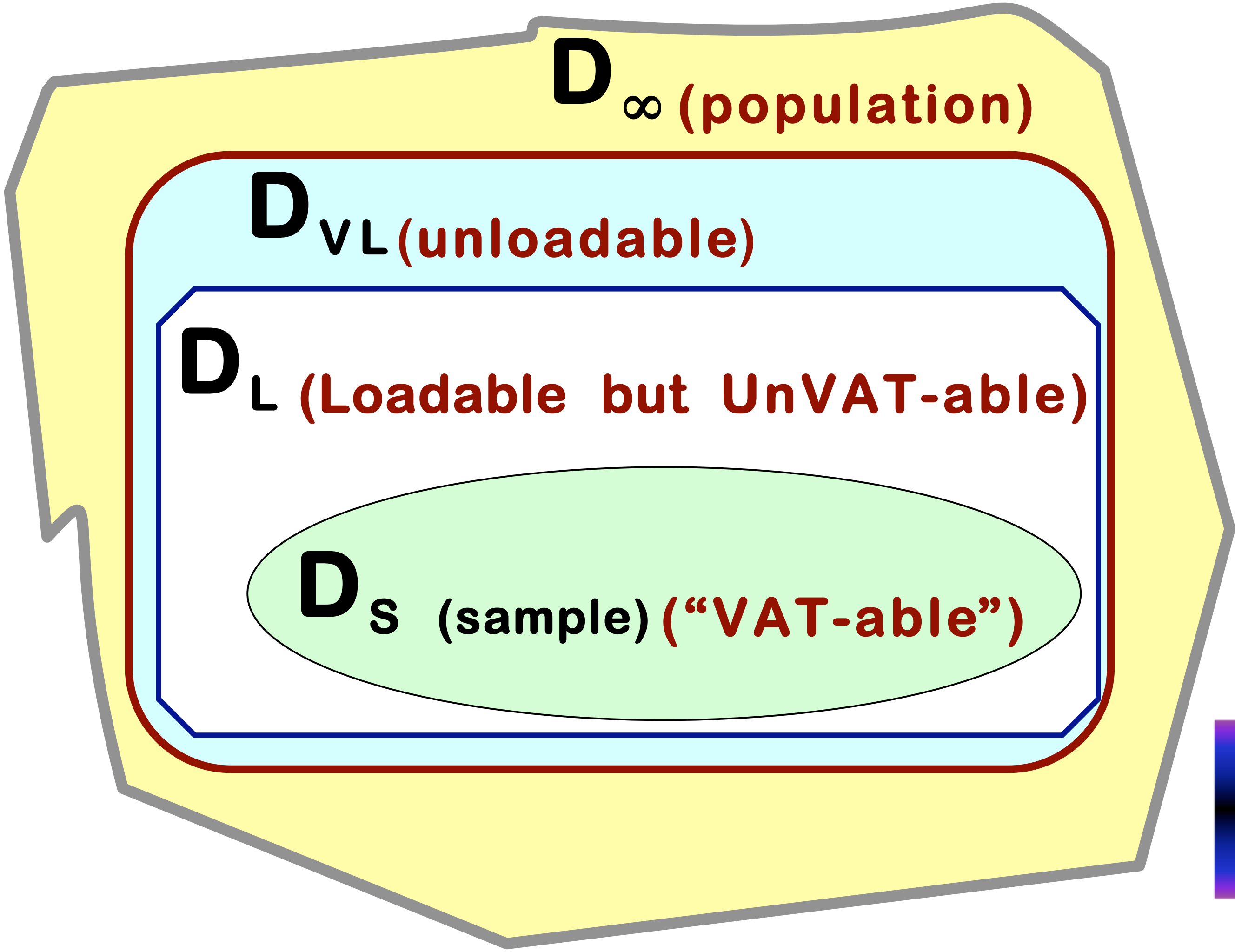
sVAT for VL Data

How Big is (VL) Data ?

Bytes	"Data Size"
10^2	Tiny
10^4	Small
10^6	Mega - medium
10^9	Giga - large
10^{10}	Huge
10^{12}	Tera - Monster
$10^{>12}$	Very Large (VL)

We can't cluster data this big (in a single computer) VL ~ "unloadable"... so ...

