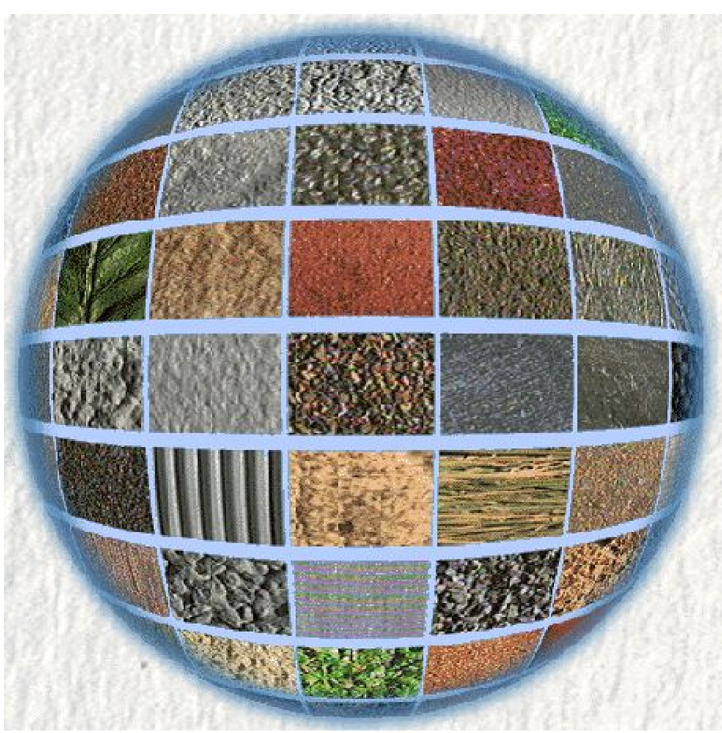
VLAD FOR TEXTURE RECOGNITION

Computational of Machine Learning
Final Defence

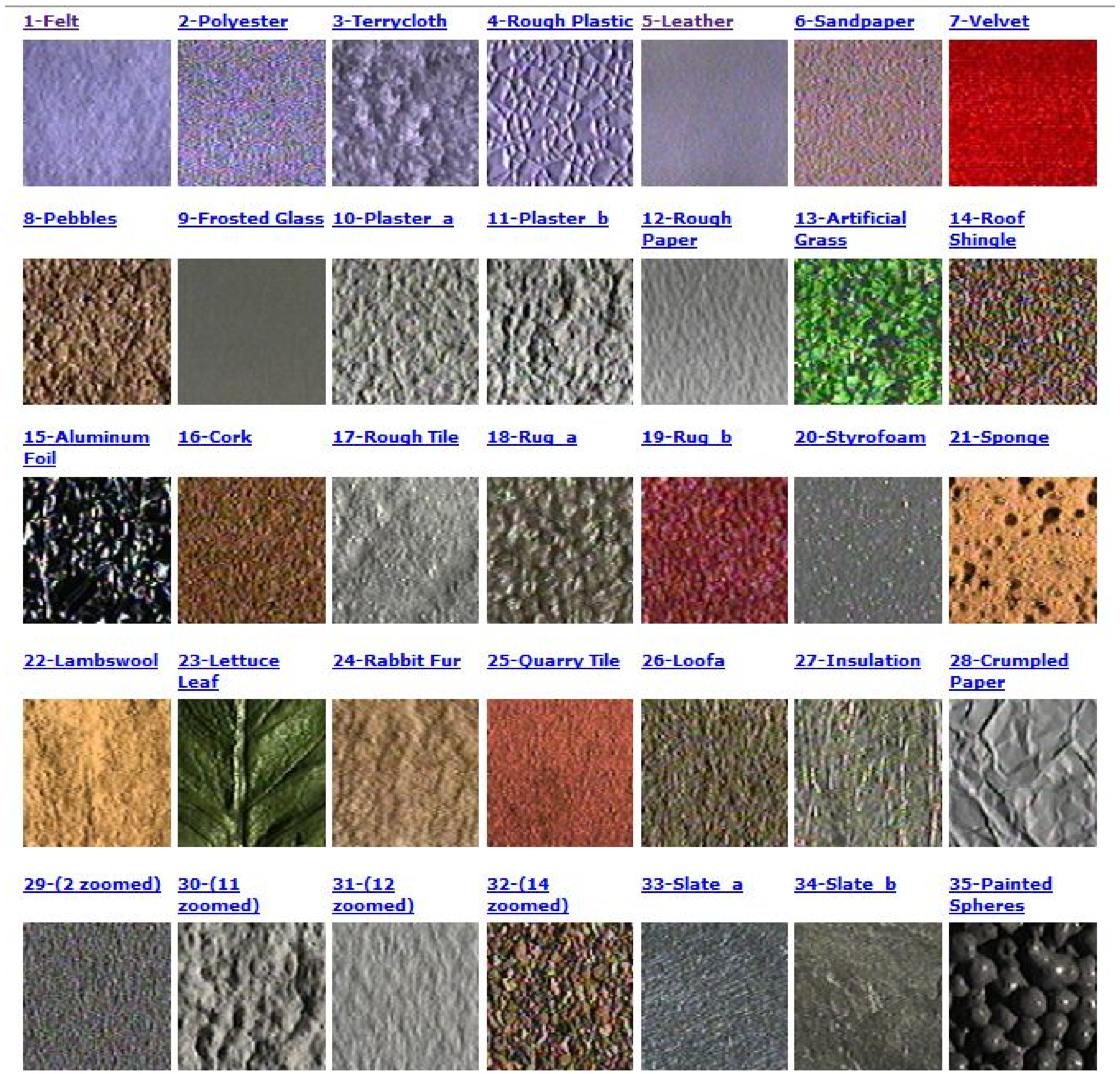


ChiaLing Wang / Tailin Lo

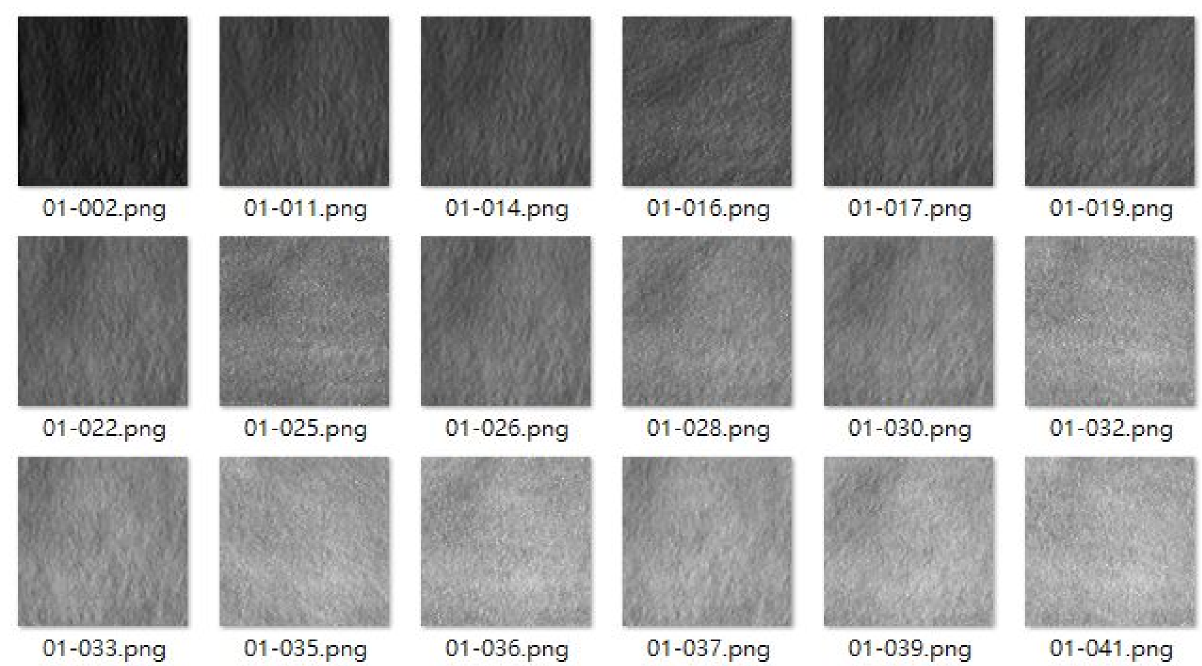
DataBase (Curet)

Columbia University and Utrecht University

61(sample sizes) x 92(images of each sample)



- centrally cropped as 200 x 200 region
- remaining backgroun is discarded
- convered into grey-scale (0-255)



http://www.cs.columbia.edu/CAVE/software/curet/html/sample.html

http://www.robots.ox.ac.uk/~vgg/research/texclass/data/curetgrey.zip

Flow Chart

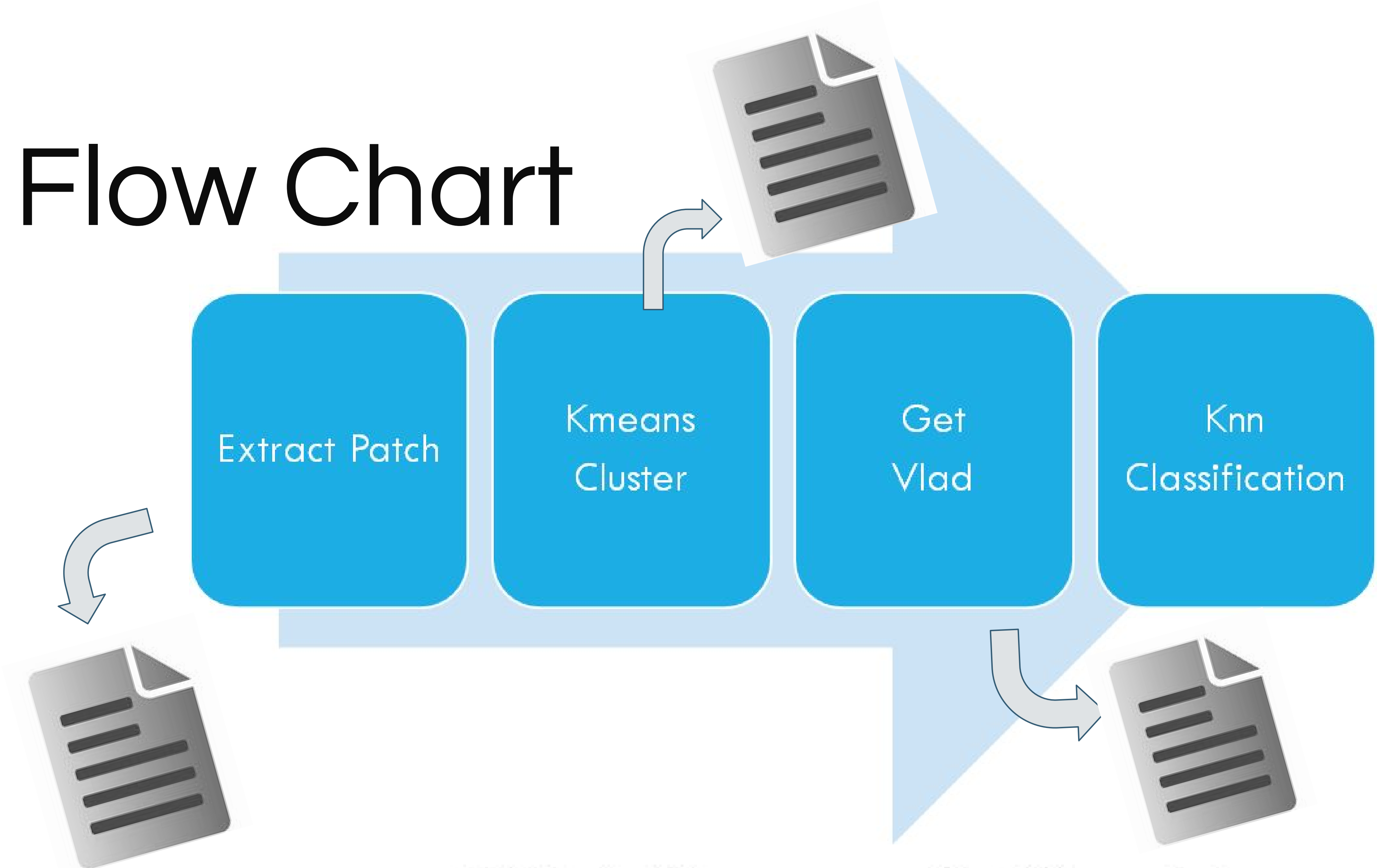


Table 1: Time-consuming (Seconds)

Process / Tools	Extract Patch	Kmeans	Get Vlad	Knn
Ours	2,928	$89 \rightarrow 13$	$70 \rightarrow 13$	NaN
Scikit Learn	NaN	12	NaN	0.1

With patch 3, class number = 10 , 2 images for each class

MINI_BATCH KMEANS Implementation

1. Decide the initial centroids
2. Choose batch set according to the batch size
3. Assign label to every point in the batch set
4. Calculate the objective value of the batch set
5. Check stop criterion
6. Assign label to every point in the data set

Get Vlad Implementation

1. Retrieve centroids and labels from previous steps
2. Calculate the difference of each data point and corresponding centroid and accumulate them.

$$\mathbf{v_{i,j}} = \sum_{\mathbf{x} \text{ such that } \mathrm{label}(\mathbf{x})=\mathbf{c_i}} \mathbf{x_j} - \mathbf{c_{i,j}}$$

3. Intra-normalization (L2 normalized each group from 2. step and then L2 normalized all group)

RunTime Challenge

- **OBJECTIVE VALUE CALCULATION :**

To calculate Euclidean Distance is a burden Intuitively, but take 90% of whole calculation.
Inner Product is the most important component.

