# Spectral Clustering

组员：刘俊浩、宁锦来、黄德东、文豪

# Code

Project Link: https://github.com/jhliu17/spectral-clustering-matlab

- Datasets
  - Toy datasets: blobs, circles, moons
  - Real datasets: seeds(普通数据), SMS Spam(文本数据), digits(图像数据)
- Graph
  - Fullyconnected, nearest_neighbor, e_neighborhood
- Metric
  - Distance metric: cosine_distances, euclidean_distances, manhattan_distances, rbf_kernel, laplacian_kernel, sigmoid_kernel, polynomial_kernel
  - Result analysis: adjusted_rand_score, similarity_matrix
- Utils
  - Normalization
- Solver
  - SpectralClustering

# Code

Scikit-Learn similar API

```
[X, y] = make_digits_dataset(300, true, false);
W = fullyconnected(X, 7.8, 'rbf');

[C, ~] = SpectralClustering(W, size(unique(y), 1), 2);
rng(568);

figure
gscatter(Y(:,1), Y(:,2), y);
grid on;
figure
gscatter(Y(:,1), Y(:,2), C);
grid on;
adjusted_rand_score(y, C)
```

# Toy datasets

[X, y] = make_blobs(200, 3);

$X \in R^{200\times 2}$

$y \in R^{200\times 1}$

2 features

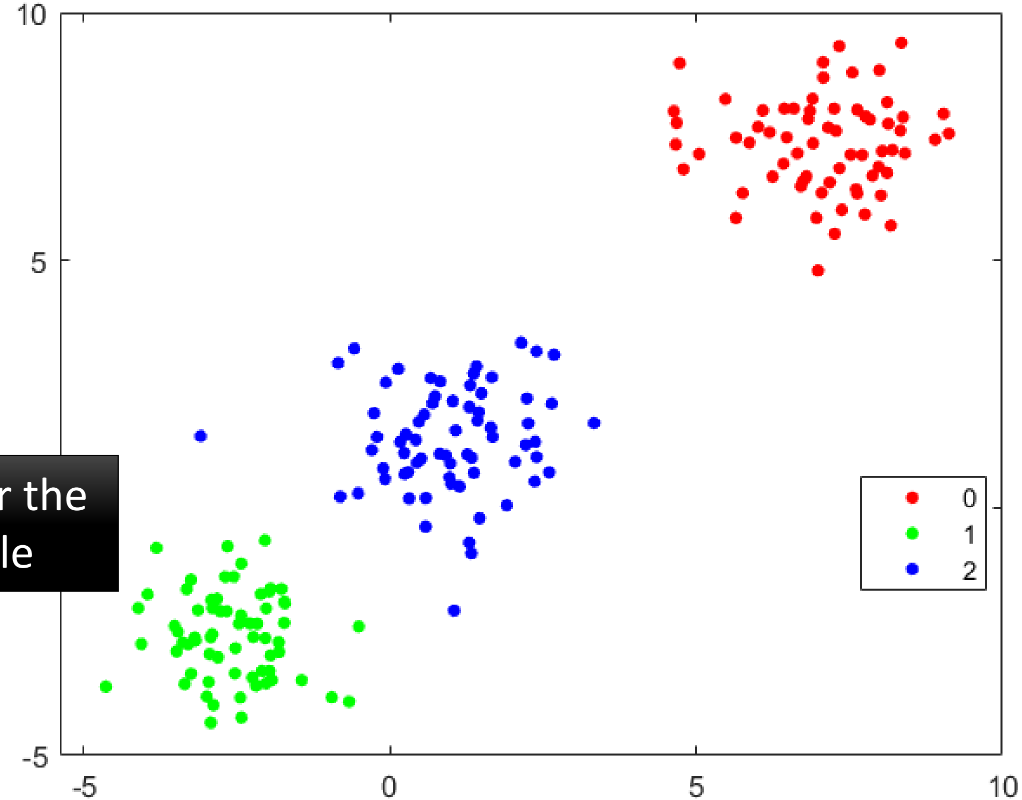
200 samples $\begin{bmatrix} 1.2 & 3.3 \\ \vdots & \vdots \\ \vdots & \vdots \\ 4.5 & -7.1 \end{bmatrix}$

$X \in R^{200\times 2}$

200 samples $\begin{bmatrix} 1 \\ \boxed{2} \\ \vdots \\ 0 \end{bmatrix}$

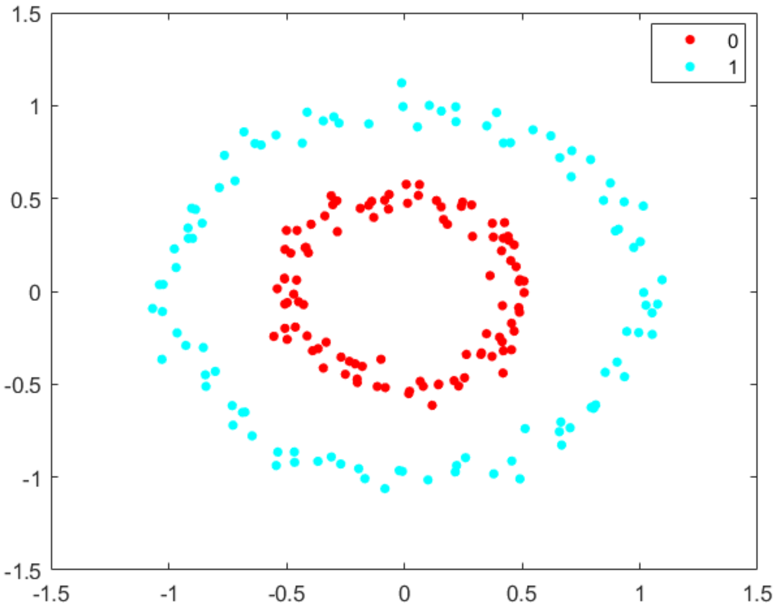
True Label for the second sample

$y \in R^{200\times 1}$

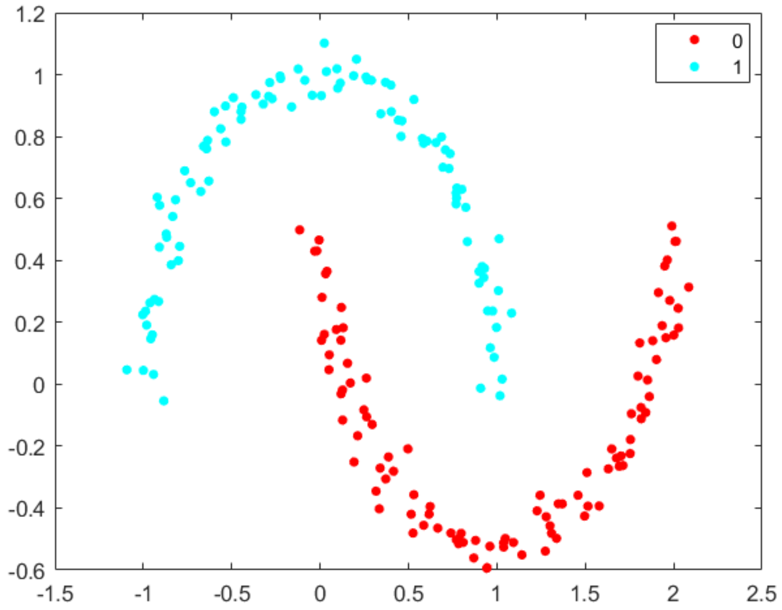


# Toy datasets

make_circles




make_moons



# Real datasets: seeds

To construct the data, seven geometric parameters of wheat kernels were measured:

1. area A,
2. perimeter P,
3. compactness C = 4*pi*A/P^2,
4. length of kernel,
5. width of kernel,
6. asymmetry coefficient
7. length of kernel groove.

All of these parameters were real-valued continuous.

| | | | | | | | label |
|---|---|---|---|---|---|---|---|
| 15.26 | 14.84 | 0.871 | 5.763 | 3.312 | 2.221 | 5.22 | 1 |
| 14.88 | 14.57 | 0.8811 | 5.554 | 3.333 | 1.018 | 4.956 | 1 |
| 14.29 | 14.09 | 0.905 | 5.291 | 3.337 | 2.699 | 4.825 | 1 |
| 13.84 | 13.94 | 0.8955 | 5.324 | 3.379 | 2.259 | 4.805 | 1 |
| 16.14 | 14.99 | 0.9034 | 5.658 | 3.562 | 1.355 | 5.175 | 1 |
| 14.38 | 14.21 | 0.8951 | 5.386 | 3.312 | 2.462 | 4.956 | 1 |
| 14.69 | 14.49 | 0.8799 | 5.563 | 3.259 | 3.586 | 5.219 | 1 |

# Real datasets: SMS Spam Collection

The collection is composed by just one text file, where each line has the correct class followed by the raw message.

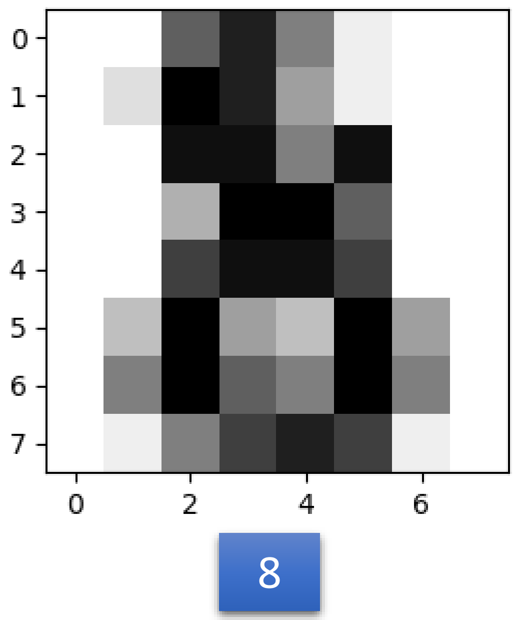
***Some examples:***

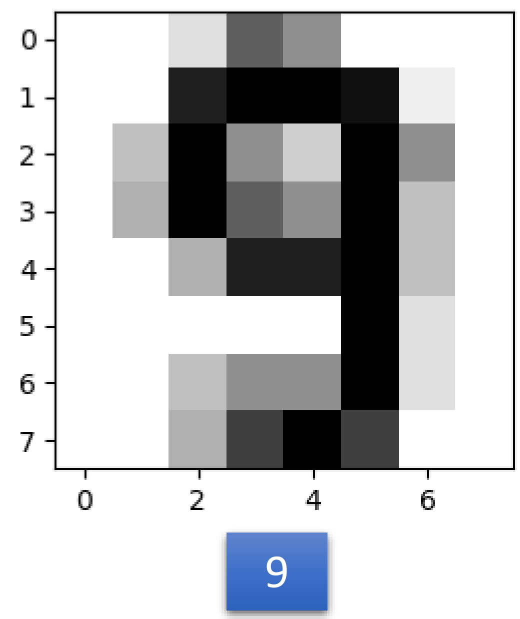
**label**

ham What you doing?how are you?
ham Ok lar... Joking wif u oni...
ham dun say so early hor... U c already then say...
ham MY NO. IN LUTON 0125698789 RING ME IF UR AROUND! H*
ham Siva is in hostel aha:-.
ham Cos i was out shopping wif darren jus now n i called him 2 ask wat present he wan lor. Then he started guessing who i was wif n he finally guessed darren lor.
spam FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk time to use from your phone now! ubscribe6GBP/ mnth inc 3hrs 16 stop?txtStop
spam Sunshine Quiz! Win a super Sony DVD recorder if you canname the capital of Australia? Text MQUIZ to 82277. B
spam URGENT! Your Mobile No 07808726822 was awarded a L2,000 Bonus Caller Prize on 02/09/03! This is our 2nd attempt to contact YOU! Call 0871-872-9758 BOX95QU

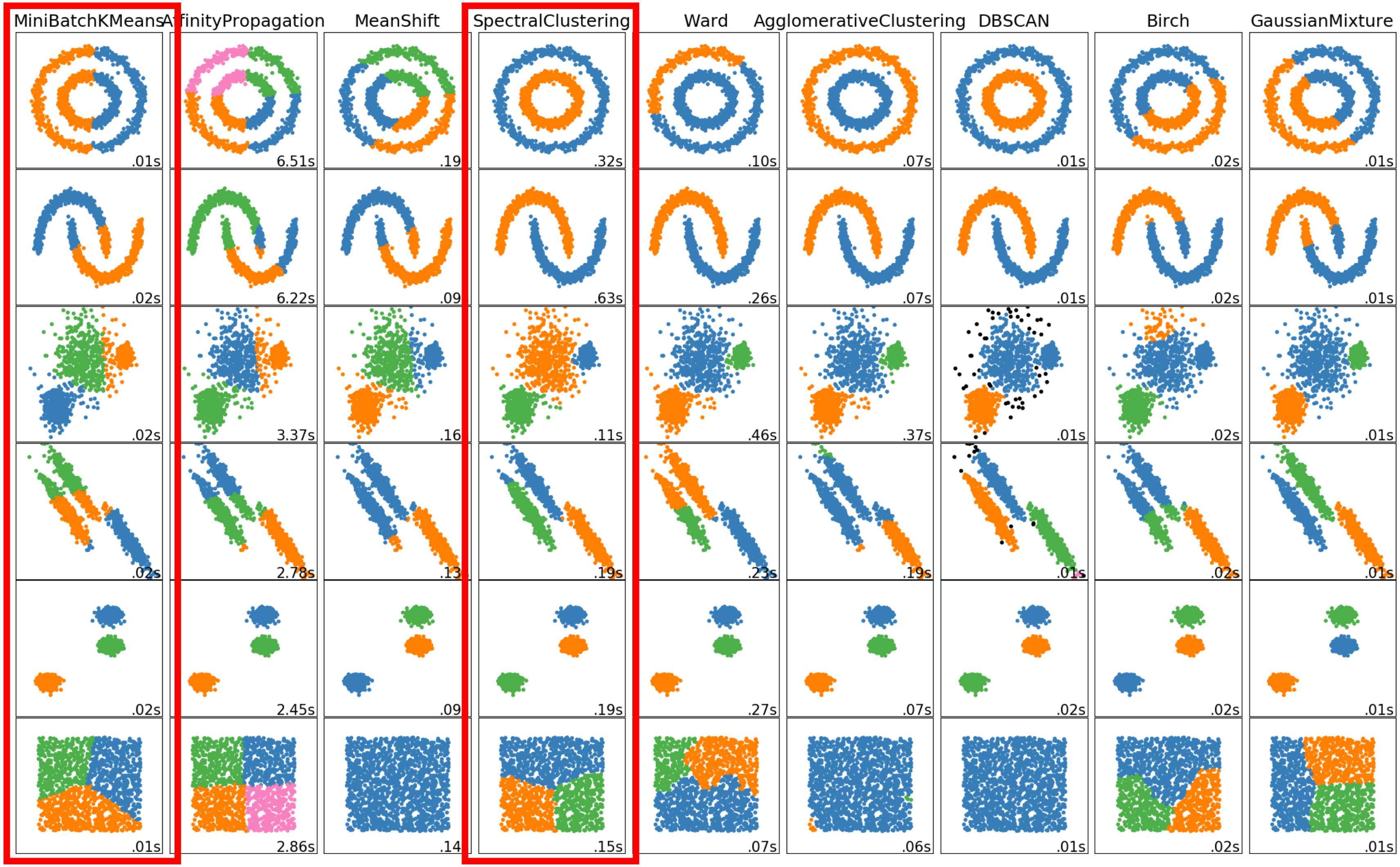
# Real datasets: Digit Dataset

This dataset is made up of 1797 8x8 images. Each image, like the one shown below, is of a hand-written digit. In order to utilize an 8x8 figure like this, we'd have to first transform it into a feature vector with length 64.



