustrative example



$$SmI = \frac{1}{5}\left(\frac{3-1}{3-1} + \frac{4-1}{4-1} + \frac{2-2}{2-1} + \frac{3-2}{3-1} + \frac{4-1}{4-1}\right)$$

i=1,2,....,5

$SmI_i^{data} = \frac{3-1}{3-1}$=*1*

$SmI_2^{data} = \frac{4-1}{4-1}$ =1

$SmI_3^{data} = \frac{2-2}{2-1}$=0

$SmI_4^{data} = \frac{3-2}{3-1}$=*0.5*

$SmI_5^{data} = \frac{4-1}{4-1}$=1

# 2. High order SmI

$Data = \{(x_i, y_i)\}_{i=1}^{m} \forall i: x_i \epsilon R^{n\times 1} \quad \mathrm{y_i} \epsilon R^{o\times 1}$

$$\mathrm{SmI^r}(Data) = \frac{1}{m}\sum_{i=1}^{m} \min_{\forall j \epsilon \{1,..,r\}} \left( \frac{\boldsymbol{d}_{imax} - \ \boldsymbol{d}_{i_j^*}}{\boldsymbol{d}_{imax} - \ \boldsymbol{d}_{imin_j}} \right) \qquad \text{r: the order of "SmI"}$$

$$i_j^* = \underset{\forall q \neq i, i_1^*, \cdots, i_{j-1}^*}{\arg} \quad min \left\| \boldsymbol{x}_i - \boldsymbol{x}_q \right\| \qquad imin_j = \underset{\forall q \neq i, imin_1, \cdots, imin_{j-1}}{\arg} \quad min \left\| \boldsymbol{y}_i - \boldsymbol{y}_q \right\|$$

$$\boldsymbol{d}_{imin_j} = \left\| \boldsymbol{y}_i - \boldsymbol{y}_{imin_j} \right\| \qquad \boldsymbol{d}_{i^*} = \left\| \boldsymbol{y}_i - \boldsymbol{y}_{i_j^*} \right\|$$

- $\mathrm{SmI^r} \in [0,1]$
- $\mathrm{SmI^r}$ considers more restricted condition of smoothness than $\mathrm{SmI^j}$ $(j < r)$.
- For each “Data” we have: $\mathrm{SmI^r} \leq \mathrm{SmI^{r-1}} \leq \cdots \leq \mathrm{SmI^1}$ $\quad$ $\mathrm{SmI^1}$=SmI

# 3. High order soft SmI

$Data = \{(\boldsymbol{x}^i, \boldsymbol{y}^i)\}_{i=1}^m$ $\forall i{:}\ x^i \epsilon R^{n\times 1}$ $y^{\mathrm{i}} \epsilon R^{o\times 1}$

$$\mathrm{SmI}_{\mathrm{soft}}^{\mathrm{r}}(Data) = \frac{1}{m\times r}\sum_{i=1}^{m}\sum_{j=1}^{r}\left(\frac{\boldsymbol{d}_{imax} - \boldsymbol{d}_{i_j^*}}{\boldsymbol{d}_{imax} - \boldsymbol{d}_{imin_j}}\right) \mathrm{j} = 1,2,\ldots,\mathrm{r} \quad \mathrm{r}{:}\ \text{the order of "SmI"}$$

- $\mathrm{SmI}_{\mathrm{soft}}^{\mathrm{r}} \in [0,1]$
- $\mathrm{SmI}_{\mathrm{soft}}^{\mathrm{r}}$ considers less restricted condition of smoothness than $\mathrm{SmI}^{\mathrm{r}}$

$\mathrm{SmI}_{\mathrm{soft}}^{\mathrm{r}} \geq \mathrm{SmI}^{\mathrm{r}}$ and $\mathrm{SmI}_{\mathrm{soft}}^{1} = \mathrm{SmI}^{1}$

# 4. Cross SmI

$Data = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$ $\quad D_{test} = \{(\check{\boldsymbol{x}}_i, \check{y}_i)\}_{i=1}^{m_{test}}$

$$SmI_{cross}(D_{test}, Data) = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} \left( \frac{\boldsymbol{d}_{imax} - \boldsymbol{d}_{i^{\#}}}{\boldsymbol{d}_{imax} - \boldsymbol{d}_{imin_j}} \right)$$

$i^{\#} = \underset{\forall q}{\arg} min \left\| \check{\boldsymbol{x}}_i - \boldsymbol{x}_q \right\|$

Cross SI measures the separation index of a test domain of dataset $D_{test}$ based on the main domain of dataset $Data$.

It can be shown that:

$$SmI_{cross}(D_{test}, Data) = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} SmI_{cross}\left( (\check{\boldsymbol{x}}_i, \check{l}_i), Data \right)$$

# Global SmI

- For the Data:$\{(\mathrm{x_i}, y_i\ )\}_{i=1}^{m}$ we have Global SmI when $SmI^{m-1}(Data) = 1.$
- For Data with Global SmI, One can show that for each example $x_i$ and two other examples $x_{i_1}$ and $x_{i_2}$:

$$if\ \|x_i - x_{i_1}\| \leq \|x_i - x_{i_2}\| \text{ then } \|y_i - y_{i_1}\| \leq \|y_i - y_{i_2}\|$$

- For Data$(\mathrm{x_i}, y_i\ )\}_{i=1}^{m}$ where $y_i = \Psi x_i$ and $\Psi$ have orthogonal columns with equal norms, we have global SmI.

# Data Node Connectivity Matrix (by SmI)

- $Node^{k}$ :$\{(\mathrm{x}_{\mathrm{i}}^{k}\ )\}_{i=1}^{m}\}$, k=1,2,…,N, $\mathrm{x}_{\mathrm{i}}^{k} \in R^{n_k}$

Connectivity matrix:

ConMat= $\left[smI_{k_1,k_2}\right]_{N\times N}$

$smI_{k_1,k_2}$=SmI ($Node^{k_1}, Node^{k_2}$)

The element indicates how $Node^{k_2}$ is affected by $Node^{k_1}$

when $smI_{k_1,k_2}$=1, the influence of $Node^{k_1}$ over $Node^{k_2}$ is maximum.

But when $smI_{k_1,k_2}$=0 the influence of $Node^{k_1}$ over $Node^{k_2}$ is minimum.

Unlike correlation matrix:

1. The matrix is not symmetric
2. The dimensions of different nodes are not necessary equal.
3. The influence is not necessary linear.

**Connectivity Matrix**

| | Node1 | Node2 | ⋯ | NodeN |
|---|---|---|---|---|
| Node1 | 1 | $SmI_{2,1}$ | ⋯ | $SmI_{1,N}$ |
| Node2 | $SmI_{1,2}$ | 1 | ⋯ | $SmI_{2,N}$ |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| NodeN | $SmI_{N,1}$ | $SmI_{N,2}$ | ⋯ | 1 |

# Data Variables Causal Matrix

Non cyclic

|  | Var1 | Var2 | ⋯ | VarN |
|---|---|---|---|---|
| Var1 | 0 | $ca_{2,1}$ | ⋯ | $ca_{1,N}$ |
| Var2 | $ca_{1,2}$ | 0 | ⋯ | $ca_{2,N}$ |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| VarN | $ca_{N,1}$ | $ca_{N,2}$ | ⋯ | 0 |

$var^k$ :$\{(\mathrm{x}_\mathrm{i}^k\ )\}_{i=1}^{m}\}$, k=1,2,…,N, $\mathrm{x}_\mathrm{i}^k \in R^1$

Causal Matrix= $\left[ca_{k_1,k_2}\right]_{N\times N}$

$ca_{k_1,k_2} \in \{1,0\}$

If $ca_{k_1,k_2}$=1 it means that $var^{k_2}$is a cause variable for $var^{k_1}$.

By using an exploration algorithm for each variable, all possible variables which make maximum possible “SmI” with a certain variable are revealed as cause of that variable and a confidence between 0 and 1 is given for that.

❖A subset of independent variables which provide the largest “SmI” for a certain variable, are indicated as the cause set of that variable.

# Similarity transformation in “SI” and “SmI”

- Show that for all possible $r$

$$SI^r(\{(x_i, l_i)\}_{i=1}^m) = SI^r(\{(\Psi_1 x_i, \Psi_2 l_i)\}_{i=1}^m)$$
$$SmI^r(\{(x_i, y_i)\}_{i=1}^m) = SmI^r(\{(\Psi_1 x_i, \Psi_2 y_i)\}_{i=1}^m)$$

where

$\Psi_h$ (h=1,2) have orthogonal columns  with equal norms.

2.1 Complexity Indices
An Unsupervised Index : 3- linear density

# Linear Density index (LDI)

LDI measures the average of linear densities of a number of clusters.
(Each cluster has a unimodal distribution around a focal point)

# 2.1. Linear Density Index (Ldi)

1. Ldi

$Data = \{(\boldsymbol{x}_i\ )\}_{i=1}^{m} \forall i: \boldsymbol{x}^i \epsilon R^{n\times 1}$

Some notes

1. Assumption: “Data” is a measured sample from a domain with high enough diversity.
2. Data has been clustered as $N$ unimodal shape clusters: $cluster_1, cluster_2, \ldots, cluster_N$ where:

$$Data = cluster_1 \cup \ldots \cup cluster_N \text{ and } \forall k_1 \neq k_2: cluster_{k_1} \cap cluster_{k_2} = 0$$

$$cluster_k = \{(\boldsymbol{x}_i^k\ )\}_{i=1}^{n_k}, n_k = number\ of\ members\ in\ cluster_k$$

Now, linear density of “Data” is defined as follows:

$$\mathrm{Ldi}(Data) = \frac{1}{n\_clusters} \sum_{k=1}^{n\_clusters} \mathrm{ldi}_k\ , \quad \mathrm{ldi}_k = \frac{n_k}{\bar{\sigma}_k}$$

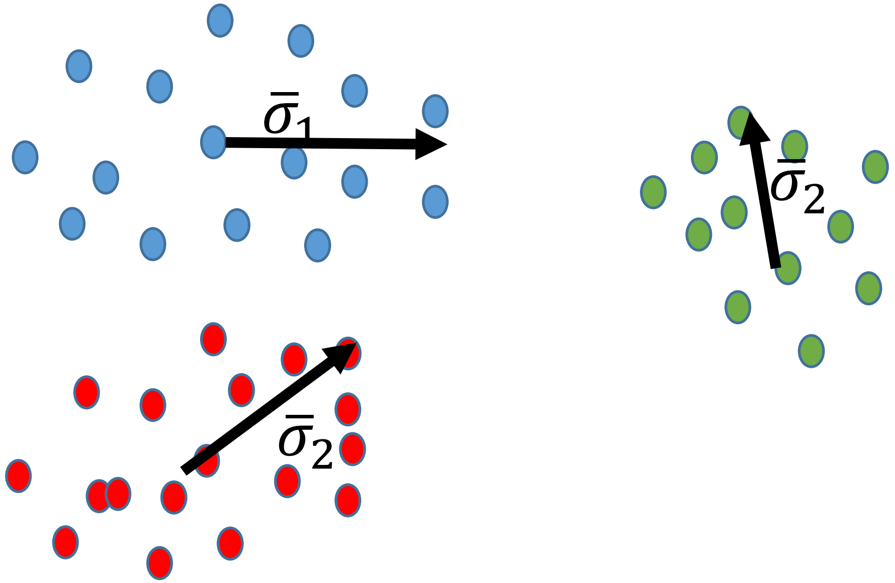
$\bar{\sigma}_k$=Maximum Singular value of the covariance matrix of $cluster_k = \{(\boldsymbol{x}_i^k\ )\}_{i=1}^{n_k}$

$Cov(cluster_j) = \frac{1}{n_k} \sum_{i=1}^{n_k} (\boldsymbol{x}_i^k - c^k))(\boldsymbol{x}_i^k - c^k)^T \qquad c^k = \frac{1}{n_k} \sum_{i=1}^{n_k} \boldsymbol{x}_i^k$

*It is assumed that the input data is normalized at each dimension just before computing Linear Density Index .