- **INNER PRODUCT :**

For two vectors with dimension d,
CPU will take d multiplications
and d-1 additions
Can we do it parallel?
In the modern CPU Architecture,
it supports SIMD
(single instruction multiple data)

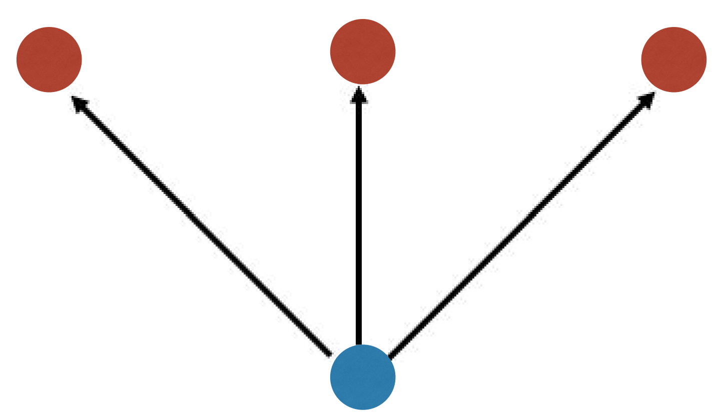
Most Time Consuming

$$\|X-Y\|_2^2 = \underbrace{\|X\|_2^2 + \|Y\|_2^2}_{\text{Pre Compute}} - 2 < X, Y >$$

from sklearn.metrics.pairwise import euclidean_distances

LABEL ASSIGN

Intuitively, calculate the Euclidean distance between each centers and each point. And assign each point to the nearest centroids.



Might have Redundant Calculation!!

- calculate the upper bound and lower bound of the distance between centroids and points.
- Use the theorem of triangle inequality to update the bound of distance between centroid and point
- Use the approach, we can reduce the calculate times of Euclidean distance as many as possible

http://users.cecs.anu.edu.au/~daa/courses/GSAC6017/kmeansicml03.pdf

Data Structure

lower bound			upper bound	cluster index

Ex: If there are 3 clusters, we need to maintain 3 lower bound of distance between centroids and the points.

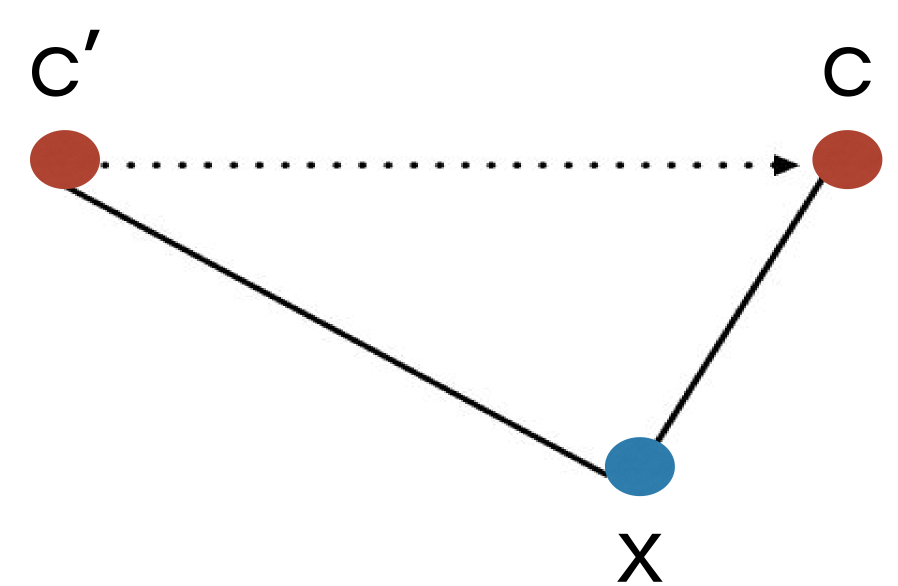
The upper bound cell stores the upper bound of the distance between the corresponding centroids and the point

The new upper bound :