...we build a VAT-sized sample, and use it instead



2 Sampling Sub-Cases

When D_L is VAT-able

$VAT[D_S] \leftrightarrow VAT[D_L]$

***Visually* Compare $VAT[D_S] \sim VAT[D_L]$**

When D_L or D_{VL} is UnVAT-able

$VAT[D_S] \nleftrightarrow VAT[D_L \text{ or } D_{VL}]$

Comparison (Quality Check) impossible

Algorithm sVAT

Inputs $D_{N\times N}$: $D_{ij} \geq 0$; $D_{ii} = 0$: $D = D^T$

Initialize

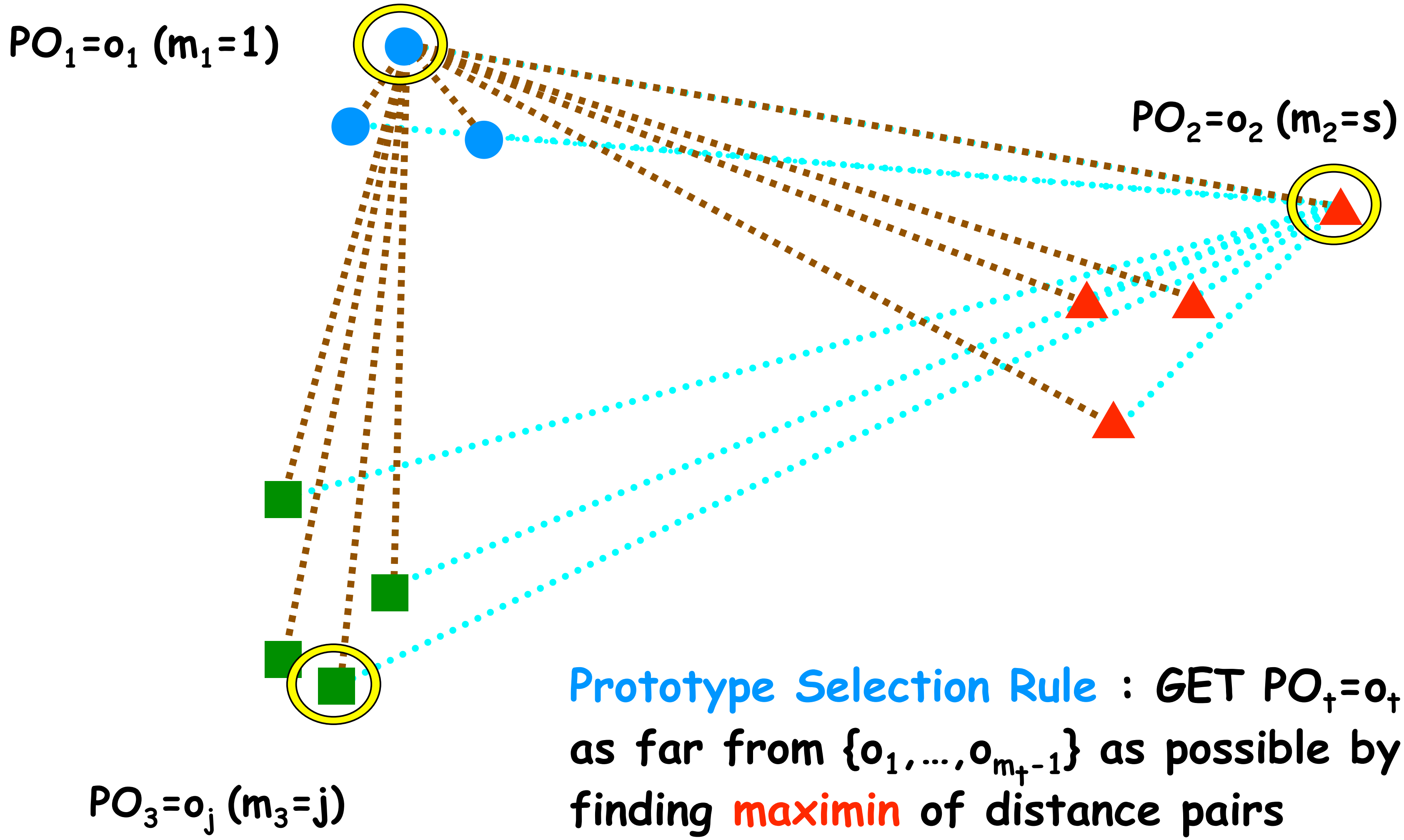
$c' \geq c$ An (OVER) estimate of c works best

$n \leq 5000$ Approximate sample size

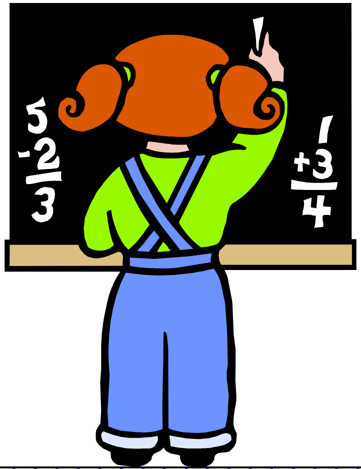
$m_1 = 1$ $o_1 = 1^{st}$ Prototype Object (PO)

$d = (d_1, \ldots d_N) = (D_{11}, \ldots D_{1N})$ search array

Get indices $\{m_i\}$ of (c') prototypes = $\{o_{m_1}, \ldots, o_{m_k}, \ldots, o_{m_{c'}}\}$



Prototype Selection Rule : GET $PO_t=o_t$ as far from $\{o_1, \ldots, o_{m_t-1}\}$ as possible by finding maximin of distance pairs



Small Result

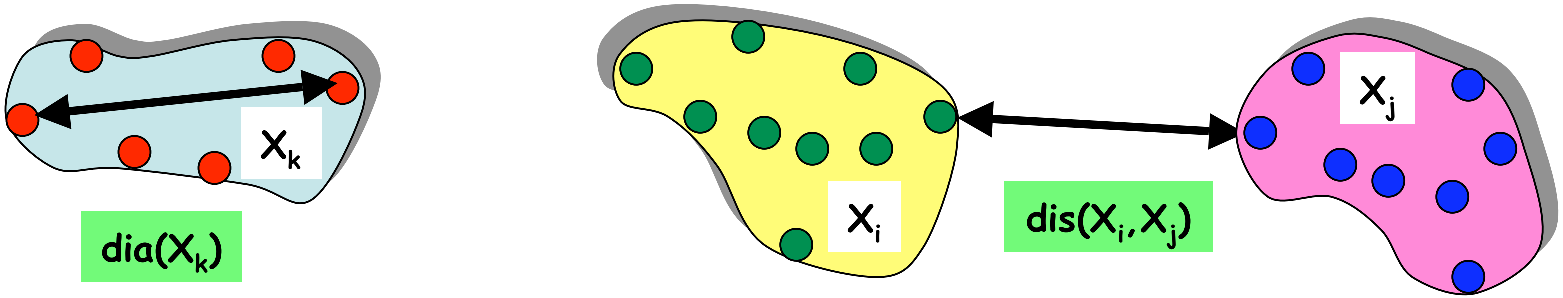
$O=\{o_1,\ldots,o_N\}$ contains $c \leq c'$ CS Clusters (Dunn, 1974)

