Topological approach for finding nearest neighbor sequence in time series

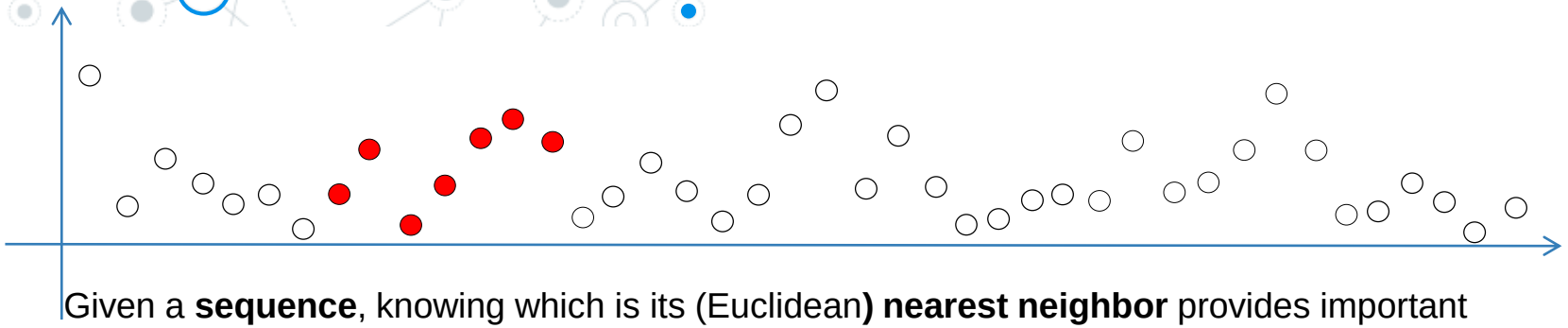
Paolo Avogadro, Matteo Dominoni

Dipartimento di Informatica, Sistemistica e Comunicazione

Università degli Studi di Milano-Bicocca

paolo.avogadro@unimib.it

Motivation



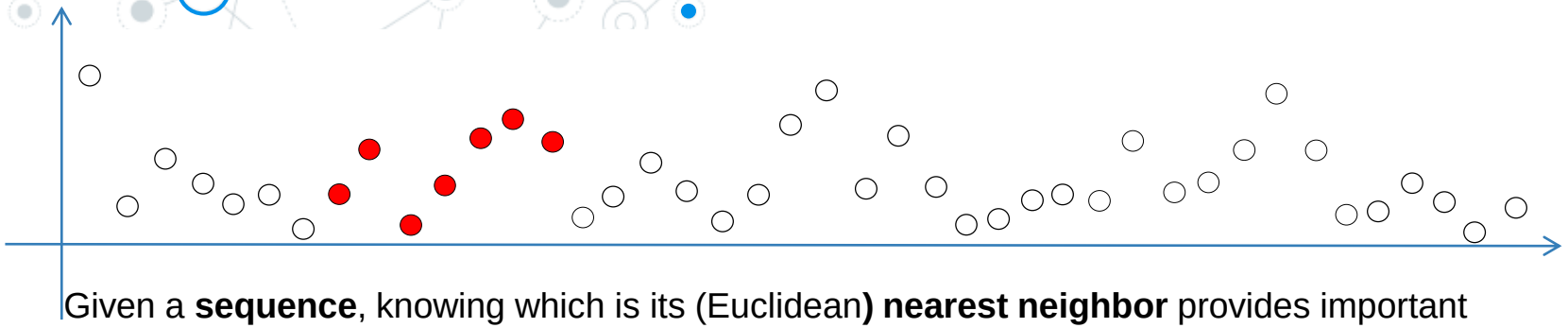
Given a **sequence**, knowing which is its (Euclidean**) nearest neighbor** provides important information, regarding:

anomalies

clusters

Recurrent patterns

Motivation



Given a **sequence**, knowing which is its (Euclidean**) nearest neighbor** provides important information, regarding:

anomalies

clusters

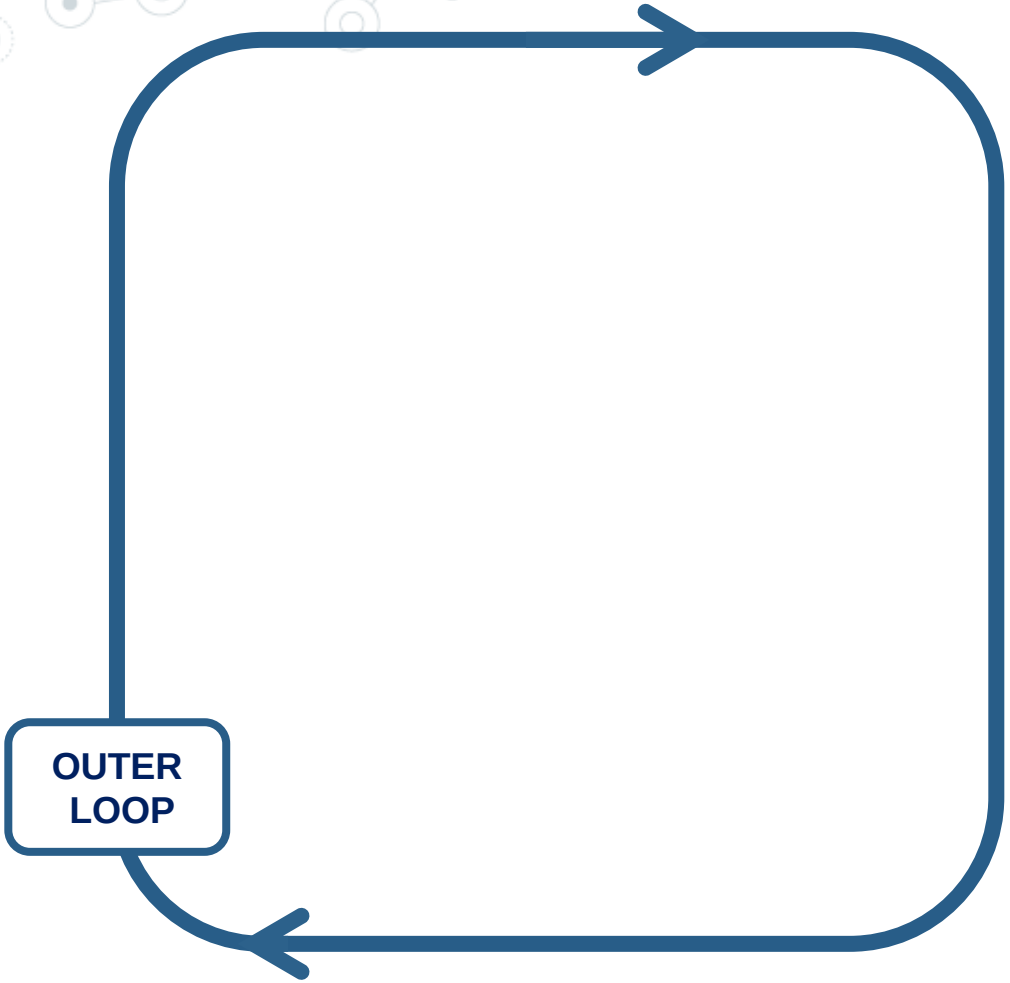
Recurrent patterns

Aim

A fast and accurate **algorithm** to find an approximate **nearest neighbor distance (nnd)** profile for **all** the sequences of a time series

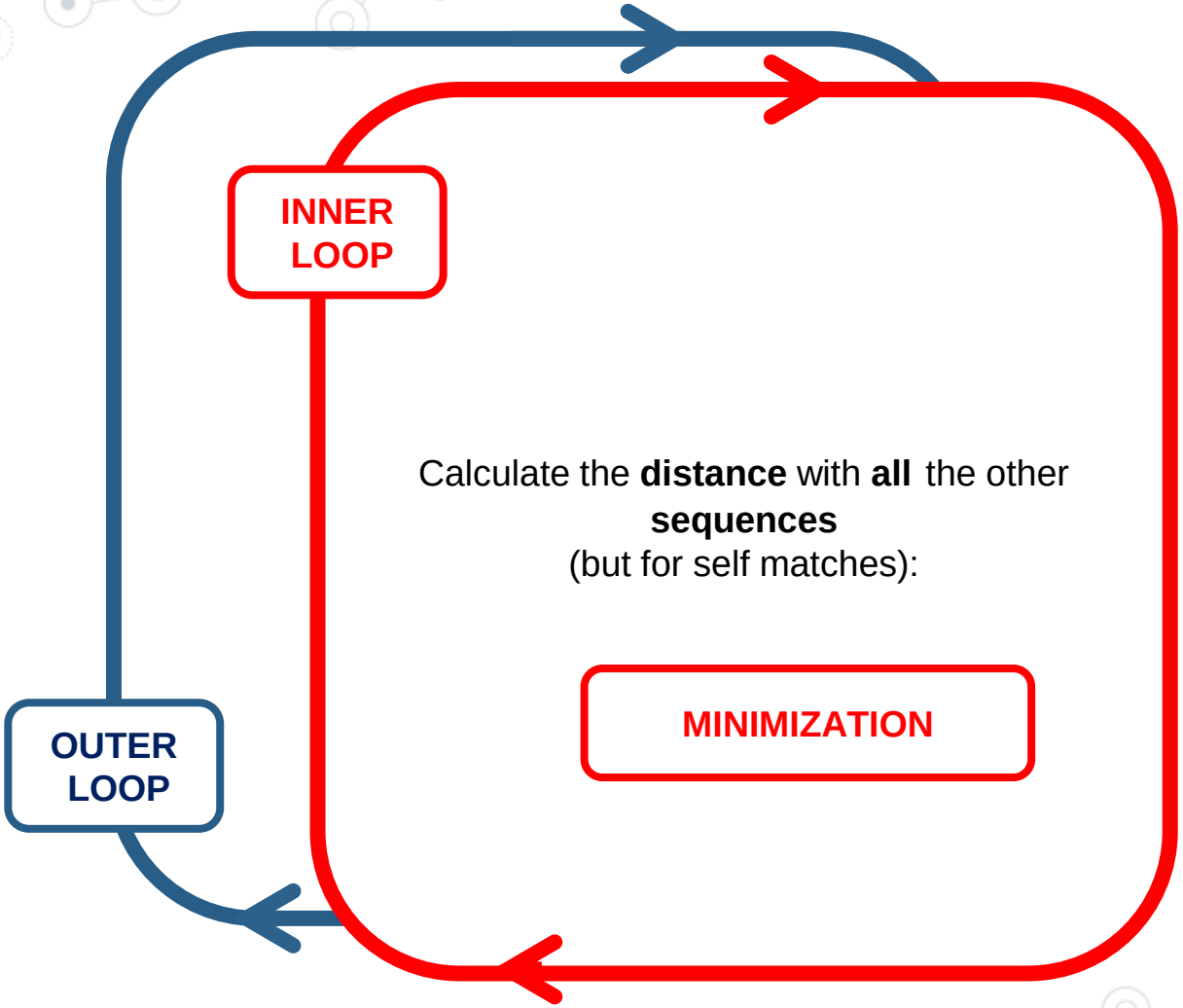
IDEA

In order to obtain the **nnd** of **all** the squences one needs **2** nested **loops**.



IDEA

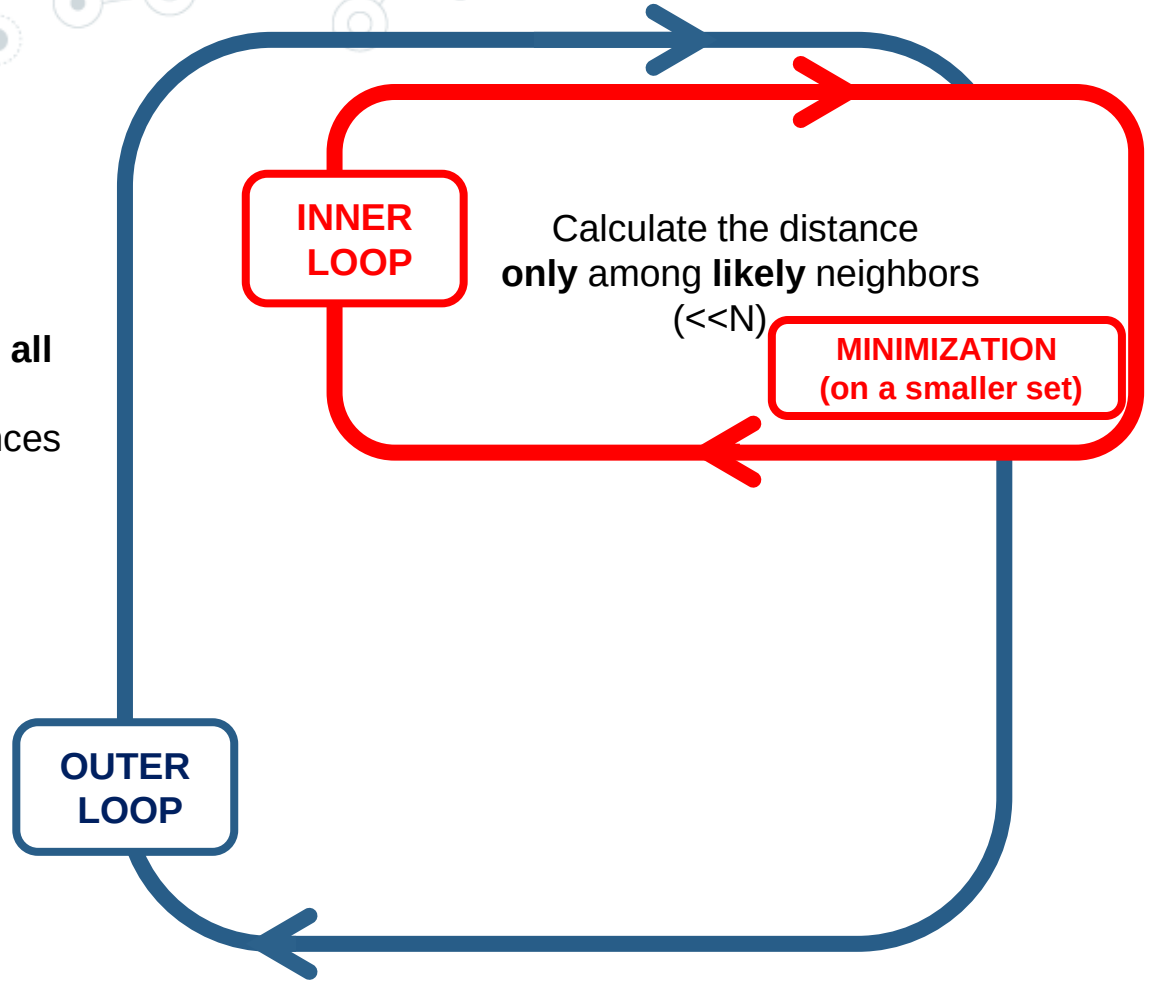
In order to obtain the **nnd** of **all** the squences one needs **2** nested **loops**.



$O(N^2)$

IDEA

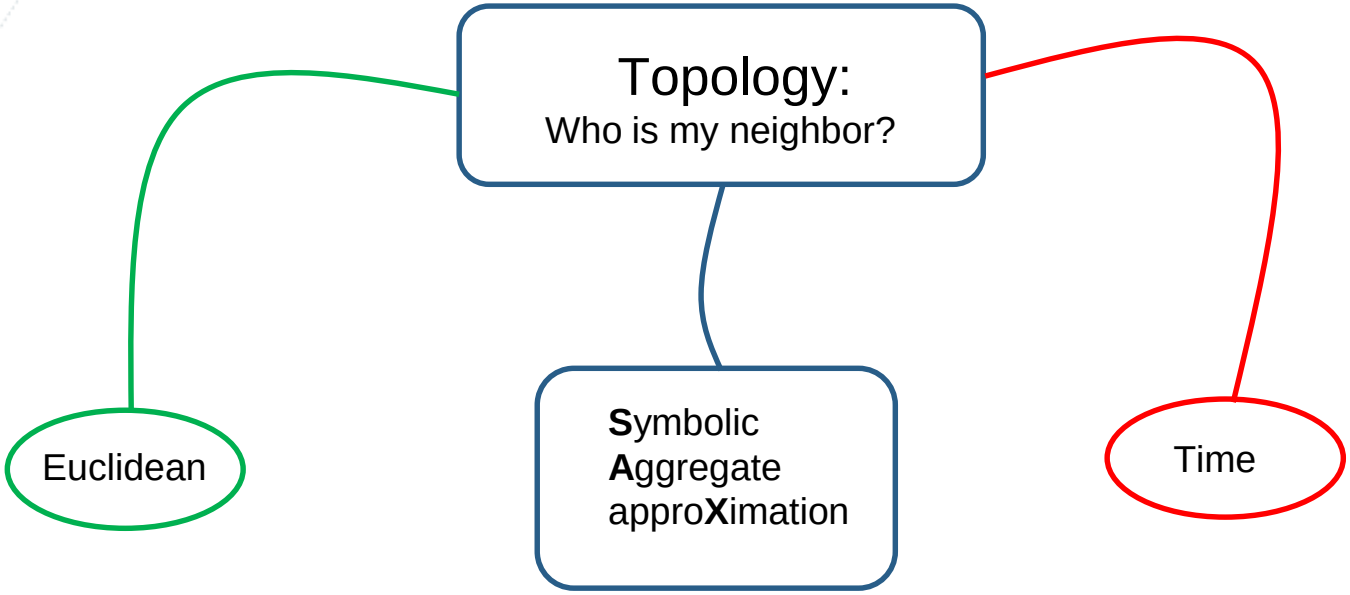
In order to obtain the **nnd** of **all** the squences one needs **2** nested **loops**.