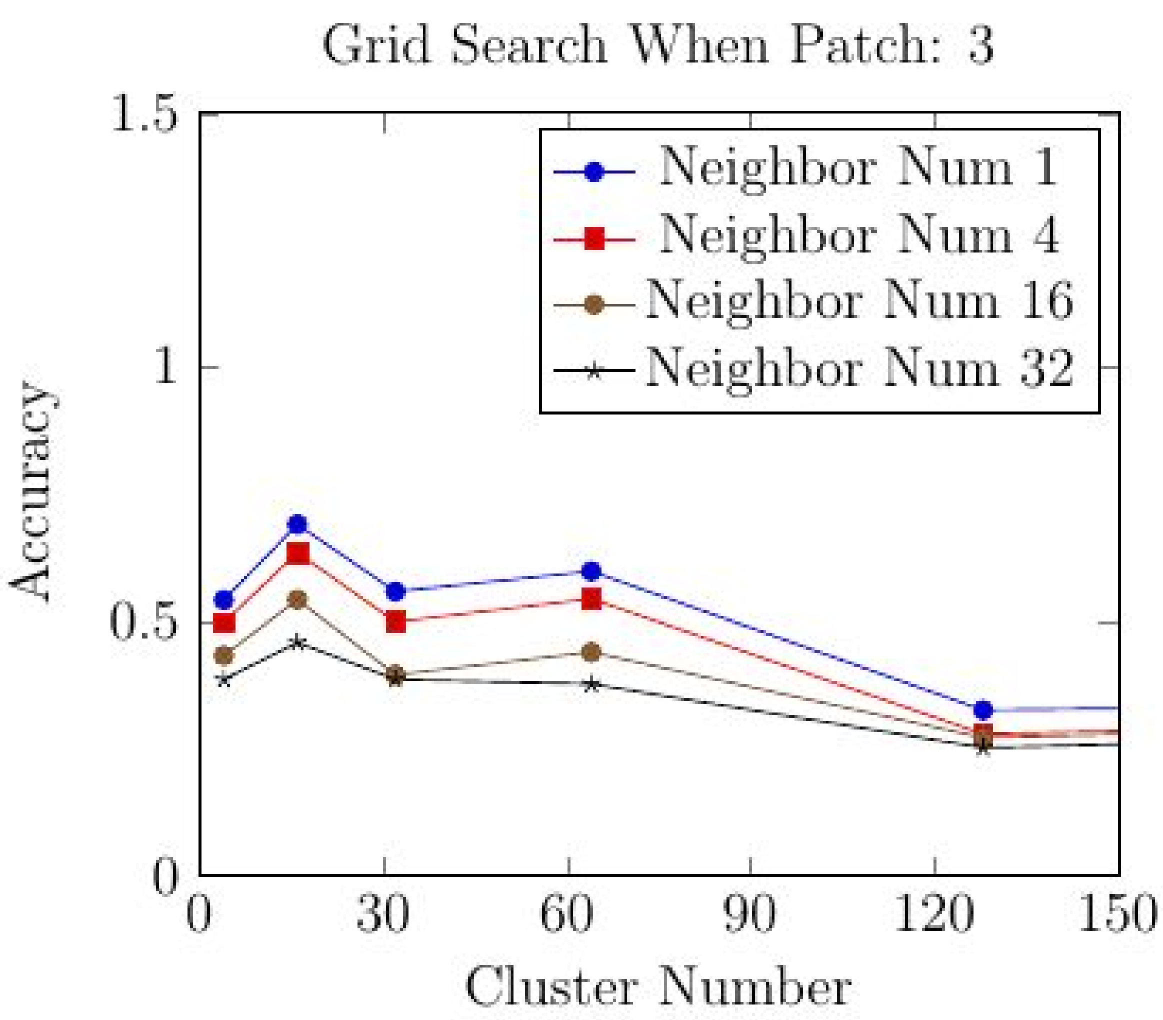
$d(x,c) = U(d(x,c')) + d(c',c)$

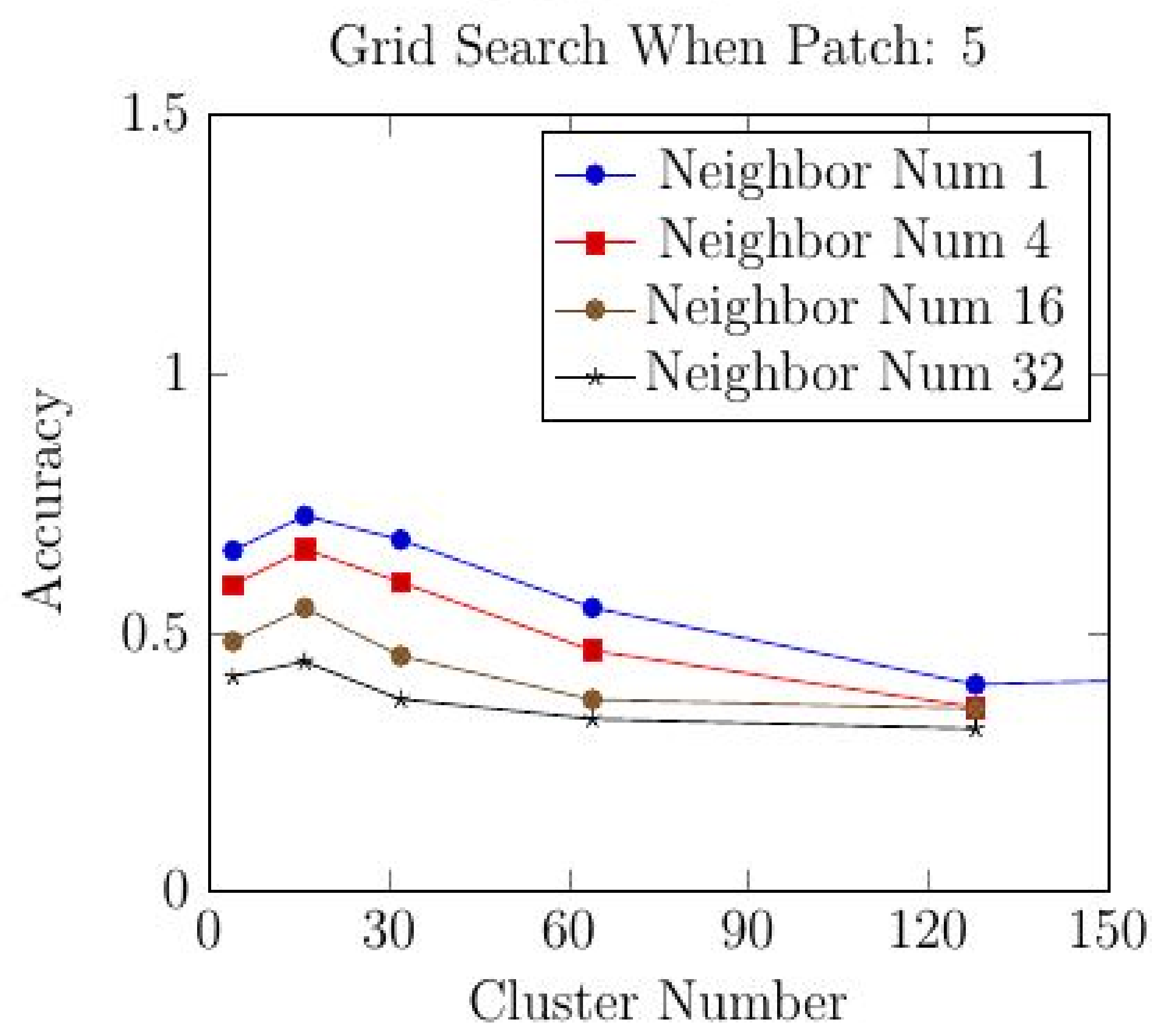
The new lower bound :