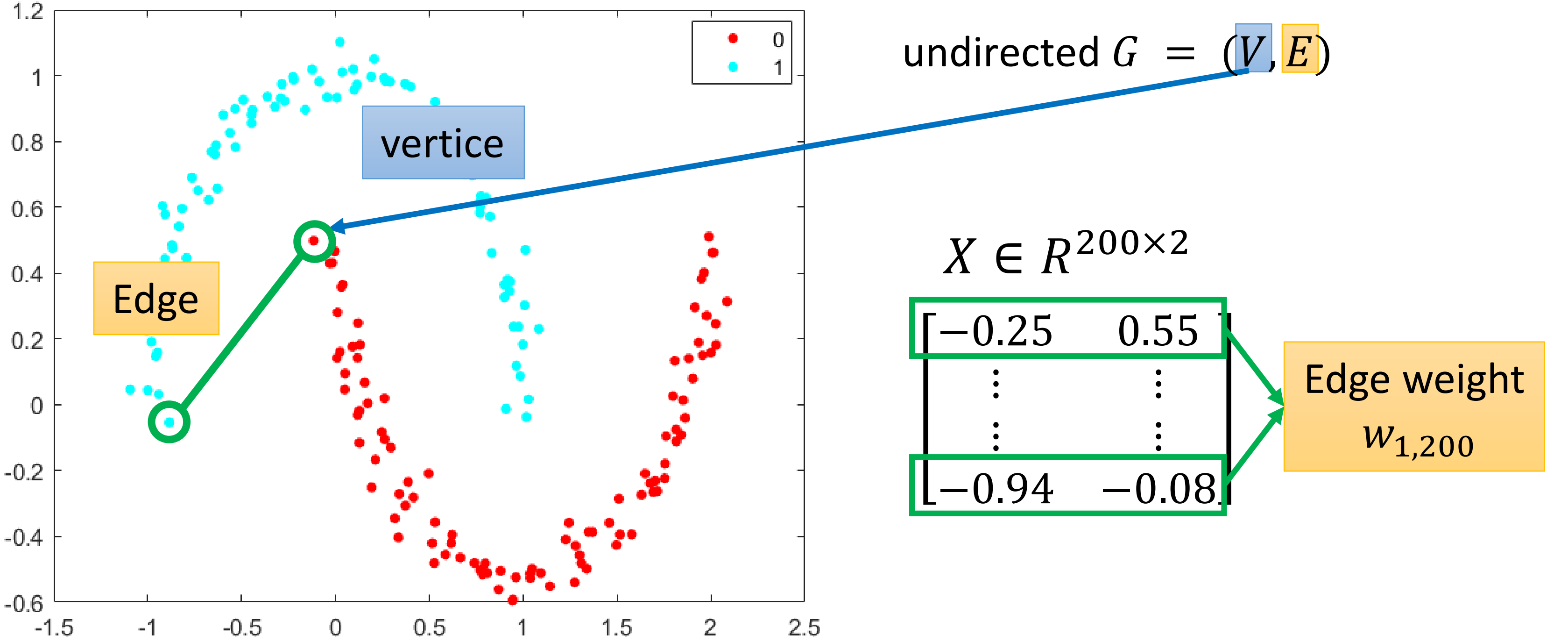



# Spectral Clustering




https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-comparison-py

# Spectral Clustering




We assume that the graph G is weighted, that is each edge between two vertices $v_i$ and $v_j$ carries a non-negative weight $w_{ij} \geq 0$. If $w_{ij} = 0$ this means that the vertices $v_i$ and $v_j$ are not connected. As G is undirected we require $w_{ij} = w_{ji}$.

# Spectral Clustering

The weighted adjacency matrix of the graph is the matrix $W = (w_{ij})_{i,j=1,\dots,n}$.

$X \in R^{200\times 2}$

$$\begin{bmatrix} \boxed{-0.25 \quad 0.55} \\ \vdots \quad\quad \vdots \\ \vdots \quad\quad \vdots \\ \boxed{-0.94 \quad -0.08} \end{bmatrix}$$

Edge weight

$w_{1,200}$

$$\begin{matrix} w_{1,1} & w_{1,2} & \dots & w_{1,200} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ w_{199,1} & w_{199,2} & \dots & w_{199,200} \\ w_{200,1} & w_{200,2} & \dots & w_{200,200} \end{matrix}$$

$W \in R^{200\times 200}$

$$\begin{matrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & d_{200} \end{matrix}$$

$D \in R^{200\times 200}$

$$d_i = \sum_{j=1}^{n} w_{i,j}$$

# Fullyconnected

$X \in R^{200\times 2}$

$$\begin{bmatrix} \boxed{-0.25 \quad 0.55} \\ \vdots \quad \vdots \\ \vdots \quad \vdots \\ \boxed{-0.94 \quad -0.08} \end{bmatrix}$$

Edge weight $w_{1,200}$

- rbf-kernel $K(x,y) = exp(-gamma * ||x-y||\text{^}2)$

- laplacian_kernel $K(x,y) = exp(-gamma * ||x-y||_1)$

# Nearest neighbor

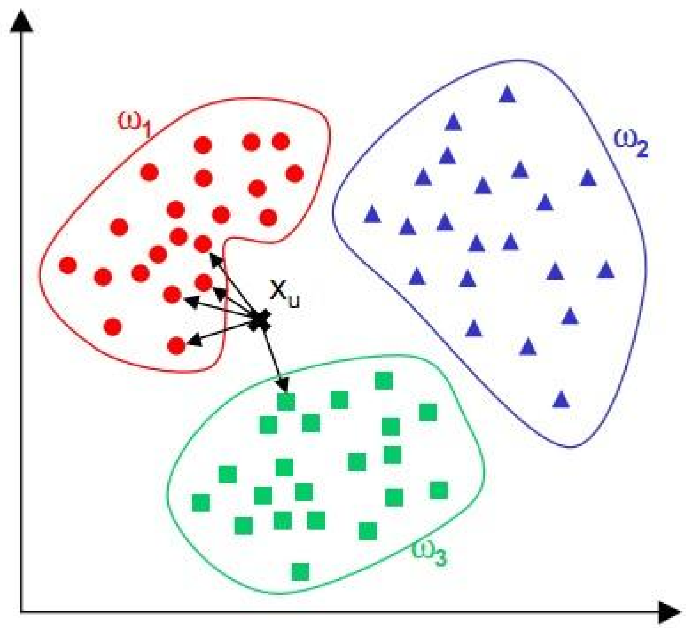
$$\begin{array}{cc} & 200 \\ 200 & \begin{bmatrix} 0 & 3.2 & \cdots & 6 & 1.3 \\ 3.2 & \ddots & 7.9 & 4.9 & 4.5 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 6 & 4.9 & 2.9 & \ddots & 6.7 \\ 1.3 & 4.5 & \cdots & 6.7 & 0 \end{bmatrix} \end{array}$$

Symmetric matrix

K=1

$$\begin{bmatrix} 0 & 0 & \cdots & 0 & 1.3 \\ 3.2 & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 2.9 & \ddots & 0 \\ 1.3 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Asymmetric matrix?



# Nearest_neighbor

$$\begin{array}{cc} & 200 \\ 200 & \begin{bmatrix} 0 & 3.2 & \cdots & 6 & 1.3 \\ 3.2 & \ddots & 7.9 & 4.9 & 4.5 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 6 & 4.9 & 2.9 & \ddots & 6.7 \\ 1.3 & 4.5 & \cdots & 6.7 & 0 \end{bmatrix} \end{array}$$

Symmetric matrix

$$\Big\downarrow \text{K=1}$$

$$\begin{bmatrix} 0 & 0 & \cdots & 0 & 1.3 \\ 3.2 & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 2.9 & \ddots & 0 \\ 1.3 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

***Mutual***

$$\begin{bmatrix} 0 & 0 & \cdots & 0 & 1.3 \\ 0 & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & 0 \\ 1.3 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

***Non-mutual***

$$\begin{bmatrix} 0 & 0 & \cdots & 0 & 1.3 \\ 3.2 & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 2.9 & \ddots & 0 \\ 1.3 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

# Nearest_neighbor

$$\begin{array}{cc} & 200 \\ 200 & \begin{bmatrix} 0 & 3.2 & \cdots & 6 & 1.3 \\ 3.2 & \ddots & 7.9 & 4.9 & 4.5 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 6 & 4.9 & 2.9 & \ddots & 6.7 \\ 1.3 & 4.5 & \cdots & 6.7 & 0 \end{bmatrix} \end{array}$$

Symmetric matrix

How to measure distance?

Cosine distances
Euclidean distances
Manhattan distances

***<u>Cosine distances</u>***

$$k(x, y) = 1 - \frac{\langle x, y \rangle}{\|x\| \times \|y\|}$$

***<u>Manhattan distances</u>***

Compute the L1 distances between the vectors in X and Y.

# Algorithm

$L = D - W$

**Normalized spectral clustering according to Shi and Malik (2000)**

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number $k$ of clusters to construct

- Construct a similarity graph by one of the ways described in Section 2. Let $W$ be its weighted adjacency matrix.
- Compute the unnormalized Laplacian $L$.
- **Compute the first $k$ eigenvectors $v_1, \dots, v_k$ of the generalized eigenproblem $Lv = \lambda Dv$.**
- Let $\boxed{V \in \mathbb{R}^{n \times k}}$ be the matrix containing the vectors $v_1, \dots, v_k$ as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the $i$-th row of $V$.
- Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the $k$-means algorithm into clusters $C_1, \dots, C_k$.

Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

$L_{rw} = D^{-1} L$

# Evaluation

## ***ARI***

Given a set $S$ of $n$ elements, and two groupings or partitions (*e.g.* clusterings) of these elements, namely $X = \{X_1, X_2, \ldots, X_r\}$ and $Y = \{Y_1, Y_2, \ldots, Y_s\}$, the overlap between $X$ and $Y$ can be summarized in a contingency table $[n_{ij}]$ where each entry $n_{ij}$ denotes the number of objects in common between $X_i$ and $Y_j$ : $n_{ij} = |X_i \cap Y_j|$.

| $X \diagdown Y$ | $Y_1$ | $Y_2$ | $\ldots$ | $Y_s$ | Sums |
|---|---|---|---|---|---|
| $X_1$ | $n_{11}$ | $n_{12}$ | $\ldots$ | $n_{1s}$ | $a_1$ |
| $X_2$ | $n_{21}$ | $n_{22}$ | $\ldots$ | $n_{2s}$ | $a_2$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $X_r$ | $n_{r1}$ | $n_{r2}$ | $\ldots$ | $n_{rs}$ | $a_r$ |
| Sums | $b_1$ | $b_2$ | $\ldots$ | $b_s$ | |

### Definition [ edit ]

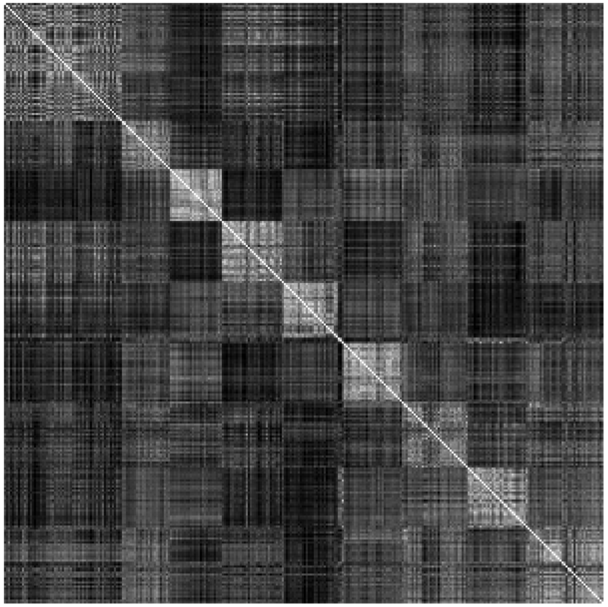
The original Adjusted Rand Index using the Permutation Model is

$$\overbrace{ARI}^{\text{Adjusted Index}} = \frac{\overbrace{\sum_{ij}\binom{n_{ij}}{2}}^{\text{Index}} - \overbrace{[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}^{\text{Expected Index}}}{\underbrace{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}]}_{\text{Max Index}} - \underbrace{[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}_{\text{Expected Index}}}$$

where $n_{ij}, a_i, b_j$ are values from the contingency table.

# Evaluation

***Similarity Matrix***



# Evaluation

## *轮廓系数*

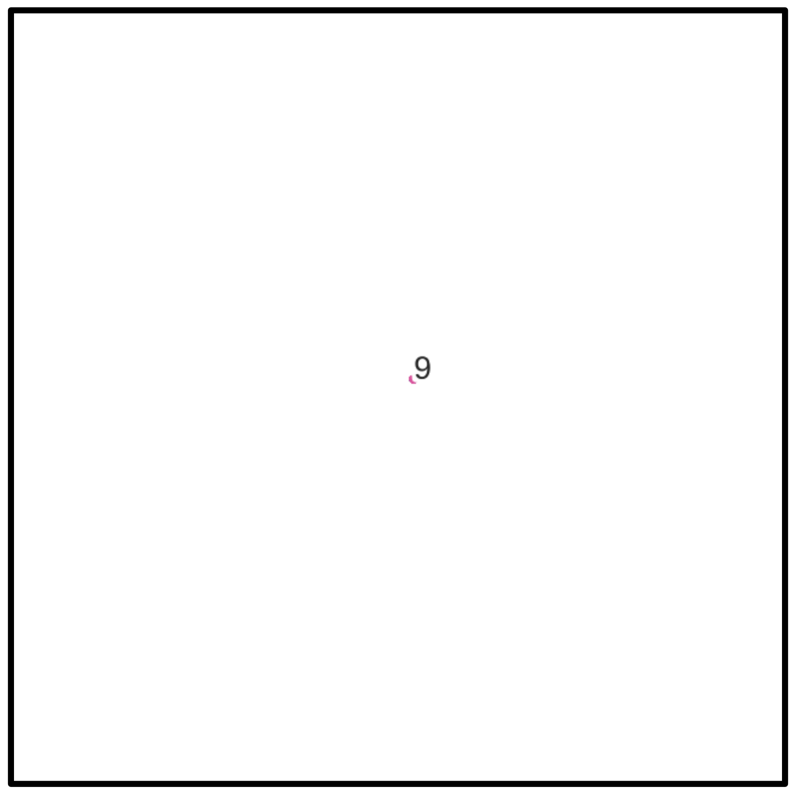
***轮廓系数*** 这个评价主要是考虑到有时我们手中并没有真实的类别归属，这时也需要有一种方法来度量聚类的性能。轮廓系数同时兼顾了凝聚度和分离度，用于评估聚类的效果并且取值范围为[-1, 1]。轮廓系数越大表示聚类效果越好。具体的计算步骤如下：1）对已聚类数据中第i个样本$x^i$，计算其与同一个类簇内的所有其他样本距离的平均值，记为$a^i$，用于量化簇群的凝聚度；2）计算$x^i$与其他簇群最近的平均距离$b^i$；3）对于每一个样本可以计算它的轮廓系数为