** Each data point such as $\boldsymbol{x}_i$may have any format (video, image, time series, etc.) ; however, to compute “ldi” for $\boldsymbol{x}_i$, it should be encoded and then being reshaped as a vector.

# An illustrative example



$n_{clusters} = N = 3$

$n_1 = 16, \bar{\sigma}_1$=1

$n_2 = 12, \bar{\sigma}_2$=0.9

$n_2 = 20, \bar{\sigma}_3$=1.5

Ldi(Data)=1/3*(16/1+12/0.9+20/1.5)=14.22

# What does the density of a distribution tell you?

- Ldi actually computes the density of a data distribution.
- In this sense, the (spatial)density is **the differential probability of observing X∈[x,x+Δx] divided by the length of the interval Δx**. So the density  represents a likelihood of observing X∈[x,x+Δx].
- larger densities reflecting a larger likelihood of observing values in that interval.
- Actually, the ldi has a direct relation with Likelihood.
- The clusters with high enough density (hot clusters) are more informative than other regions because they provide larger likelihood for observation.

# Ldi and Entropy

- Gibs Entropy Formula: $Entropy = -kB \sum_j \ p_j \log(p_j)$ s.t. $\sum_j \ p_j$=1 $p_j: microstate$
- In distributions where we have larger ldi, one can say the entropy is decreased and we have more informative data
- Higher densities mean **more deterministic, less randomness and hence more accuracy**.
- Actually, the ldi has an inverse relation with Entropy.

# Ldi in classification and regression (problems)

- In a classification problem, it is desired to get feature space that within distances among examples of a class decrease and between distances among examples from different classes increase.
- According to the above property, the examples of each class form one or a few clusters with high ldi.
- The number of clusters must be equal or larger than number of classes.
- However, in a regression problem, it is not expected that the input space become as a number of clusters with high densities.
- The desired distribution of input examples strongly depend to the distribution of target examples.

# Relative Density

$Data = \{(\boldsymbol{x}_i)\}_{i=1}^{m}$

*1. Cluster Data space by ldi:*

$Data$=Union$(cluster_j) \quad j = 1,2, \ldots ,J$

$c_j = center\ of\ clustyer\ j$

*2. Compute Relative Density*

$$RL(Data) = \frac{1}{m}\sum_{i=1}^{m} rd(x_i)$$

$rd(x_i) = 1 - \frac{l_{i_{j1}}}{l_{ij_2}}$ *(rd: relative density)*

l$_{\text{ij}}$ $= (\|x_i - c_j\|)$ $\qquad l_{ij_1} \lhd l_{ij_2} \lhd l_{ij_3} \ldots \lhd l_{ij_J}$

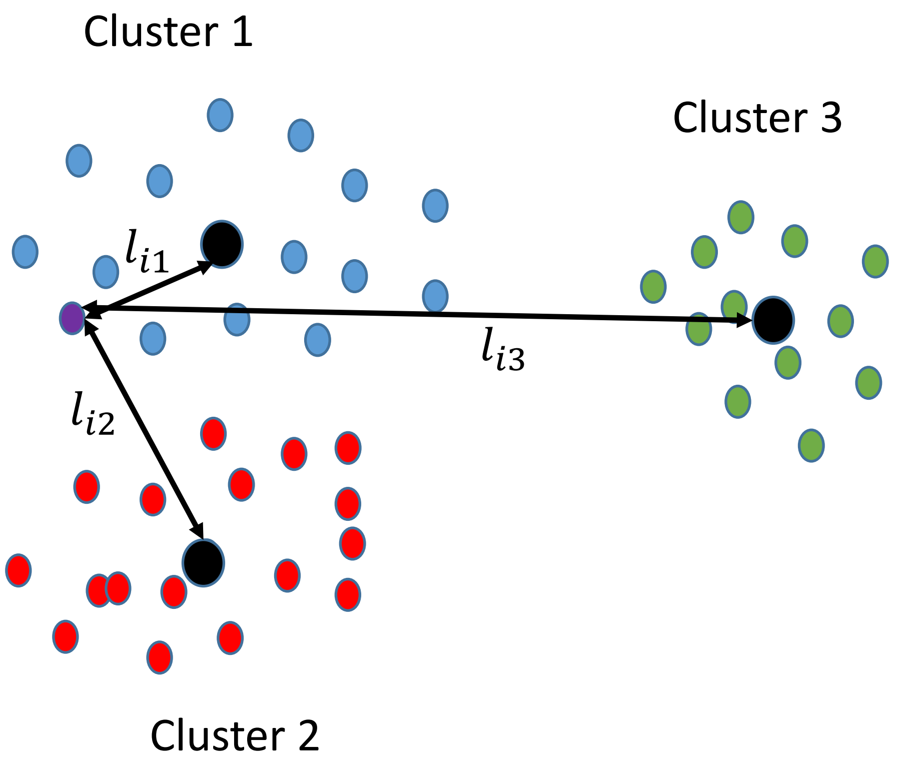
*Relative density is an unsupervised normalized index*

$$0 \le RL(data) \le 1$$

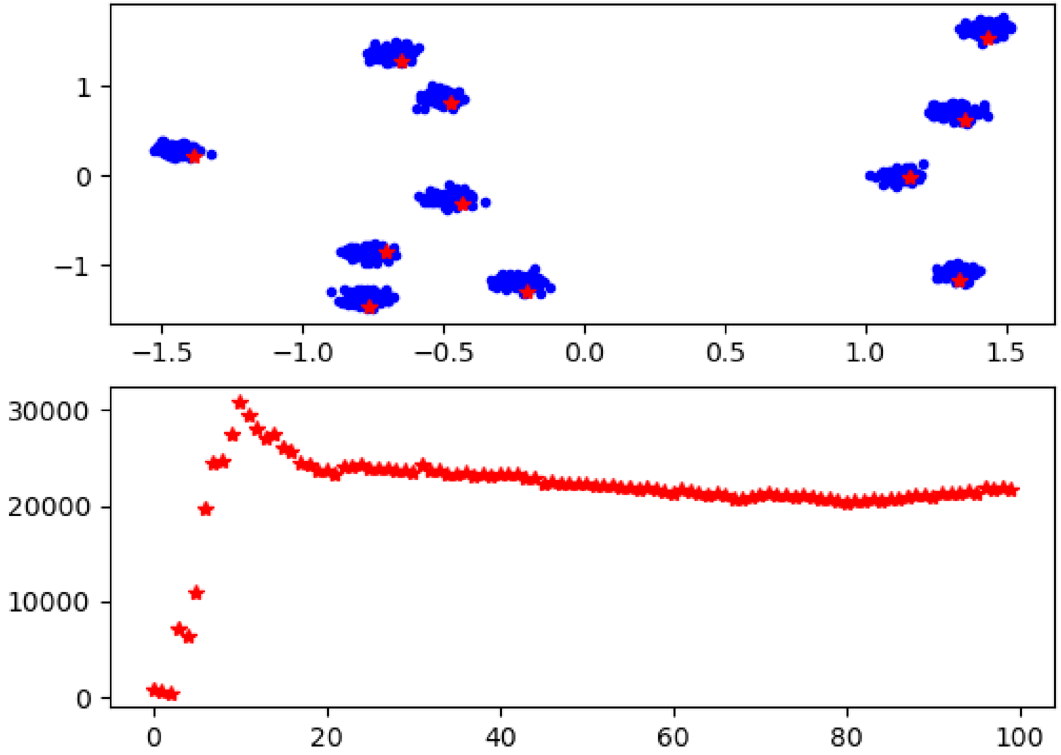
*For when* $RL(Data) \to 1$ *we have a data distribution with very dense and separated clusters*

*For when* $RL(Data) \to 0$ *we have a data distribution with very near clusters.*