$d(x,c) = \max(0, L(d(x,c')) - d(c',c))$

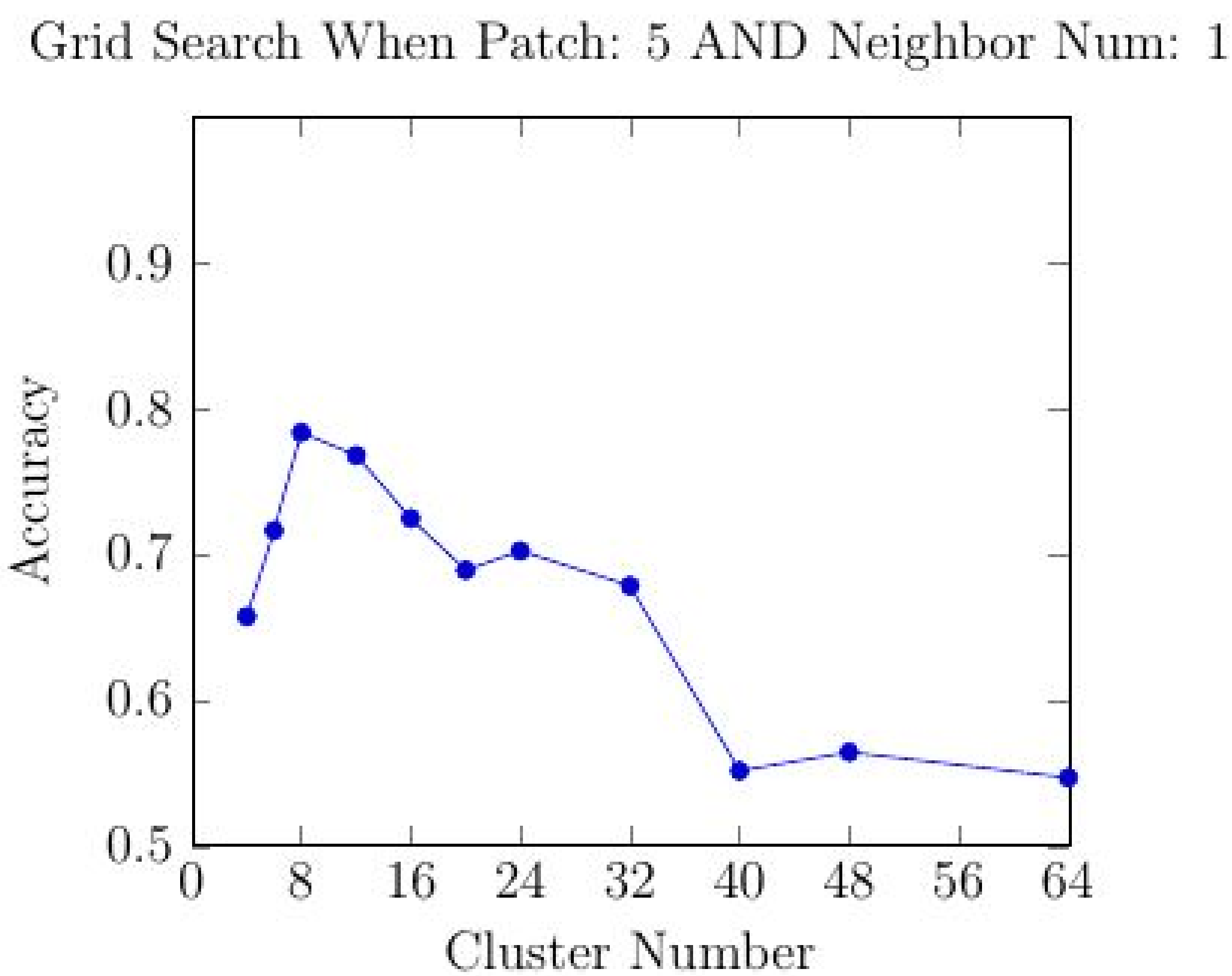


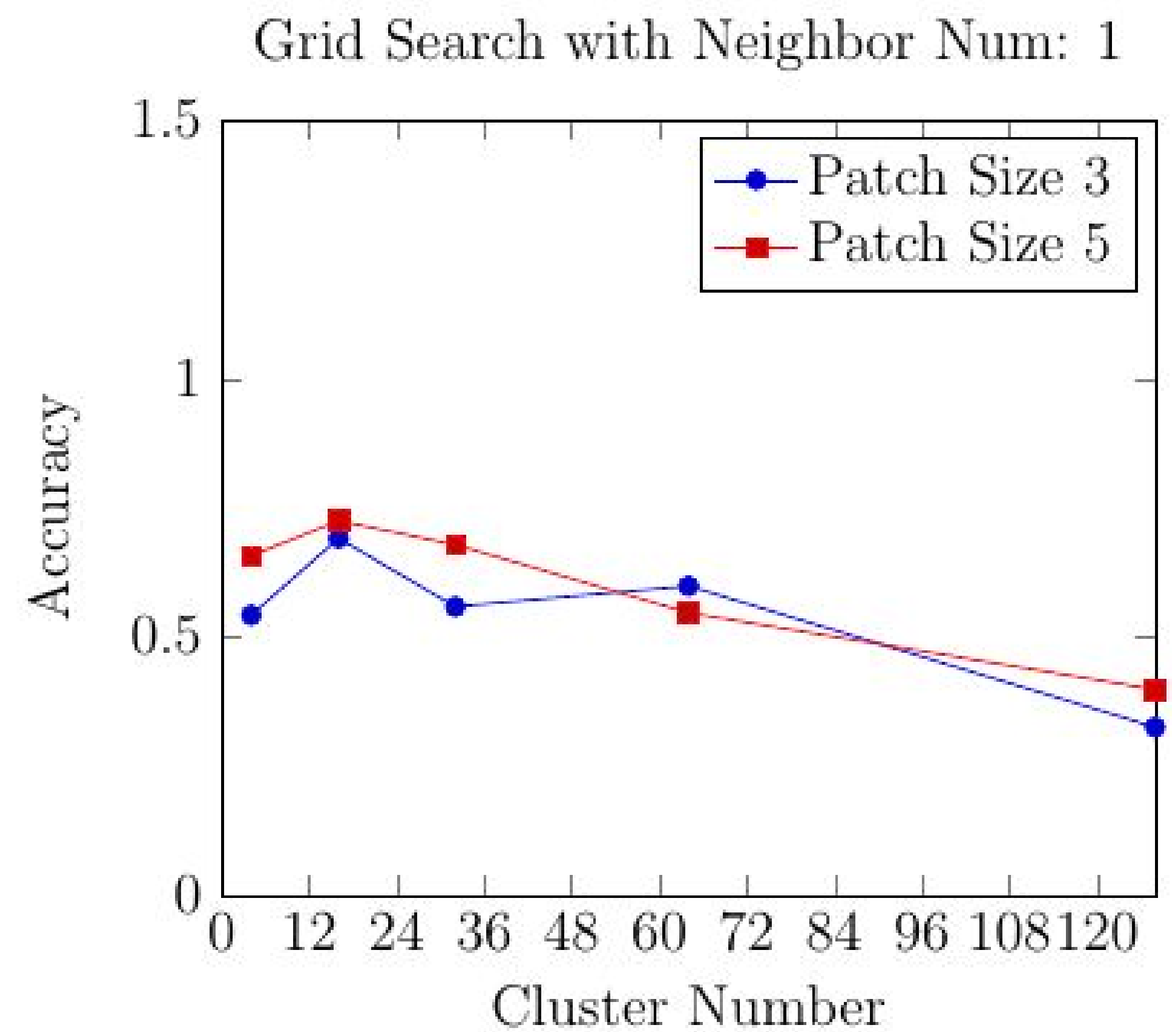
EXPERIMENTS DATA (Based on all dataset 61*92)





EXPERIMENTS DATA (Based on all dataset 61*92)





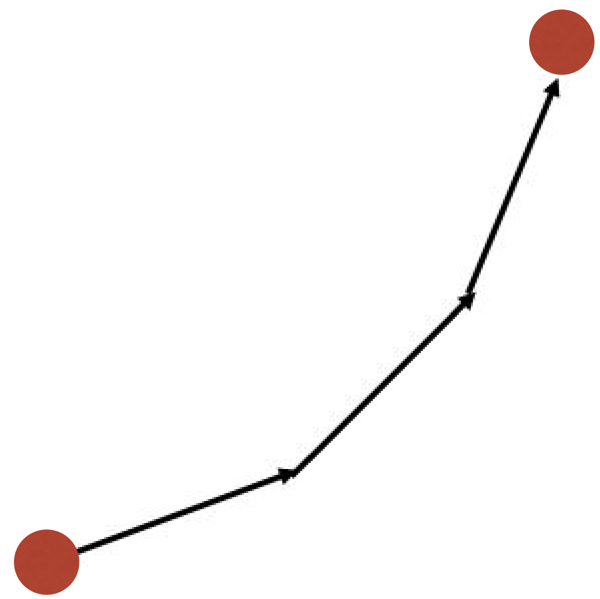
For now the highest accuracy is 78.4%
(Patch Size : 5 , Clust Number : 8 , Neighbor Number : 1)

Thank you !