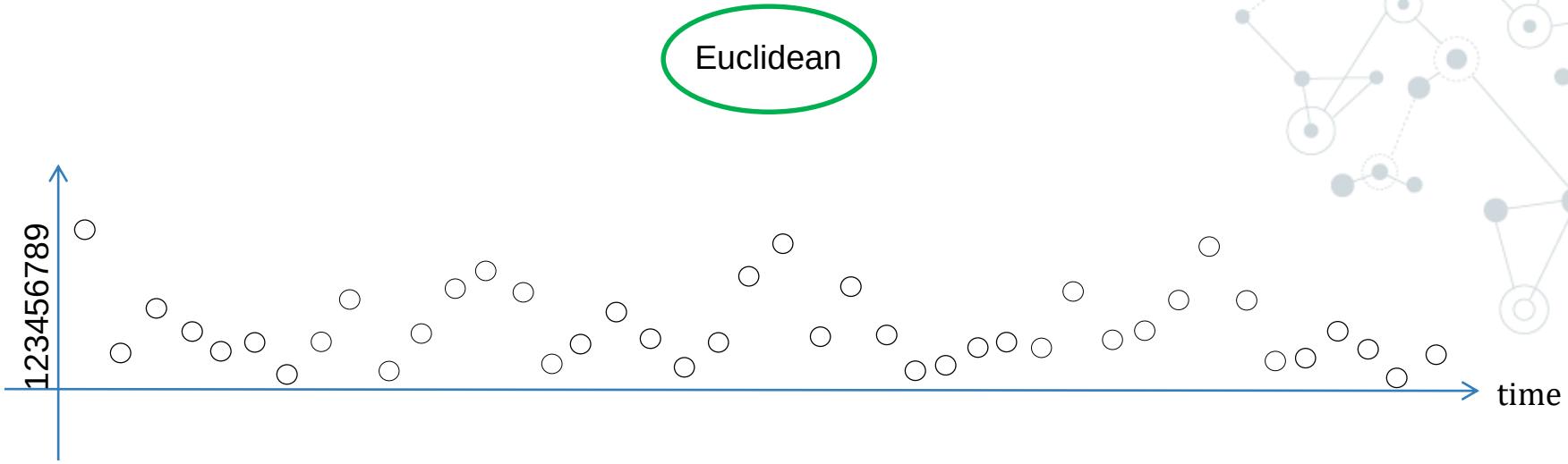
Run on **all** the N sequences

Let's **prune** the **inner loop**, thanks to **topologies**

Topology

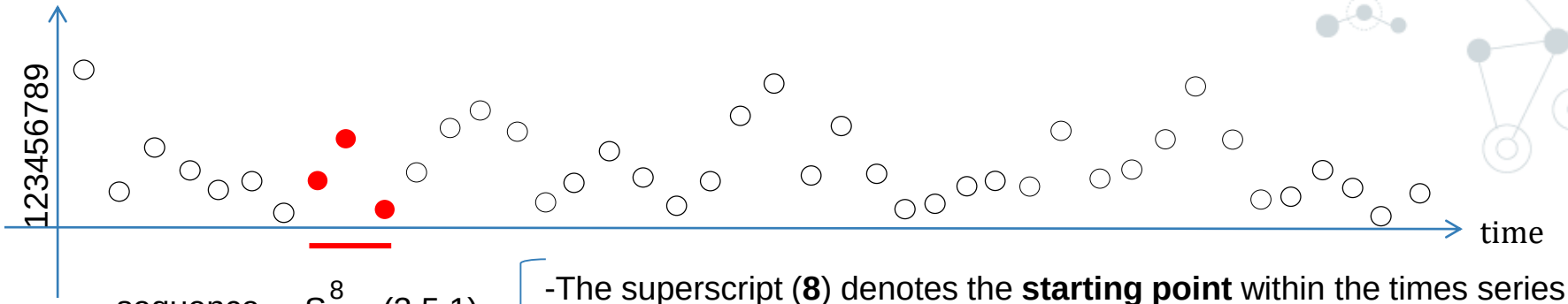

Topology:
Who is my neighbor?
Euclidean
Symbolic
Aggregate
approXimation
Time

Topology



Topology

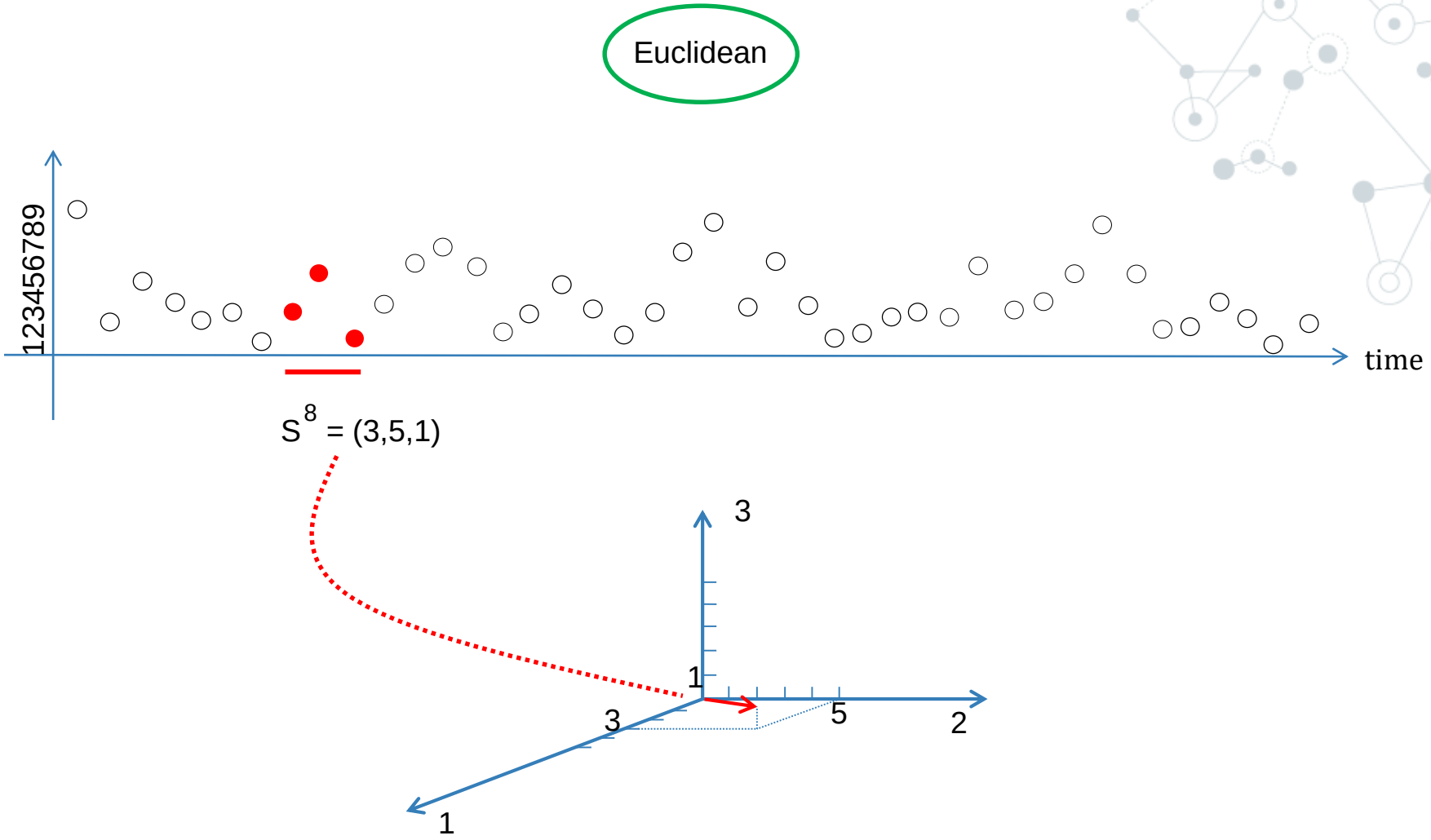
Euclidean



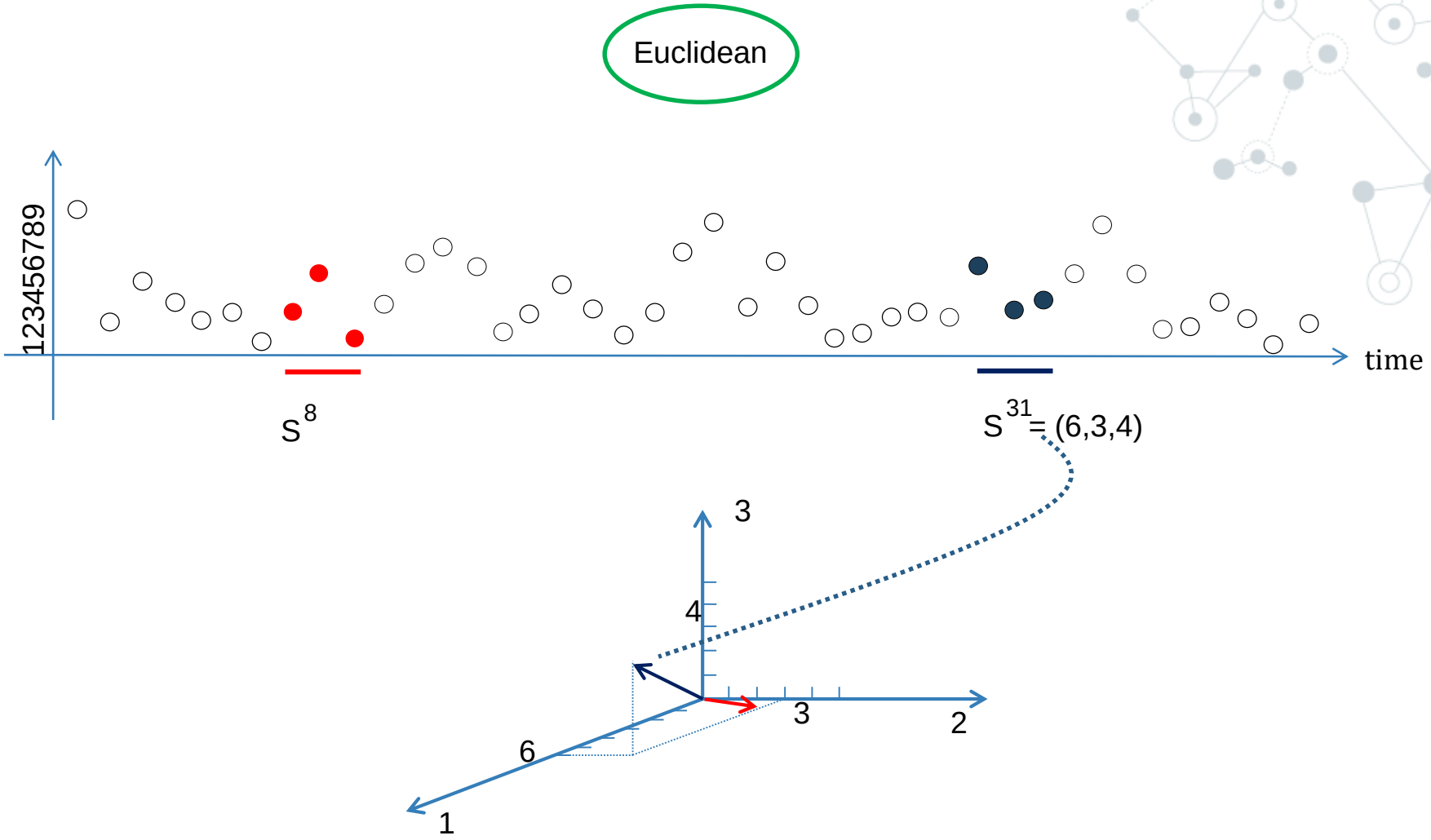
sequence $S^8 = (3,5,1)$

- The superscript (**8**) denotes the **starting point** within the times series
- s_2^8 is the **2**nd point of sequence S^8, here $s_2^8 = 5$