$$sc^i = \frac{b^i - a^i}{\max(b^i, a^i)} \tag{8}$$

4）对所有样本的轮廓系数取均值为当前聚类结果的整体轮廓系数。

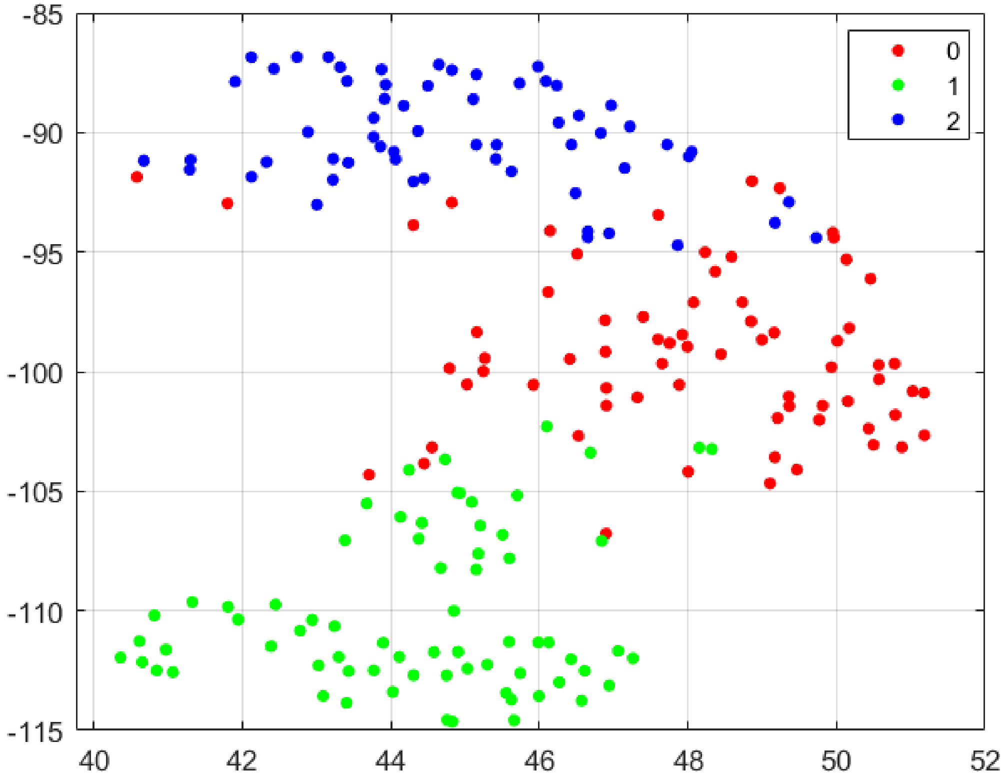
# Seeds dataset

- We use t-Distributed Stochastic Neighbor Embedding (t-SNE) based on Euclidean distances to visualize high dimensional data.

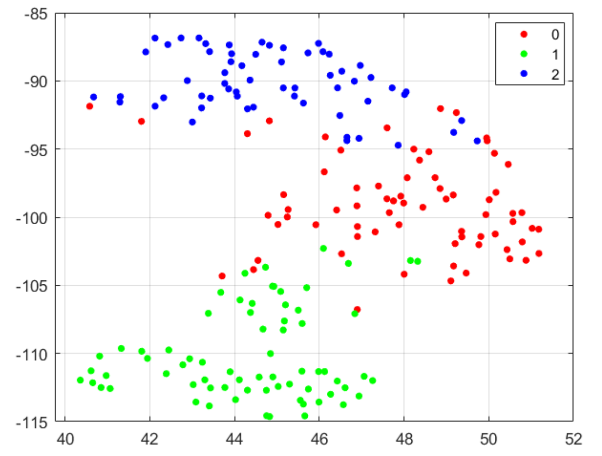


# Seeds dataset

- We use t-Distributed Stochastic Neighbor Embedding (t-SNE) based on Euclidean distances to visualize high dimensional data.



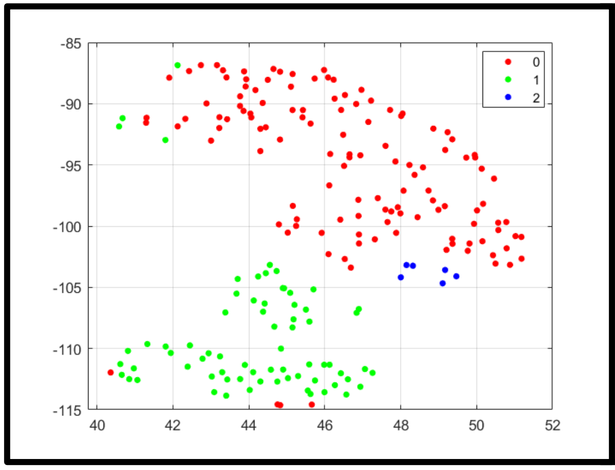
# Seeds dataset - Fullyconnected

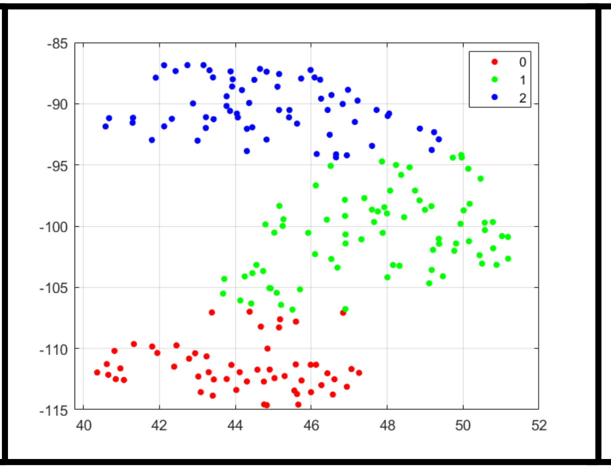


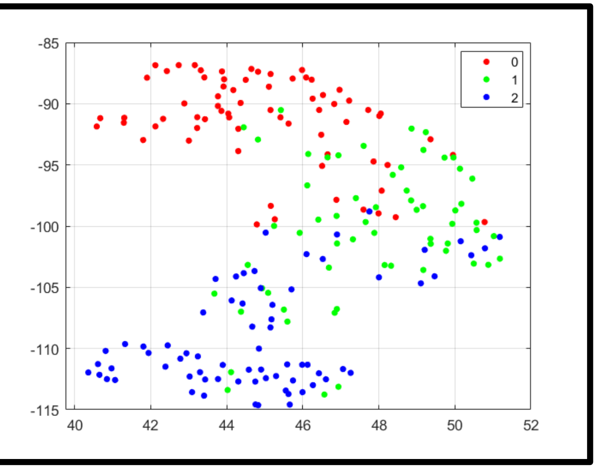

***RBF kernel***

$$K(x, y) = \exp\left(\frac{\|x - y\|^2}{-2\sigma^2}\right)$$

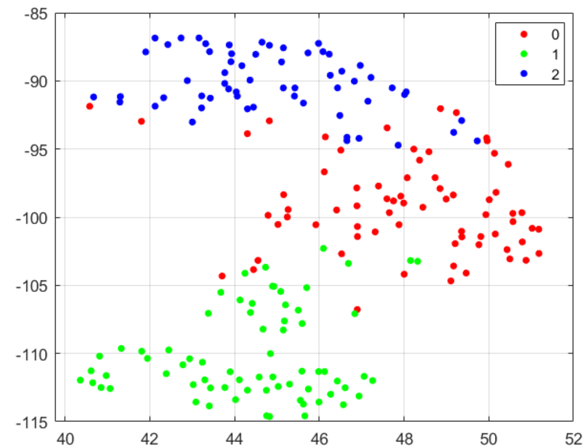
$\sigma = 0.1$ $\sigma = 1$ $\sigma = 10$



# Seeds dataset - Fullyconnected




***RBF kernel***

$$K(x,y) = \exp(\frac{\|x-y\|^2}{-2\sigma^2})$$

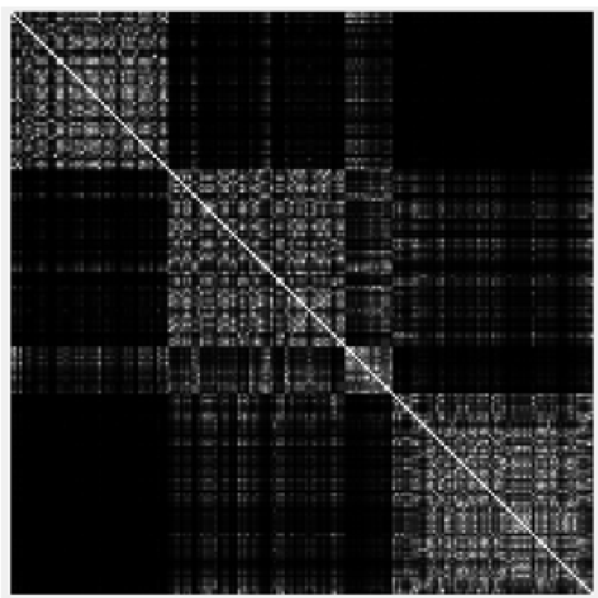
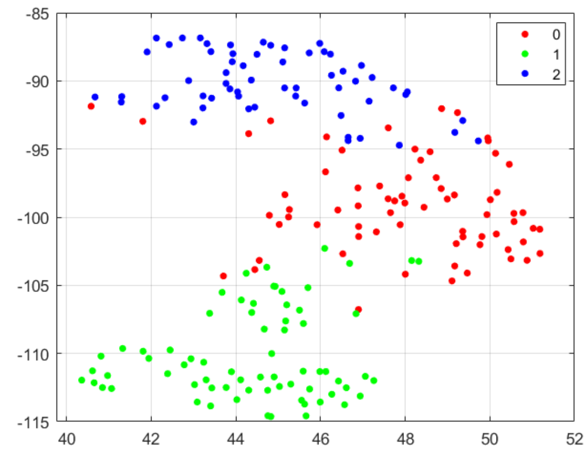