$\Downarrow$

MMI selects at least one object from each of the CS clusters

aside **What *are* CS clusters ?**

$U \in M_{hcn} \leftrightarrow \{X_1, \ldots, X_c\}$ is any crisp partition of X



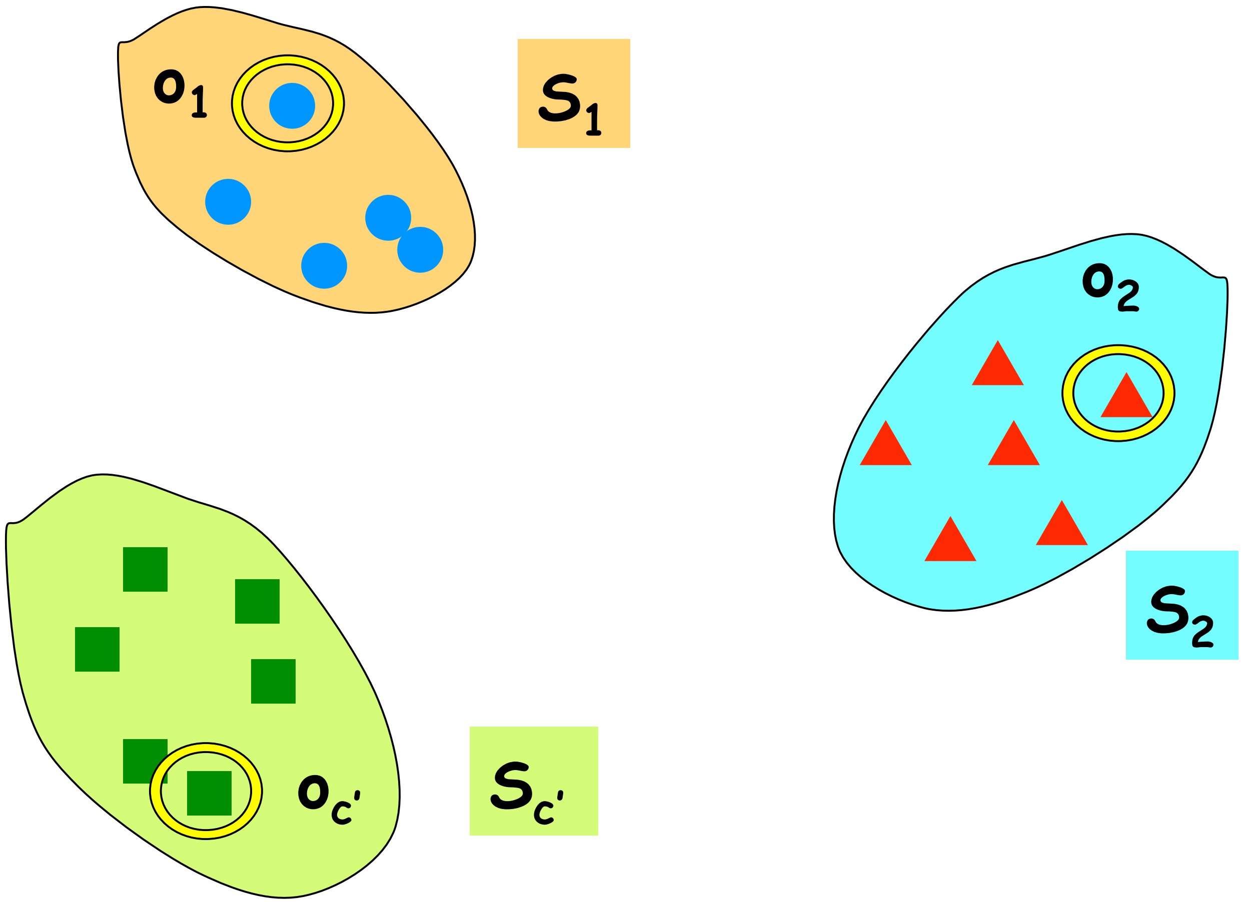
$$V_{Dunn}(U) = \underbrace{\min}_{1 \le i \le c} \left\{ \underbrace{\min}_{\substack{1 \le j \le c \\ j \ne i}} \left\{ \frac{dis(X_i, X_j)}{\underbrace{\max}_{1 \le k \le c} \left\{ dia(X_k) \right\}} \right\} \right\}$$