Topology

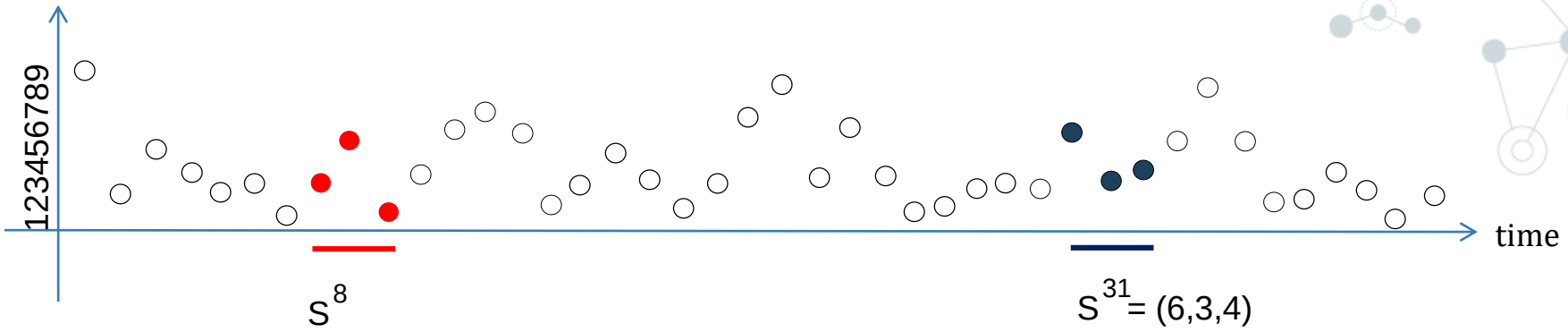


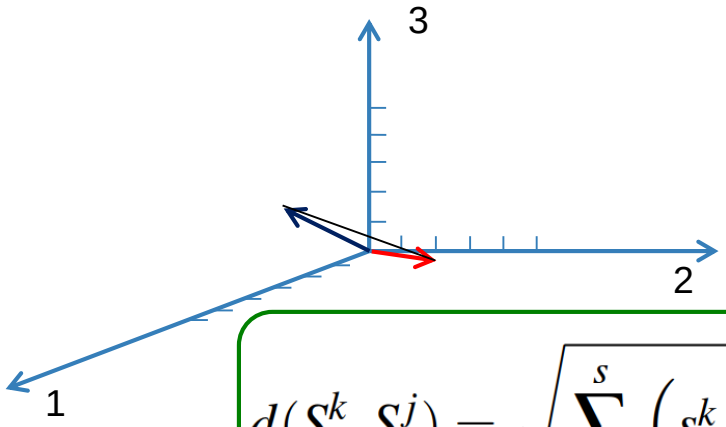
Topology



Topology

Euclidean

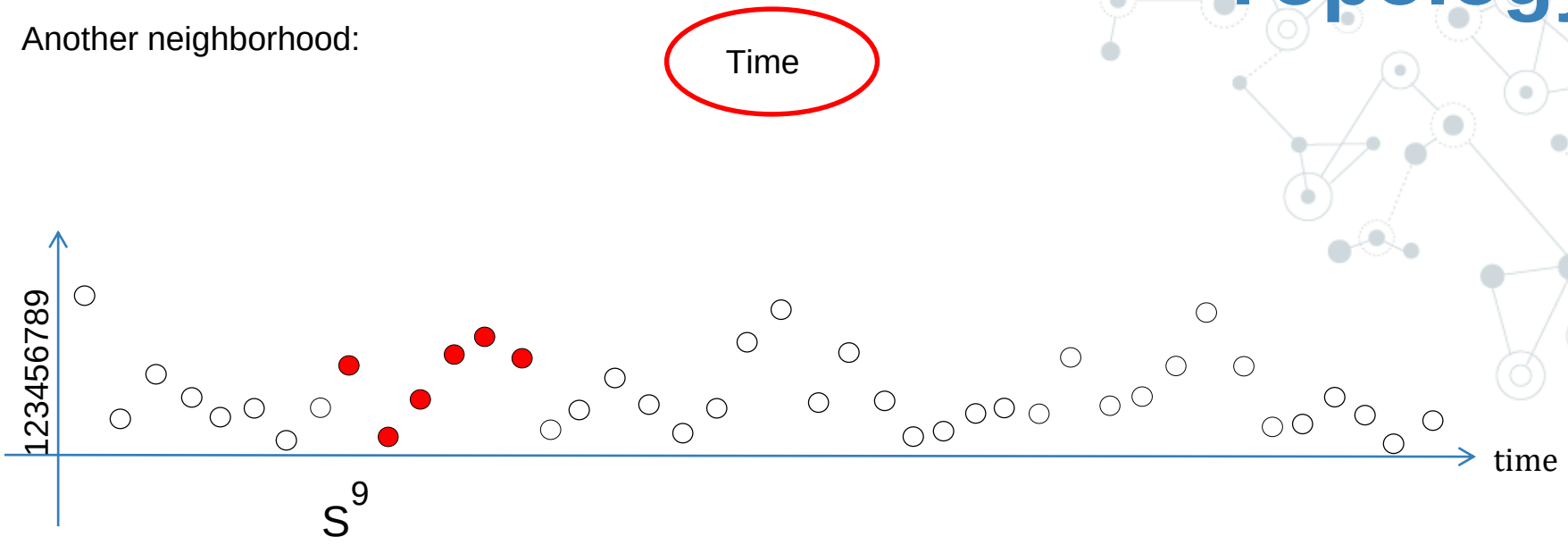




$$d(S^k, S^j) = \sqrt{\sum_{n=1}^{s} \left(s_n^k - s_n^j\right)^2}$$

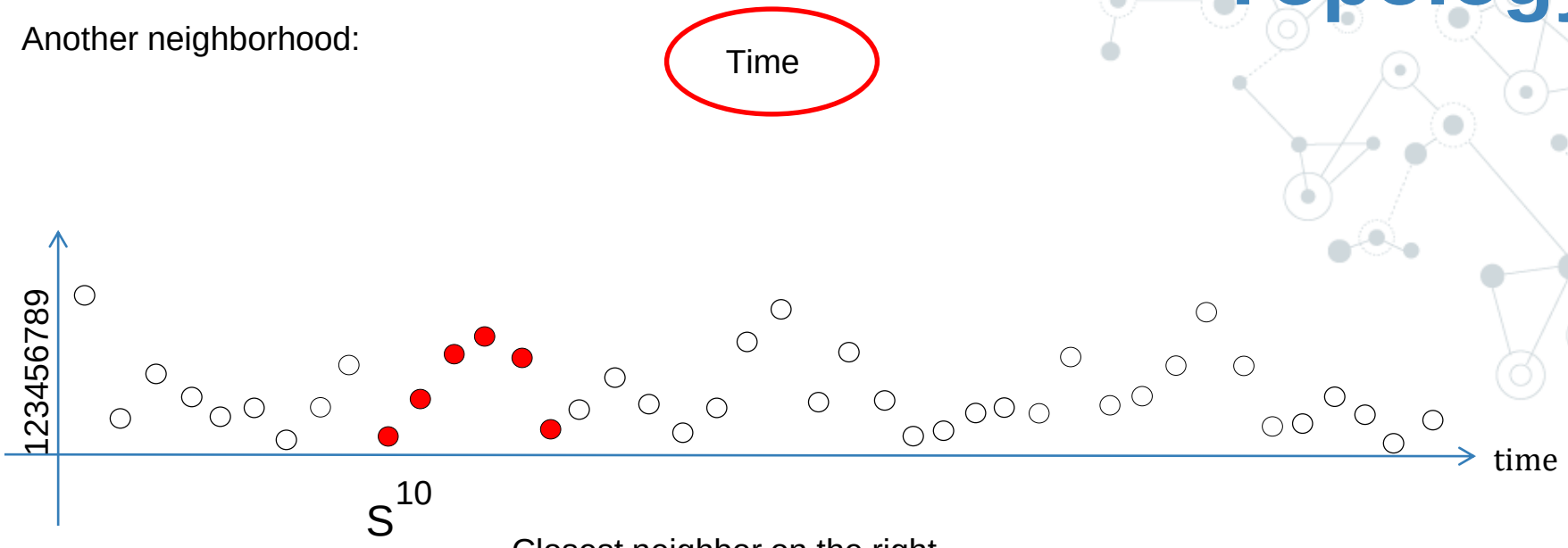
Topology

Another neighborhood:



Topology

Another neighborhood:

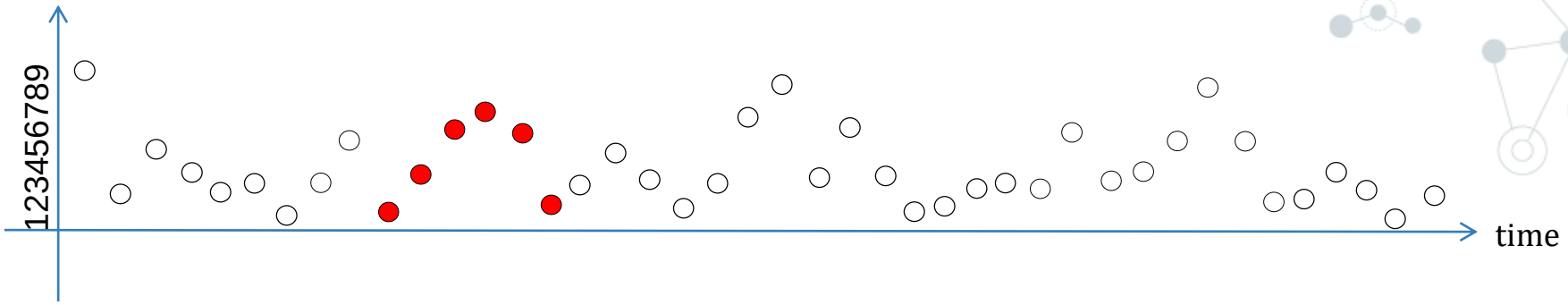


Closest neighbor on the right

Topology

Another neighborhood:

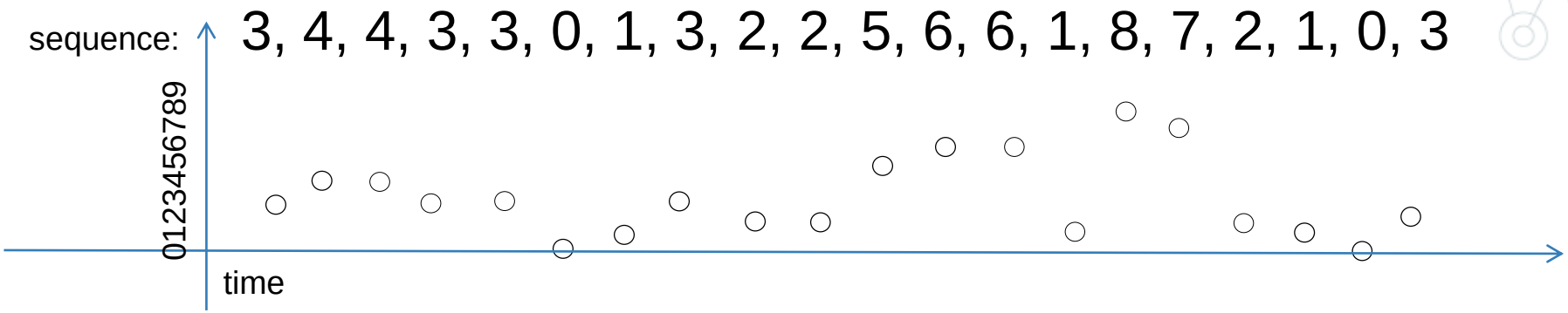
Time



$$d_t(S^k, S^j) = |k - j|$$

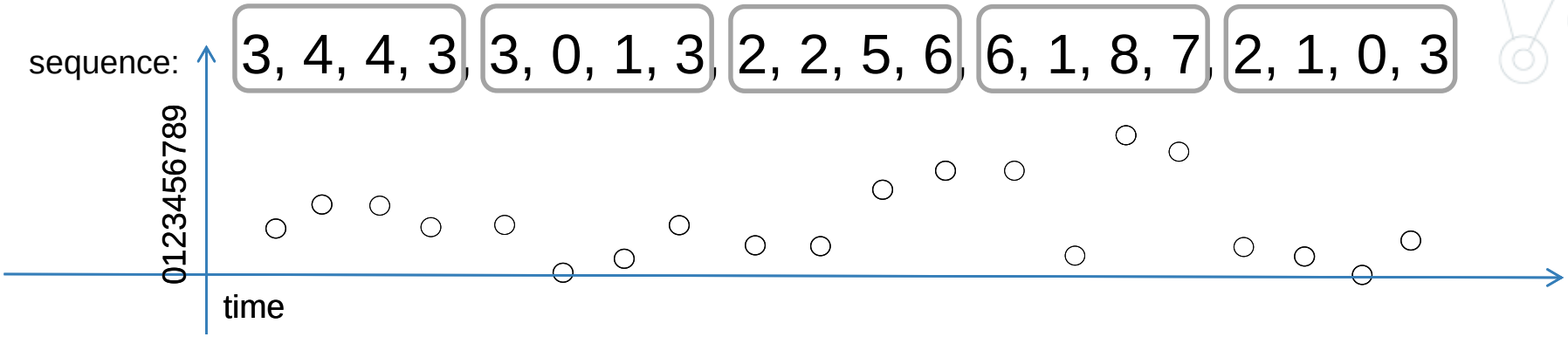
Topology

Symbolic
Aggregate
appro**X**imation



Topology

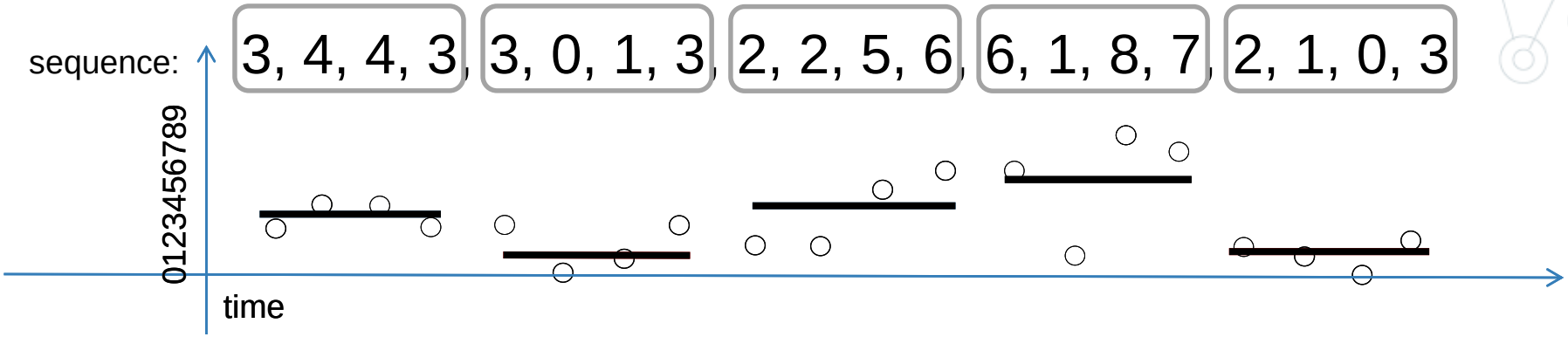
Symbolic
Aggregate
appro**X**imation



divide the sequence in parts

Topology

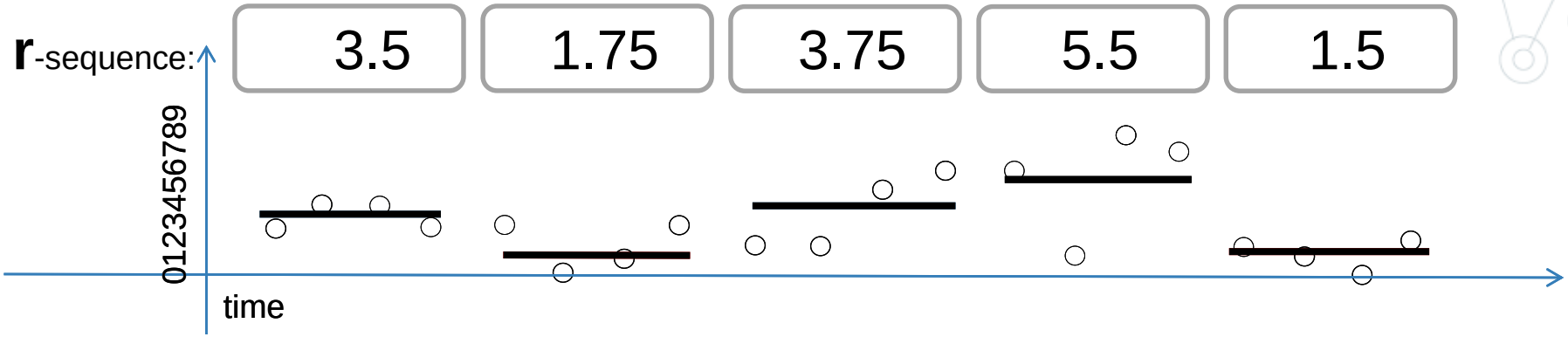
Symbolic
Aggregate
appro**X**imation



Piecewise Aggregate Approximation (PAA)

Topology

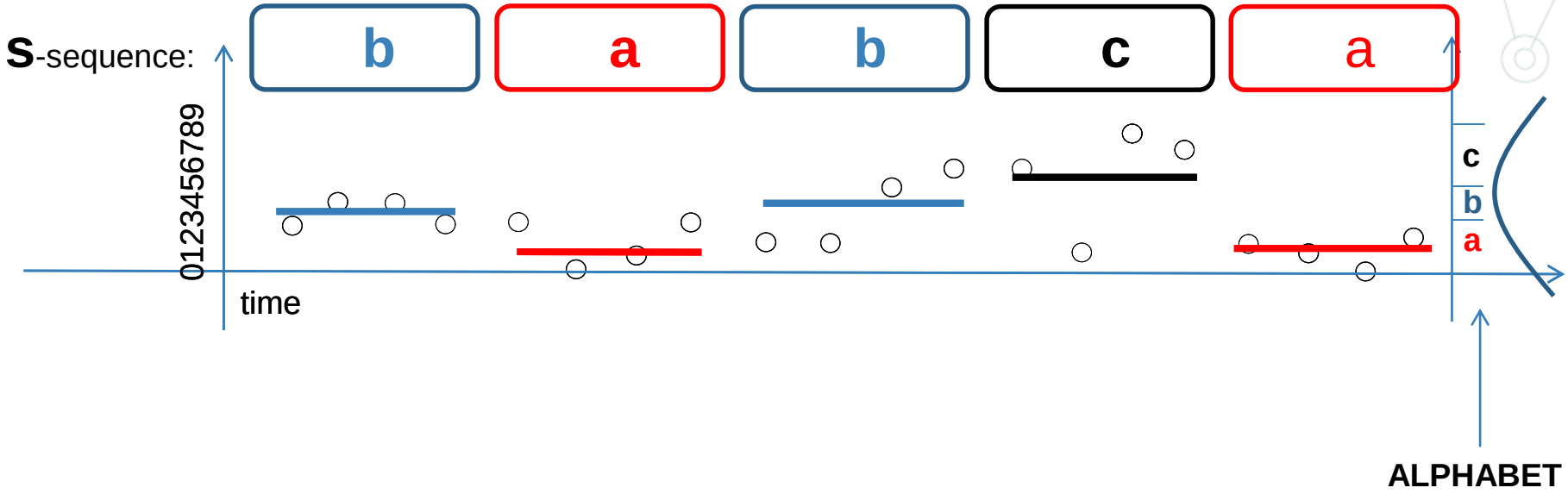
Symbolic
Aggregate
appro**X**imation



Piecewise Aggregate Approximation (PAA)