图 16 种子数据集拉普拉斯核$\sigma = 1$时的相似性矩阵

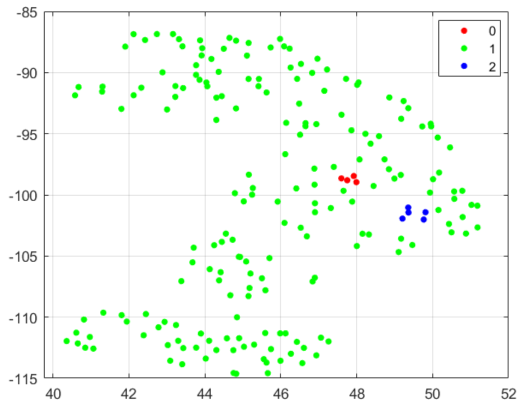
# Seeds dataset - Fullyconnected



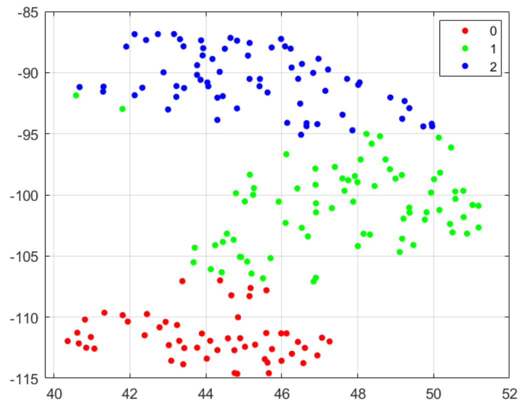

***Laplacian kernel***

$$K(x, y) = \exp\left(\frac{\|x - y\|_1}{-2\sigma^2}\right)$$

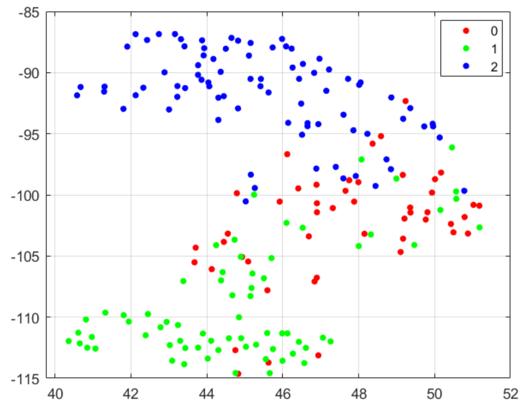
$\sigma = 0.1$



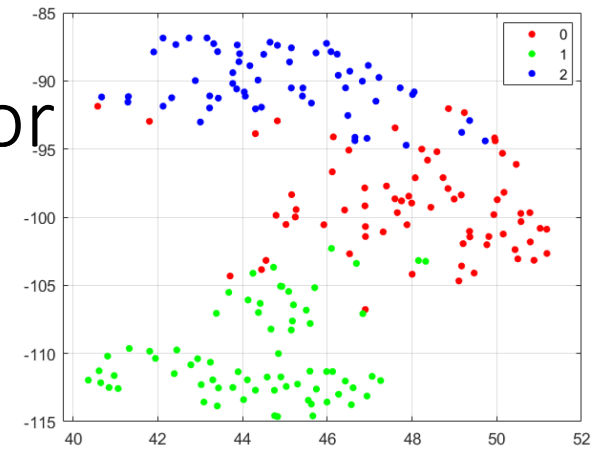

$\sigma = 1$



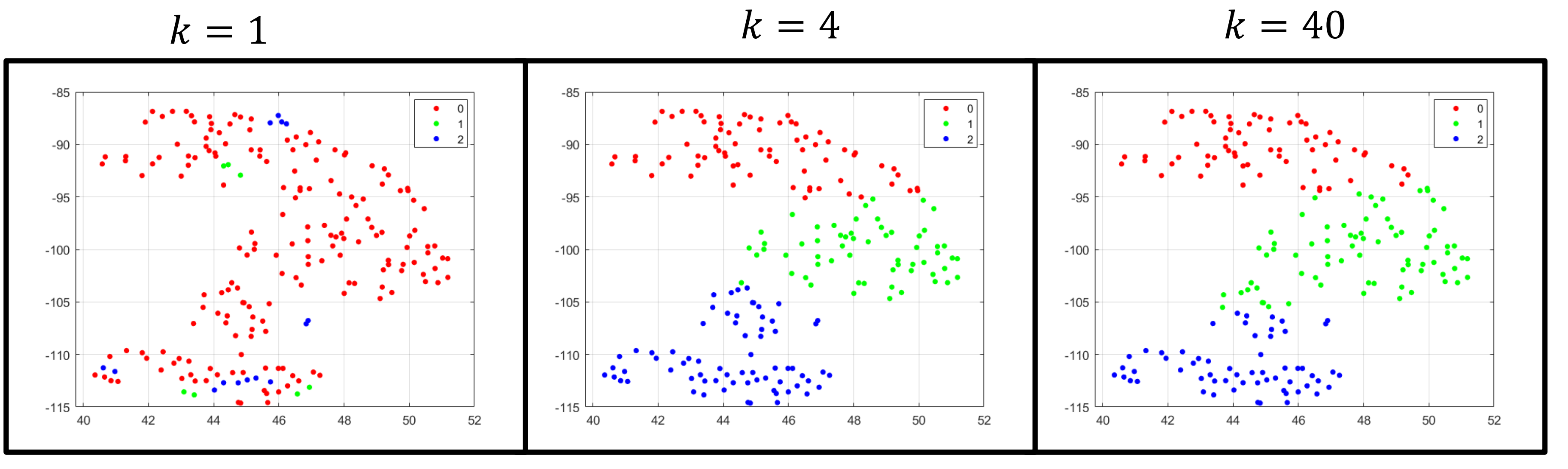

$\sigma = 10$



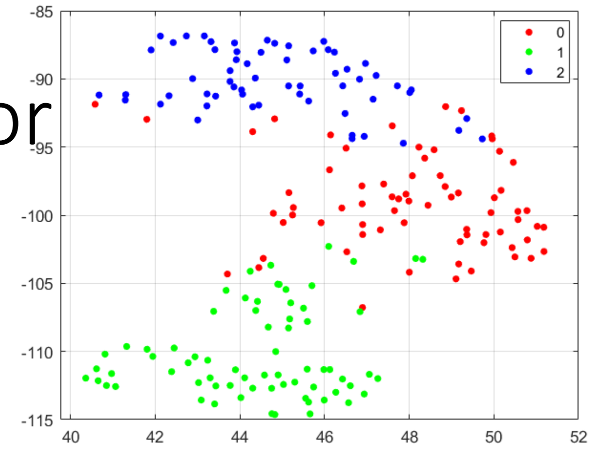
# Seeds dataset - Nearest neighbor




***Euclidean distances (non-mutual)***

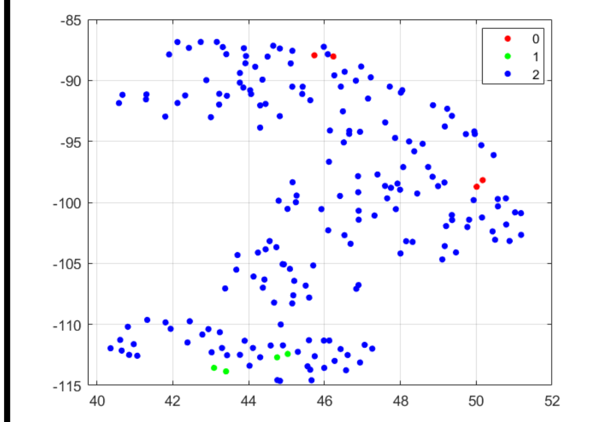


# Seeds dataset - Nearest neighbor

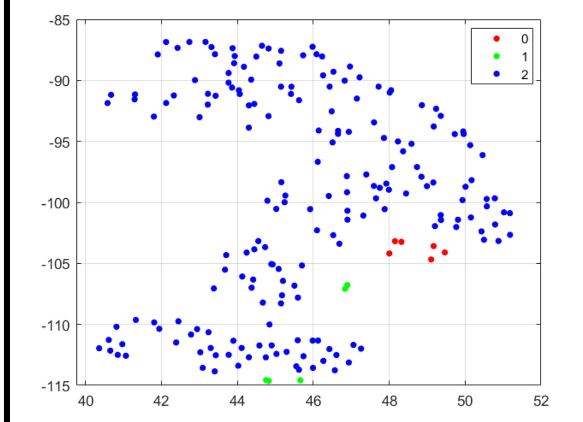



***Nearest neighbor:***
***Euclidean distances (mutual)***

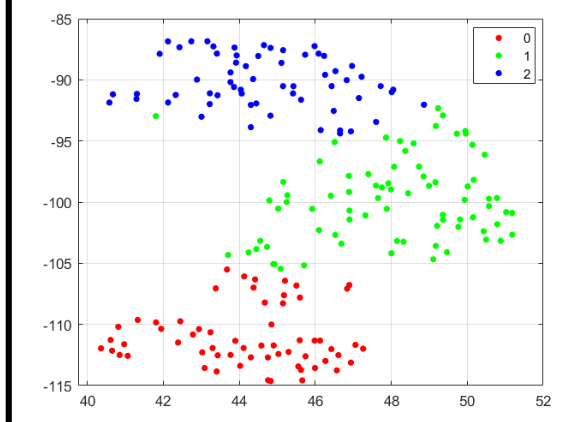
$k = 1$



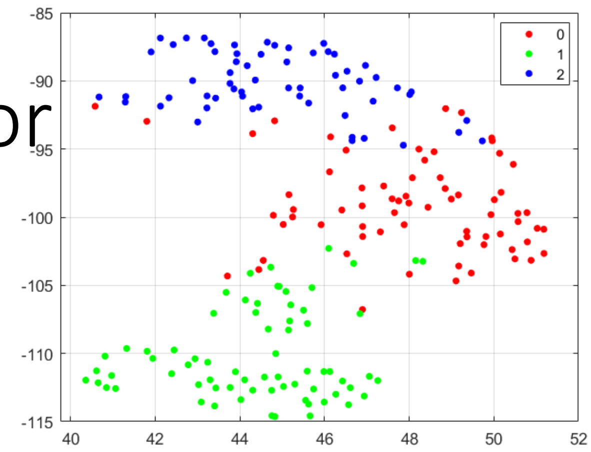

$k = 4$



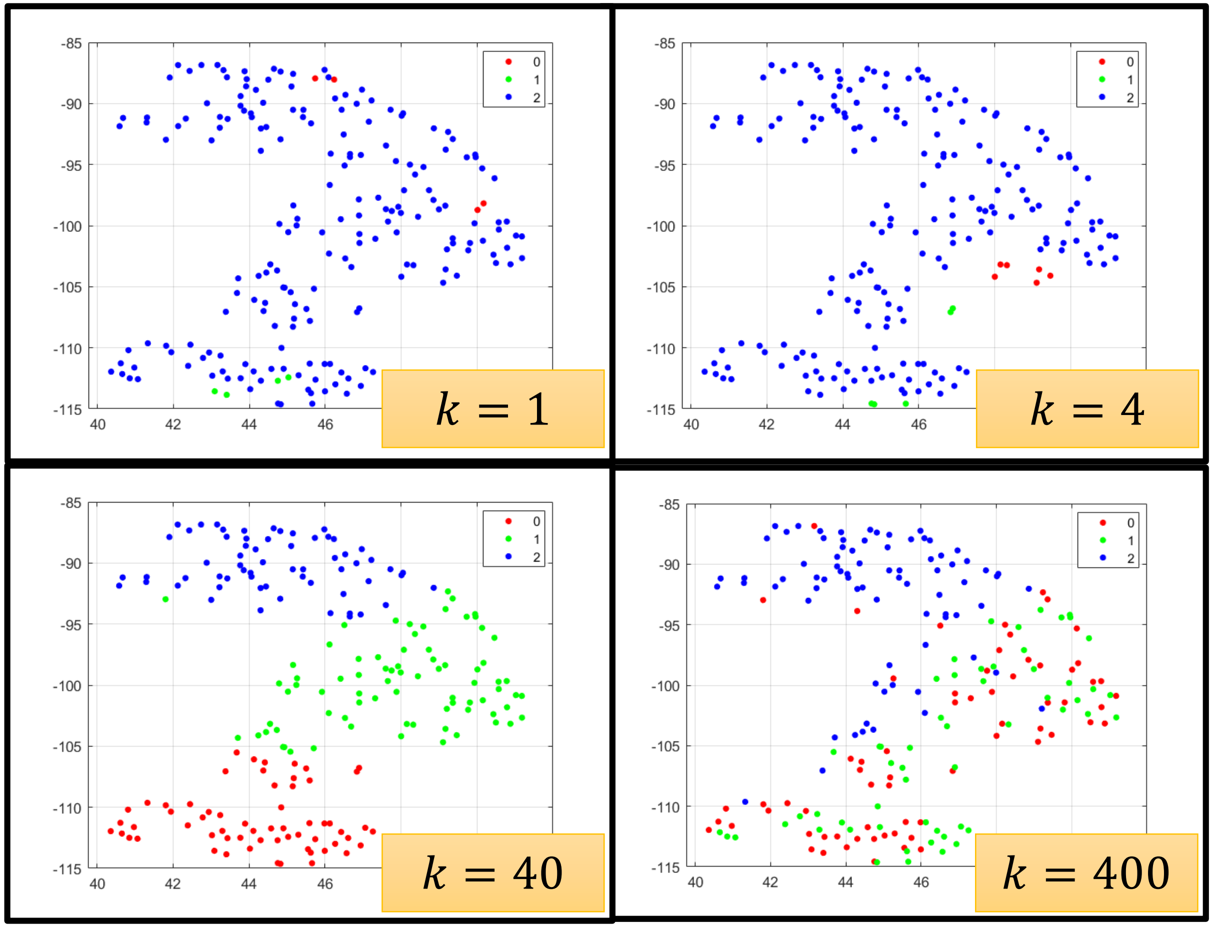

$k = 40$



# Seeds dataset - Nearest neighbor




***Nearest neighbor:***
***Cosine distances(non-mutual)***



# SMS Spam dataset: preprocessing

***Some examples:***

label

ham What you doing?how are you?
ham Ok lar... Joking wif u oni...
ham dun say so early hor... U c already then say...
ham MY NO. IN LUTON 0125698789 RING ME IF UR AROUND! H*
ham Siva is in hostel aha:-.
ham Cos i was out shopping wif darren jus now n i called him 2 ask wat present he wan lor. Then he started guessing who i was wif n he finally guessed darren lor.
spam FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk time to use from your phone now! ubscribe6GBP/ mnth inc 3hrs 16 stop?txtStop
spam Sunshine Quiz! Win a super Sony DVD recorder if you canname the capital of Australia? Text MQUIZ to 82277. B
spam URGENT! Your Mobile No 07808726822 was awarded a L2,000 Bonus Caller Prize on 02/09/03! This is our 2nd attempt to contact YOU! Call 0871-872-9758 BOX95QU

Convert text to feature vectors

Extract labels from text

***Processing label***

1. Use Regex to extract labels
2. Binary label

***Processing text***

1. Tokenize
2. Bag-of-word(TF-IDF method)
3. N-gram model
4. Word to vector
5. Document to vector

# SMS Spam dataset: preprocessing

***Processing text***

1. Tokenize
2. **Bag-of-word(TF-IDF method)**

Assign a fixed integer id to each word occurring in any document of the training set (for instance by building a dictionary from words to integer indices).For each document #i, count the number of occurrences of each word w and store it in X[i, j] as the value of feature #j where j is the index of word w in the dictionary.

What you doing?how are you? {What:0, you:1, doing?how:2, are:3 , ?:4 }

$[1 \quad 2 \quad 1 \quad 1 \quad 1]$ this number is typically larger than 100,000.

If n_samples == 10000, storing X as a NumPy array of type float32 would require 10000 x 100000 x 4 bytes = 4GB in RAM which is barely manageable on today's computers.

# SMS Spam dataset: preprocessing

***Processing text***

1. Tokenize
2. **Bag-of-word(TF-IDF method)**

Assign a fixed integer id to each word occurring in any document of the training set (for instance by building a dictionary from words to integer indices).For each document #i, count the number of occurrences of each word w and store it in X[i, j] as the value of feature #j where j is the index of word w in the dictionary.

What you doing?how are you?

{What:0, you:1, doing?how:2, are:3 , ?:4 }

$$[1 \quad 2 \quad 1 \quad 1 \quad 1]$$

What are you?

$$[1 \quad 1 \quad 0 \quad 1 \quad 1]$$

$$\mathrm{T}F - IDF(i,j) = \frac{n_{i,j}}{\sum_k n_{k,j}} lg \frac{|D|}{\left|\{j: t_i \in d_j\}\right|}$$

# SMS Spam dataset: preprocessing

***Processing text***

1. Tokenize
2. Bag-of-word(TF-IDF method)
3. **N-gram model**

What are you? → [1 1 1]

Bi-gram: {What are:0, are you:1, you?:2 }

## 7.2 DOCUMENT CLASSIFICATION: TOPIC CLASSIFICATION

In the Topic Classification task, we are given a document and need to classify it into one of a predefined set of topics (e.g., Economy, Politics, Sports, Leisure, Gossip, Lifestyle, Other).

Here, the letter level is not very informative, and our basic units will be words. Word order is not very informative for this task (except maybe for consecutive word pairs such as bigrams). Thus, a good set of features will be the *bag-of-words* in the document, perhaps accompanied by a *bag-of-word-bigrams* (each word and each word-bigram is a core feature).

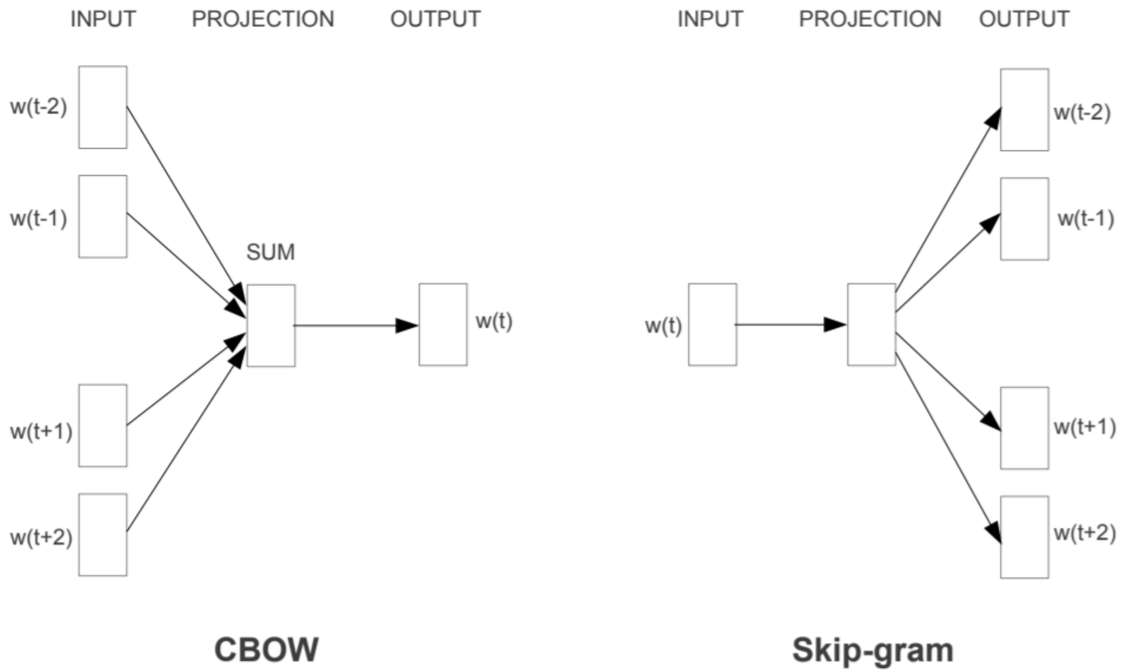
Yoav Goldberg. Neural Network Methods for Natural Language Processing

# SMS Spam dataset: preprocessing

***Processing text***

## 4. Word to vector

The word2vec algorithms include skip-gram and CBOW models.