X has CS clusters

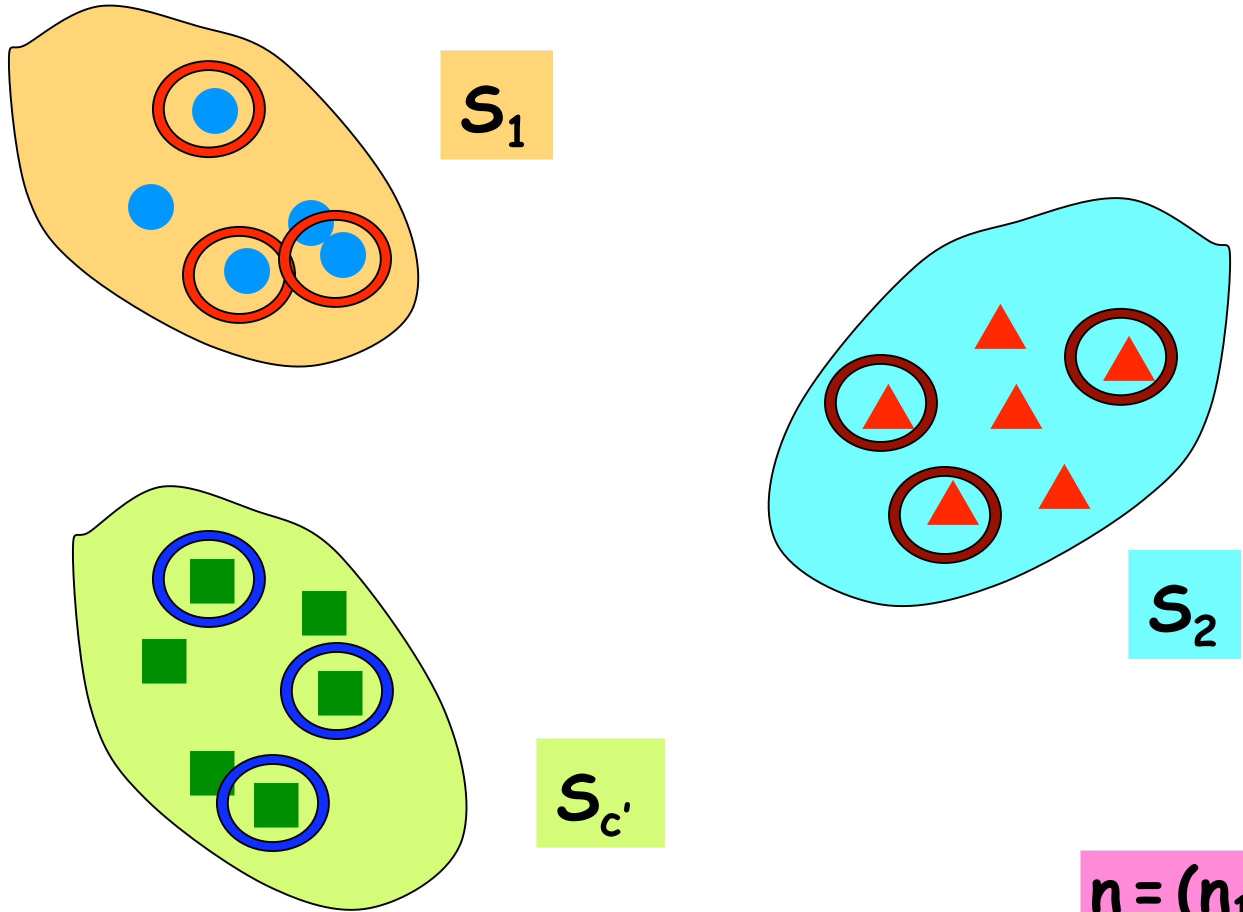
$$\Updownarrow$$

$$\underbrace{\max}_{U \in M_{hcn}} \left\{ V_{Dunn}(U) \right\} > 1$$

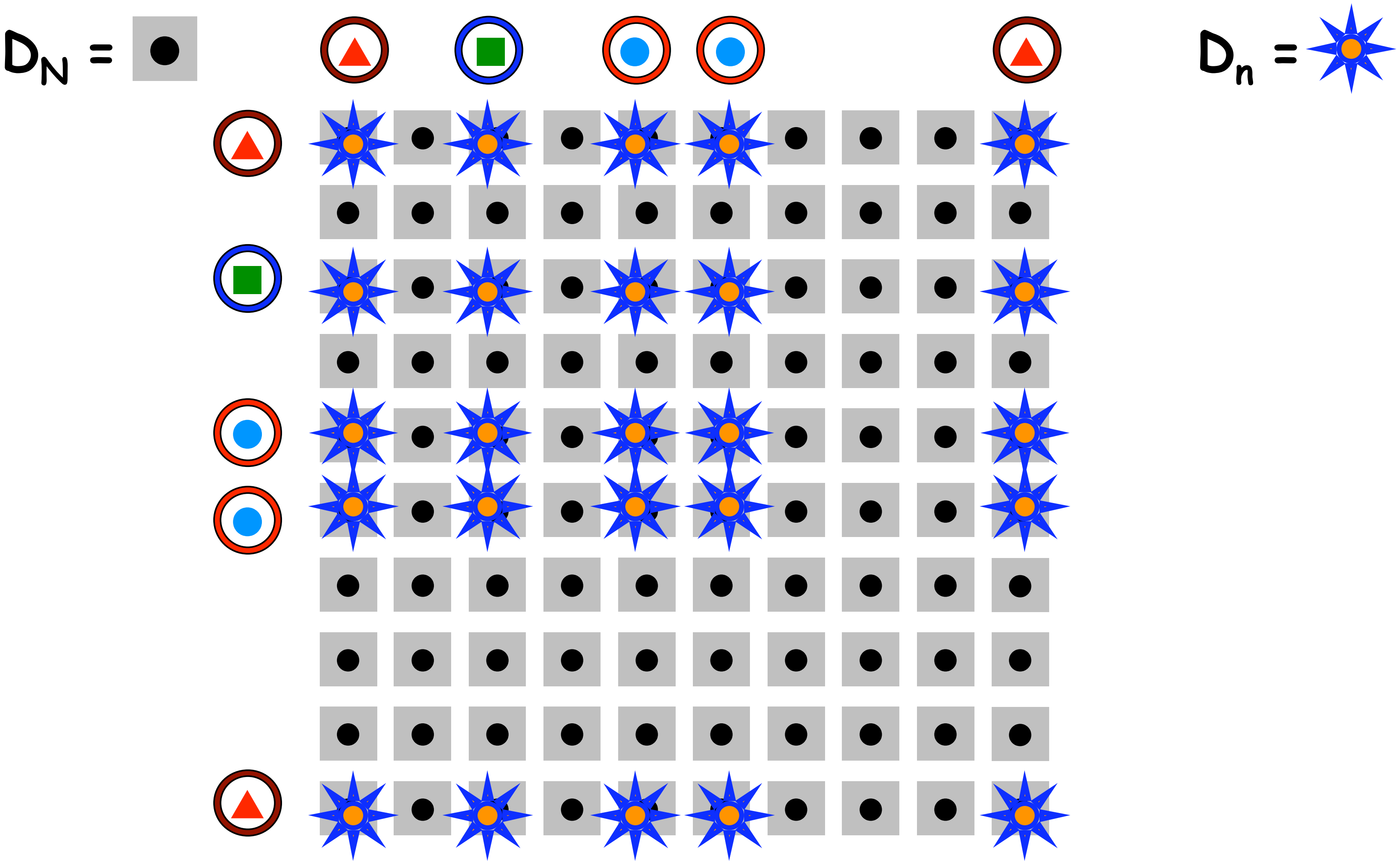
Get crisp 1-np clusters of $\{o_{m_k}\}$



Randomly Sample cluster S_t $n_t = \left\lceil \frac{n|S_t|}{N} \right\rceil$ times



$n = (n_1 + \cdots + n_{c'})$



Display VAT(D_n)

Three results for sVAT

sVAT is $O(\max\{c'N, (c'+n)^2\})$; linear in N>>n

O_N contains (c) CS (Dunn) clusters and ($c' \geq c$),

$\Downarrow$

sVAT selects at least 1 DO from each cluster

O_N contains (c) CS (Dunn) clusters, ($c' \geq c$), $\forall$ t

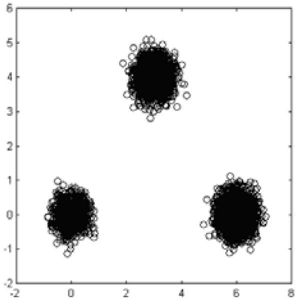
$n|S_t|/N$ is integer,

$\Downarrow$

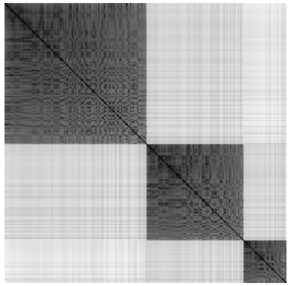
$n_i = N_i$ for $1 \leq i \leq c$

σ^2 = 0.1
"Clean"

Example 1
(c'=5,c=3)



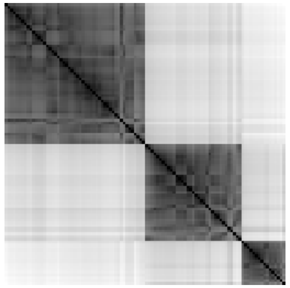
VAT



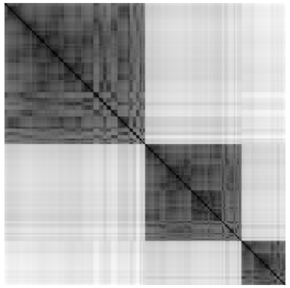
N = 5000
t = 156 s

COMPARE !

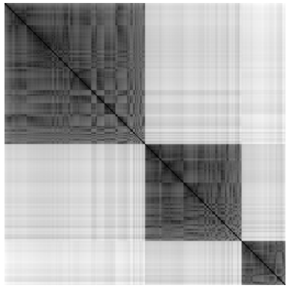
sVAT



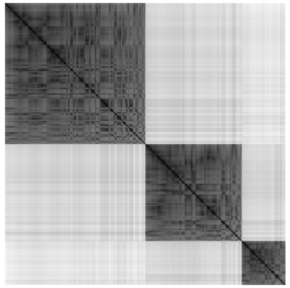
n = 100
t = 0.3 s



n = 300
t = 0.8 s



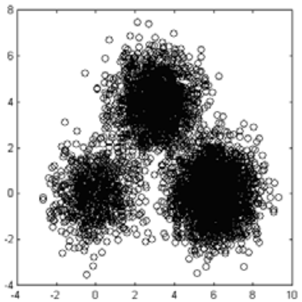
n = 500
t = 1.8 s



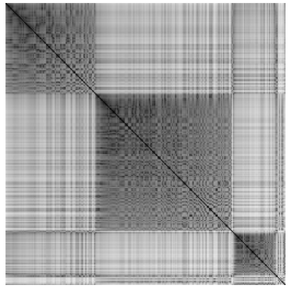
n = 1000
t = 6.5 s

$\sigma^2 = 1$
"Noisy"