Topology

Symbolic
Aggregate
appro**X**imation



Topology

Symbolic
Aggregate
appro**X**imation

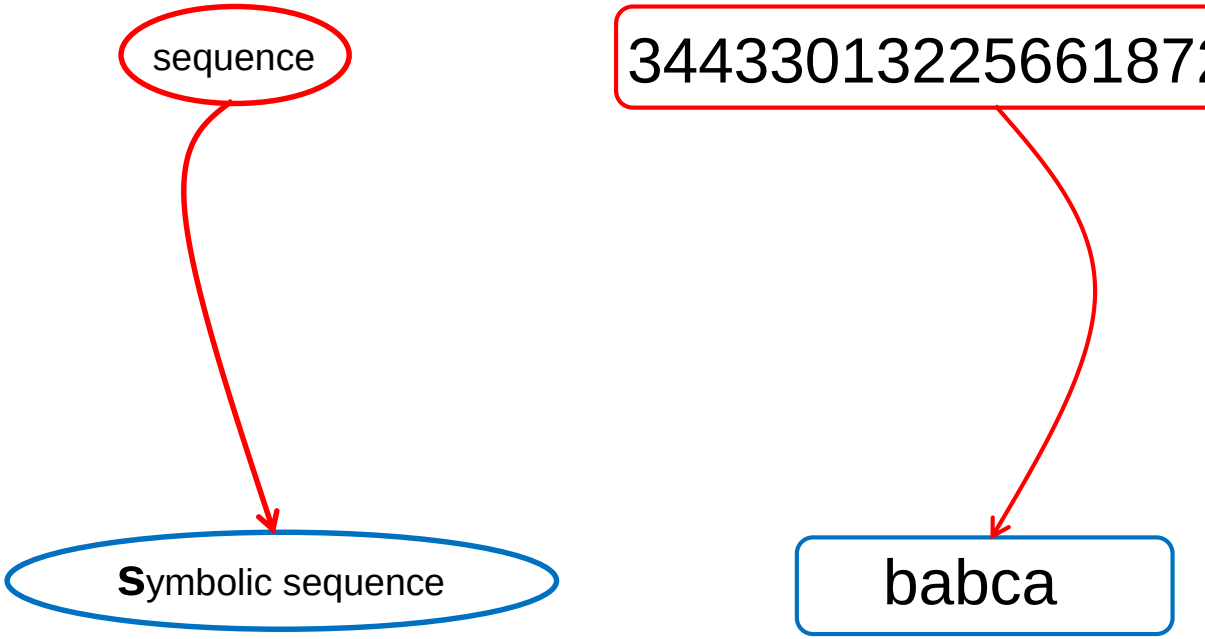
sequence

34433013225661872103

with **2** parameters:
-the number of sub-sequences
- the number of symbols

Topology

Symbolic
Aggregate
appro**X**imation



Topology

A different form of Neighborhood:

Symbolic
Aggregate
appro**X**imation

Many sequences

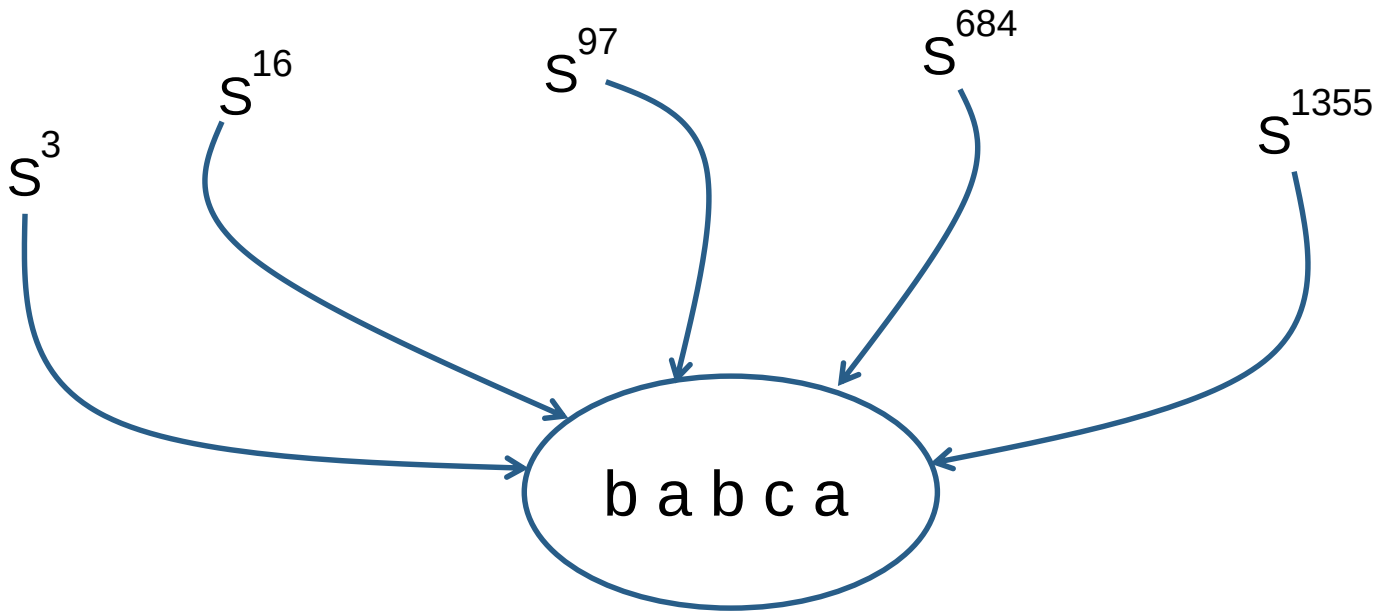
S^{3} S^{16} S^{97} S^{684} S^{1355}

Topology

A different form of Neighborhood:

Symbolic
Aggregate
appro**X**imation

Many sequences



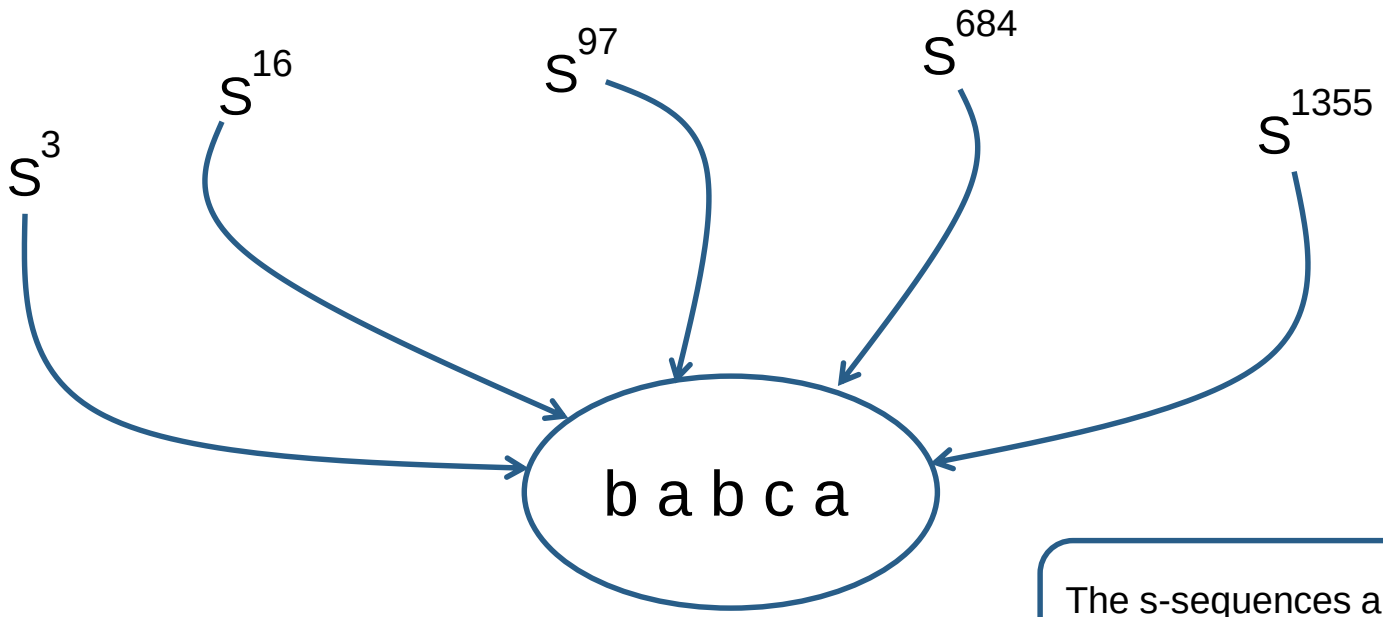
Symbolic-sequence

Topology

A different form of Neighborhood:

Symbolic
Aggregate
appro**X**imation

Many sequences



Symbolic-sequence

The s-sequences are natural **clusters** of sequences!

Topology

Topology:
Who is my neighbor?

Euclidean

$$d(S^k, S^j) = \sqrt{\sum_{n=1}^{s} \left(s_n^k - s_n^j\right)^2}$$

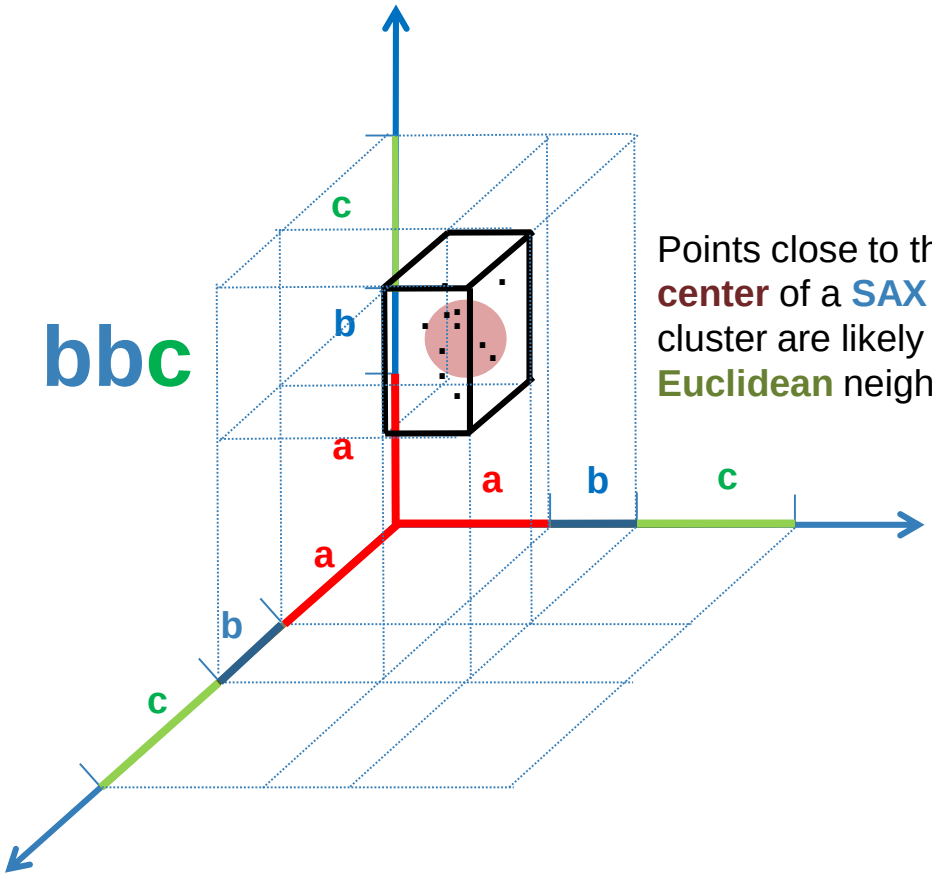
Symbolic
Aggregate
appro**X**imation

"SAX neighbors"
are sequences belonging to
same s-sequence

Time

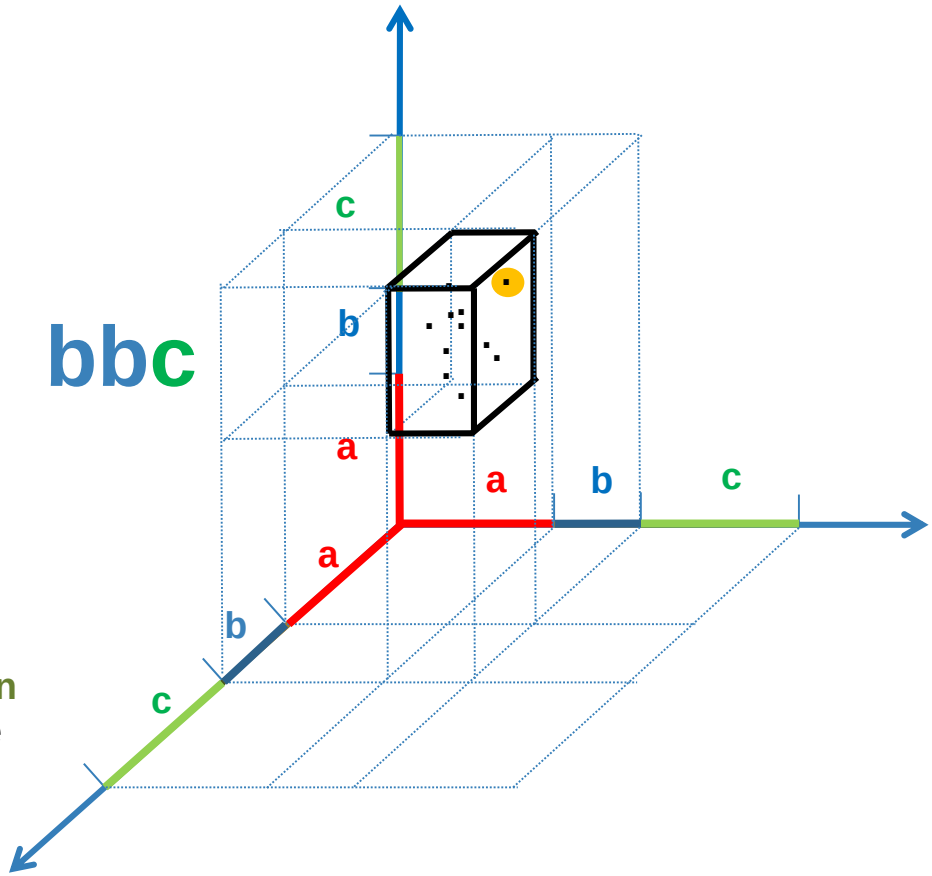
$$d_t(S^k, S^j) = |k - j|$$

Topology



Points close to the **center** of a **SAX** cluster are likely **Euclidean** neighbors

Topology



For points close to the **border**, the **SAX**-**Euclidean** connection is more loose!

HOT SAX, an algorithm for quickly finding **the** sequence with the **highes**t *nnd*

SAX **EUCLIDEAN**

Algorithm

SAX EUCLIDEAN Algorithm

HOT SAX, an algorithm for quickly finding **the** sequence with the **highes**t *nnd*

Outer loop:

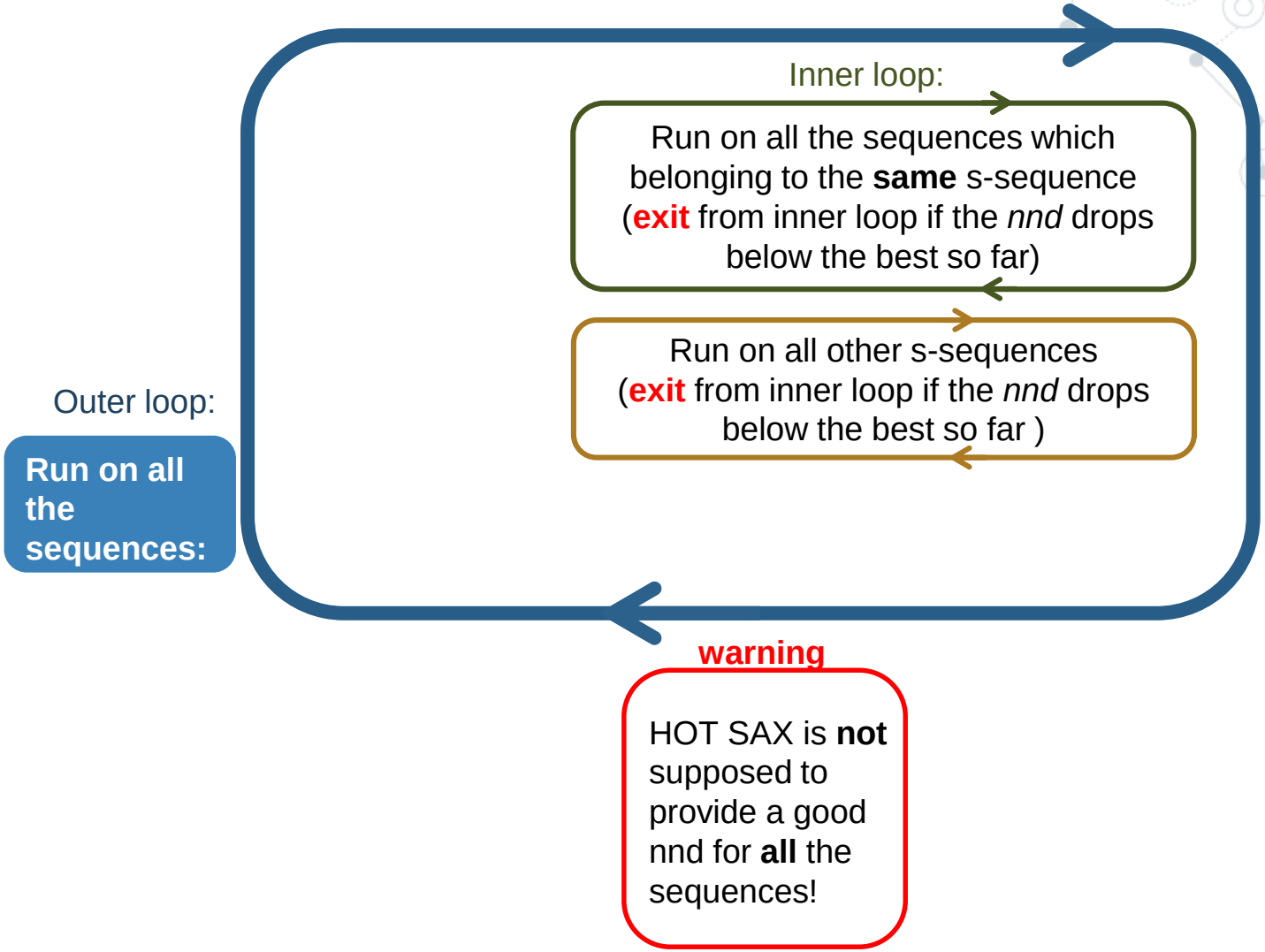
Run on all the sequences:

Inner loop:

Run on all the sequences which belonging to the **same** s-sequence (**exit** from inner loop if the *nnd* drops below the best so far)

Run on all other s-sequences (**exit** from inner loop if the *nnd* drops below the best so far)

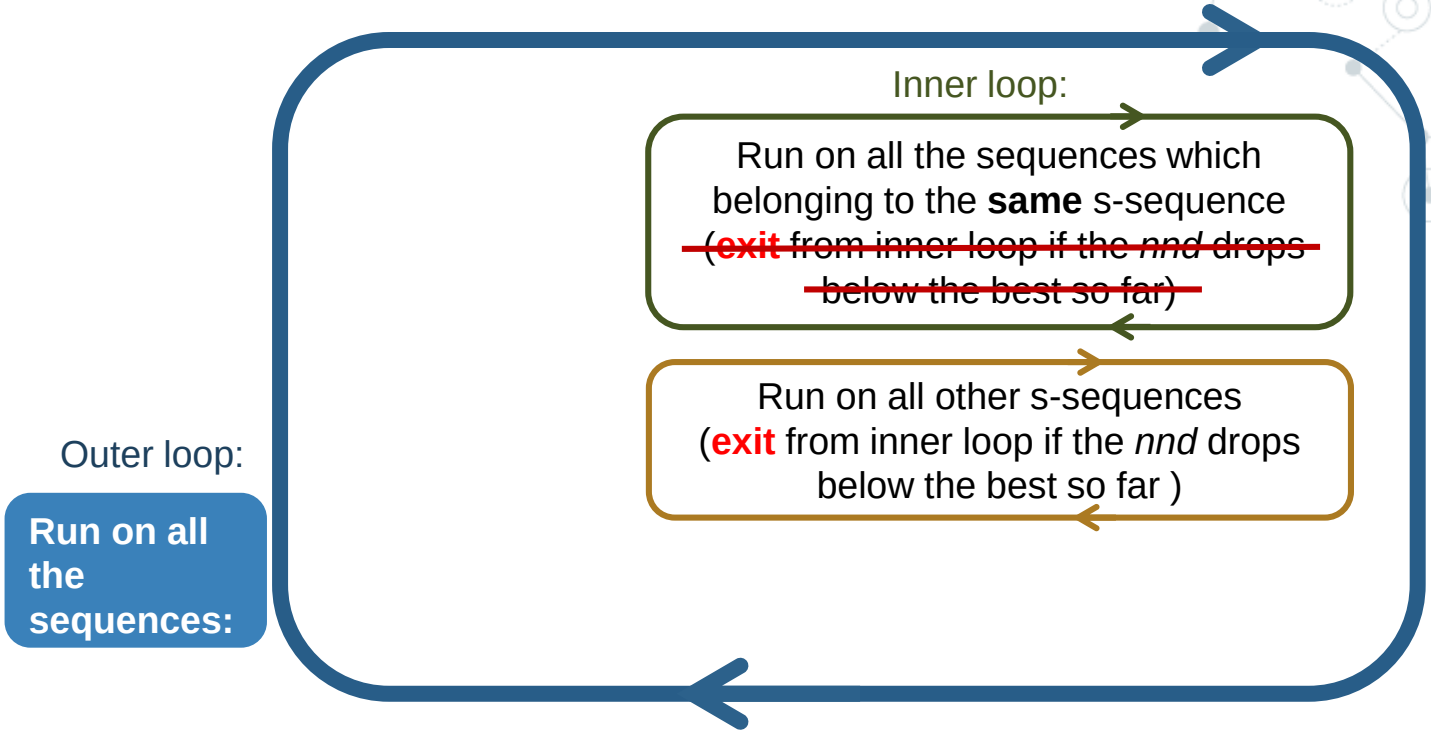
SAX EUCLIDEAN Algorithm



warning

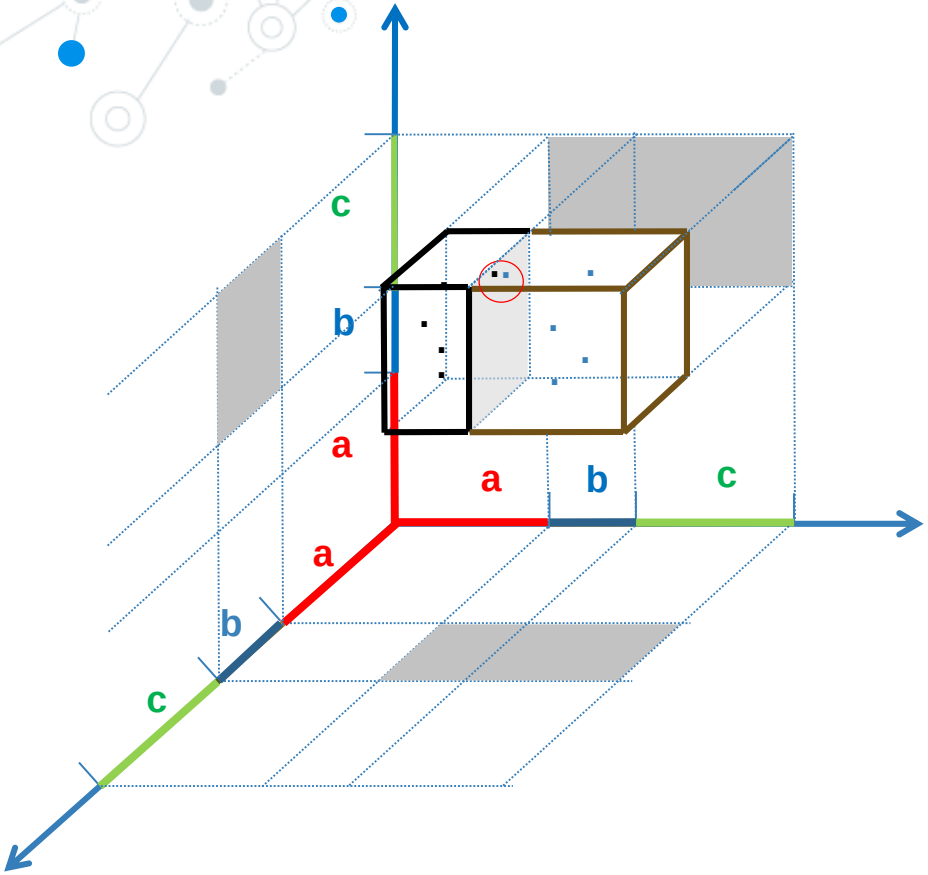
HOT SAX is **not** supposed to provide a good nnd for **all** the sequences!

Algorithm

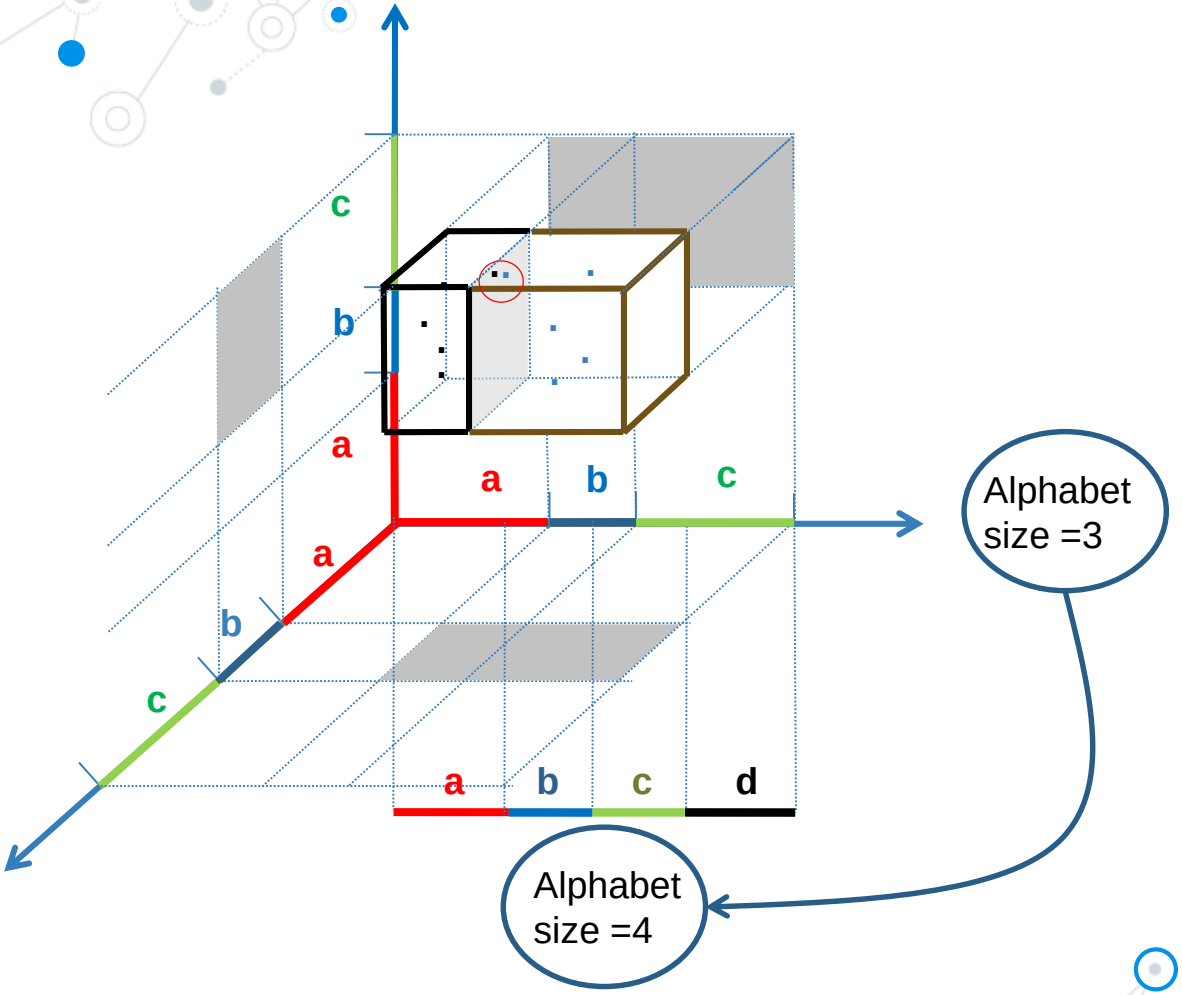


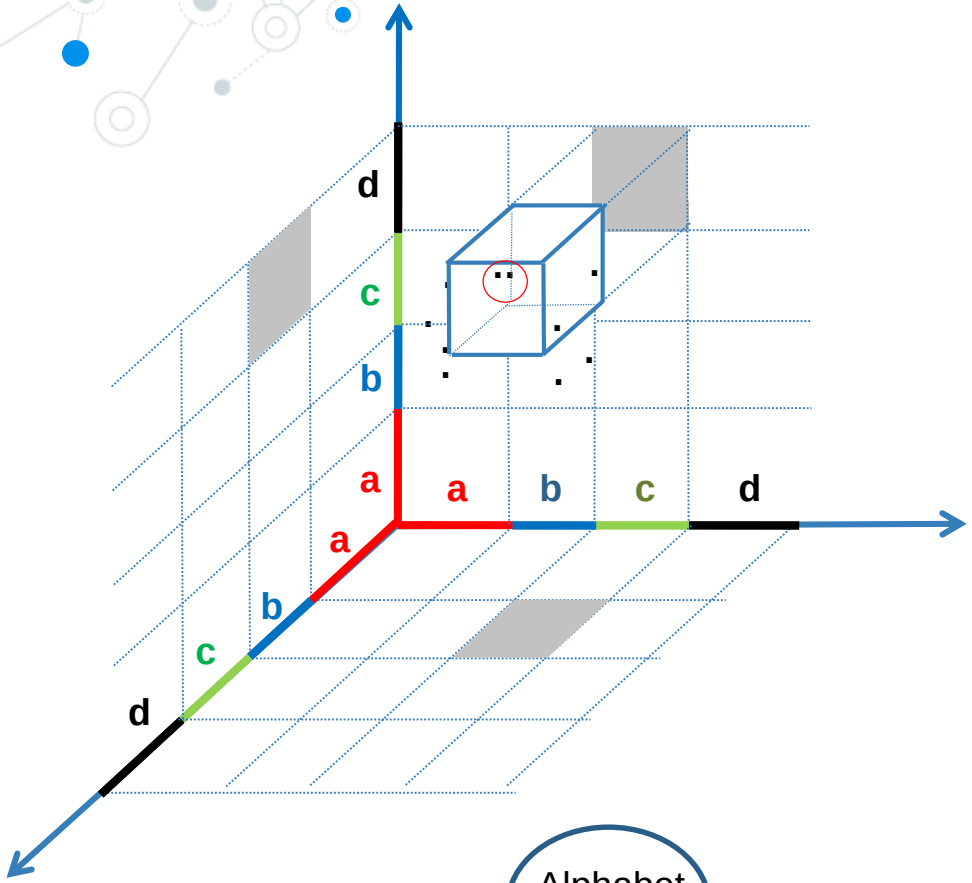
1

Let's force a full run on the same cluster!



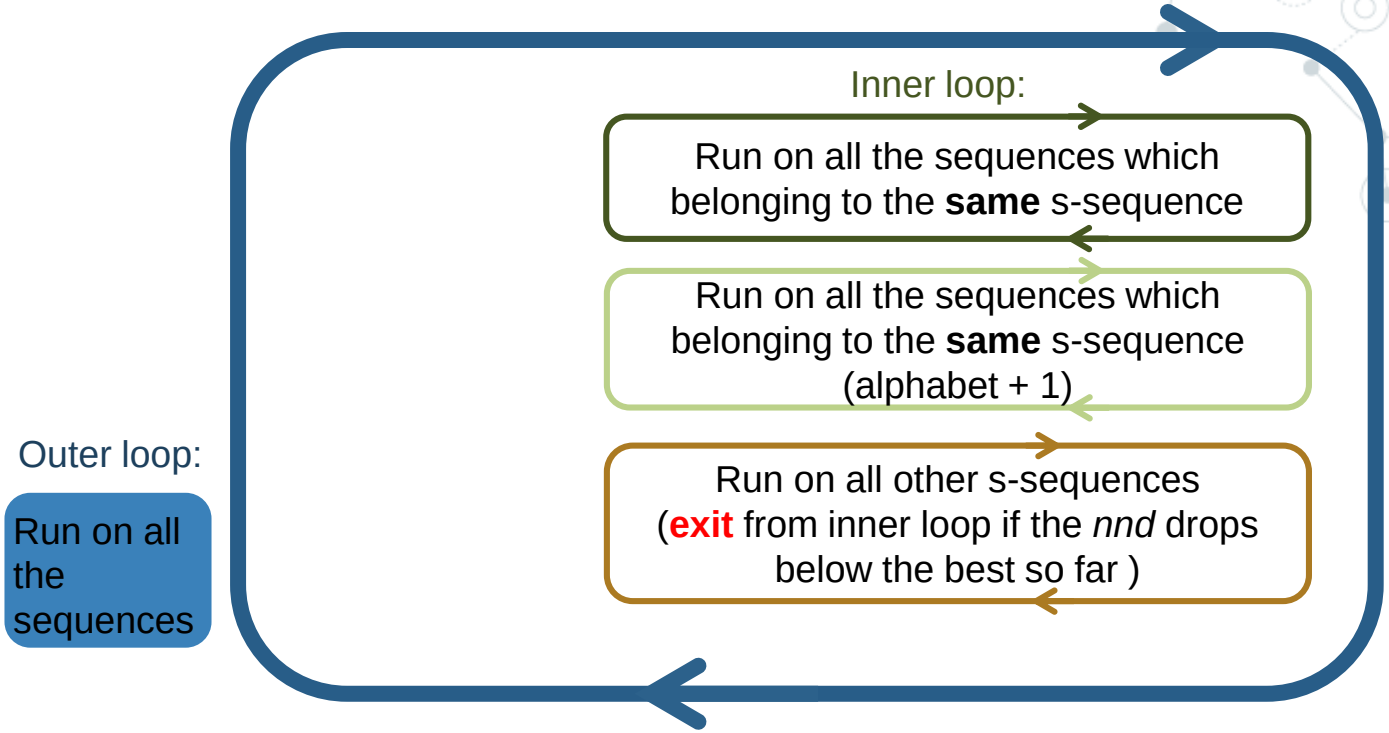
Alphabet
size =3


c
b
a
a
b
c
a
b
c
a
b
c
d
Alphabet
size =3
Alphabet
size =4



Alphabet size =4

Algorithm



2

Let's force a full run on the same cluster, but with alphabet +1

Unfortunately as the **dimension** of the clusters increases, more and more points are doomed to be **close to the borders**!