flair

gensim

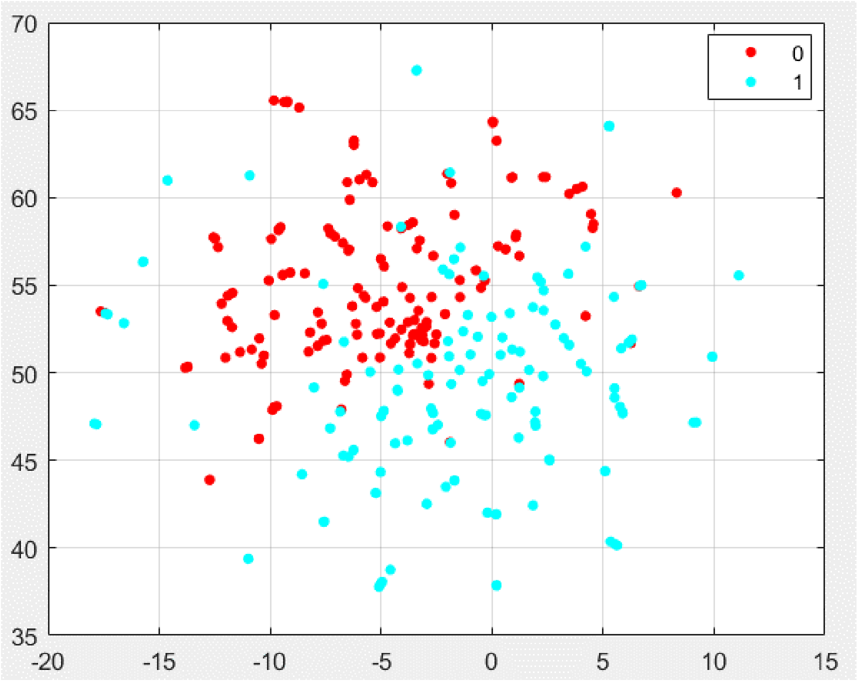
Sum over all word vectors within one sentence to represent the sentence

Tomas Mikolov et al. Efficient Estimation of Word Representations in Vector Space