Example 2
($c' = 5, c = 3$)



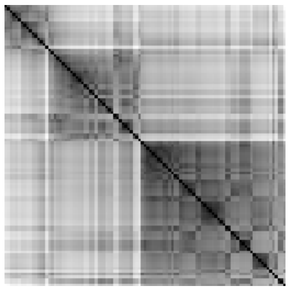
VAT



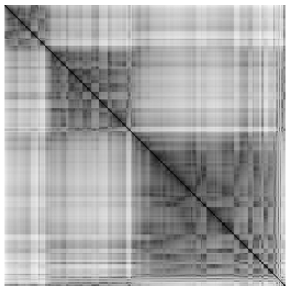
N = 5000
t = 155 s

COMPARE !

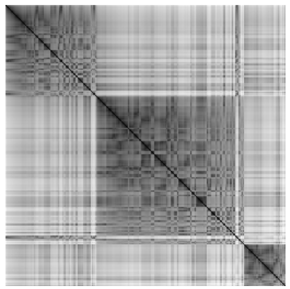
sVAT



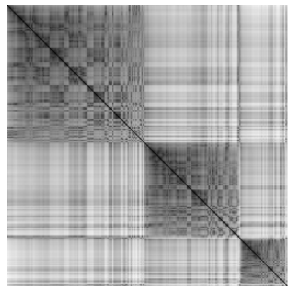
n = 100
t = 0.3 s



n = 300
t = 0.8 s



n = 500
t = 1.9 s



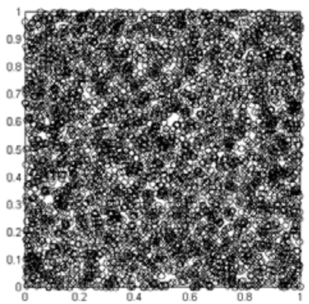
n = 1000
t = 6.9 s

- Uniform -

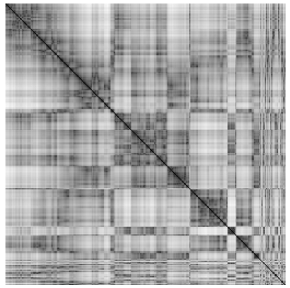
No Clusters !

Example 3

(c'=5,c=0)



VAT

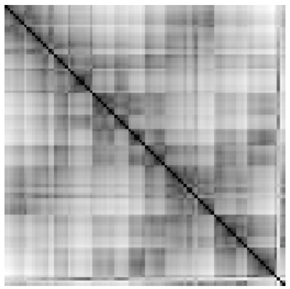


N = 5000

t = 161 s

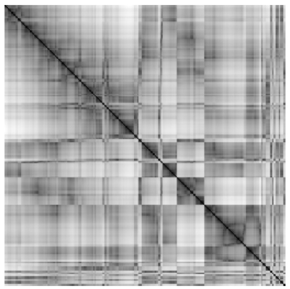
COMPARE !

sVAT



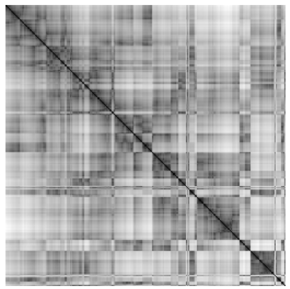
n = 100

t = 0.3 s



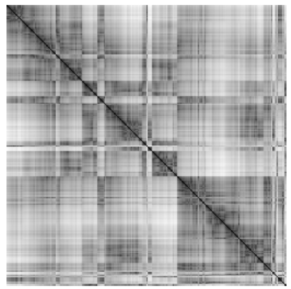
n = 300

t = 0.8 s



n = 500

t = 1.8 s

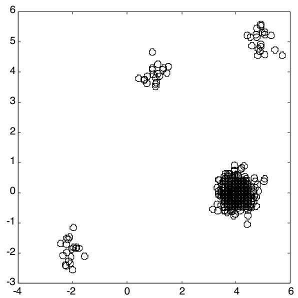


n = 1000

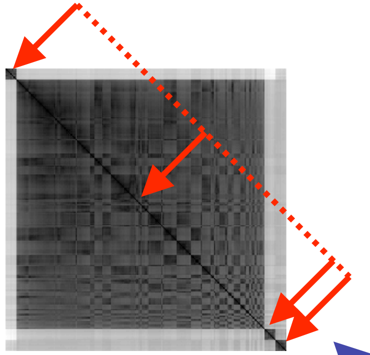
t = 6.5 s

Unequal Populations
$n_1 = n_2 = n_3 = 20$; $n_4 = 440$

Example 4
(c'=varies,c=4)



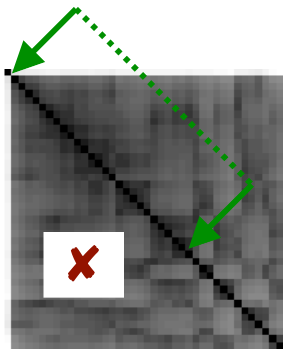
VAT



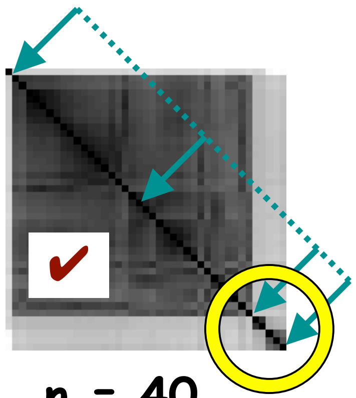
N = 500

COMPARE !