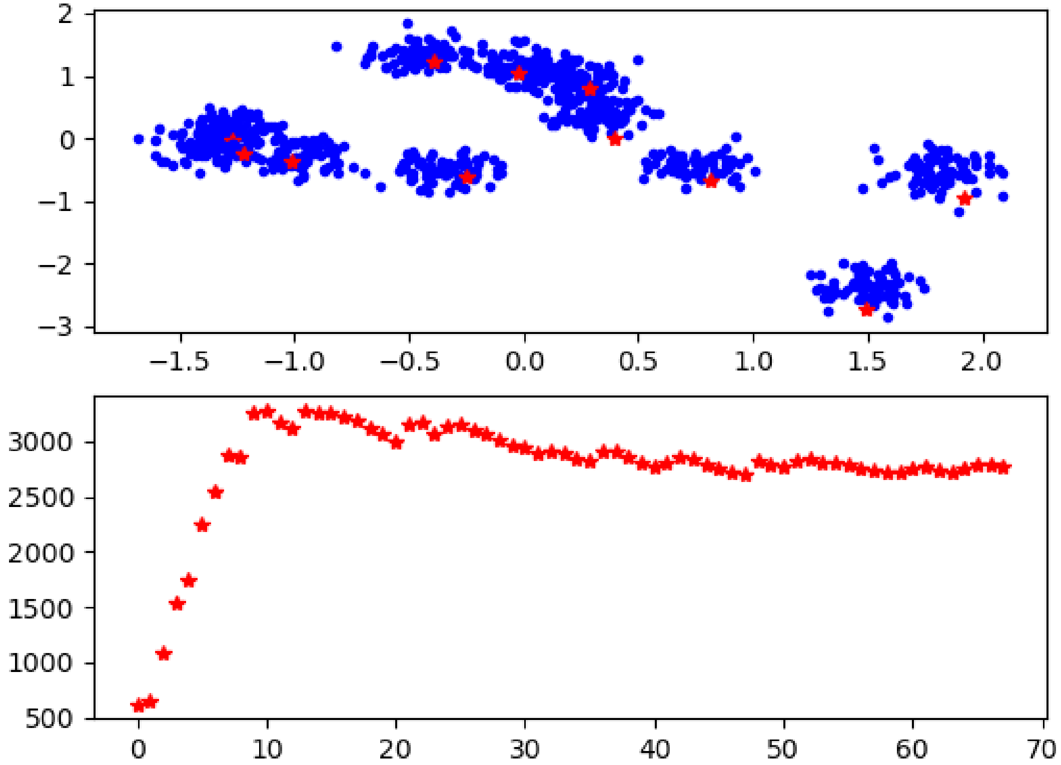
-



# illustrative examples



RD(data)=0.8551, mm=880,
ncluster=11



RD(data)=0.525
M=880, n_cluster=11

# Cross Relative Density

$Data = \{(\boldsymbol{x}_i)\}_{i=1}^{m}$ $\quad D_{test} = \{(\breve{\boldsymbol{x}}_i\ )\}_{i=1}^{m_{test}}$

*1. Cluster Data space by ldi:*

$Data = \text{Union}\left(cluster_j\right) \quad j = 1,2,\ldots,J$

$c_j = center\ of\ clustyer\ j$

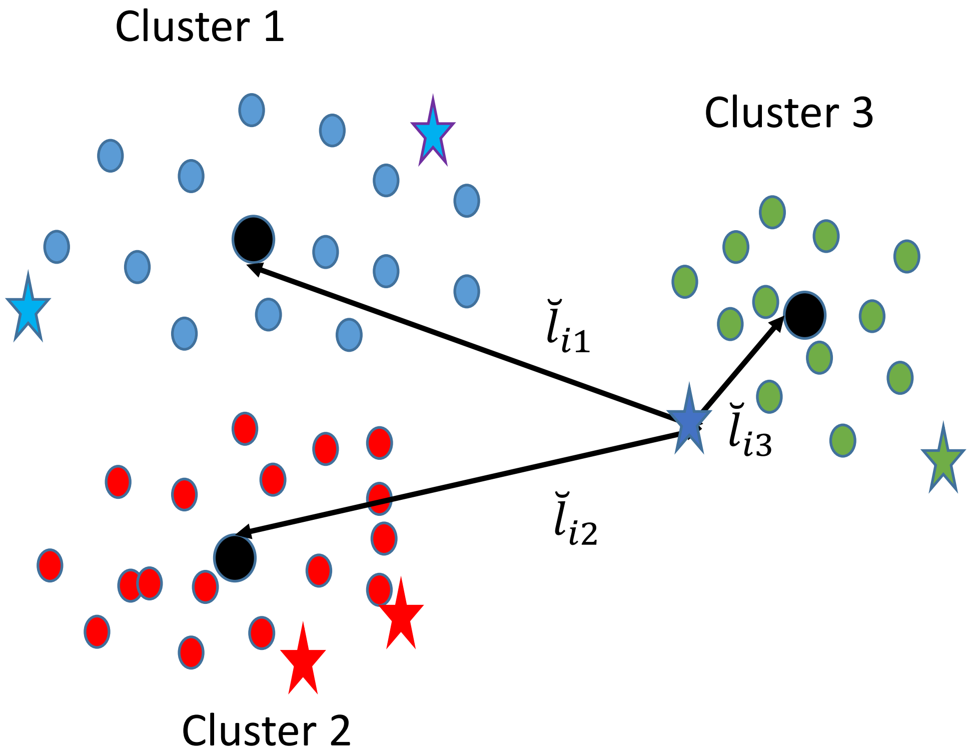
*2. Compute Cross Relative Density*

$$cross\_RD(Dtest, Data) = \frac{1}{mtest}\sum_{i=1}^{mtest} cross\_rd(\breve{\boldsymbol{x}}_i)$$

$$\text{cross\_rd}(\breve{\boldsymbol{x}}_i) = 1 - \frac{\breve{l}_{i_{j1}}}{\breve{l}_{i_{j2}}}$$

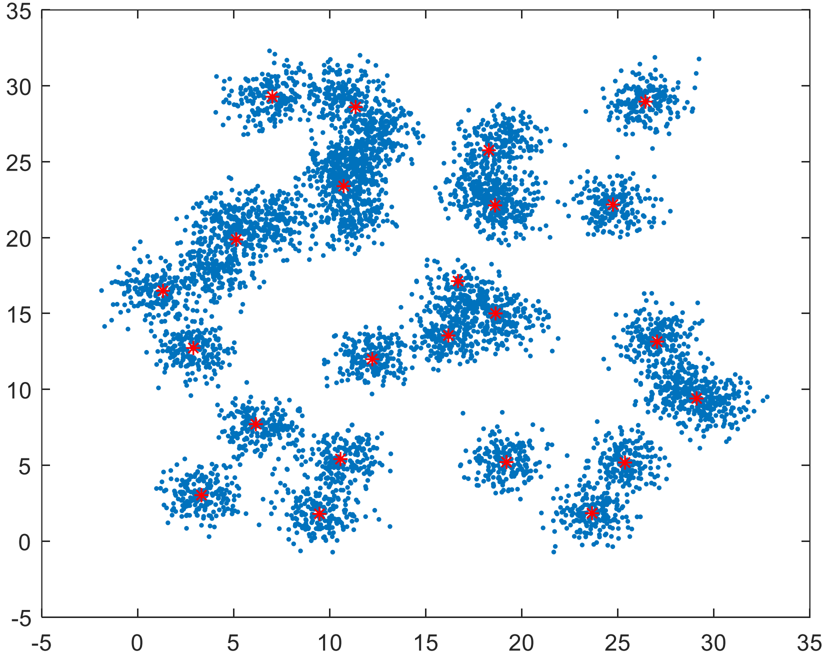
$\breve{l}_{ij} = (\|\breve{\boldsymbol{x}}_i - c_j\|) \quad \breve{l}_{ij_1} < \breve{l}_{ij_2} < \breve{l}_{ij_3} \ldots < \breve{l}_{ij_J}$

-



# Some applications of using “ldi”

N=30 clusters
with a few overlaps

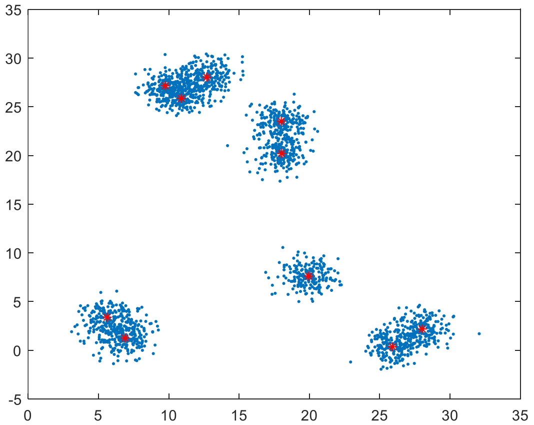


ldi

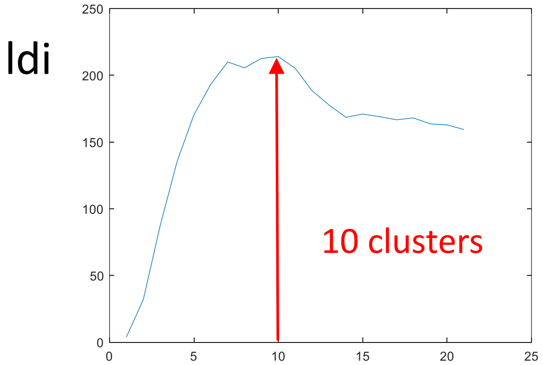
## 1. Data Clustering

Clustering is an important task in the process of knowledge discovery in data mining.

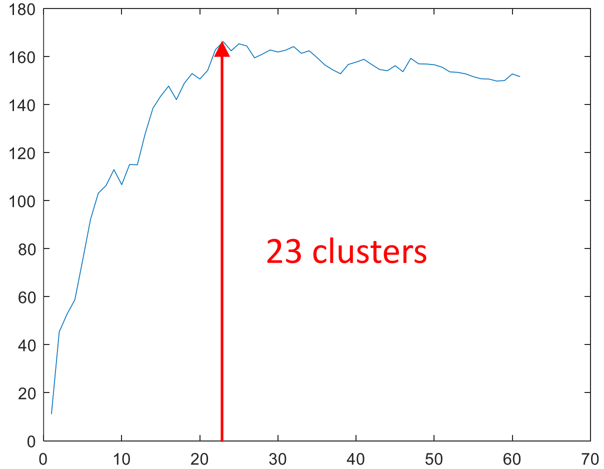
suppose that a distribution of data is formed as a mixture of Gaussian shape clusters where they have almost the same size and their overlap is pretty low. One can use “ldi” to find all clusters and their members. In fact, the ldi for when the predicted clusters are the same defined clusters, is maximum.