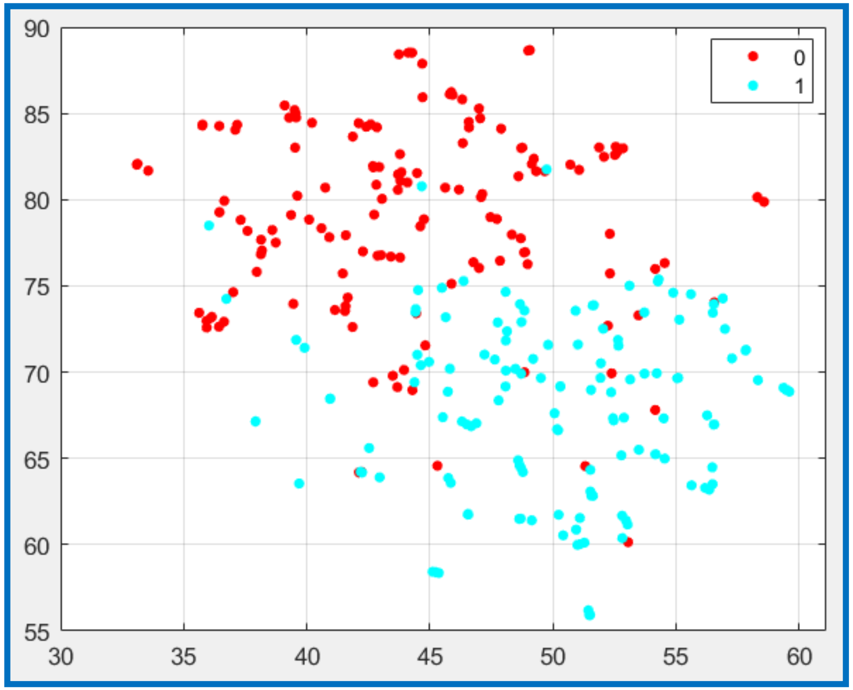
# SMS Spam dataset: visualization

***Bag-of-word*** NumPCAComponents=50

Euclidean distances



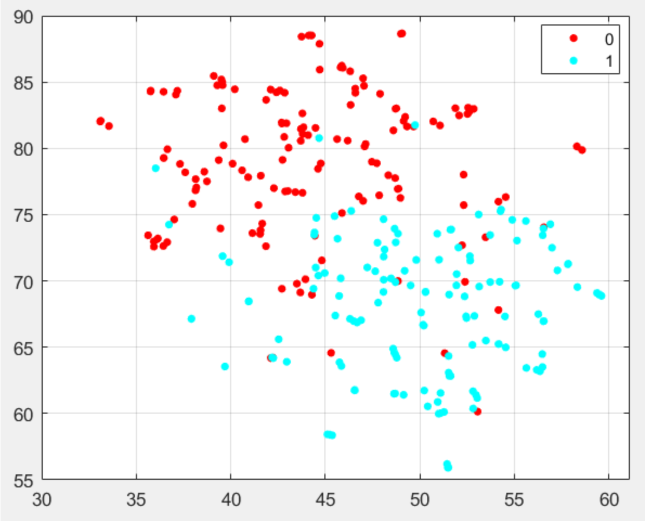

Cosine distances



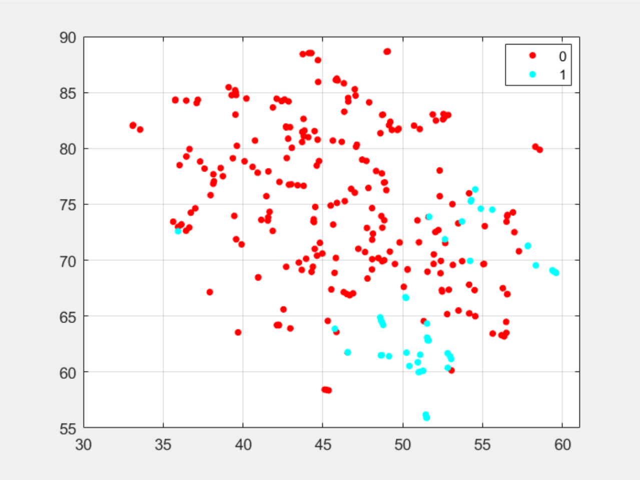
# SMS Spam dataset: clustering

***Bag-of-word***

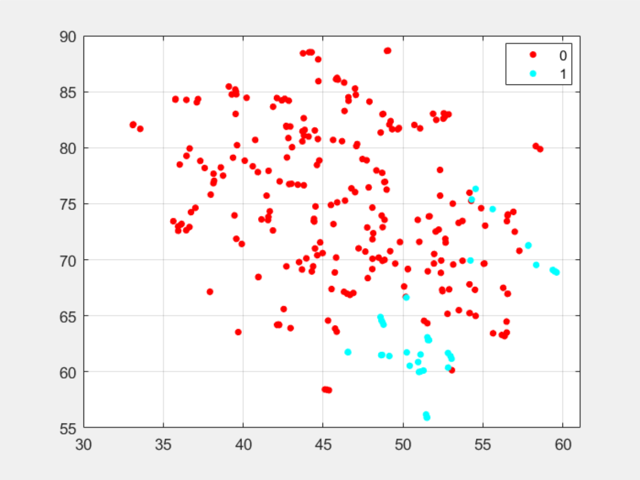
NumPCAComponents=50, Cosine distances



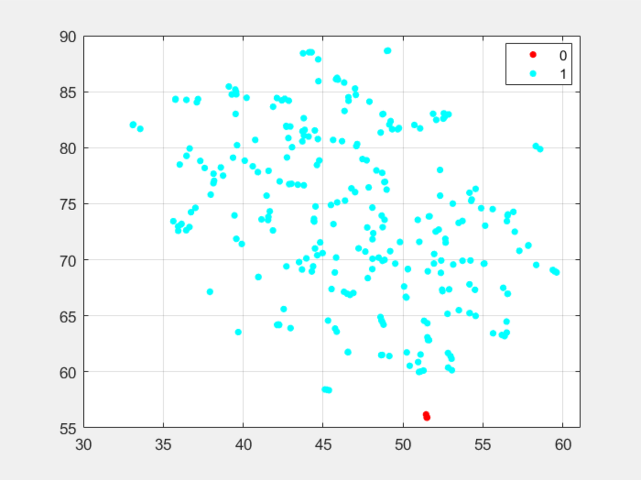

Euclidean distances




Cosine distances



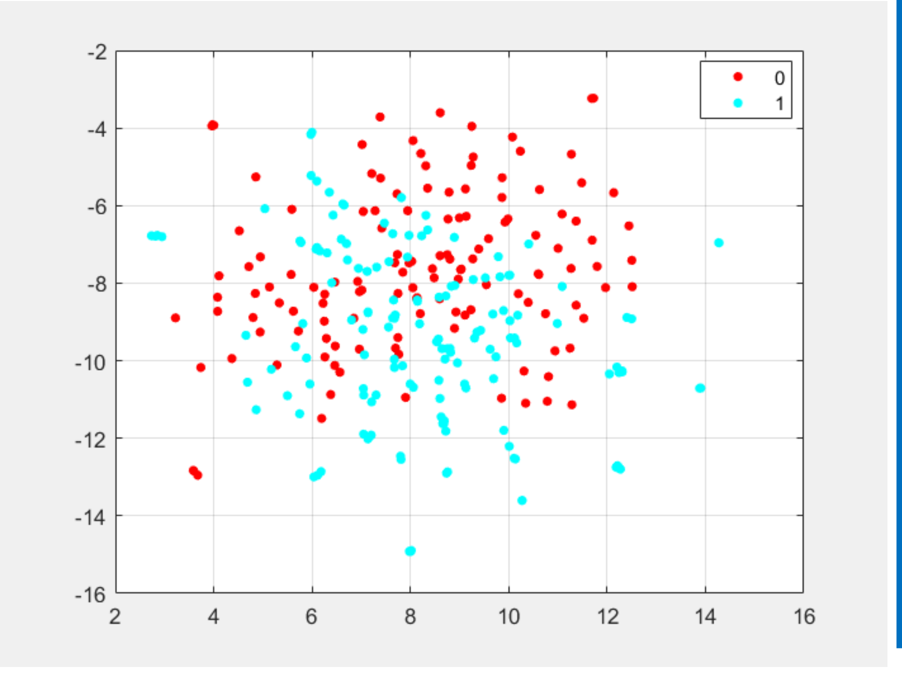

Manhattan distances



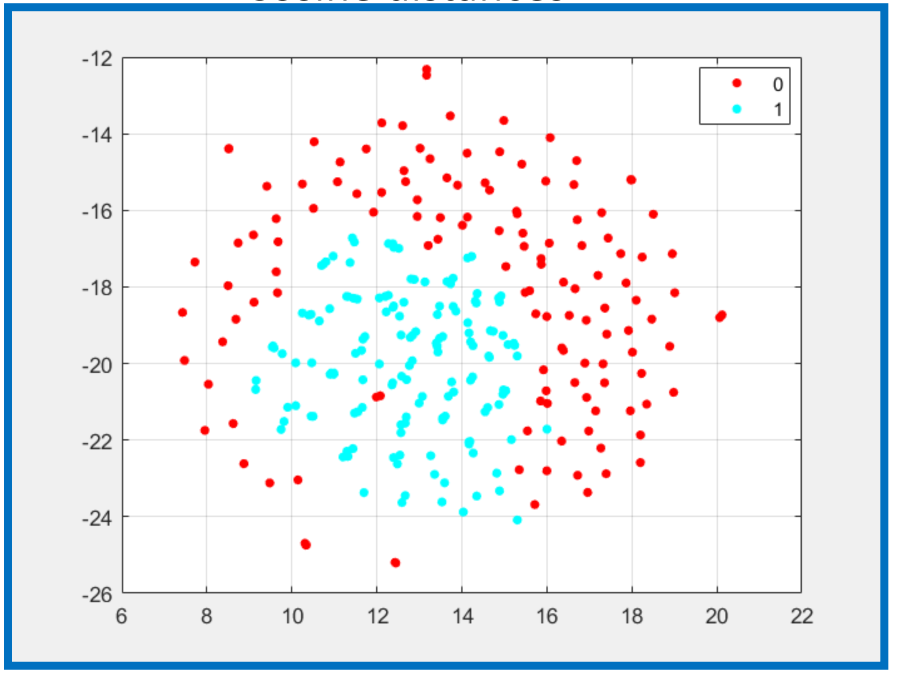
# SMS Spam dataset: visualization

***<u>Bag-of-word(bi-gram)</u>*** NumPCAComponents=2000

Euclidean distances



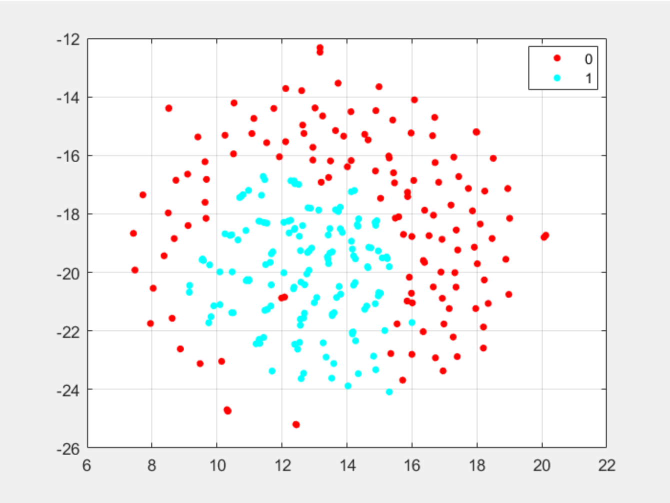

Cosine distances



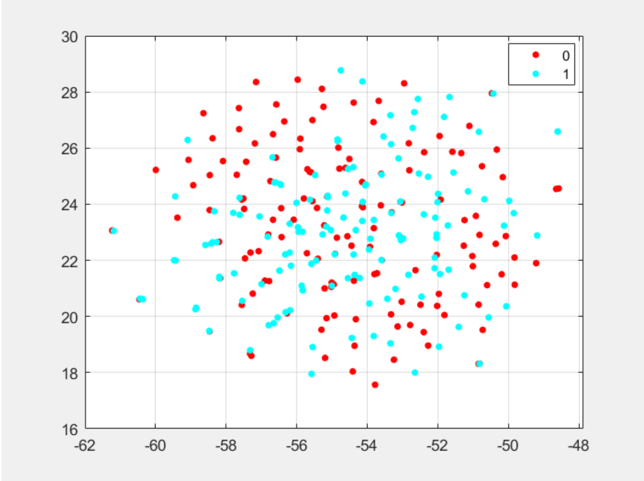
# SMS Spam dataset: clustering

***Bag-of-word(bi-gram)***

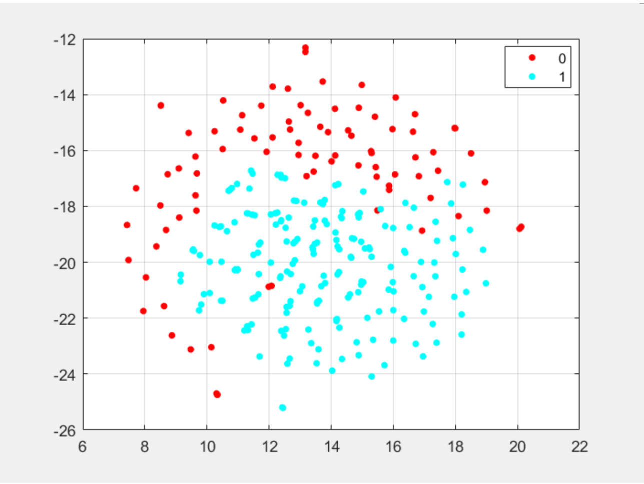
NumPCAComponents=2000_Cosine distances



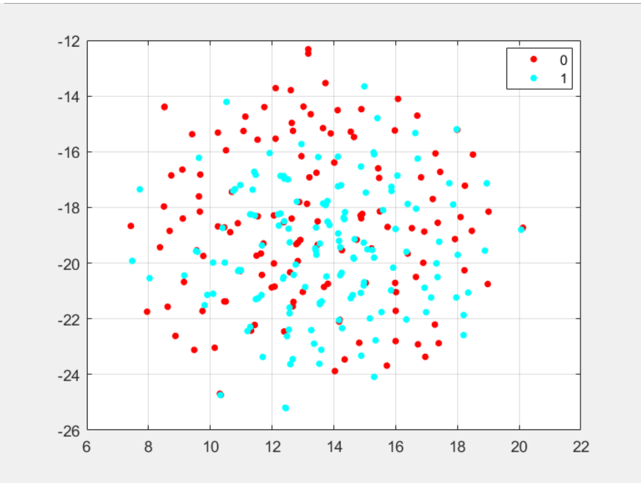

Euclidean distances




Cosine distances



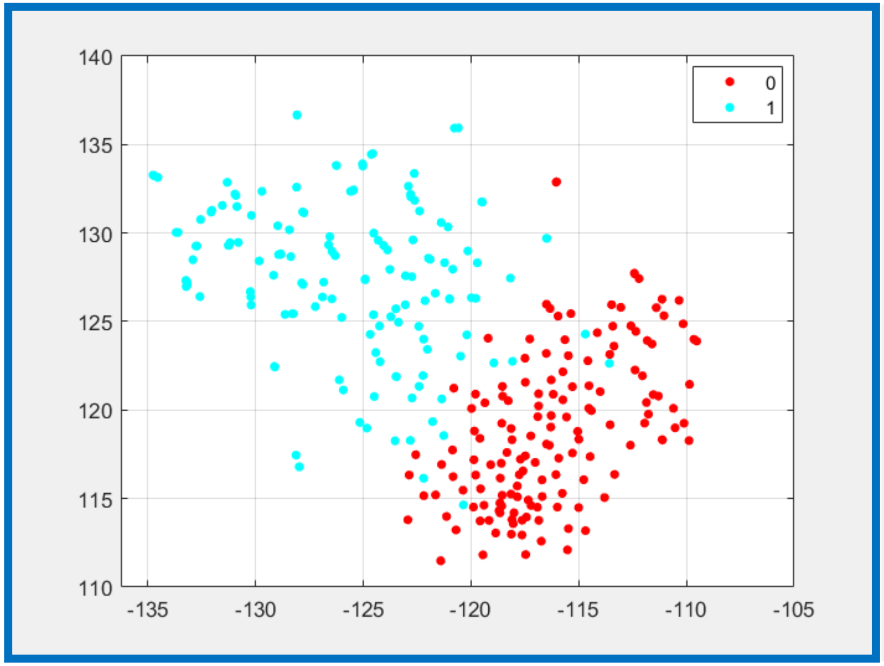

Manhattan distances



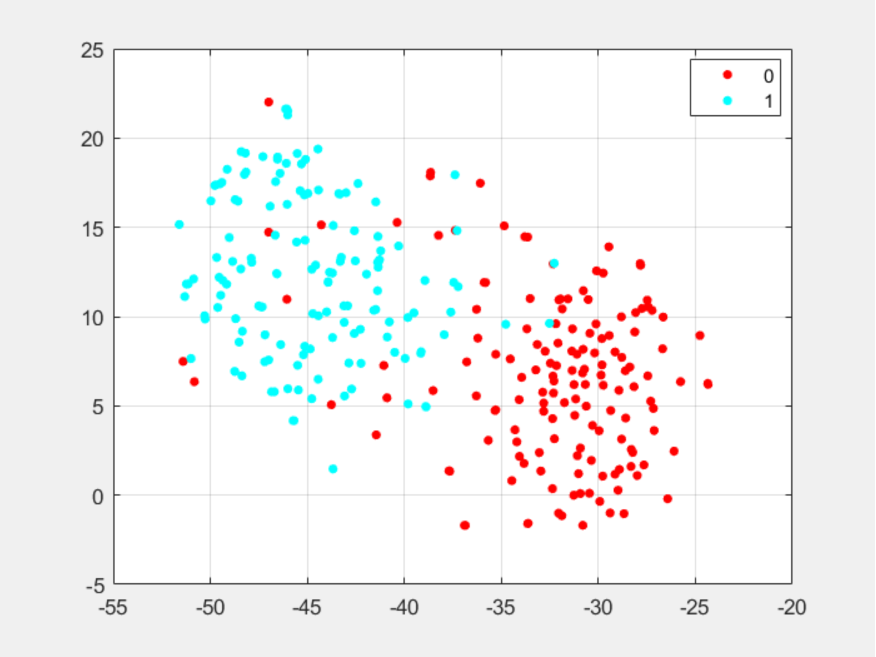
# SMS Spam dataset: visualization

***Word2vec*** NumPCAComponents=50

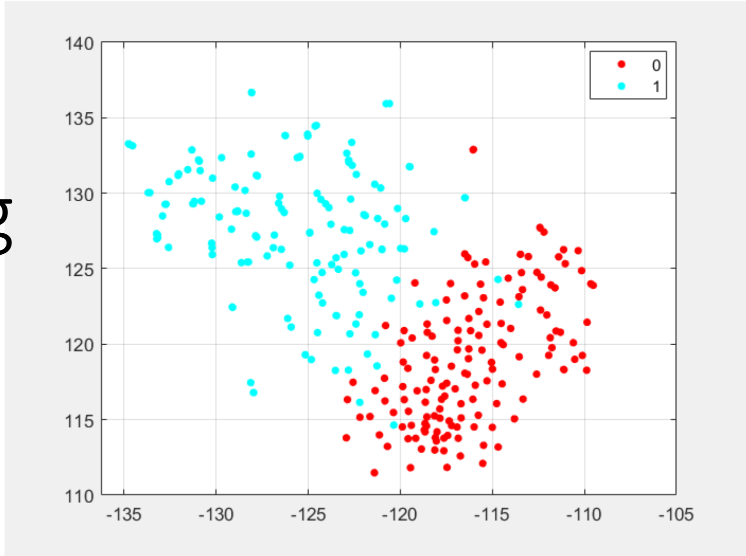
Euclidean distances