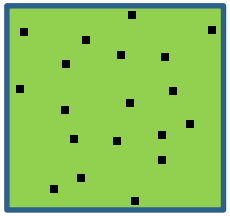
CURSE

s-sequences ⟷ hypercubes

CURSE

Unfortunately as the **dimension** of the clusters increases more and more points are doomed to be **close to the borders**!

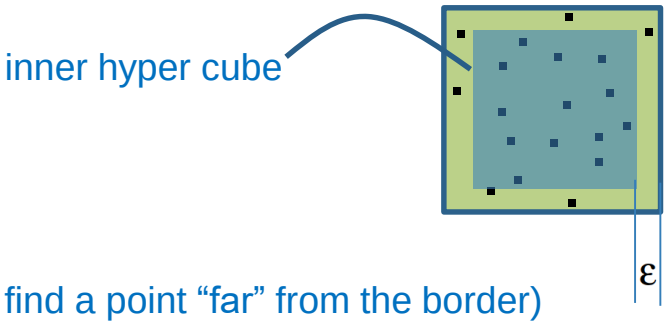
s-sequences ⟷ hypercubes



CURSE

Unfortunately as the **dimension** of the clusters increases more and more points are doomed to be **close to the borders**!

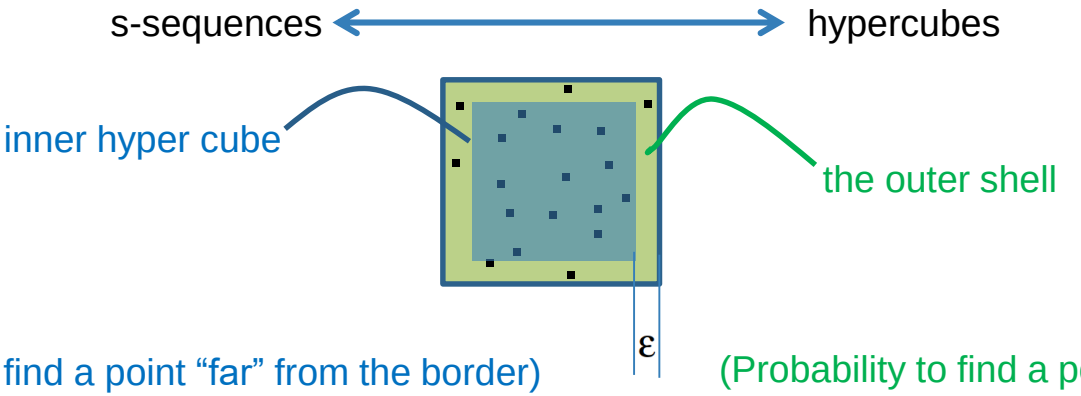
s-sequences ⟷ hypercubes



(probability to find a point “far” from the border)

CURSE

Unfortunately as the **dimension** of the clusters increase more and more points are doomed to be **close to the borders**!

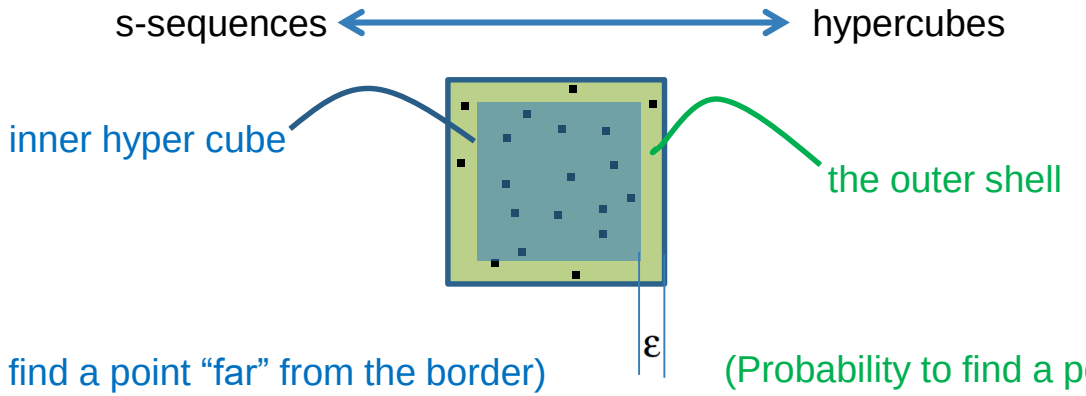


(probability to find a point “far” from the border)

(Probability to find a point close to the border)

CURSE

Unfortunately as the **dimension** of the clusters increase more and more points are doomed to be **close to the borders**!



(probability to find a point “far” from the border)

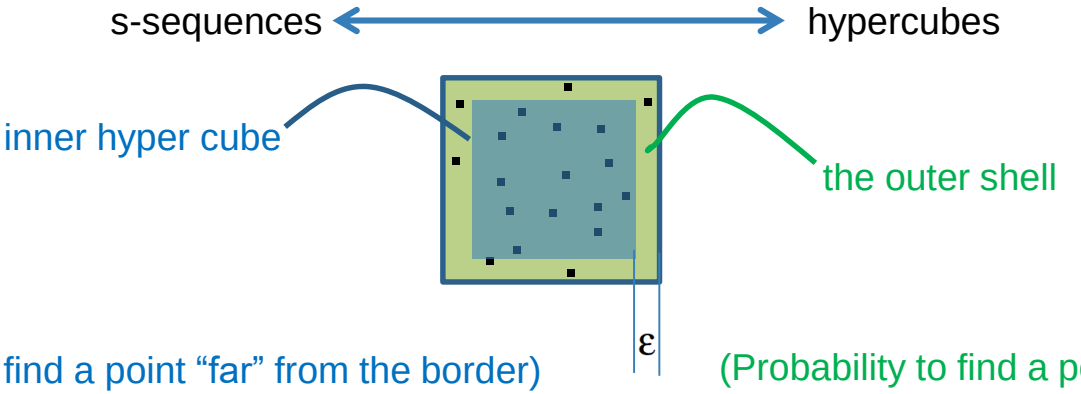
(Probability to find a point close to the border)

$$\frac{\text{inner volume}}{\text{volume}} = \left(\frac{l-2\varepsilon}{l}\right)^n \xrightarrow{n\to\infty} 0$$

For higher dimensions most of the volume of a hypercube is close to its borders!

CURSE

Unfortunately as the **dimension** of the clusters increase more and more points are doomed to be **close to the borders**!

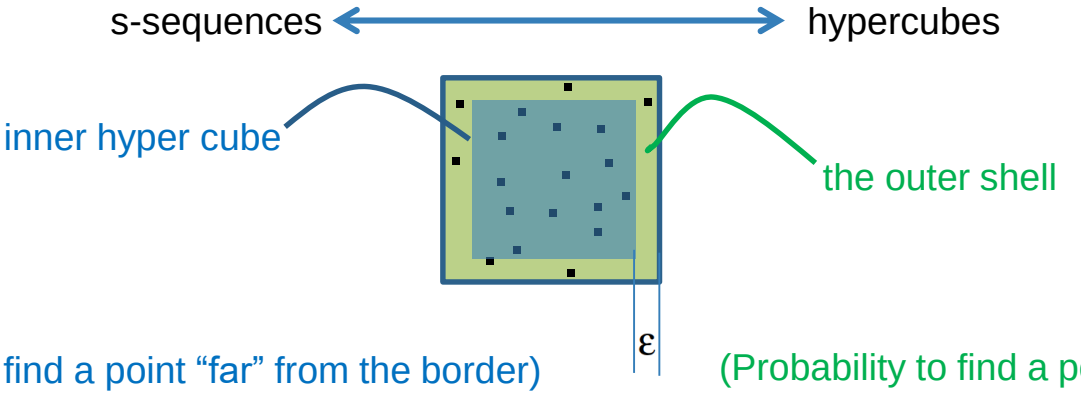


Curse of dimensionality

As n grows the **SAX**-**Euclidean** connection neighborhood becomes looser

CURSE

Unfortunately as the **dimension** of the clusters increase more and more points are doomed to be **close to the borders**!



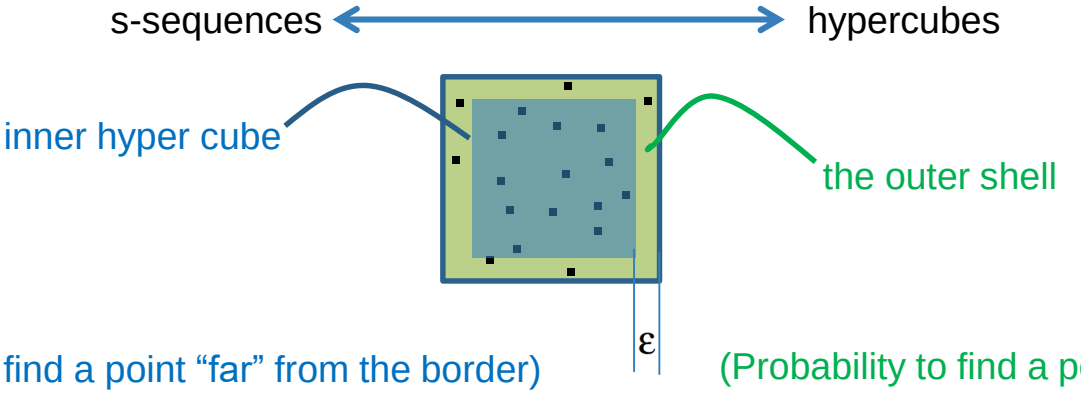
Curse of dimensionality

As n grows the **SAX**-**Euclidean** connection neighborhood becomes looser

There are too many close cluster!

CURSE

Unfortunately as the **dimension** of the clusters increase more and more points are doomed to be **close to the borders**!



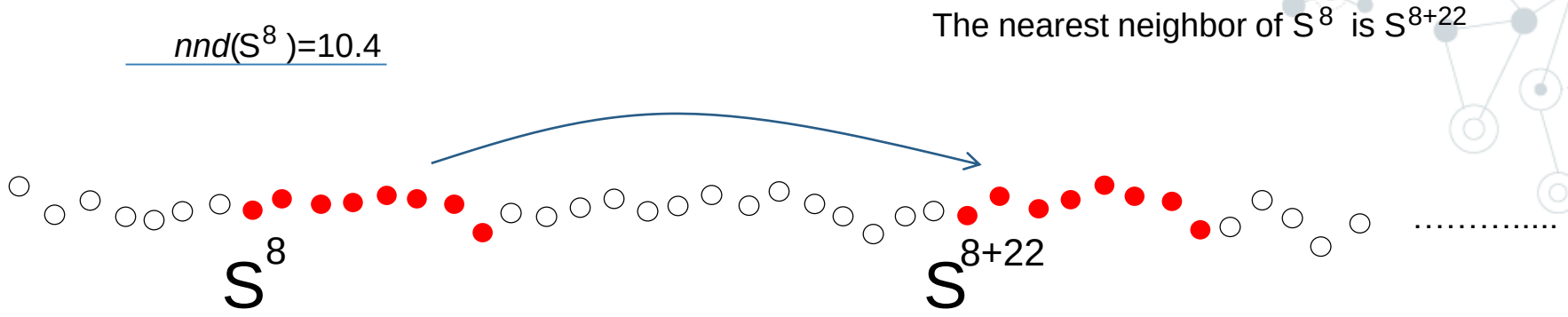
Curse of dimensionality

As n grows the **SAX**-**Euclidean** connection neighborhood becomes looser

There are too many close cluster!

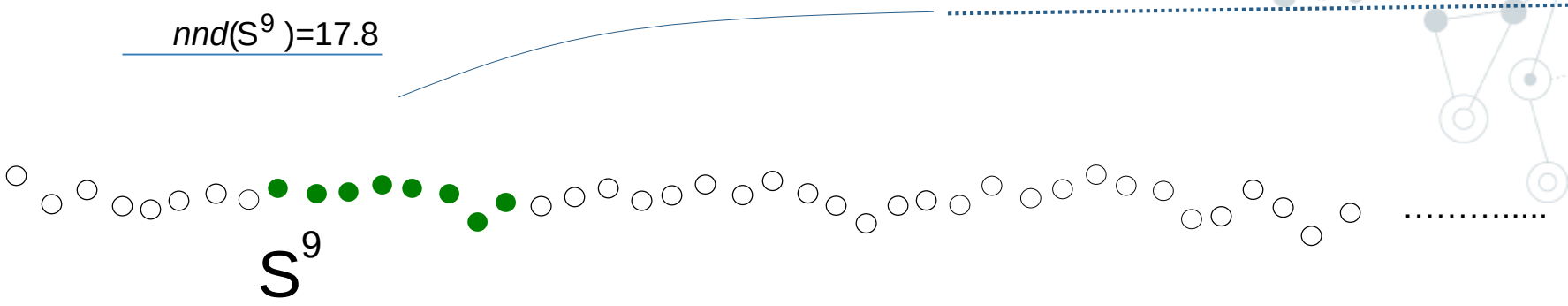
Next?