Q & A

BACKUP

When To Stop ?



If do infinite iterations, it will finally get the ideal center

In Mini-Batch cases, objective value go up and down because we select data randomly

If an objective value sequence like this 1,2,1,2,1,2,...
when does the algorithm stop?

MOVING AVERAGE

Consider a simple case: 1,2,1,2,1,2,…
Given a window size 2.
For window1, the average is (1+2)/2 = 1.5.
For the next window, the average is (2+1)/2 = 1.5.
Thus, it seems to be stable, we stop at 3rd iteration.

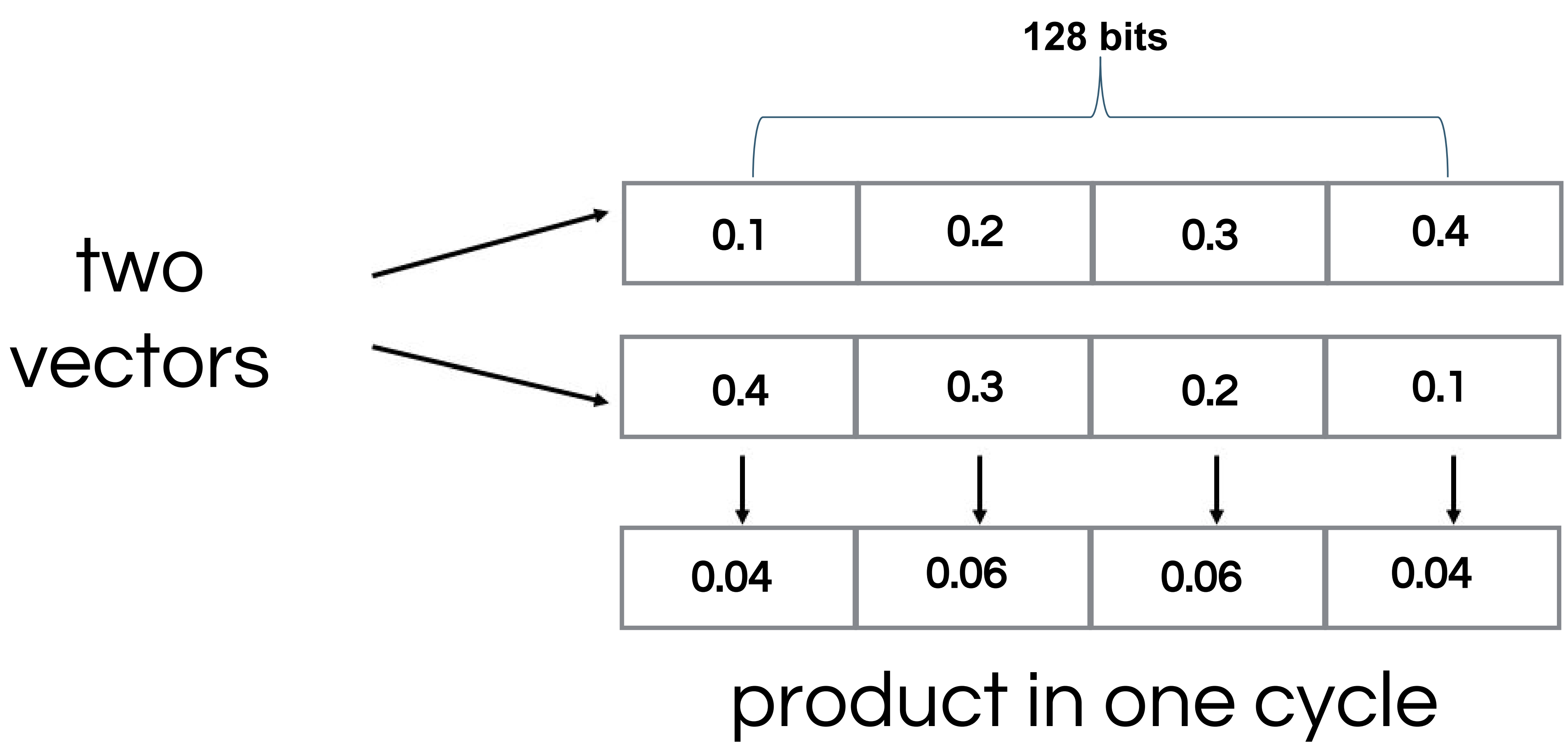
Consider a simple case: 1,2,3,1,2,3,1,2,3,…
Given a window size 3.
For window1, the average is (1+2+3)/3 = 2.
For the next window, the average is (2+3+1)/3 = 2.
Thus, it seems to be stable, we stop at 4-th iteration.

WEIGHTED MOVING AVERAGE (Studied Scikit-Learn)

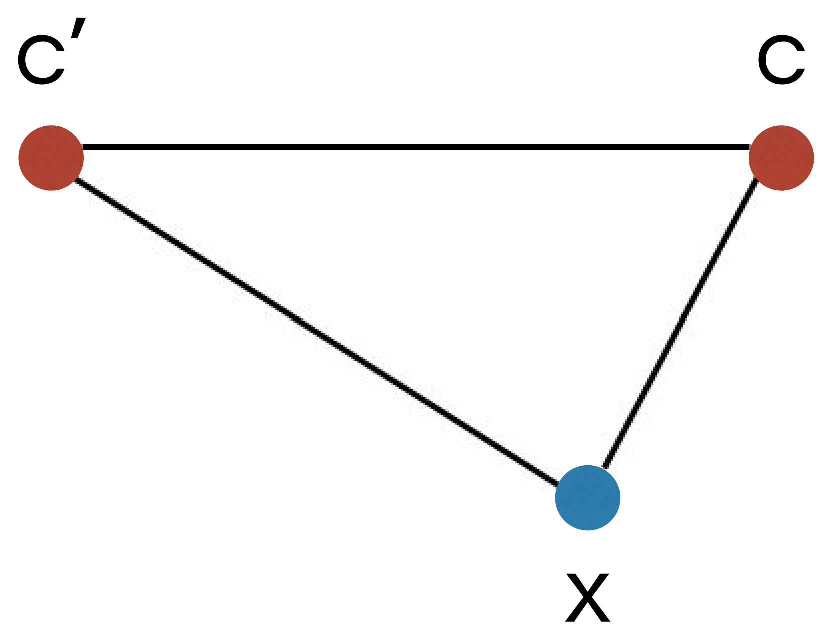
Sometimes, we may multiply a value at every elements, i.e
In Scikit-Learn, it uses exponential weighted moving average(the previous element with fewer weight)