N=10 clusters



num of predicted clusters



num of predicted clusters

## 2. Unsupervised Feature Selection

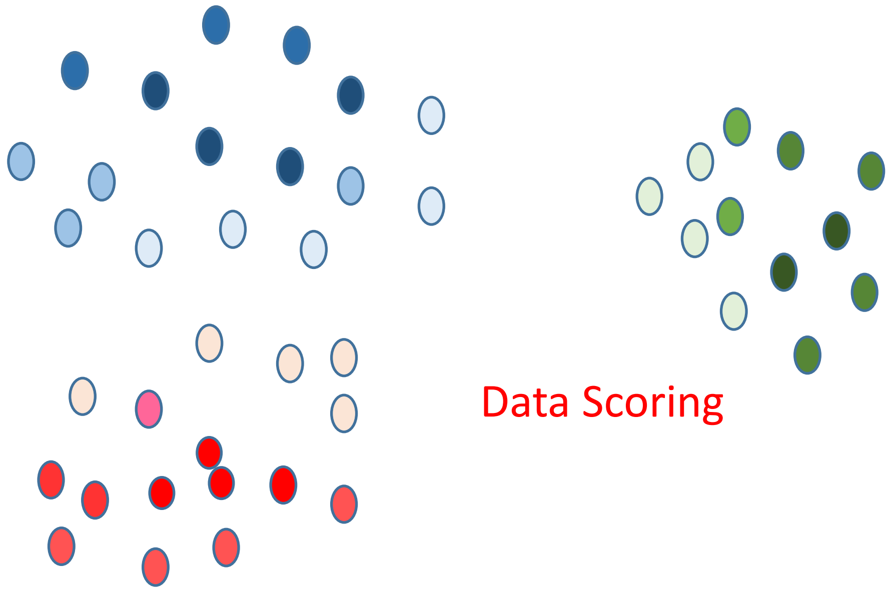
After removing some correlated features by an encoder or using Sml, One can choose a subset of features which provide maximum sum of “ldi” over clusters or equally with (maximum N*ldi) (N= number of clusters).

## 3. Feature Representation (Self Supervised Learning)

- In self supervised learning process, the space which has more sum of “ldi” over clusters, has more information

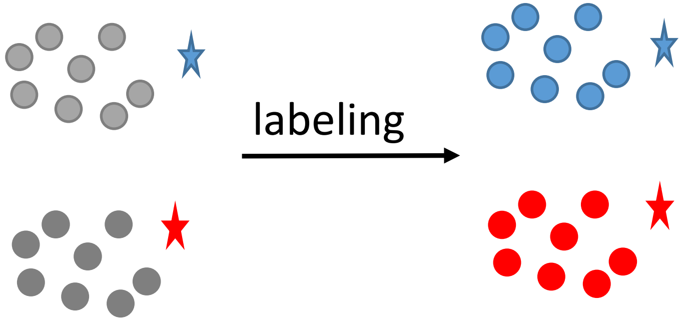
## 4. Unsupervised Data Scoring

By using the concept of “ldi” one can find the clusters and then score each data point with respect to its nearness to the center of clusters. Data example with high scores are near to center of a cluster and far from centers of other clusters.



## 5. Data Labeling with a few labeled data

Assume that there are a few data points with known labels (rare labeled data points). After clustering of the unlabeled data points with ldi, assign the labels of all member of each cluster as the label of a rare labeled data point which has minimum distance to its center.



## 6. Unsupervised (Self-Cross) Data domain scoring

How one can score a domain of data points with respect to its distribution or distribution of another domain of data points. Actually, by using the concept of ldi and clustering the data points one can compute the self or cross score for a domain dataset.

# 2.2. Data Analysis

2.2.1 Dataset evaluation and Scoring

2.2.2 Supervised Feature Selection

2.2.3 Data Clustering

2.2.4 Unsupervised Feature Selection

2.2.5 Data Connectivity Matrix (Smi Table)

## 2.2.1 Dataset evaluation and Scoring

# Dataset Evaluation

- Assume that a dataset: $Data = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{m}$ is provided for training a model in a classification $(\boldsymbol{y}_i \equiv l_i)$ or regression problem.
- We would like to know how such a dataset is challenging and which model is more appropriate for it.

**Algortihm1:**

**(To score the complexity of the dataset and to suggest an appropriate deep or a shallow model)**

1. Compute $SI(Data)$ ($SmI\ (Data)$) of the dataset.
2. If $SI(Data)$ ($SmI\ (Data)$) is nearer to one than to zero, the provided data is less challenging and a shallow ANN is suggested to model it.
3. If $SI(Data)$ ($SmI\ (Data)$) is nearer to zero, the provided data is less challenging and a deep learning ANN with high enough complexity is suggested to model it.

Fair

# SI index for some known datasets

$Data = \{(\boldsymbol{x}_i, l_i)\}_{i=1}^{m} \;\; m = 50000$

| DataSet | N. Of Classes | Sepration Index | SI_random |
|---|---|---|---|
| MNIST Digits | 10 | 0.9722 | 0.10 |
| Fashion MNIST | 10 | 0.85072 | 0.10 |
| Cifar10 | 10 | 0.35086 | 0.10 |
| Cifar100 | 100 | 0.17446 | 0.01 |

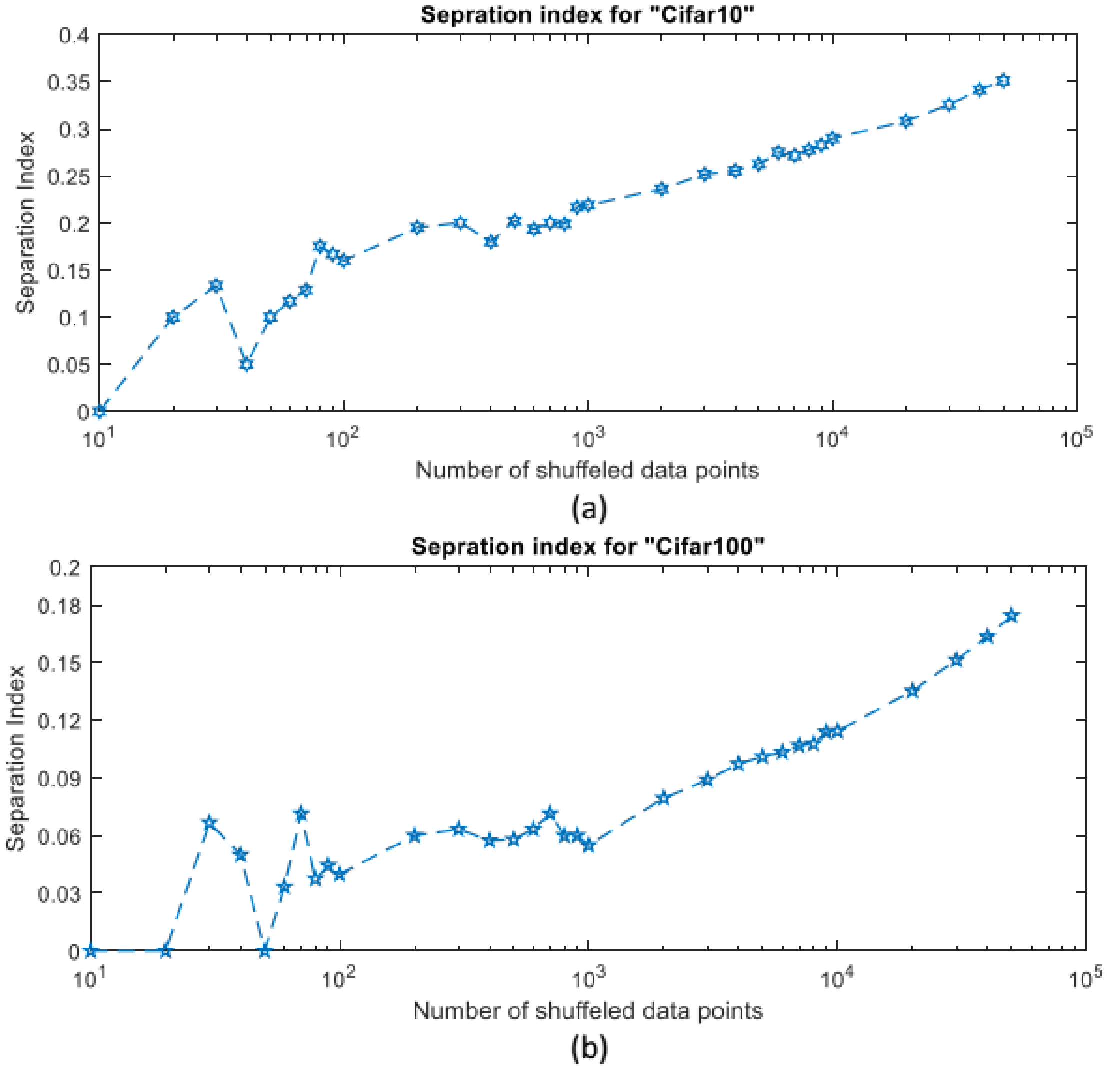
*The expected SI is equal to $SI_w = 1/n_C$ for when (1) each class has equal number of examples and (2) all examples are distributed with uniform random variable.

** to have fair comparison among SI of different data set the it is suggested to normalize in number of classes ($n_C$)

$$A\ sugestion\colon\ SI_n = \frac{SI - 1/n_C}{1 - 1/n_C}$$

# The sensitivity of SI to the number of data points in a data-set

- Actually, the SI(SmI) is suggested to be used for a standard data-set with high enough diversity.
- For a data-set with a very low number of data points (insufficient diversity), SI (SmI) changes non-smoothly versus number of data points. In such a state, it does not show the true complexity of the data, and the sensitivity to variation of number of data points is high.
- For a data-set, while the number of data points is high enough (sufficient diversity), the SI (SmI) changes more smoothly versus number of data points (low sensitivity) .



(a)

(b)

Fig. 6. The plot of separation index versus different number of shuffled data points in both “cifar10” (a) and “cifar100” (b).

# Dataset ranking

- Computing $SI(Data)$ ($SmI\ (Data)$) provides a solution to rank and compare standard provided datasets from challenging view point.

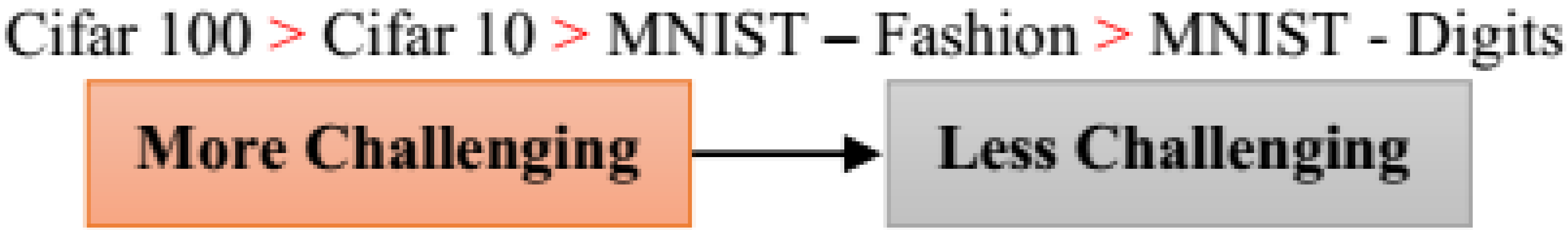


Fig. 4. The ordered classification data sets from more challenging to less challenging.