Solution: time topology

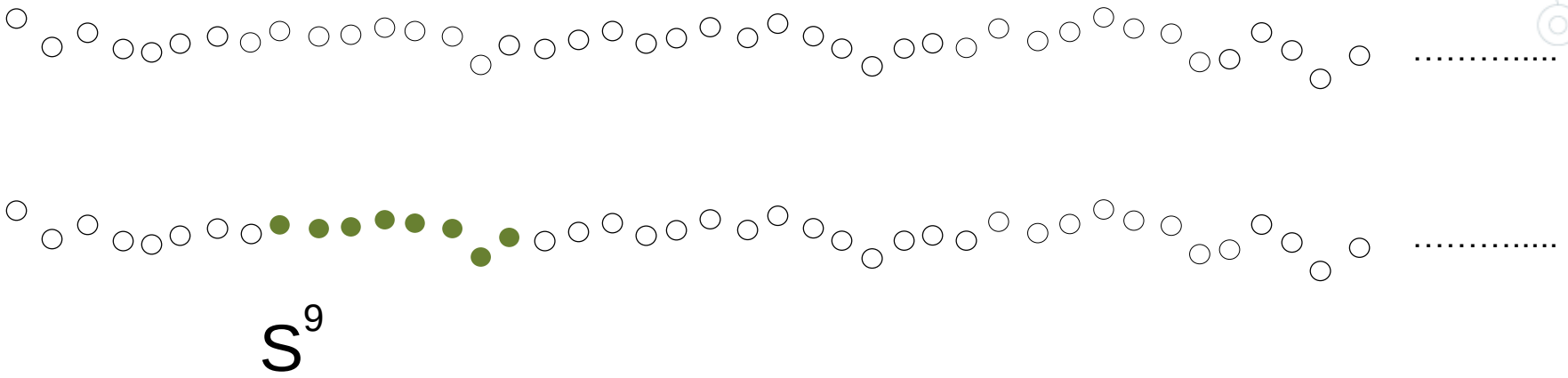


Solution: time topology

The nearest neighbor of S^9 is S^{453}



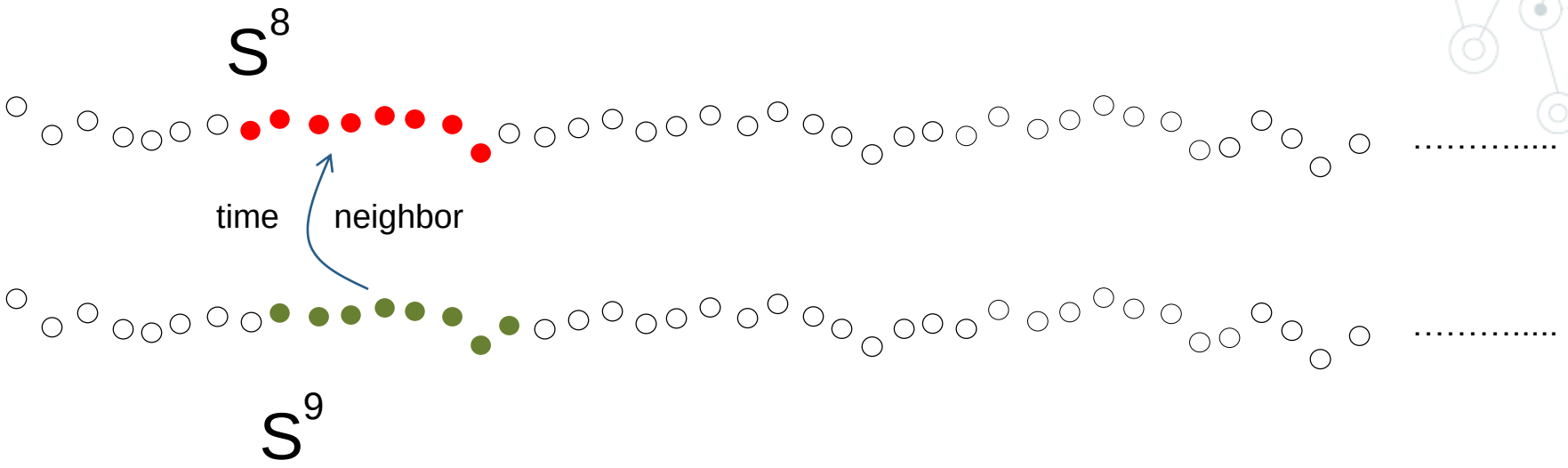
Solution: time topology



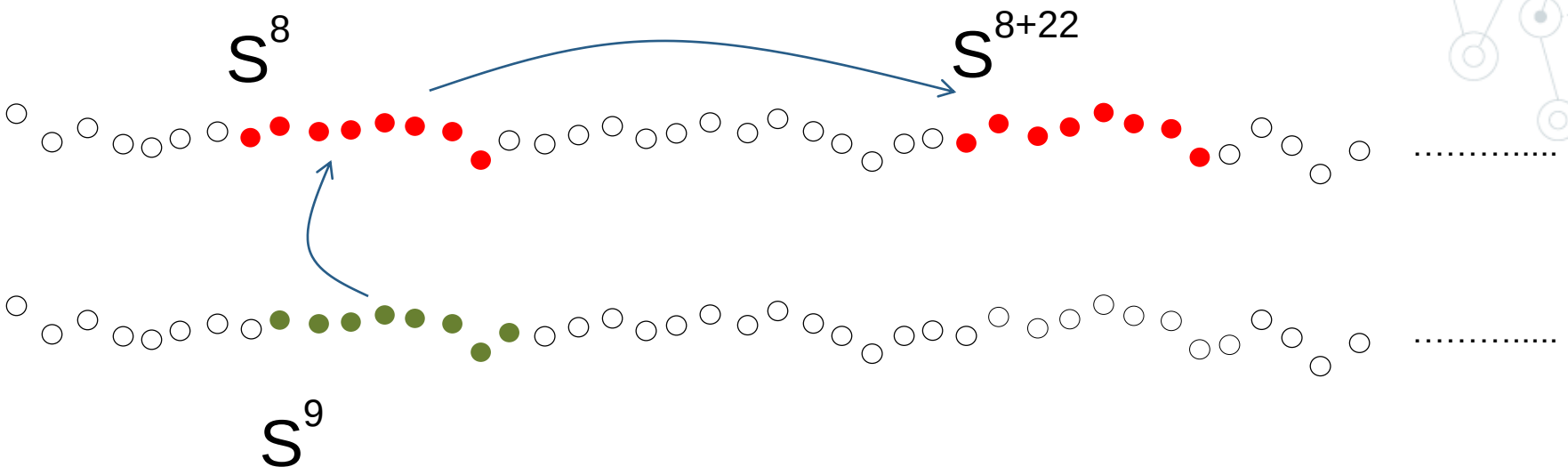
S^9

Solution: time topology

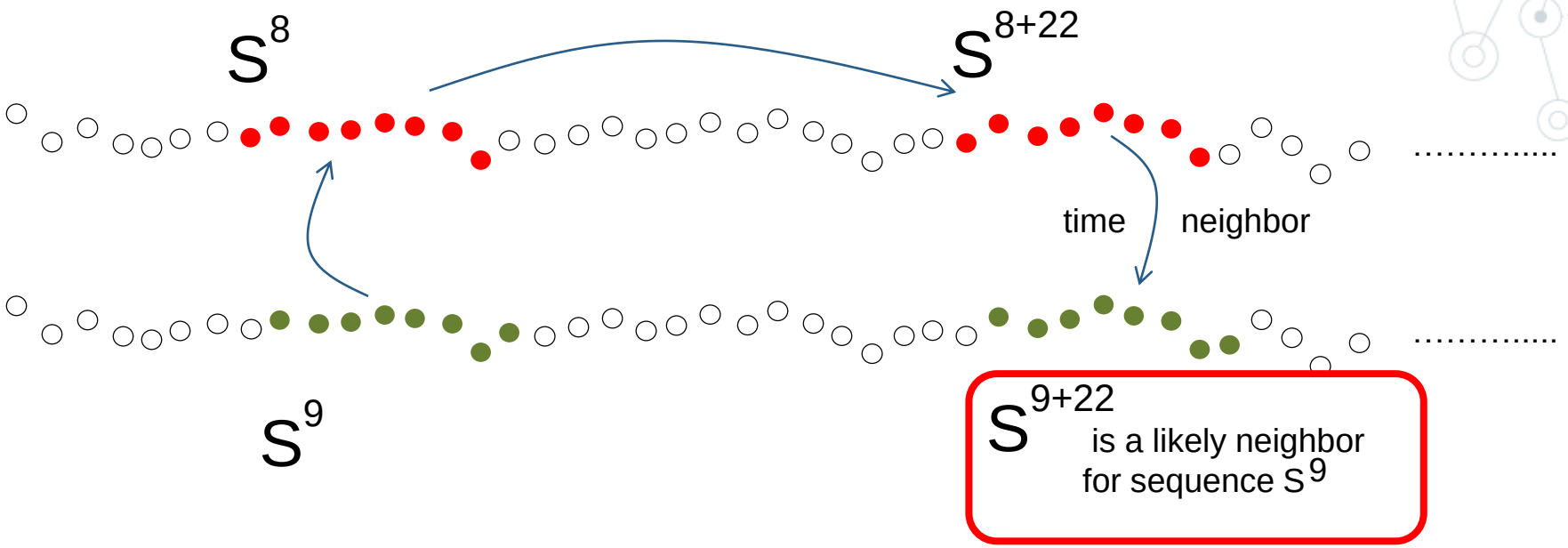
Since $nnd(S^8) << nnd(S^9)$



Solution: time topology

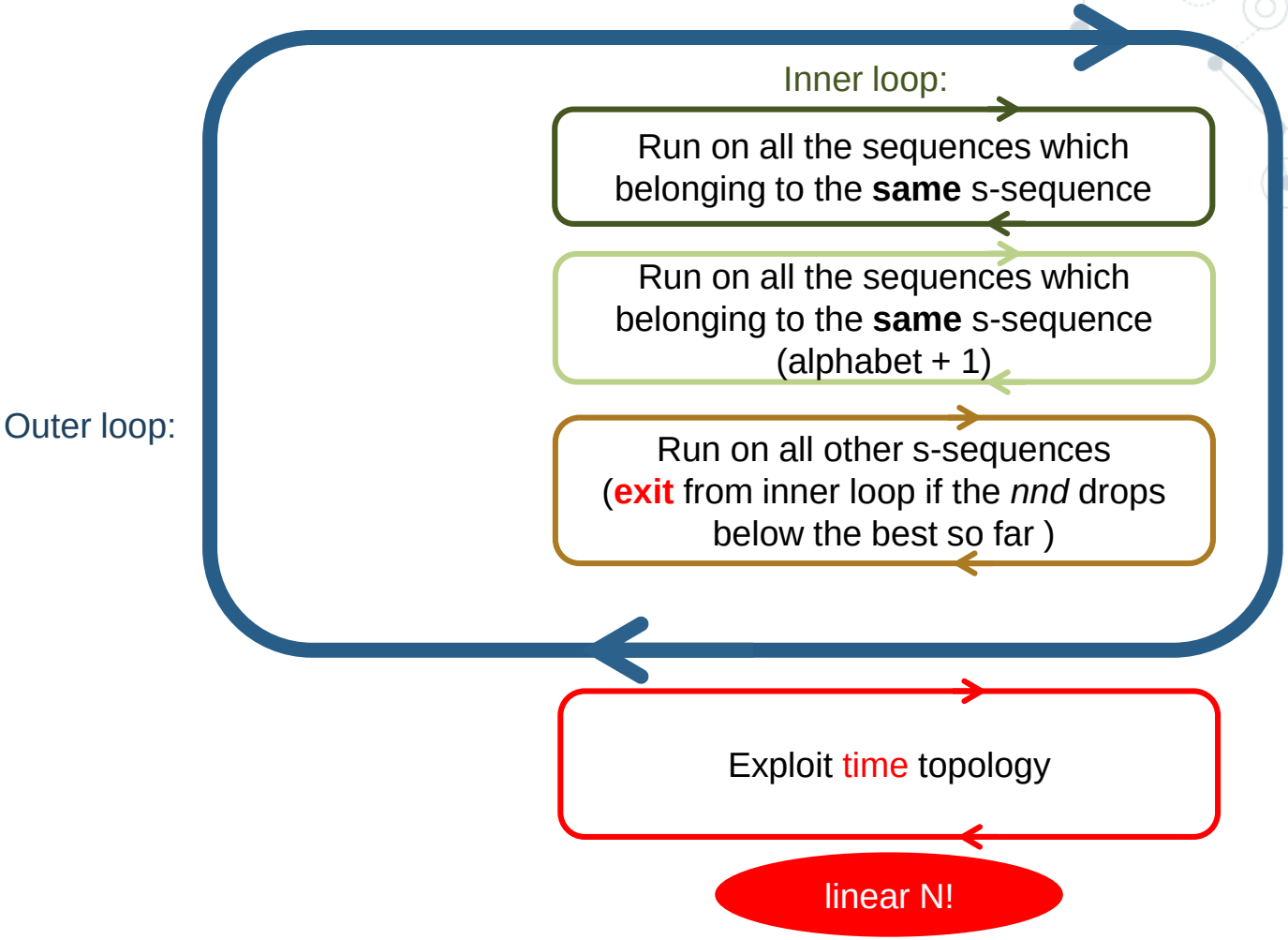


Solution: time topology



$$nnd(S^{i+1}) \leq d_2(S^{i+1}, S^{i+k+1}) = nnd(S^i) + (p_{i+s} - p_{i+s+k})^2 - (p_i - p_{i+k})^2$$