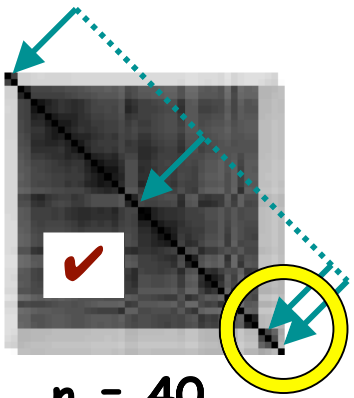
sVAT



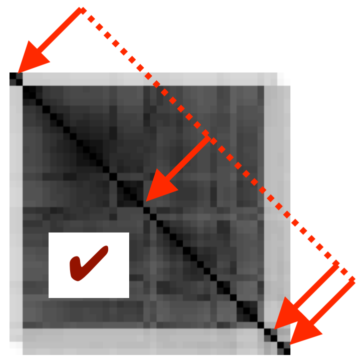
n = 40
c' = 1
c = 2



n = 40
c' = 2
c = 4

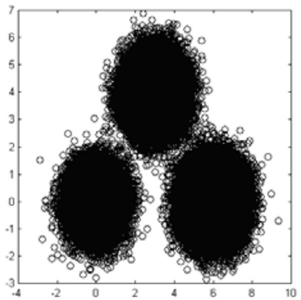


n = 40
c' = 3
c = 4



n = 40
c' = 4
c = 4

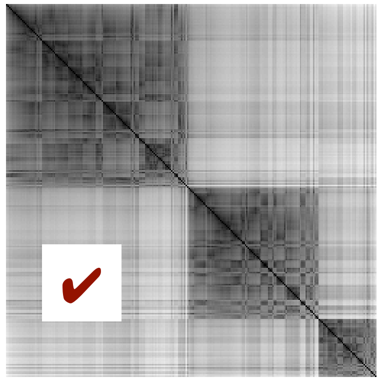
$\sigma^2 = 0.5$
$N = 100{,}000$



Example 5
(c′=5, c=3)

CAN NOT COMPARE !

sVAT still works !



$n = 500$
$t = 6.81$ s

Loadable (but Un VAT-able) D_L

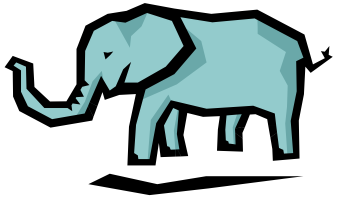
- ✗ CPU time ~ 36 hrs
- ✗ Storage ~ 38 gb
- ✗ |D| > maxint in Matlab

Object Vectors $X_N = \{x_1, \ldots, x_{435}\} \subset \Re^{16}$

Example 6

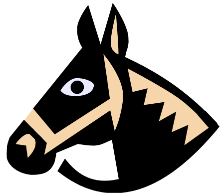
Euclidean distances yield $D_{435 \times 435}$

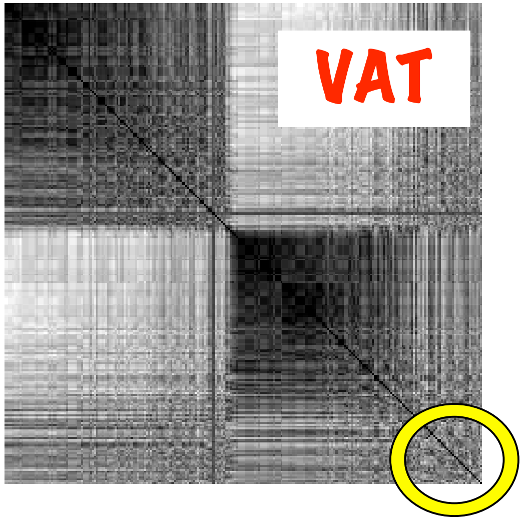
The Data? 1984 Voting Records of 435 US Representatives on 16 bills



54.8% Republicans

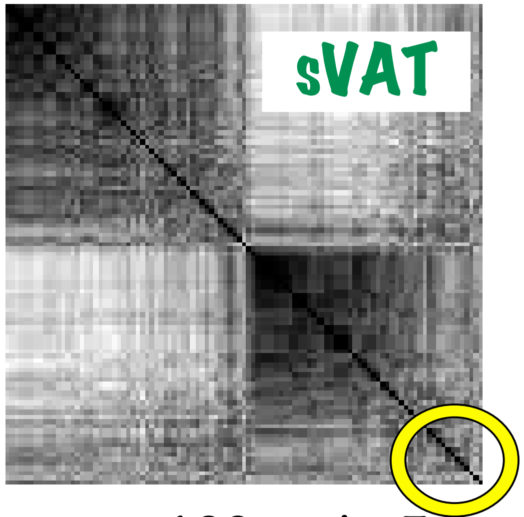
45.2% Democrats





COMPARE !

c = 2



n = 100 ; c′= 5

sVAT es Bueno !

Builds VAT image on a sample of D_N for any N

- ✔ Linear Complexity : O(c'N)
- ✔ Guarantees representation of all CS clusters
- ✔ Can guarantee that # samples n_i = # N_i in O_i

Gets right answer in all tests … … so far!

sVAT es Malo !

Requires *square* D_{NxN} matrix

Ex.6 : *Raw Data* is *rect.* D_{435x16}

Que es proximo?

coVAT

(Rectangular Data)

coVAT : The Input Data

Rectangular Dissimilarity Data Matrix : D_{mxn}

$$D_{m\times n} = \begin{array}{c|c} & \begin{array}{cccc} o_{m+1} & o_{m+2} & & o_{m+n} \end{array} \\ \hline \begin{array}{c} o_1 \\ o_2 \\ \\ o_m \end{array} & \begin{bmatrix} d(o_1,o_{m+1}) & d(o_1,o_{m+2}) & \cdots & d(o_1,o_{m+n}) \\ d(o_2,o_{m+1}) & d(o_2,o_{m+2}) & \cdots & d(o_2,o_{m+n}) \\ \vdots & \vdots & \ddots & \vdots \\ d(o_m,o_{m+1}) & d(o_m,o_{m+2}) & \cdots & d(o_m,o_{m+n}) \end{bmatrix} \end{array}$$