# Cross domain dataset evaluation

## 1. Classification Problems

$Data = \{(\boldsymbol{x}_i, l_i)\}_{i=1}^{m}$ $D_{test} = \{(\breve{\boldsymbol{x}}_i, \check{l}_i)\}_{i=1}^{m_{test}}$

$SI_{cross}(D_{test}, Data) = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} \delta(\check{l}_i, l_{i^{\#}})$ ,$i^{\#}$=$\underset{\forall q}{\arg min} \|\breve{\boldsymbol{x}}_i - \boldsymbol{x}_q\|$

❖if $SI_{cross}(D_{test}, Data) \gg SI(Data)$ , then it is expected that *the training model* (with "*Data*") will have high *generlization for* $D_{test}$.

❖if $SI_{cross}(D_{test}, Data) \ll SI(Data)$ , then it is expected that *the training model* (with "*Data*") will have low *generlization for* $D_{test}$.

❖The test data set is called homogenous with the training dataset when $SI_{cross}(D_{test}, Data) \approx SI(Data)$

| DataSet (m=50000) | N. Of Classes | Sepration Index | Cross Sep. Index |
|---|---|---|---|
| MNIST Digits | 10 | 0.9722 | 0.9666 |
| Fashion MNIST | 10 | 0.85072 | 0.844 |
| Cifar10 | 10 | 0.35086 | 0.3539 |
| Cifar100 | 100 | 0.17446 | 0.1755 |

## 2. Regression Problems

$Data = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{m}$ $\quad D_{test} = \{(\breve{\boldsymbol{x}}_i, \breve{\boldsymbol{y}}_i)\}_{i=1}^{m_{test}}$

$$SmI_{cross}(D_{test}, Data) = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} \left( \frac{\boldsymbol{d}_{imax\#} - \ \boldsymbol{d}_{i^\#}}{\boldsymbol{d}_{imax\#} - \ \boldsymbol{d}_{imin\#}} \right)$$

$i^\#$=$\arg \underset{\forall q}{min} \left\| \breve{\boldsymbol{x}}_i - \boldsymbol{x}_q \right\|$ $\quad \boldsymbol{d}_{imax\#} = \max_{\forall q} \left\| \breve{\boldsymbol{y}}_i - \boldsymbol{y}_q \right\|$ $\quad \boldsymbol{d}_{imin\#} = \min_{\forall q} \left\| \breve{\boldsymbol{y}}_i - \boldsymbol{y}_q \right\|$ $\quad \boldsymbol{d}_{i^\#}$=$\left\| \breve{\boldsymbol{y}}_i - \boldsymbol{y}_{i^\#} \right\|$

❖if $SmI_{cross}(D_{test}, Data) \gg SmI(Data)$ , then it is expected that $the\ training\ model$ (with “$Data$”) will have high $generlization\ for\ D_{test}$.

❖if $SmI_{cross}(D_{test}, Data) \ll SmI(Data)$ , then it is expected that $the\ training\ model$ (with “$Data$”) will have low $generlization\ for\ D_{test}$.

❖The test data set is called homogenous with the training dataset when $SmI_{cross}(D_{test}, Data) \approx SmI(Data)$

# Dataset evaluation for some Regression cases

$Data = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$ $D_{test} = \{(\check{\boldsymbol{x}}_i, \check{y}_i)\}_{i=1}^{m_{test}}$

| DataSet | N. Of data points | Sml linear | Smi mean |
|---|---|---|---|
| Diabets | (m=353,n=10) | 0.7286 | 0.4230 |
| Car Price | (m=174, n=63) | 0.9340 | 0.7784 |
| California housing | (m=16512,n=8) | 0.7303 | 0.4005 |
| Sinc function | (m=900,n=2) | 0.9840 | 0.8027 |

California Housing > Diabets > Car price> Sinc-Function

More challenging-----------------------------Less Challenging

# Cross domain dataset evaluation

## 1. Regression Problems

$Data = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$ $D_{test} = \{(\breve{\boldsymbol{x}}_i, \breve{y}_i)\}_{i=1}^{m_{test}}$

$SmI_{cross}(D_{test}, Data) = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} \delta(\check{l}_i, l_{i^{\#}})$ ,$i^{\#}$=$\underset{\forall q}{\arg min} \|\breve{\boldsymbol{x}}_i - \boldsymbol{x}_q\|$

- if $SmI_{cross}(D_{test}, Data) \gg SmI(Data)$ , then it is expected that *the training model* (with "*Data*") will have high $generlization\ for\ D_{test}$.
- if $SmI_{cross}(D_{test}, Data) \ll SmI(Data)$ , then it is expected that *the training model* (with "*Data*") will have low $generlization\ for\ D_{test}$.
- The test data set is called homogenous with the training dataset when $SmI_{cross}(D_{test}, Data) \approx SmI(Data)$

| DataSet | N. Of data points | Sml linear | Smi mean | Cr. Smi linear | Cr. Smi mean |
|---|---|---|---|---|---|
| Diabets | (m=353,n=10) (mtest=89,n=10) | 0.7286 | 0.4230 | 0.7635 | 0.4739 |
| Car Price | (m=174, n=63),(mtest=31-n=63) | 0.9340 | 0.7784 | 0.9291 | 0.7741 |
| California housing | (m=16512,n=8) (m=4128, n=8) | 0.7303 | 0.4005 | 0.7323 | 0.4061 |
| Sinc function | (m=900,n=2) (mtest=100,n=2) | 0.9840 | 0.8027 | 0.9828 | 0.8295 |

# Algortihm2:
## (To check that if a test dataset is less or more challenging with the main dataset(Cross domain score)

1. Compute $SI\ (SmI, or\ relative\ density(rd))$ of the main data.

2. Compute cross_SI (cross_smi, cross_rd) for the test datset in comparison to main dataset,

3. If cross_SI<<SI (cross_SmI<<SmI, cross_rd<<rd) then test dataset is more challenging than the main dataset.

4. If cross_SI>>SI (cross_SmI>>SmI, cross_rd>>rd) then test domain data set is less challenging than the main dataset.

5. If cross_SI ≅ SI (cross_SmI ≅ SmI, cross_rd≅rd) then test domain data set is homogenous with the main dataset.

# Data dividing for test and training datasets

- To have high enough generalization, divide an available dataset to test and training sets in order that the $SI_{cross}(SmI\ _{cross})$ of test dataset becomes almost equal to $SI(SmI)$ of the training dataset.

$$Data_{available} \rightarrow \{D_{test}, Data\}$$

1. For classification problems

$$SI_{cross}(D_{test}, Data) \sim SI(Data)$$

2. For regression problems

$$SmI_{cross}(D_{test}, Data) \sim SmI(Data)$$

❖ **Domain Score for train data set= Domain Score for test data set**

# Data Point Scoring

1. Classification $Data = \{(\boldsymbol{x}_i, l_i)\}_{i=1}^{m}$

$$Score(x_i) = si(x_i)\ \text{(1}^{\text{st}}\text{ order or any variants)}$$

$$si(x_i) \in \{0,1\}$$

2. Regression $Data = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$

$$Score(x_i) = smi(x_i)\ \text{(1}^{\text{st}}\text{ order or any variants)}$$

$$smi(x_i) \in [0,1]$$

3. Unsupervised $Data = \{(\boldsymbol{x}_i)\}_{i=1}^{m}$

$$Score(x_i) = rd(x_i)$$

$$rd(x_i) \in [0,1]$$

## Some notes

- Hard Examples are Examples which have lower scores
- One can use score data (1) to determine risky data, (2)to clean data, or (3) to weight data in learning process.
- The score of data domain is the average of scores of all data points (SI, SmI, RD)
- The cross score of test data domain is the average of scores of all test data points in the distribution of the train data domain (cross_SI, cross_SmI, cross_RD)

### 3.2.2 Supervised Feature Selection

Among available features, which ones should be selected?

Among different observations which ones should be integrated?

How one can make a suitable fusion for some available data sources with different modalities?

# Algortihm3:

## (Subset Selection among distinct features by SI(SmI))

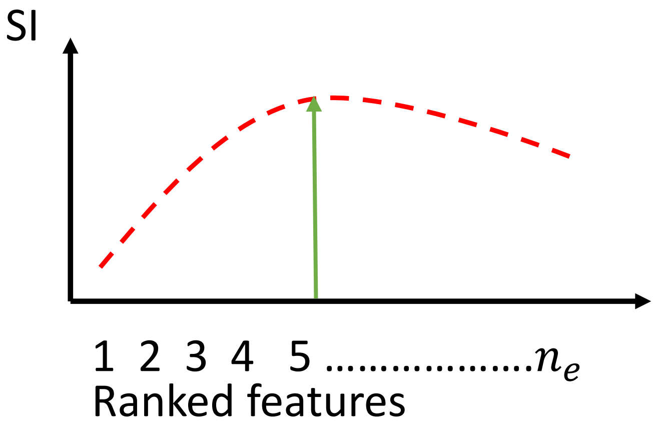
- Assume there is $x_{available} = \{x_1, \dots, x_{ne}\}$ with $n_e$features.
- Among available $n_e$ inputs, select a subset $\boldsymbol{x} \subseteq x_{available}$ and define:

$$Data = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{m}$$

- To decrease the complexity, select $\boldsymbol{x}$ in a way that the $SI(Data)$ $(SmI(Data),)$ becomes maximum.
- It is aimed to remove all non-relevant, correlated inputs and noise, which decrease the SI(SmI) or do not increase it.
- “Forward selection”, “backward elimination” or any other exploration algorithm can be used for this purpose.