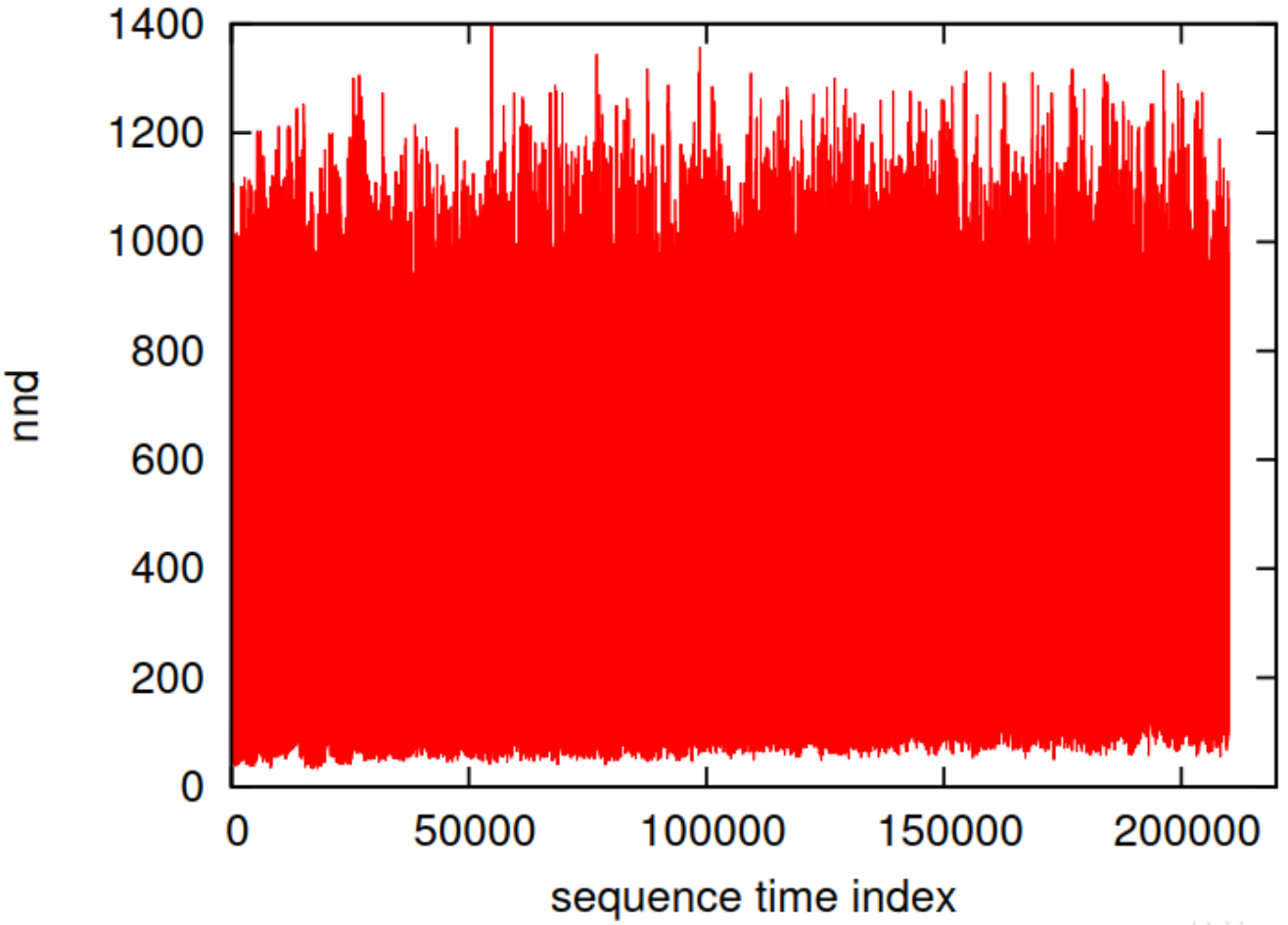
Algorithm



3

Improve the *nnd*s with the time topology

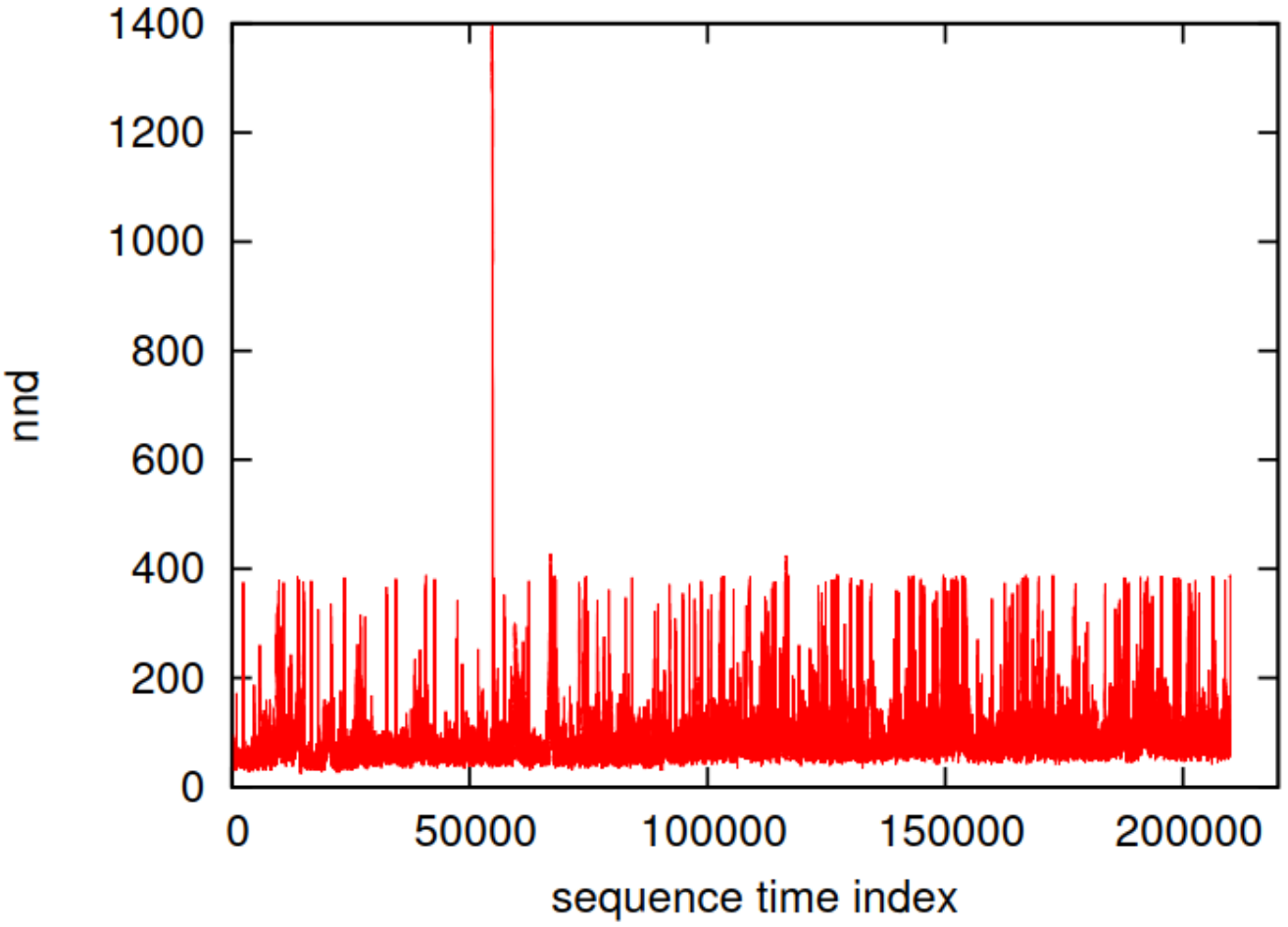
VALIDATION



0

The approx. *nnd*s obtained with HOT SAX

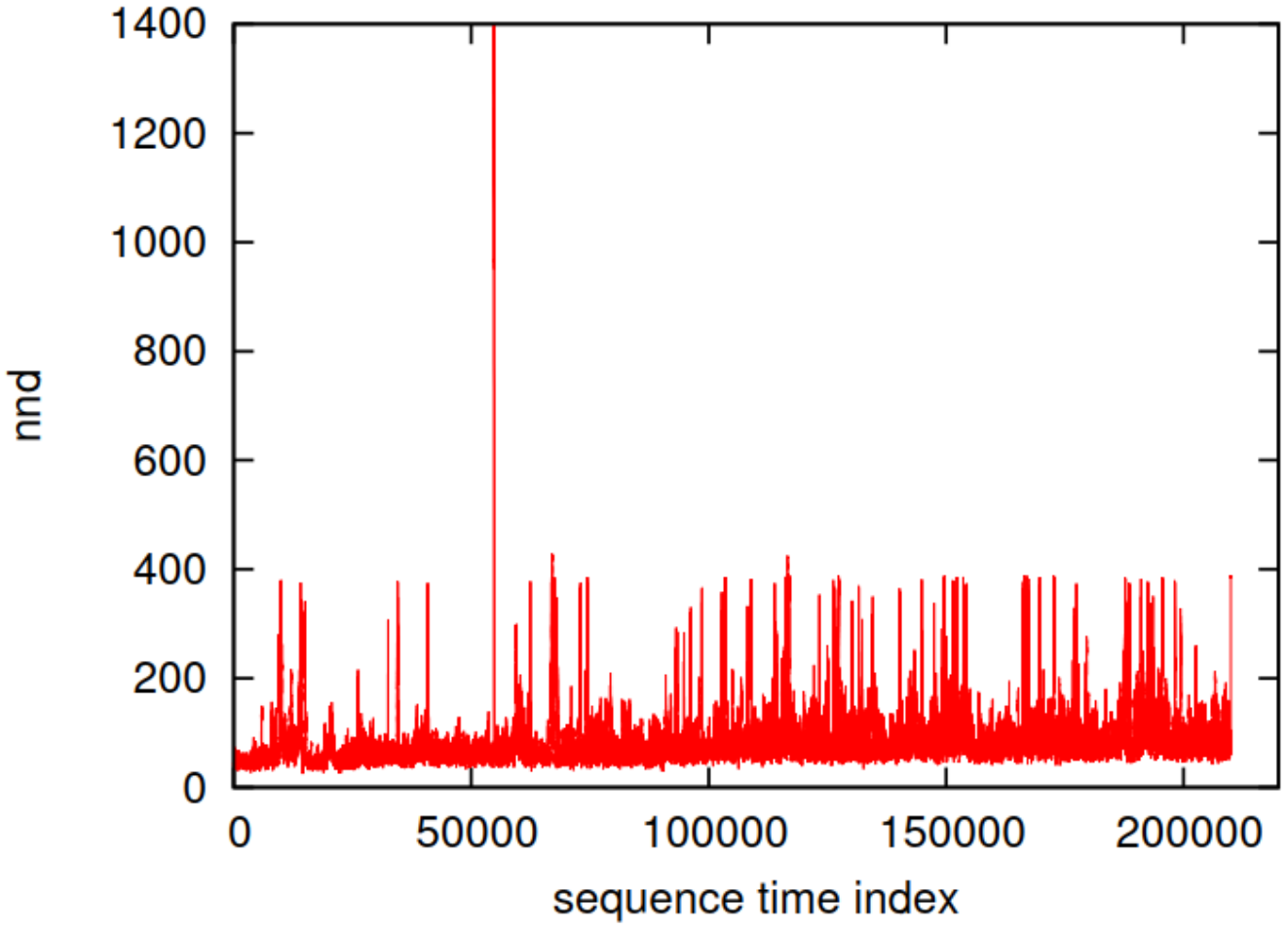
VALIDATION



1

Let's force a full run on the same cluster!

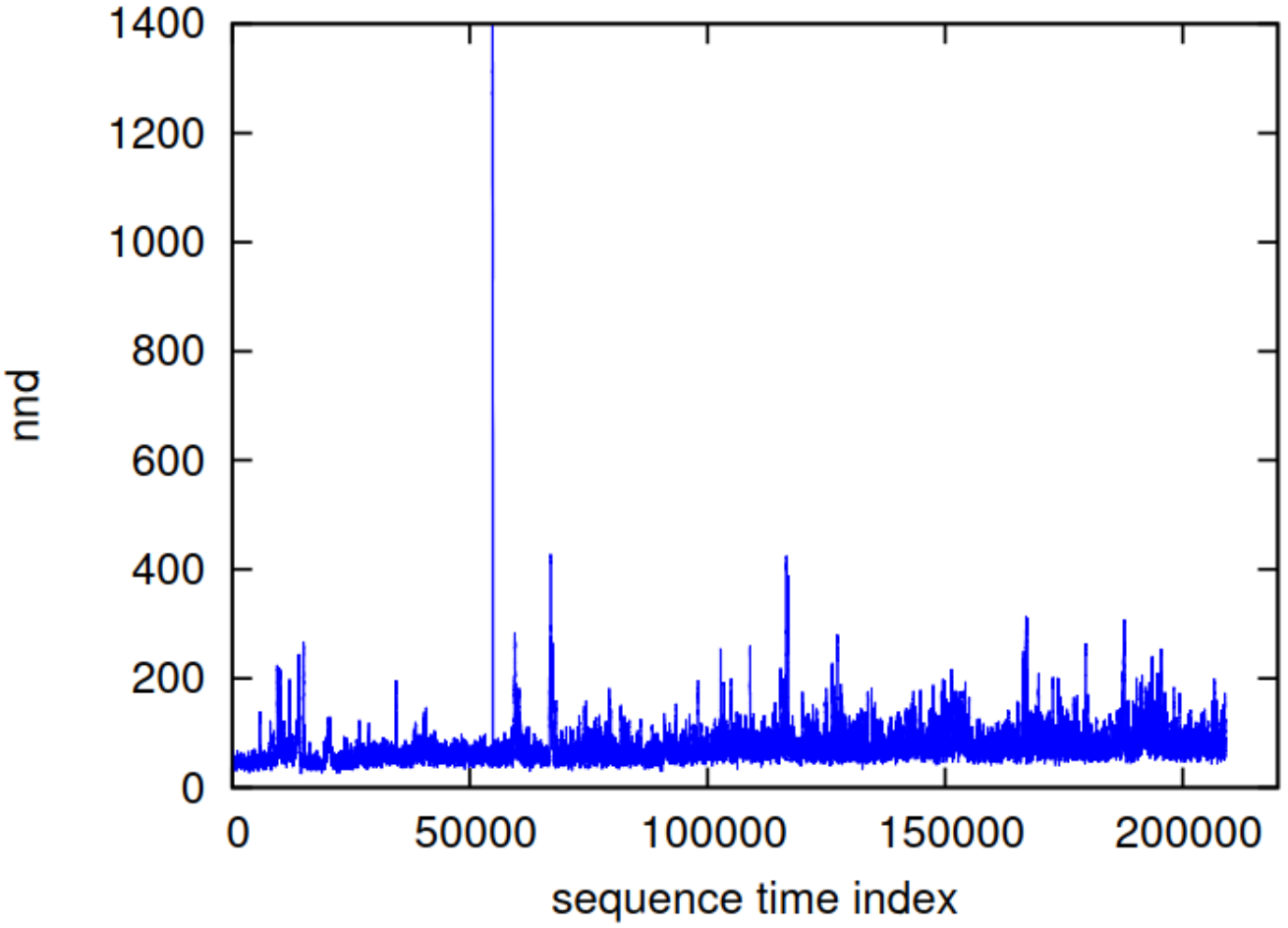
VALIDATION



2

Let's force a full run on the same cluster, but with alphabet +1

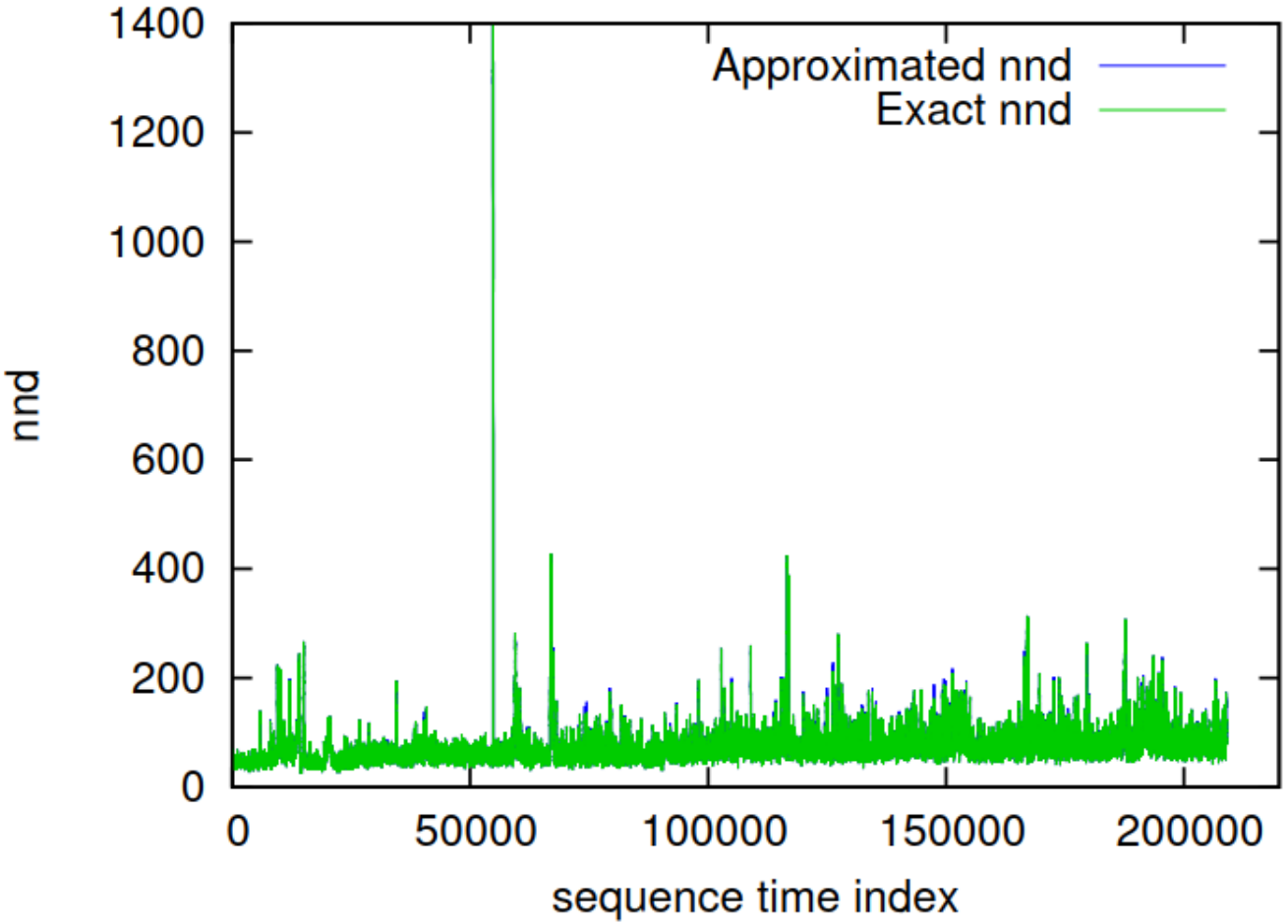
VALIDATION



3

Improve the *nnd*s with the time topology

VALIDATION


Approximated nnd
Exact nnd
nnd
sequence time index
0
200
400
600
800
1000
1200
1400
0
50000
100000
150000
200000

VALIDATION

file name	% of exact *nnds*	Err	speedup
$sel0606_2$	99.6	0.05	83
$sel0606_3$	99.5	0.05	116
$sel102_2$	98.8	0.05	40
$sel102_3$	99.7	0.06	26
$sel123_2$	98.8	0.05	32
$sel123_3$	99.2	0.04	14
$bidmc15_2$	99.6	0.15	5
$bidmc15_3$	99.8	0.09	30
$bidmc15_4$	98.4	0.05	40
$bidmc15_5$	98.9	0.06	58

VALIDATION

Err=Average Relative Error

file name	% of exact *nnds*	Err	speedup
$sel0606_2$	99.6	0.05	83
$sel0606_3$	99.5	0.05	116
$sel102_2$	98.8	0.05	40
$sel102_3$	99.7	0.06	26
$sel123_2$	98.8	0.05	32
$sel123_3$	99.2	0.04	14
$bidmc15_2$	99.6	0.15	5
$bidmc15_3$	99.8	0.09	30
$bidmc15_4$	98.4	0.05	40
$bidmc15_5$	98.9	0.06	58

$$Err = \frac{1}{N_a} \sum_{i=1}^{N} \frac{nnd_a(S^i) - nnd(S^i)}{nnd(S^i)}$$

approximate *nnd*

exact *nnd*

N_a= number of sequences for which only an approximate *nnd* has been found

VALIDATION

file name	% of exact *nnds*	Err	speedup
$sel0606_2$	99.6	0.05	83
$sel0606_3$	99.5	0.05	116
$sel102_2$	98.8	0.05	40
$sel102_3$	99.7	0.06	26
$sel123_2$	98.8	0.05	32
$sel123_3$	99.2	0.04	14
$bidmc15_2$	99.6	0.15	5
$bidmc15_3$	99.8	0.09	30
$bidmc15_4$	98.4	0.05	40
$bidmc15_5$	98.9	0.06	58

Conclusions

1. By interweaving **Euclidean**, **SAX**, and **Time** topologies, (starting from a discord search algorithm) we obtain a new algorithm for finding an **approximate** *nnd* profile for all the sequences of a time series

Conclusions

1. By interweaving **Euclidean**, **SAX**, and **Time** topologies, (starting from a discord search algorithm) we obtain a new algorithm for finding an **approximate** *nnd* profile for all the sequences of a time series

2. The evaluation we performed returns more than 98% of the **exact nnds,** and the values of the approximate *nnd*s are within 15% in respect to the exact ones**.**

Conclusions

1. By interweaving **Euclidean**, **SAX**, and **Time** topologies, (starting from a discord search algorithm) we obtain a new algorithm for finding an **approximate** *nnd* profile for all the sequences of a time series

2. The evaluation we performed returns more than 98% of the **exact nnds,** and the values of the approximate *nnd*s are within 15% in respect to the exact ones**.**

3. The speedups are between **1 and 2 orders** of magnitude in respect to the brute force

Conclusions

1. By interweaving **Euclidean**, **SAX**, and **Time** topologies, (starting from a discord search algorithm) we obtain a new algorithm for finding an **approximate** *nnd* profile for all the sequences of a time series

2. The evaluation we performed returns more than 98% of the **exact nnds,** and the values of the approximate *nnd*s are within 15% in respect to the exact ones**.**

3. The speedups are between **1 and 2 orders** of magnitude in respect to the brute force

4. The algorithm **assures** to find the sequences with the highest ***nnd*** *(discords)*

Conclusions

1. By interweaving **Euclidean**, **SAX**, and **Time** topologies, (starting from a discord search algorithm) we obtain a new algorithm for finding an **approximate** *nnd* profile for all the sequences of a time series

2. The evaluation we performed returns more than 98% of the **exact nnds,** and the values of the approximate *nnd*s are within 15% in respect to the exact ones**.**

3. The speedups are between **1 and 2 orders** of magnitude in respect to the brute force

4. The algorithm **assures** to find the sequences with the highest ***nnd*** *(discords)*

Future

1. Use faster algorithms to calculate the distance between sequences

2. Add the second *nnd*, third *nnd*, etc.

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA
KEOD KDIR KMIS
IC3K 2019
11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management
Vienna - Austria 17 - 19 September, 2019
DISCo DIPARTIMENTO DI INFORMATICA SISTEMISTICA E COMUNICAZIONE