Practically, no such beautiful sequence, we still need to set a threshold to compare the previous average objective value with current average objective value

SIMD FOR PRODUCT



TRIANGLE INEQUALITY



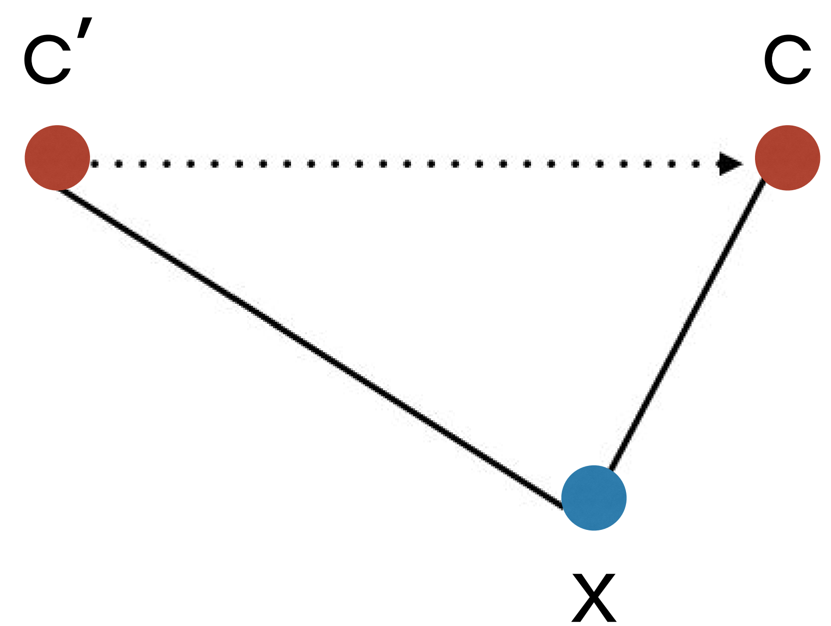
Theorem:

1. If d(c,c’) >= 2d(x,c), then d(x,c’) >= d(x,c)
2. d(x,c’) >= max(0, d(x,c)-d(c,c’))

CENTER MOVEMENT

scenario:

if the label of x doesn't change, and the center moves



Assume c' is the previous center, and c is the current center

the new upper bound of d(x,c) = U(d(x,c')) + d(c',c)

the new lower bound of d(x,c) = max(0, L(d(x,c')) - d(c',c))

Note: U means upper bound, and L means lower bound

CENTER MOVEMENT

From the above, it’s clear that we extend the upper bound and
lower bound of the point in each round if that point doesn’t change its cluster.

d(x,c)

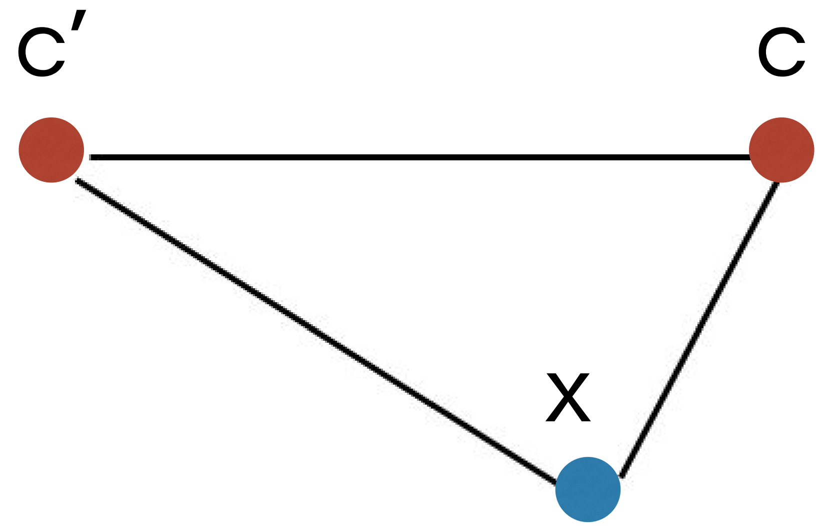
d(x,c)

Extend the bound

CHANGE CLUSTER

When does x change its owner ?

what is the check condition?



Assume c is the center of the cluster x belongs to.

And c' is the center of other cluster.

Use theorem (1), if the upper bound of d(x,c) <= d(c,c') /2,

then we can guarantee the upper bound of d(x,c) <= d (x,c')

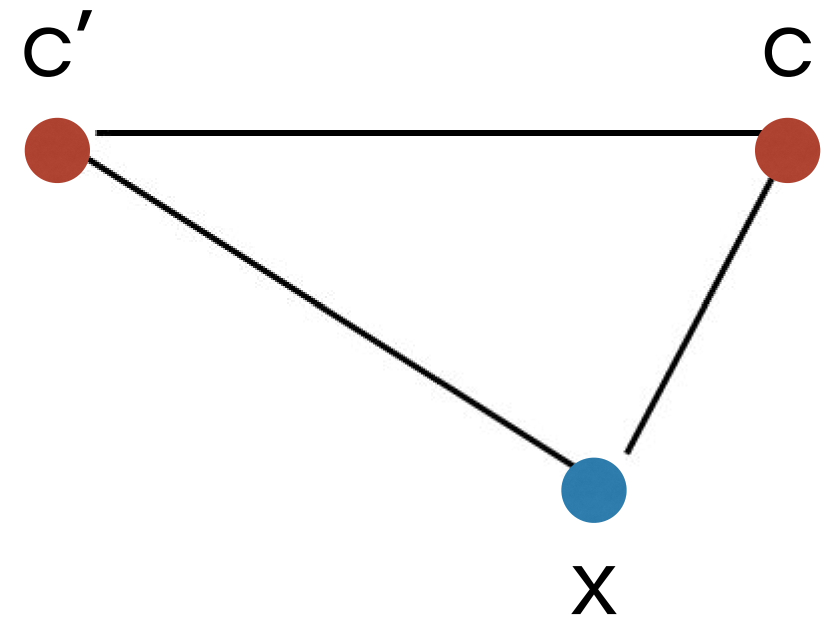
Theorem:

1. If d(c,c') >= 2d(x,c), then d(x,c') >= d(x,c)

CHANGE CLUSTER

When does x change its owner ?

what is the check condition?