# (Greedy)Forward Selection Algorithm

```
repeat="True"; best_set=[];rest_set=[1,2..,$n_e$];
 index_max=0; index_opt=0;
While repeat:
• For j in range(len(rest_fea)):
    • fea=concat(best_fea,rest_fea[j])
    • If index(Data(fea))>index_max:
        • index_max=index(Data(fea))
        • best_fea=rest_set[j]
• best_set:=concat(best_set, best_fea)
• rest_set:=rest_set-best_fea
• If index_max<index_opt:
    • index_opt:=index_max
• else:
    • Repeat="False"
• If len(rest+fea)==0:
    • Repeat="False"
---------End
```



Some notes:

1. best_set includes the selected features
2. index=SI, SmI or any other index
3. For the cases you think there will be oscillations in the index plot per epoch, you can repeat the  the stop condition to see maximum absolute index

# Choosing effective inputs by Smoothness Index
## Example1 (illustrative)

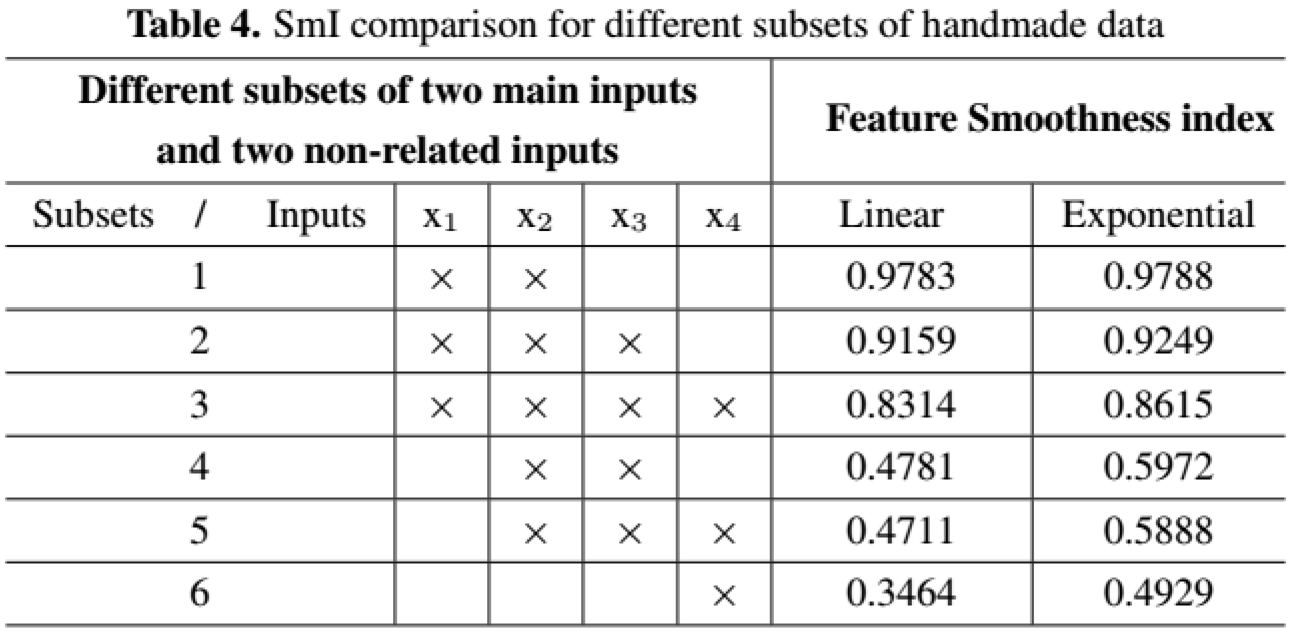
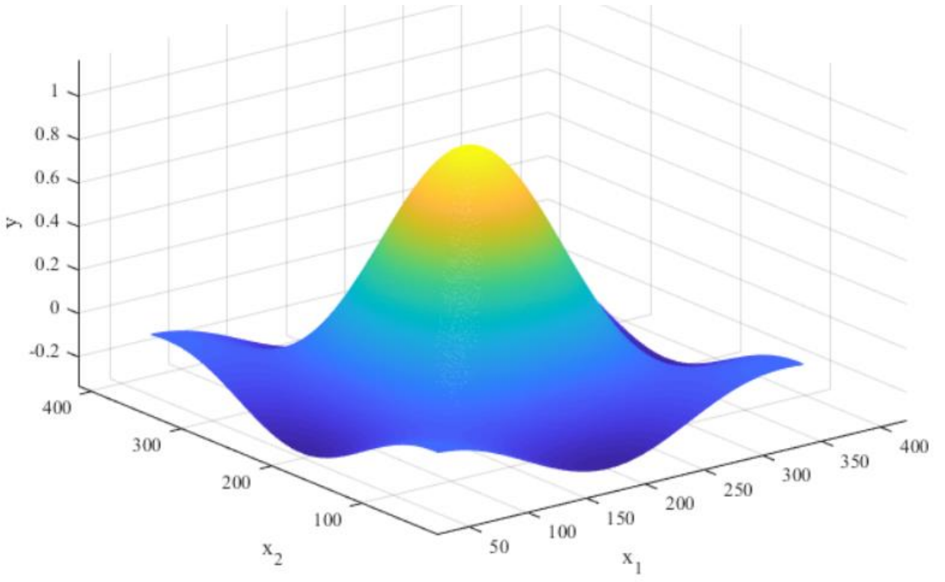


**Table 4.** SmI comparison for different subsets of handmade data

| **Different subsets of two main inputs and two non-related inputs** | | | | | **Feature Smoothness index** | |
|---|---|---|---|---|---|---|
| Subsets / Inputs | $x_1$ | $x_2$ | $x_3$ | $x_4$ | Linear | Exponential |
| 1 | × | × | | | 0.9783 | 0.9788 |
| 2 | × | × | × | | 0.9159 | 0.9249 |
| 3 | × | × | × | × | 0.8314 | 0.8615 |
| 4 | | × | × | | 0.4781 | 0.5972 |
| 5 | | × | × | × | 0.4711 | 0.5888 |
| 6 | | | | × | 0.3464 | 0.4929 |

Two-dimensional synchronous function of 1000 randomly generated data points.



$$y = \frac{sin\,(x_1)\,sin\,(x_2)}{x_1 x_2},$$

$$0 < |x_1| \leq 5, 0 < |x_2| \leq 5, 0 < |x_3| \leq 5, 0 < |x_4| \leq 5$$

white noise variables with X3, and X4 features have uniform distribution

While we have relevant inputs the SmI is maximum so the subset selection by SmI reveals the relevant inputs.

# Choosing effective inputs by Smoothness Index

## Example2

Yearly residential water consumption data, along with climatic characteristics, and socioeconomic factors of rural areas of Isfahan, Iran are aggregated.

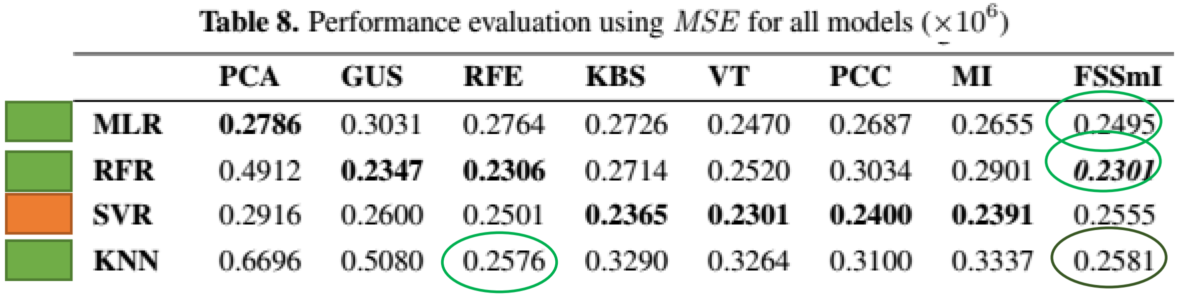


**Table 8.** Performance evaluation using *MSE* for all models ($\times 10^6$)

| | | PCA | GUS | RFE | KBS | VT | PCC | MI | FSSmI |
|---|---|---|---|---|---|---|---|---|---|
| 🟩 | MLR | **0.2786** | 0.3031 | 0.2764 | 0.2726 | 0.2470 | 0.2687 | 0.2655 | 0.2495 |
| 🟩 | RFR | 0.4912 | **0.2347** | **0.2306** | 0.2714 | 0.2520 | 0.3034 | 0.2901 | ***0.2301*** |
| 🟧 | SVR | 0.2916 | 0.2600 | 0.2501 | **0.2365** | **0.2301** | **0.2400** | **0.2391** | 0.2555 |
| 🟩 | KNN | 0.6696 | 0.5080 | 0.2576 | 0.3290 | 0.3264 | 0.3100 | 0.3337 | 0.2581 |

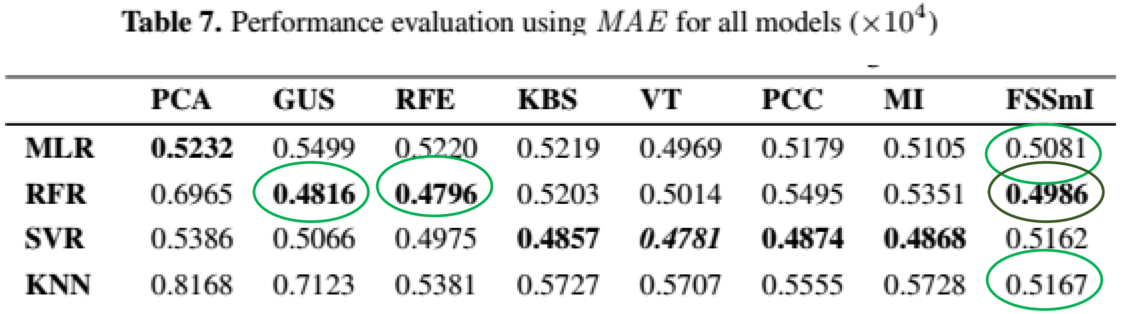


**Table 7.** Performance evaluation using *MAE* for all models ($\times 10^4$)

| | PCA | GUS | RFE | KBS | VT | PCC | MI | FSSmI |
|---|---|---|---|---|---|---|---|---|
| MLR | **0.5232** | 0.5499 | 0.5220 | 0.5219 | 0.4969 | 0.5179 | 0.5105 | 0.5081 |
| RFR | 0.6965 | **0.4816** | **0.4796** | 0.5203 | 0.5014 | 0.5495 | 0.5351 | **0.4986** |
| SVR | 0.5386 | 0.5066 | 0.4975 | **0.4857** | ***0.4781*** | **0.4874** | **0.4868** | 0.5162 |
| KNN | 0.8168 | 0.7123 | 0.5381 | 0.5727 | 0.5707 | 0.5555 | 0.5728 | 0.5167 |

**Selectin Algorithms**

Forward **selection based SmI (FSSmI)**

Principle Component Analysis (PCA)

Recursive feature elimination (RFE)

Generic uni-variant selection (GUS)

Mutual Information (MI)

K-best selection (KBS)

Pearson correlation coefficient (PCC)

Variance threshold (VT)

Models

Support vector regression (SVR)

Multiple linear regression (MLR)

K nearest neighbors (KNN)

Random forest regression (RFR)