m Row Objects : $O_r=\{o_1,\ldots o_m\}$

n Column Objects : $O_c=\{o_{m+1},\ldots o_{m+n}\}$

4 Types of Clusters in D_{mxn}

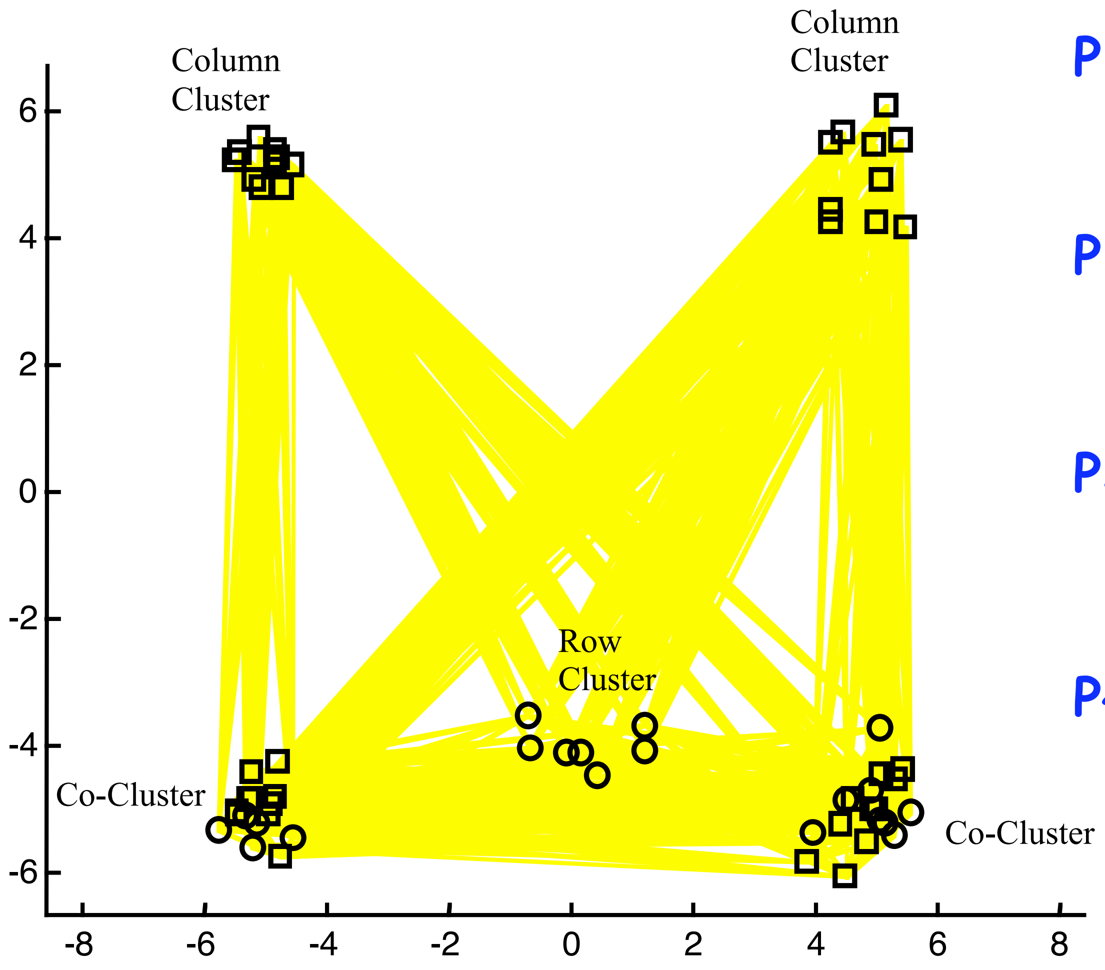
P1: c_r clusters in the row objects O_r

P2: c_c clusters in the column objects O_c

P3: $c_{r\cup c}$ All clusters in $O_r \cup O_c$

P4: c_{co} (mixed) co-clusters $O_r \cup O_c$

Ex.1 Data D_{20x40} : 4 Types of Clusters



P1: c=3 clusters

in rows O_r

P2: c=4 clusters

in cols O_c

P3: c=5 clusters

in $O_r \cup O_c$

P4: c=2 (mixed) co-clusters in $O_r \cup O_c$

Our Basic Assumption

$D_{m \times n}$ is a known rectangular submatrix

$$\left[D_{r\cup c}\right]_{N\times N} = \begin{bmatrix} \left[D_r\right]_{m\times m} & \left[D\right]_{m\times n} \\ \left[D^T\right]_{n\times m} & \left[D_c\right]_{n\times n} \end{bmatrix}$$

of an unknown square matrix $D_{N \times N}$, $N = m+n$

Our Dilemma

VAT is undefined on two rectangular matrices

$$\left[D_{r \cup c}\right]_{N \times N} = \begin{bmatrix} \left[D_r\right]_{m \times m} & \left[D\right]_{m \times n} \\ \left[D^T\right]_{n \times m} & \left[D_c\right]_{n \times n} \end{bmatrix}$$

VAT works on three unknown square matrices

Our Solution

Regard the cols of D as feature vectors of the col. objects in R^m

$$
[D]_{m\times n} = \begin{array}{c} o_1 \\ \vdots \\ o_i \\ \vdots \\ o_m \end{array}
\begin{bmatrix} \leftarrow & d_{1*} & \rightarrow \\ & \vdots & \\ \leftarrow & d_{i*} & \rightarrow \\ & \vdots & \\ \leftarrow & d_{m*} & \rightarrow \end{bmatrix}
= \begin{array}{ccc} o_{m+1} & o_{m+2} & o_{m+n} \\ \left[\begin{array}{c} \uparrow \\ d_{*1} \\ \downarrow \end{array}\right. & \begin{array}{c} \uparrow \\ d_{*j} \\ \downarrow \end{array} & \left.\begin{array}{c} \uparrow \\ d_{*n} \\ \downarrow \end{array}\right] \end{array}
$$

Regard the rows of D as feature vectors of the row objects in R^n

Impute missing values for D_r and D_c

$$\left[D_r\right]_{ij} = \gamma_r \left\| \mathbf{d}_{i*} - \mathbf{d}_{j*} \right\| \qquad \left[D_c\right]_{ij} = \gamma_c \left\| \mathbf{d}_{*i} - \mathbf{d}_{*j} \right\|$$

γ_r and γ_c scale D_r and D_c to improve constrast

Save VAT($D_{r\cup c}$) indices
$P_{r\cup c}$ = (P(1),…, P(m+n))
for coVAT algorithm

$$\left[D_{r\cup c}\right]_{N\times N} = \begin{bmatrix} \left[D_r\right]_{m\times m} & \left[D\right]_{m\times n} \\ \left[D^T\right]_{n\times m} & \left[D_c\right]_{n\times n} \end{bmatrix}$$

Apply VAT to 3 (estimated) square matrices

Algorithm coVAT : Build ordered version of $D_{m \times n}$

1. Unshuffle VAT($D_{r \cup c}$) indices $P_{r \cup c} = (P(1), \ldots, P(m+n))$

   For t = 1 to m+n

   If $P(t) \le m$: rc=rc+1; RP(rc)=P(t)

   Else cc=cc+1; CP(cc)=P(t)