Cosine distances

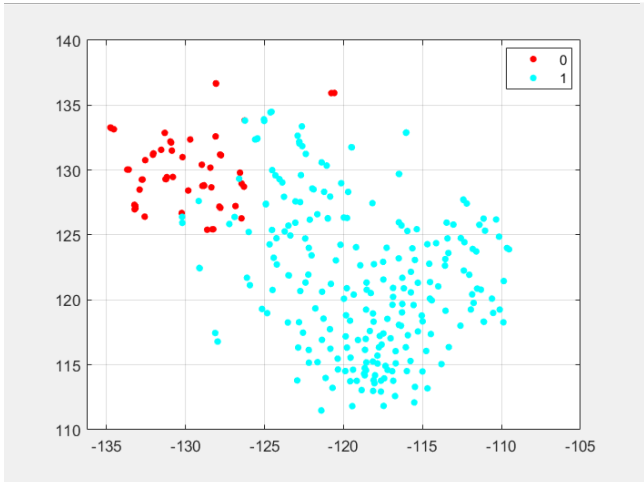


# SMS Spam dataset: clustering

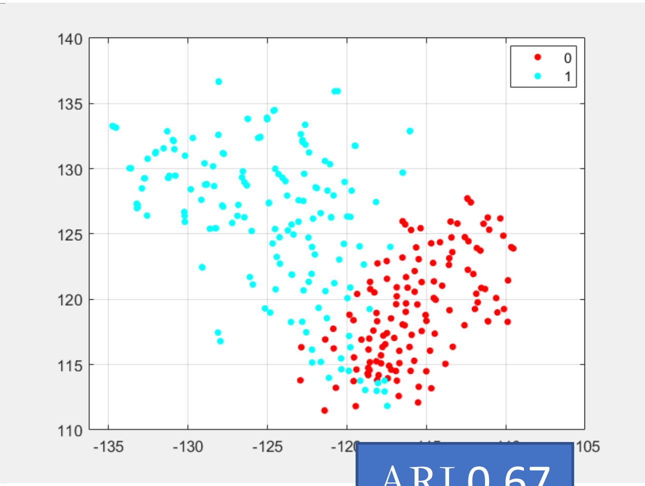



***Word2vec***
NumPCAComponents=50 Euclidean distances

Euclidean distances

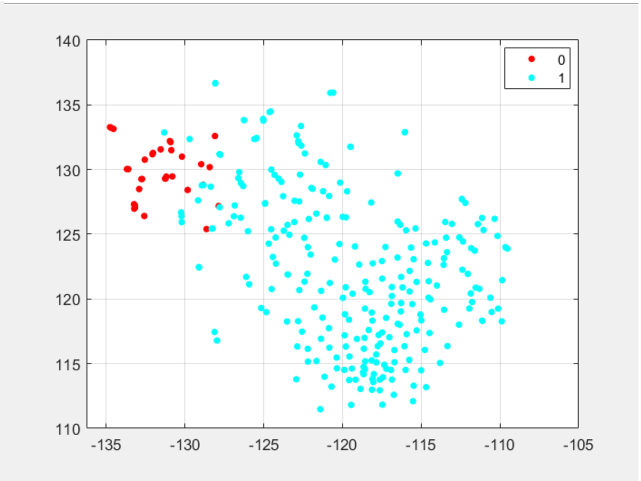



Cosine distances

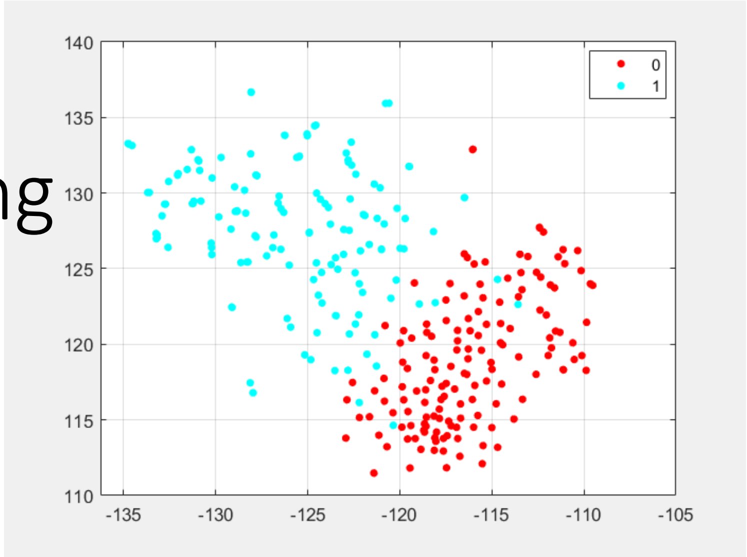



ARI 0.67

Manhattan distances



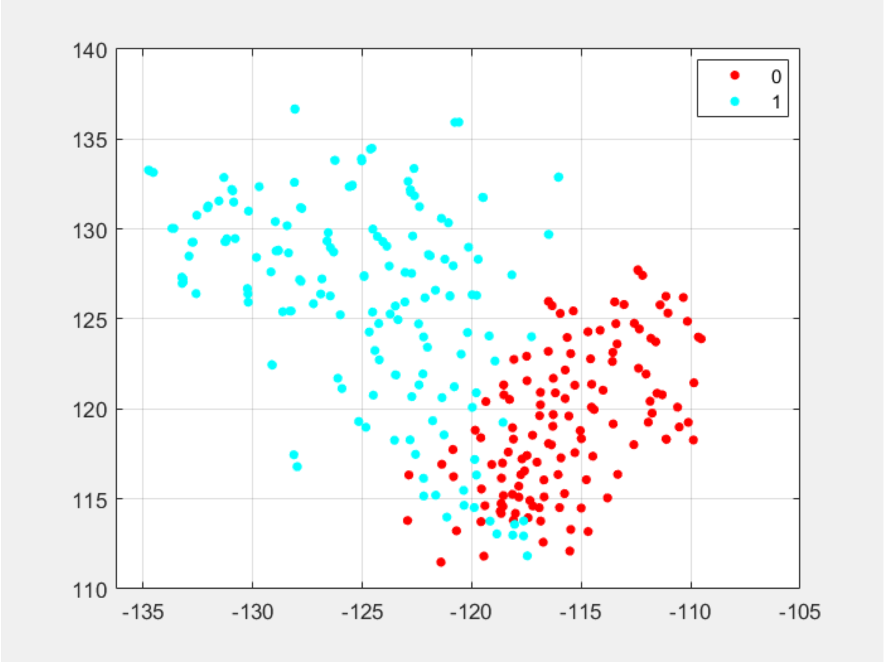
# SMS Spam dataset: clustering

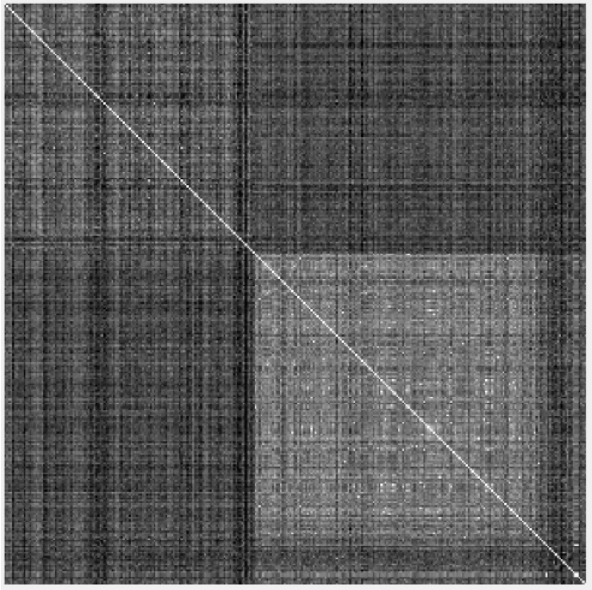



***Word2vec***

NumPCAComponents=50 Euclidean distances

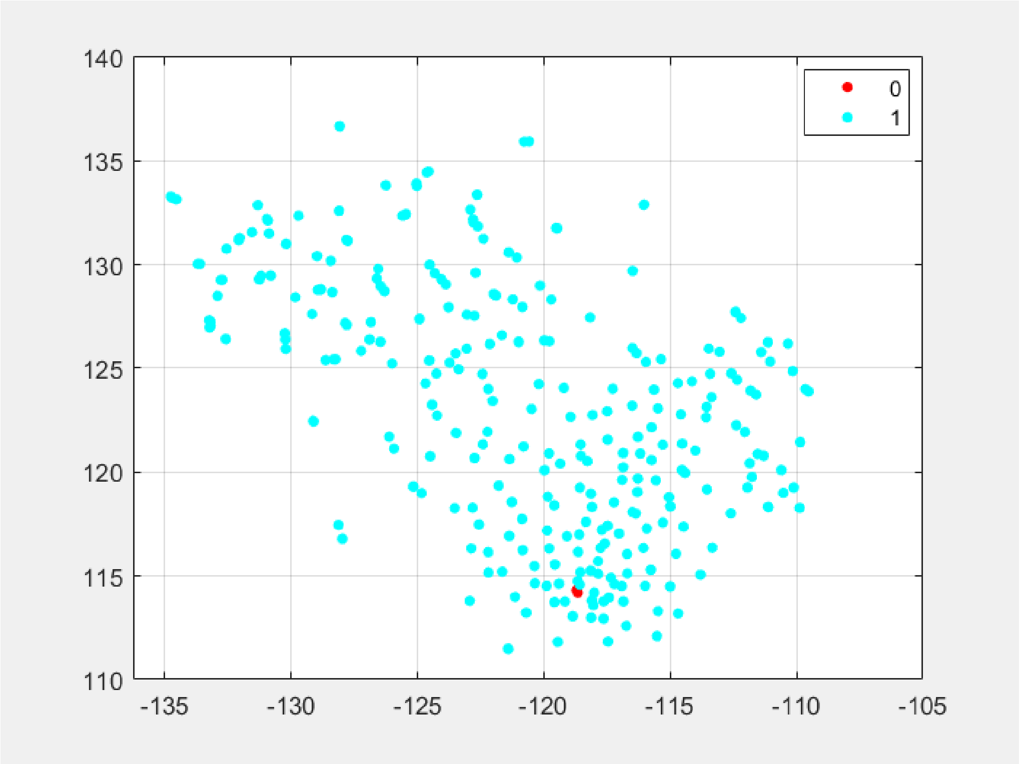
Cosine distances



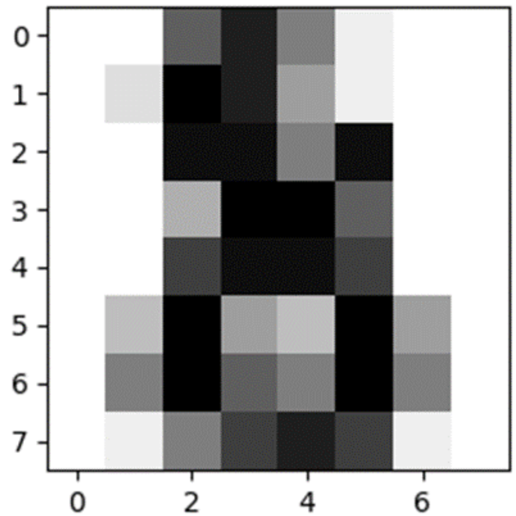



# SMS Spam dataset: clustering

***<u>Fully connected</u>***



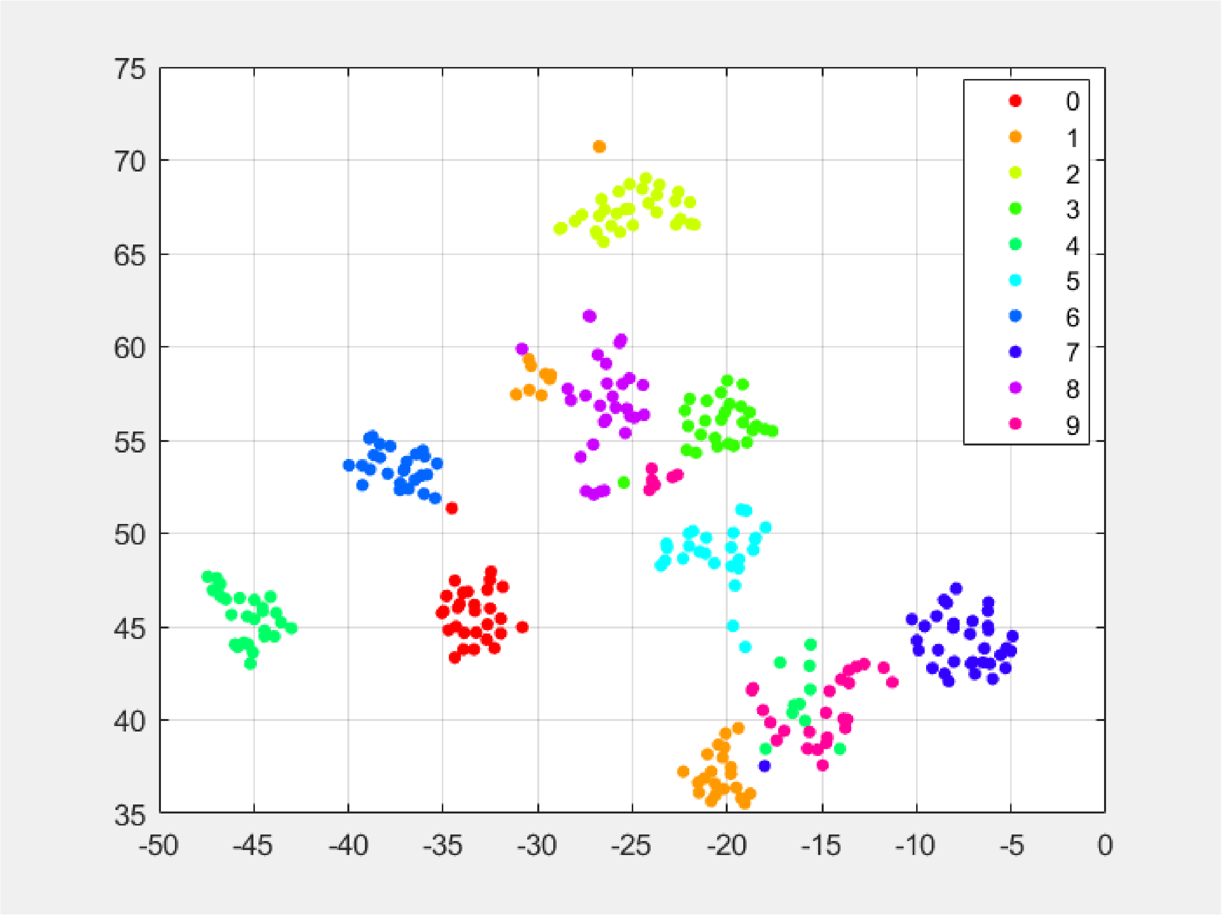
# Digits datasets: preprocessing



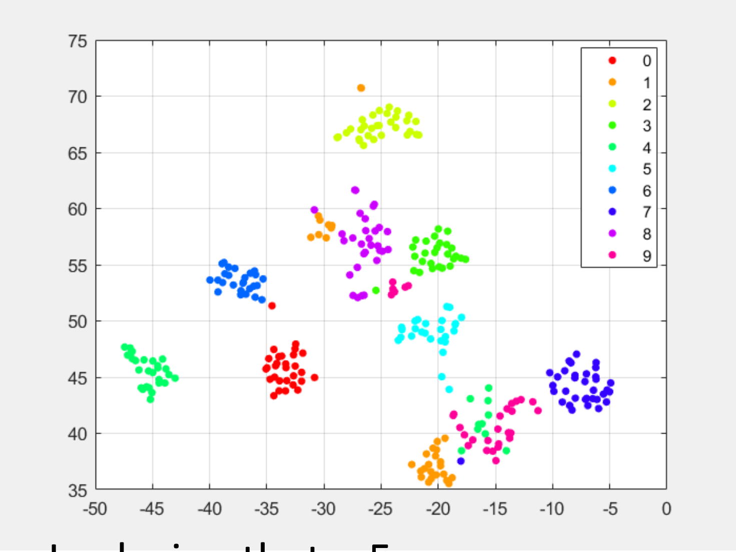

$$\begin{bmatrix} 12 \\ 2 \\ \vdots \\ 8 \end{bmatrix}$$

64 dimensional vector

# Digits datasets: visualization

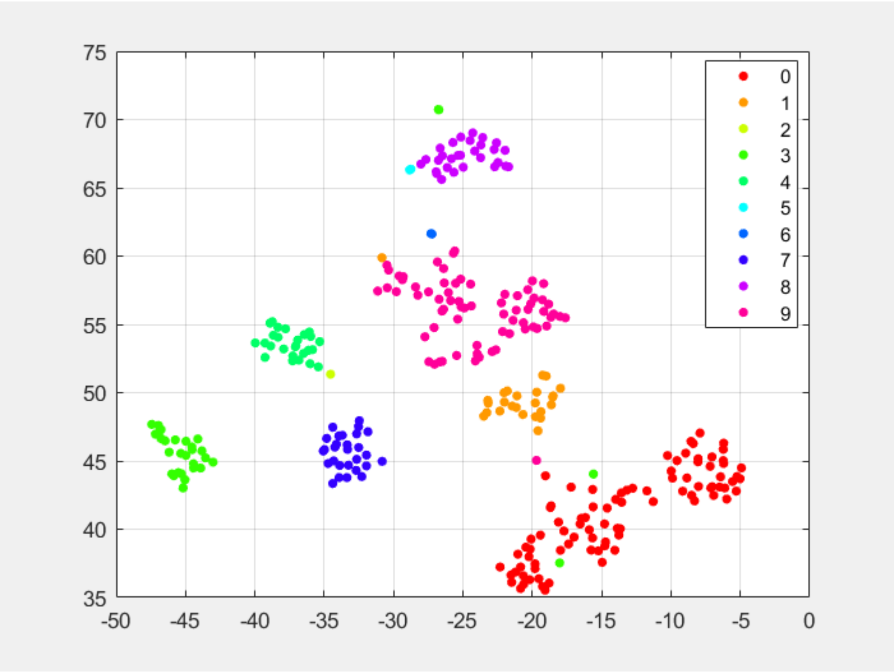


# Digits datasets: clustering

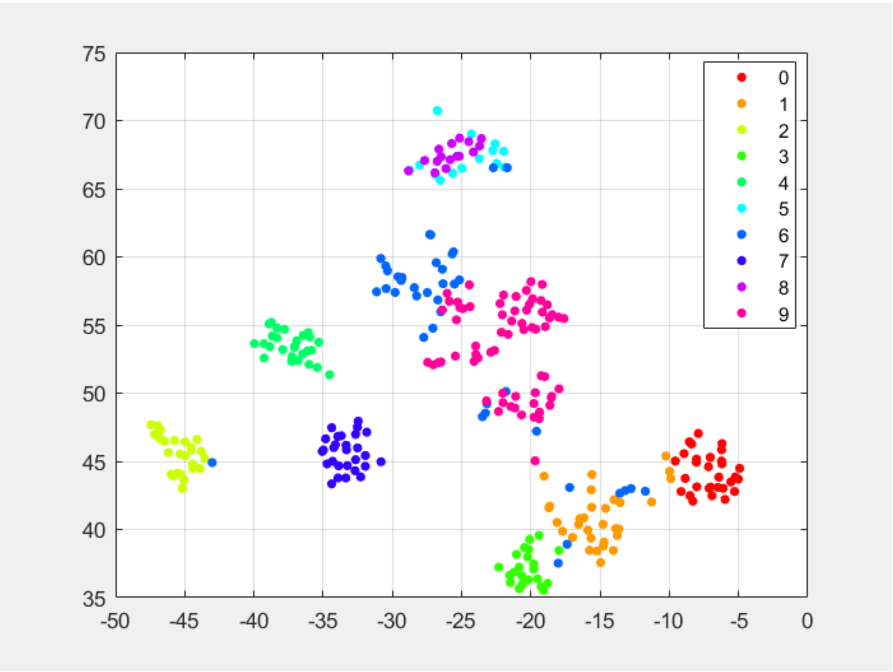



***Fully connected***

Rbf theta=7.8



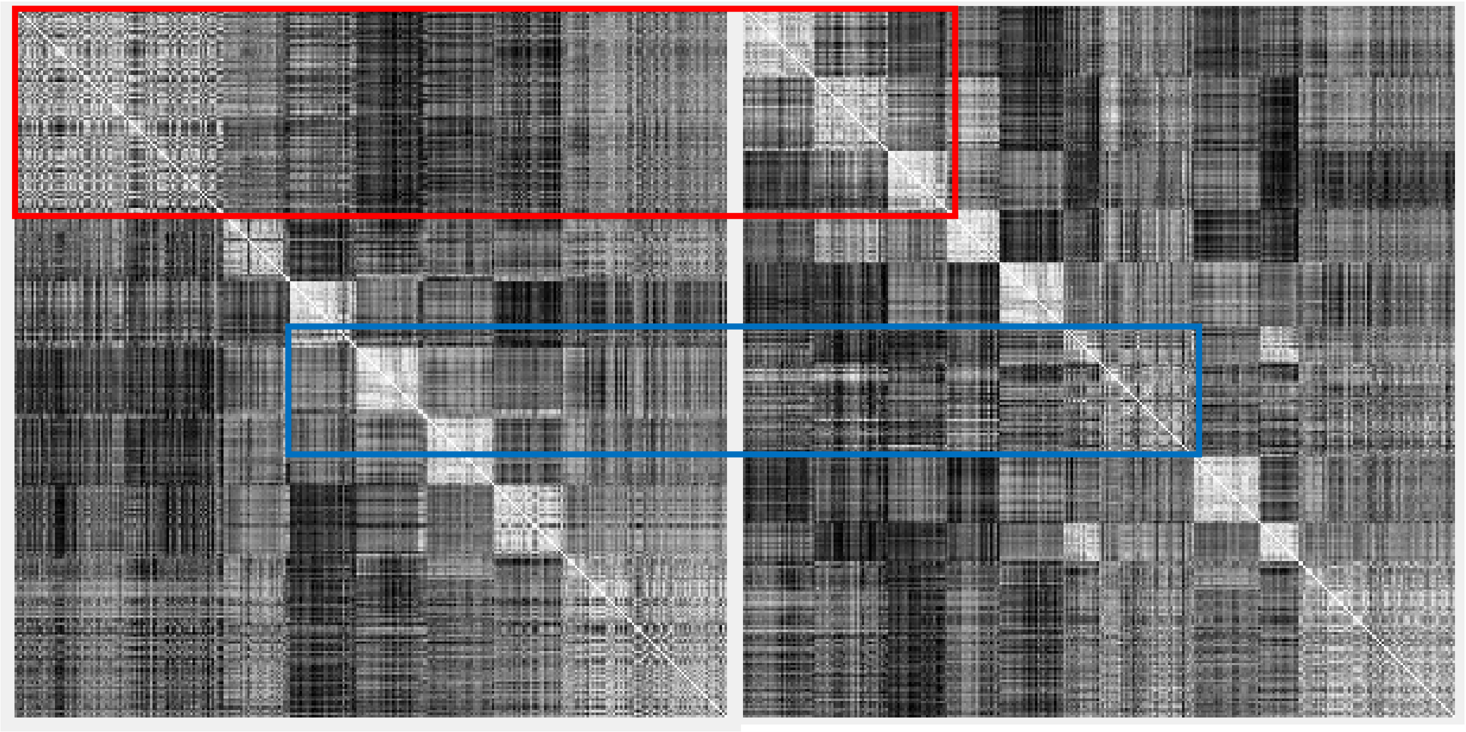

Laplacian theta=5



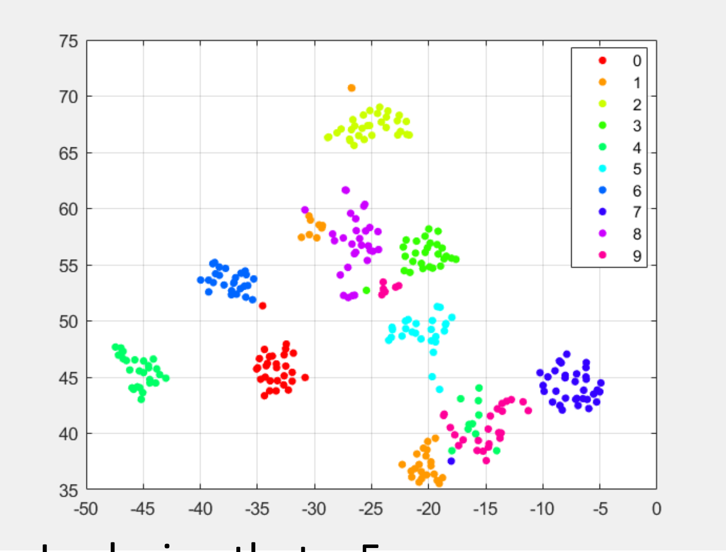
# Digits datasets: clustering

***Fully connected***

Rbf theta=7.8 Laplacian theta=5