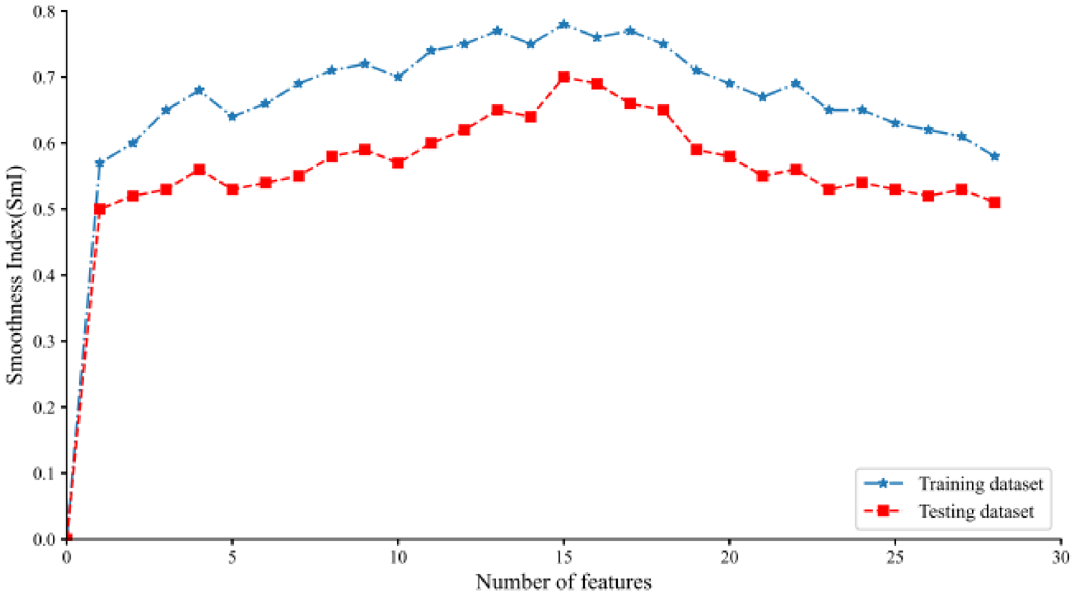
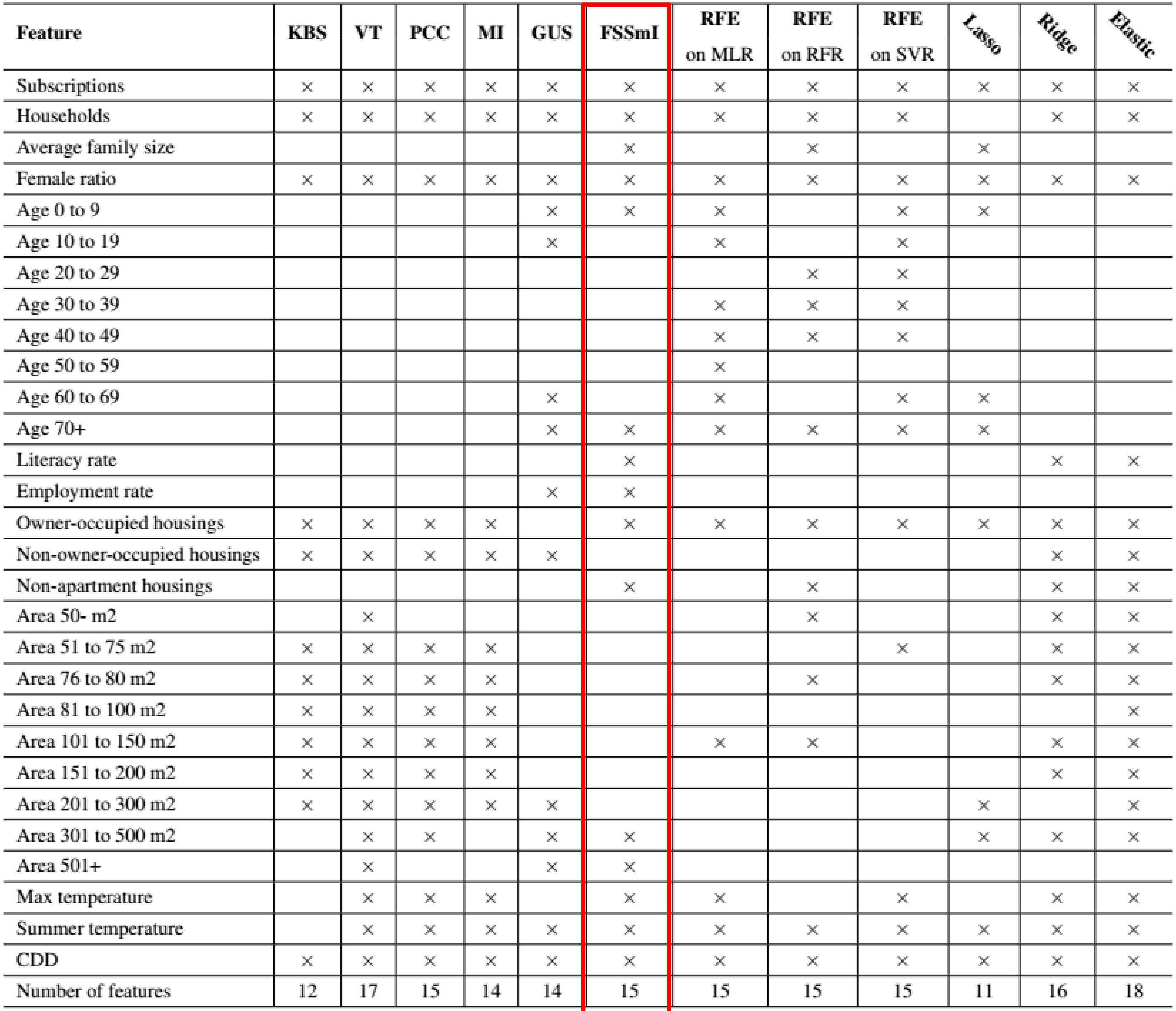


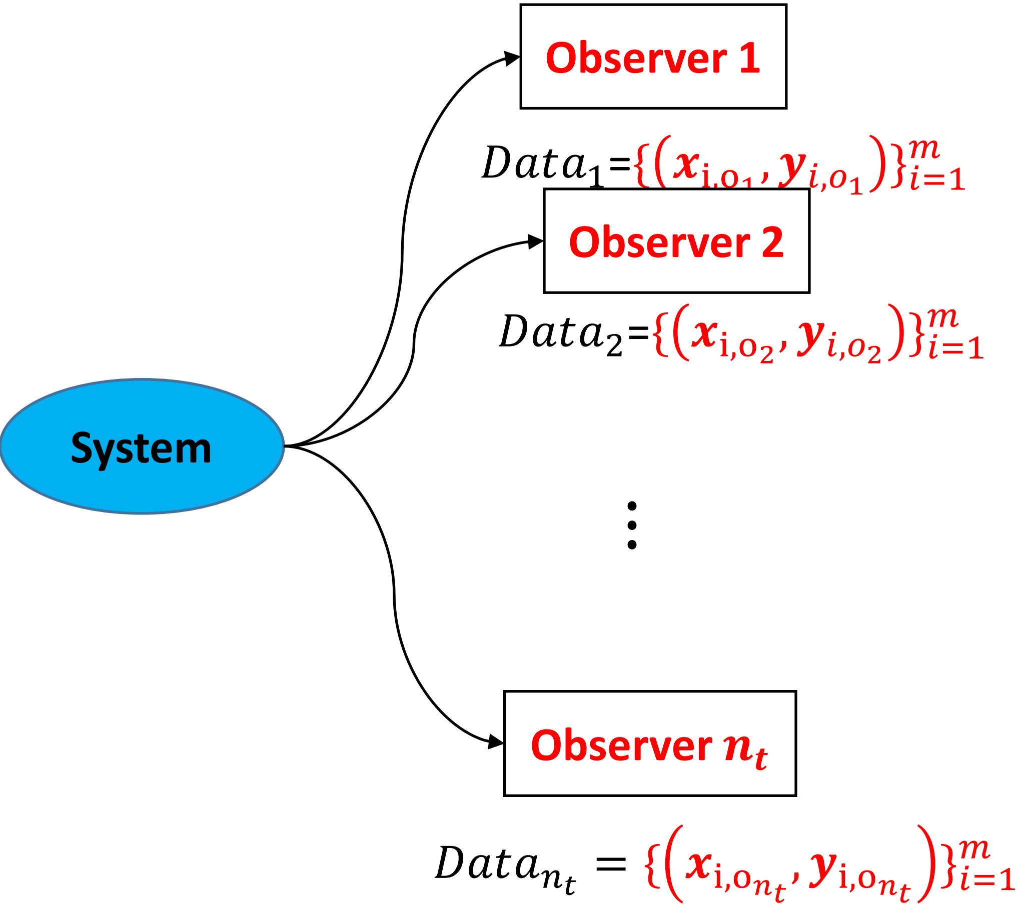
**Fig. 5.** Smoothness Index (SmI) based on number of features. There is a good correlation between the smoothness charts of training and test datasets, i.e., the selected features based on the absent data are the same as those that give the highest SmI in the training dataset.