Assume c is the center of the cluster x belongs to.

And c’ is the center of other cluster.

Use theorem (1), if the upper bound of d(x,c) <= d(c,c’)/2,

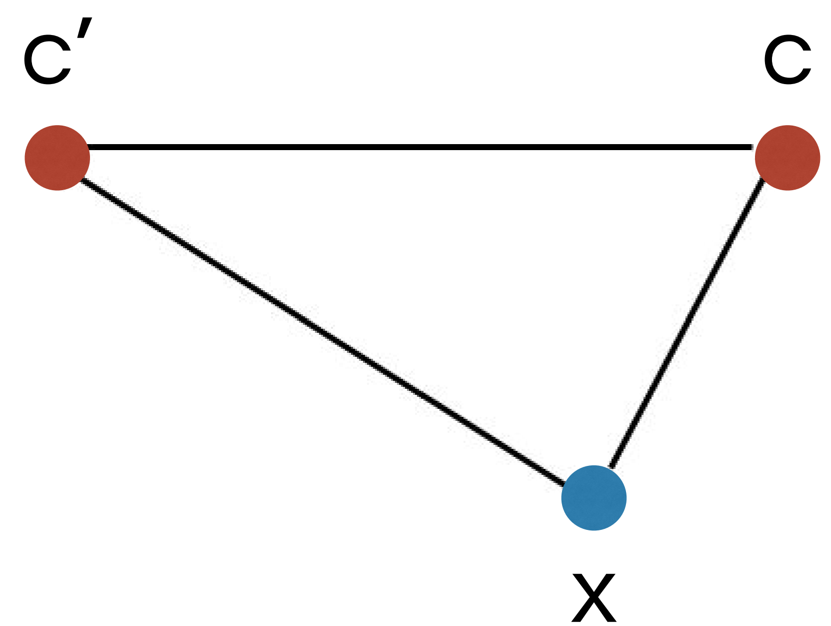
then we can guarantee the upper bound of d(x,c) <= d(x,c’)

Theorem:

1. If d(c,c’) >= 2d(x,c), then d(x,c’) >= d(x,c)

CHANGE CLUSTER

When does x change its owner ?
what is the check condition?



What if the upper bound of d(x,c) > d(c′,c)/2 and > the lower bound of d(x,c′) ?

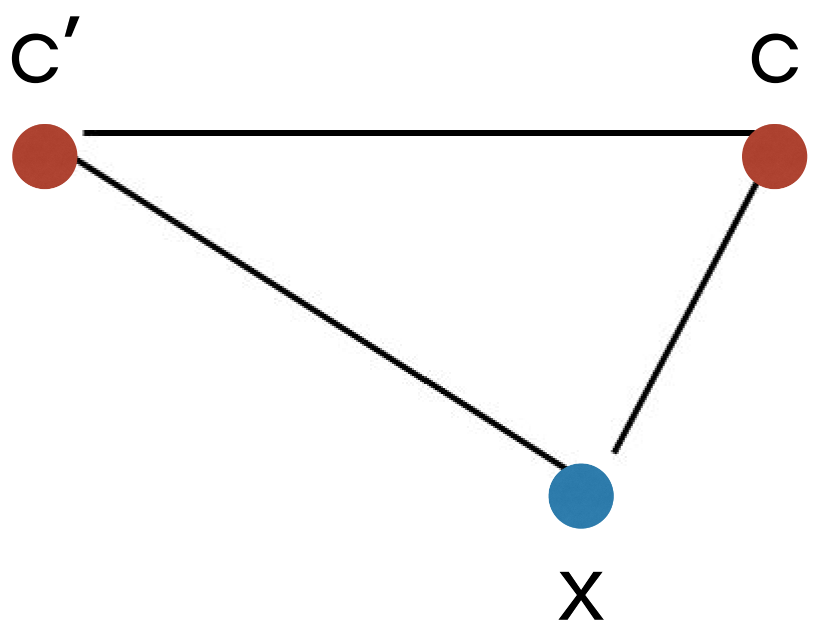
Theorem:

1. If d(c,c′) >= 2d(x,c), then d(x,c′) >= d(x,c)

CHANGE CLUSTER

When does x change its owner ?

what is the check condition?



What if the upper bound of d(x,c) > d(c′,c)/2 and > the lower bound of d(x,c′) ?

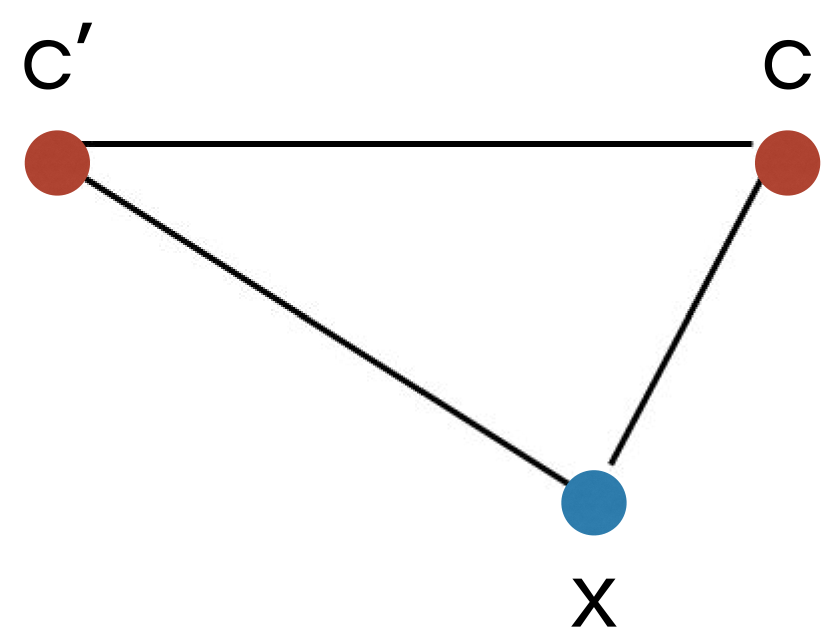
we need to compare d(x,c′) with d(x,c) precisely — calculate the Euclidean distance

Theorem:

1. If d(c,c′) >= 2d(x,c), then d(x,c′) >= d(x,c)

CHANGE CLUSTER

When does x change its owner ?
what is the check condition?



if $d(x,c') < d(x,c)$, then re-assign x to c′ and set
lower bound of $d(x,c') = d(x,c')$
upper bound of $d(x,c') = d(x,c)$

Theorem:

1. If $d(c,c') >= 2d(x,c)$, then $d(x,c') >= d(x,c)$