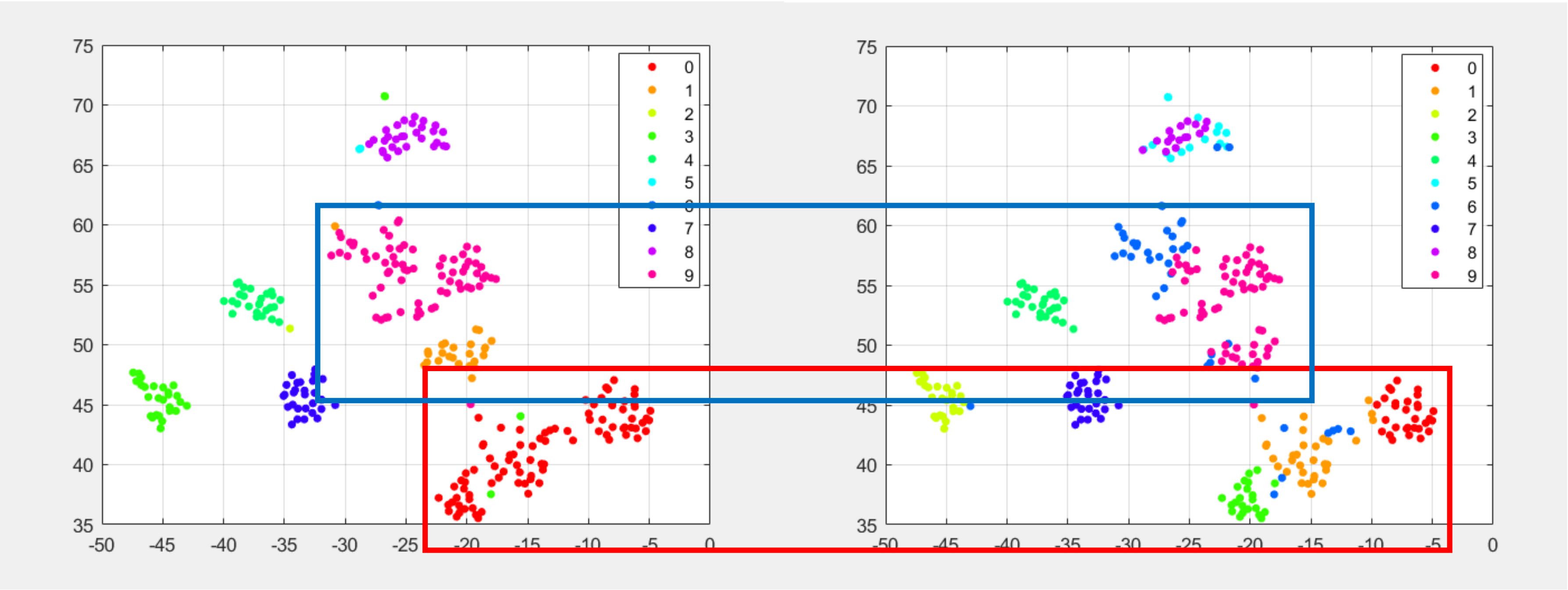
# Digits datasets: clustering




***__Fully connected__***

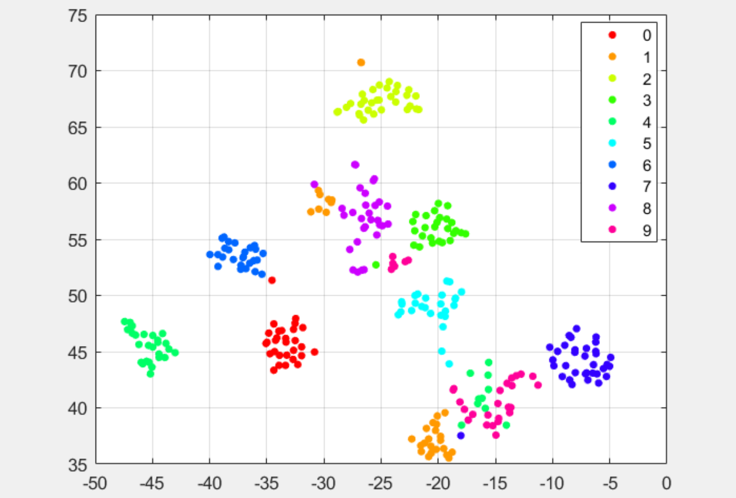
Rbf theta=7.8

Laplacian theta=5

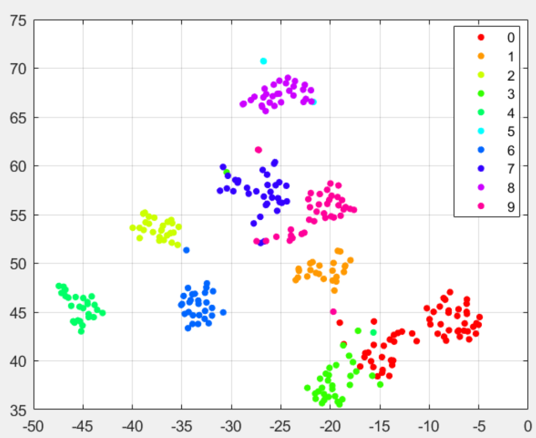


# Digits datasets: clustering

***Nearest neighbor***

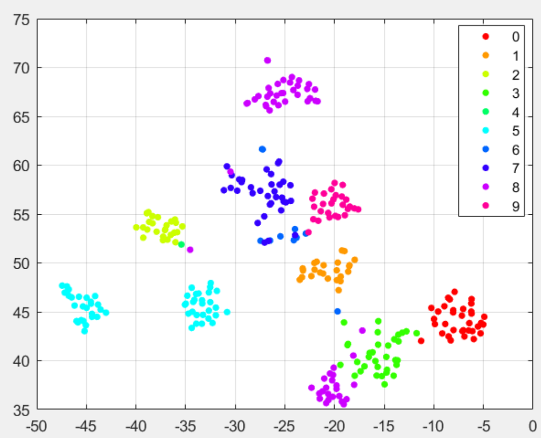



Euclidean distances 8



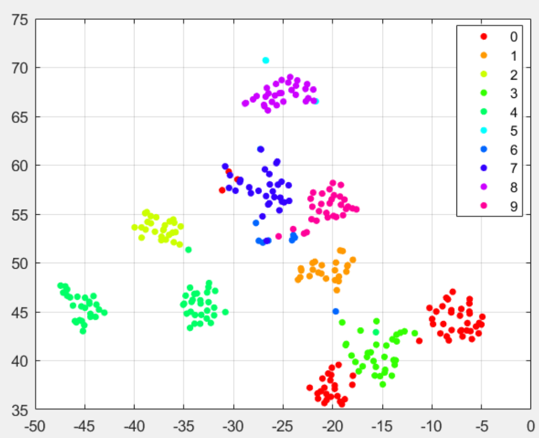

0.6878

Cosine distances 8




0.6490

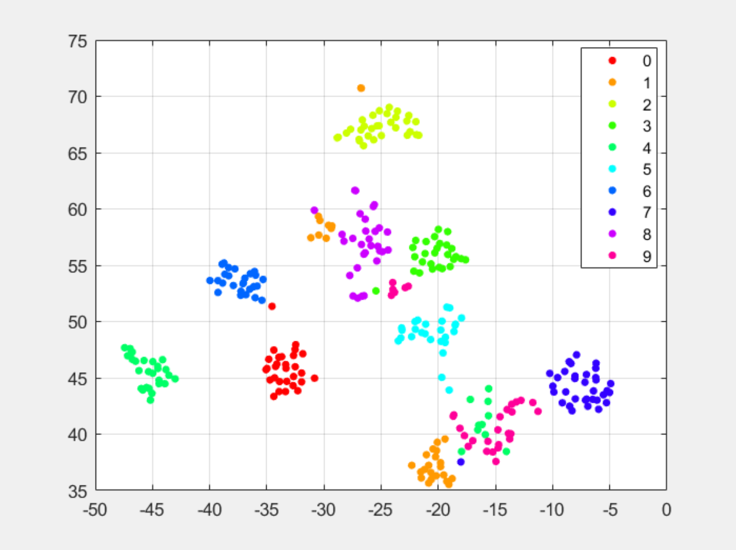
Manhattan distances 8



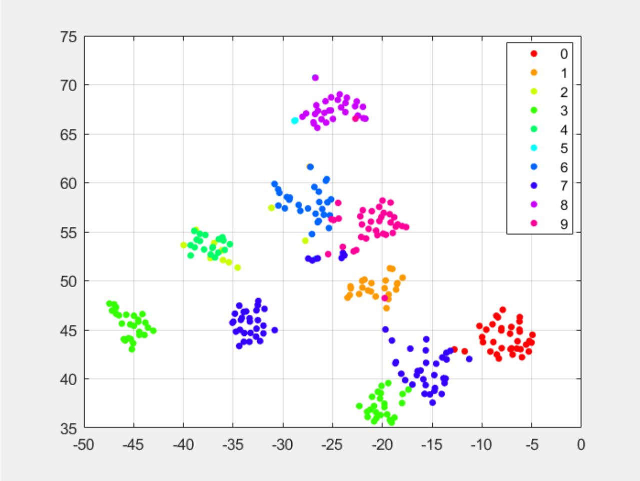

0.6707

# Digits datasets: clustering

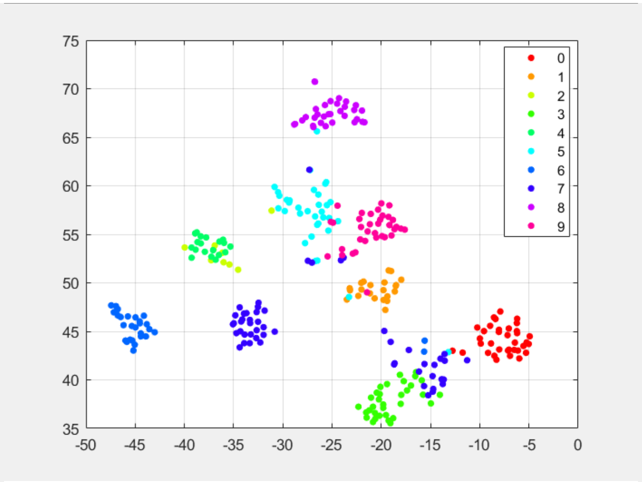
***Nearest neighbor mutual***



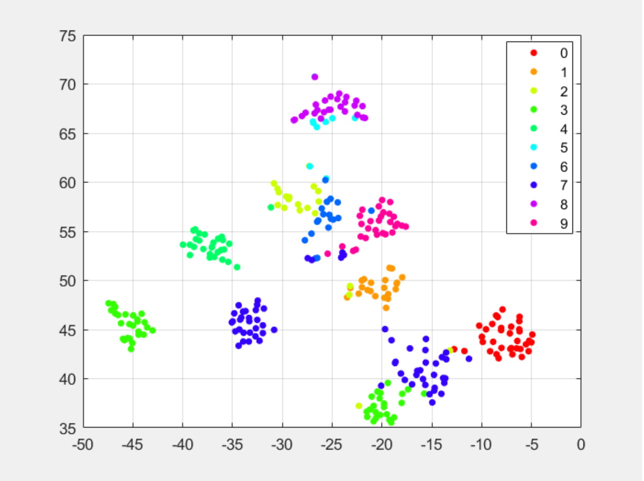

Euclidean distances 40




Cosine distances 40



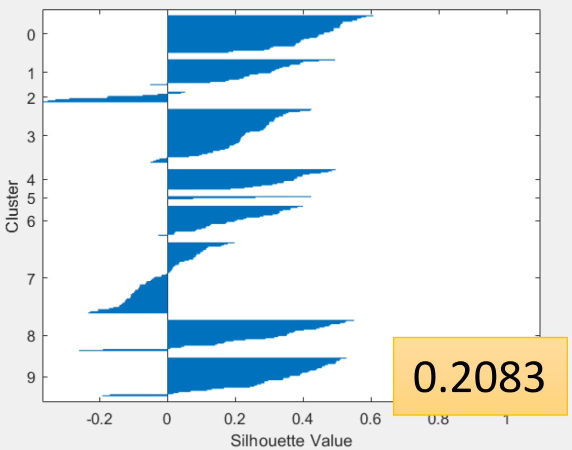

Manhattan distances 40



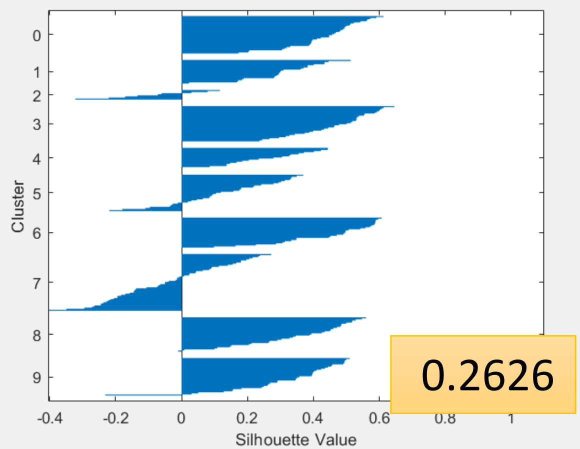
# Digits datasets: clustering

***Nearest neighbor mutual***

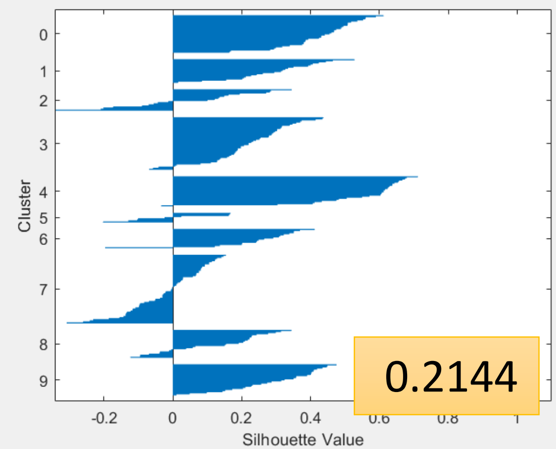
Euclidean distances 40




Cosine distances 40




Manhattan distances 40



# 结果分析

通过前面的实验设计、运算、分析的过程我们可以知道，聚类算法并不是对任何数据集都是放之四海而皆准的；聚类效果通常与数据集的特征提取过程以及相似性度量的意义有着很大的关系。倘若提取的特征不能很好的反映数据本身的模式，那么接下来再强的聚类算法也将无计可施；另外，从文本数据中，我们可以很明显的看到维度爆炸带来的严重危害。随后，在相似图的构建中，我们也应当去选择最能度量数据相似性的参数去完成聚类过程，这样才能取得更好的效果。