**Table 5.** Selected features.
The features, households, subscriptions, and female ratio, are selected by all the feature selection methods that show their influences on regression



| Feature | KBS | VT | PCC | MI | GUS | FSSmI | RFE on MLR | RFE on RFR | RFE on SVR | Lasso | Ridge | Elastic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Subscriptions | × | × | × | × | × | × | × | × | × | × | × | × |
| Households | × | × | × | × | × | × | × | × | × |  | × | × |
| Average family size |  |  |  |  |  | × |  | × |  | × |  |  |
| Female ratio | × | × | × | × | × | × | × | × | × | × | × | × |
| Age 0 to 9 |  |  |  |  | × | × | × |  | × | × |  |  |
| Age 10 to 19 |  |  |  |  | × |  | × |  | × |  |  |  |
| Age 20 to 29 |  |  |  |  |  |  |  | × | × |  |  |  |
| Age 30 to 39 |  |  |  |  |  |  | × | × | × |  |  |  |
| Age 40 to 49 |  |  |  |  |  |  | × | × | × |  |  |  |
| Age 50 to 59 |  |  |  |  |  |  | × |  |  |  |  |  |
| Age 60 to 69 |  |  |  |  | × |  | × |  | × | × |  |  |
| Age 70+ |  |  |  |  | × | × | × | × | × | × |  |  |
| Literacy rate |  |  |  |  |  | × |  |  |  |  | × | × |
| Employment rate |  |  |  |  | × | × |  |  |  |  |  |  |
| Owner-occupied housings | × | × | × | × |  | × | × | × | × | × | × | × |
| Non-owner-occupied housings | × | × | × | × | × |  |  |  |  |  | × | × |
| Non-apartment housings |  |  |  |  |  | × |  | × |  |  | × | × |
| Area 50- m2 |  | × |  |  |  |  |  | × |  |  | × | × |
| Area 51 to 75 m2 | × | × | × | × |  |  |  |  | × |  | × | × |
| Area 76 to 80 m2 | × | × | × | × |  |  |  | × |  |  | × | × |
| Area 81 to 100 m2 | × | × | × | × |  |  |  |  |  |  |  | × |
| Area 101 to 150 m2 | × | × | × | × |  |  | × | × |  |  | × | × |
| Area 151 to 200 m2 | × | × | × | × |  |  |  |  |  |  | × | × |
| Area 201 to 300 m2 | × | × | × | × | × |  |  |  |  | × |  | × |
| Area 301 to 500 m2 |  | × | × |  | × | × |  |  |  | × | × | × |
| Area 501+ |  | × |  |  | × | × |  |  |  |  |  |  |
| Max temperature |  | × | × | × |  | × | × |  | × |  | × | × |
| Summer temperature |  | × | × | × | × | × | × | × | × | × | × | × |
| CDD | × | × | × | × | × | × | × | × | × | × | × | × |
| Number of features | 12 | 17 | 15 | 14 | 14 | 15 | 15 | 15 | 15 | 11 | 16 | 18 |

# Subset selection among distinct observations



$Data_1 = \{(\boldsymbol{x}_{\mathrm{i},\mathrm{o}_1}, \boldsymbol{y}_{i,o_1})\}_{i=1}^{m}$

$Data_2 = \{(\boldsymbol{x}_{\mathrm{i},\mathrm{o}_2}, \boldsymbol{y}_{i,o_2})\}_{i=1}^{m}$

$Data_{n_t} = \{(\boldsymbol{x}_{\mathrm{i},\mathrm{o}_{n_t}}, \boldsymbol{y}_{\mathrm{i},\mathrm{o}_{n_t}})\}_{i=1}^{m}$

- It is aimed to select $n_s$ observations from available $n_t$ observations and then concatenate them in order have maximum SI (SmI).

Best concatenation $\boldsymbol{x}_i^* = [\boldsymbol{x}_{i,o_1^*},\ldots,\boldsymbol{x}_{i,o_{n_s}^*}]$, $\boldsymbol{y}_{i,o_1} = \boldsymbol{y}_{i,o_2} \ldots = \boldsymbol{y}_{i,o_{n_t}}$

*For classification problems*

$$\mathrm{SI}(\{(\boldsymbol{x}_{\mathrm{i}}^*, l_i)\}_{i=1}^{m}) \geq \mathrm{SI}(\{(\check{\boldsymbol{x}}_i, l_i)\}_{i=1}^{m})$$

or for *regression problems*

$$\mathrm{SmI}(\{(\boldsymbol{x}_{\mathrm{i}}^*, \boldsymbol{y}_i)\}_{i=1}^{m}) \geq \mathrm{SmI}(\{(\check{\boldsymbol{x}}_i, \boldsymbol{y}_i)\}_{i=1}^{m})$$

**where** $\check{\boldsymbol{x}}$ denotes any other concatenation from available $n_t$ different observations.

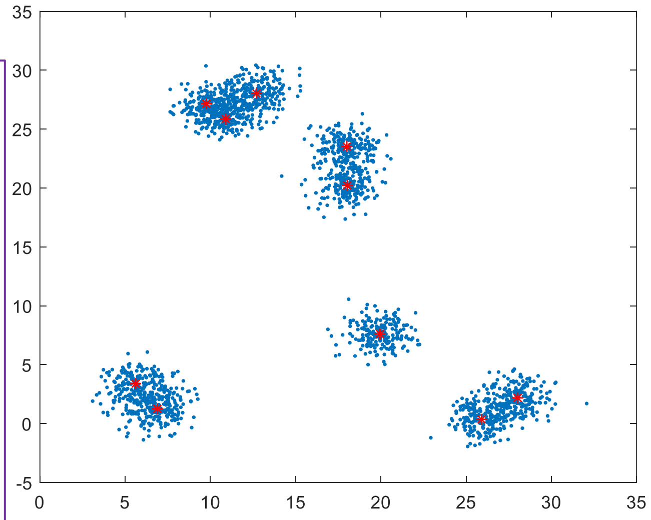
- “Forward selection”, “backward elimination” or any other exploration algorithms can be used for this purpose.

# 2.2.3 Data Clustering

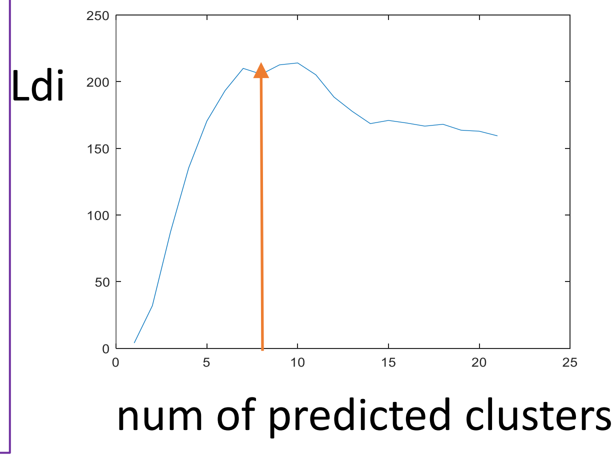
K-Splits: improved K-means clustering algorithm to automatically detect the number of clusters, SO Mohammadi, A Kalhor, H Bodaghi, Computer Networks, Big Data and IoT: Proceedings of ICCBI 2021, 197-213

## Algorithm4 Ksplits

```
A greedy version of Ksplit Algorithm

Def  density_scatter(cluster):
        $Cov$ =Covariance_Matrix(cluster)
        Find $(v^{*}, \sigma^{*})$ as the pair of maximum Eigne vector and its corresponding Engen value of $Cov$
        density=num(cluster)/$\sigma^{*}$,   scatter=num(cluster)× $\sigma^{*}$

      return   density,   scatter

=================================

0- Initiate the number of clusters: N=1 and define $cluster_1$=(whole)Data

1-if N==1: compute the density of $cluster_1$ as $density_1$ and put $ldi(1) = density_1$

2. For all available m clusters choose the worst one which has maximum scatter:

            $j_w = \arg\min scatter_j \,, j = 1,2,3, \ldots, N$

3. Split $cluster_{jw}$into two new clusters ($cluster_{jw}$, $cluster_{N+1}$)  by using k_means clustering algorithm and then
put: N:=N+1

4- Compute the linear density of all clusters as $ldi(N) = \sum_{j=1}^{N} density_j$

5. If $ldi(N) > ldi(N-1)$ appove the last split operation and jump to step 1, else reject the split, put N:= $N - 1$
and go to the next step.

6.  $m$ clusters are $\{cluster_j\}_{j=1}^{N}$
```



N=10 clusters



## 2.2.4 Unsupervised Feature Selection

**Algorithm5**

- Stage1: Remove every feature which has correlation with other features:

(a) consider a feature as an output and other features as possible inputs, then by using Algorithm3, find features which make maximum SmI.

(b) If the the maximized SmI is larger than a threshold, remove the considered feature.

(c) Repeat steps (a) and (b) for other features until a set of independent features remain.

- Stage2: Choose all Features which maximizes the number of clusters.

(a) Use a forward selection or backward elimination or any other exploration algorithm to find features which maximizes the number of clusters.

# 2.2.5 Data Connectivity Matrix (Smi Table)

There are $K$ data nodes: $Node^k$ :{$\left(\mathrm{x}_\mathrm{i}^k\right)\}_{i=1}^m$}, k=1,2,...,N

where each node has its dimension: $\mathrm{x}_\mathrm{i}^k \in R^{n_k}$.

It is demanded to define a $K \times K$ matrix as Data Connectivity Matrix by SmI.

```
Algorithm 6
for $k_1$ in range($K$):
    for $k_2$ in range (K):
        $smI_{k_1,k_2}$=SmI ($Node^{k_1}$, $Node^{k_2}$)
        put the computed element at ConMat= $\left[smI_{k_1,k_2}\right]_{N\times N}$
#-------
```

when $smI_{k_1,k_2}$=1, the influence of $Node^{k_1}$ over $Node^{k_2}$ is maximum.

But when $smI_{k_1,k_2}$=0 the influence of $Node^{k_1}$ over $Node^{k_2}$ is minimum.

Unlike correlation matrix:

1. The matrix is not symmetric
2. The dimensions of different nodes are not necessary equal.
3. The influence is not necessary linear.

**Connectivity Matrix**

| | Node1 | Node2 | ⋯ | NodeN |
|---|---|---|---|---|
| Node1 | 1 | $SmI_{2,1}$ | ⋯ | $SmI_{1,N}$ |
| Node2 | $SmI_{1,2}$ | 1 | ⋯ | $SmI_{2,N}$ |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| NodeN | $SmI_{N,1}$ | $SmI_{N,2}$ | ⋯ | 1 |

# An illustrative Example

California_Housing

Data Set Characteristics:

Number of Instances: 20640

Number of Nodes: 8

Nodes: MedInc median income in block group - HouseAge median house age in block group - AveRooms average number of rooms per household - AveBedrms average number of bedrooms per household - Population block group population - AveOccup average number of household members - Latitude block group latitude - Longitude block group longitude

Connectivity Matrix=

[1.0000, 0.0000, 0.2751, 0.0312, 0.0035, 0.0401, 0.0259, 0.0726],

[0.0151, 1.0000, 0.0400, 0.0126, 0.0464, 0.0195, 0.1210, 0.1439],

[0.2651, 0.1694, 1.0000, 0.1382, 0.0135, 0.0804, 0.0568, 0.0960],

[0.0842, 0.1603, 0.2555, 1.0000, 0.0859, 0.0484, 0.0248, 0.0964],

[0.0553, 0.1339, 0.0106, 0.0845, 1.0000, 0.0442, 0.0000, 0.0719],

[0.0722, 0.1935, 0.0597, 0.1066, 0.0000, 1.0000, 0.1867, 0.2552],

[0.0158, 0.0265, 0.0368, 0.0047, 0.0100, 0.0422, 1.0000, 0.7952],

[0.0131, 0.0450, 0.0336, 0.0045, 0.0043, 0.0407, 0.7763, 1.0000]

# End of Chapter 2

Thank you