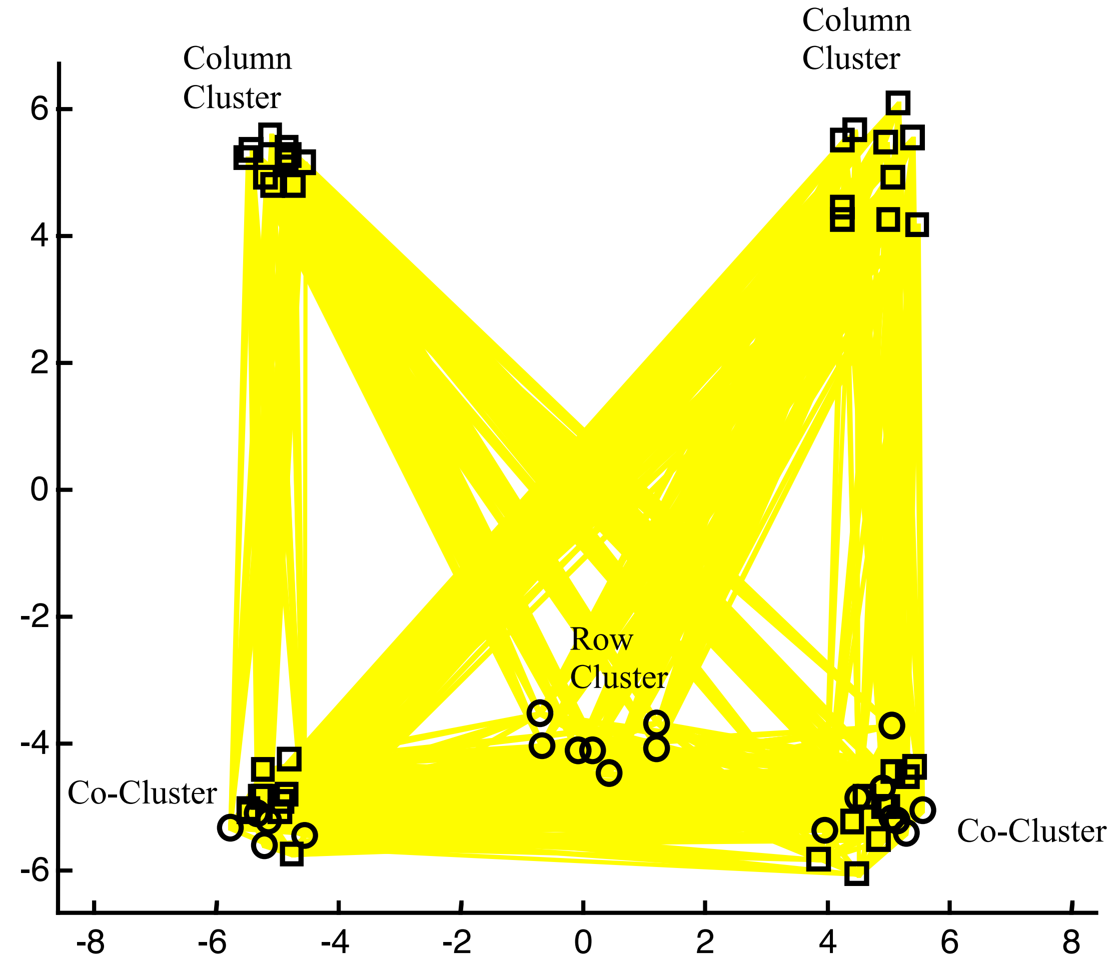
2. Reorder rectangular input matrix D = [d_{ij}]

$$\left[\tilde{D}_{m \times n}\right] = \left[\tilde{d}_{ij}\right] = \left[d_{RP(i)CP(j)}\right]$$

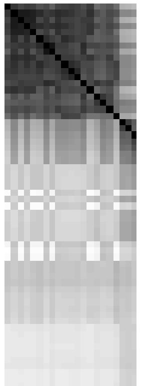
3. Display rectangular coVAT image $I(\tilde{D})$

Ex. 1: coVAT images for $D_{20 \times 40}$

P1: c=

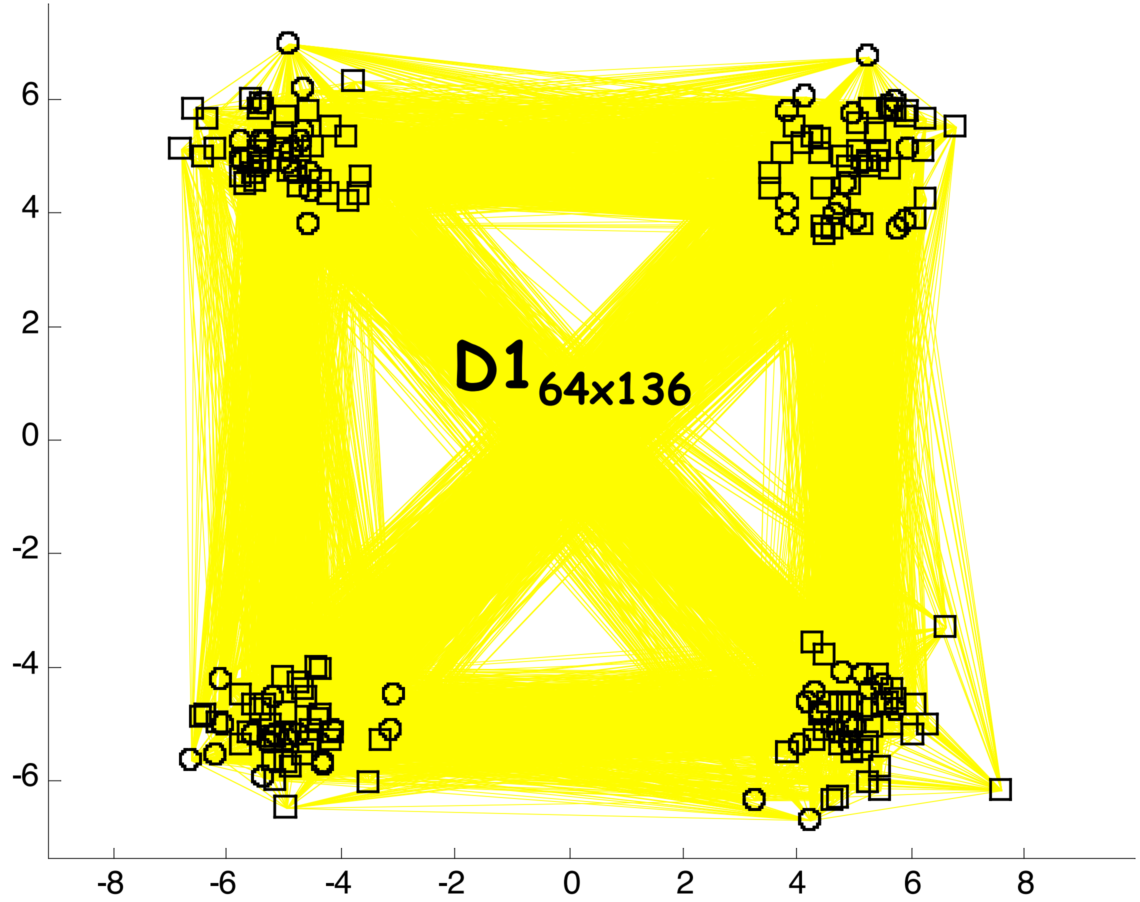


4 in O_c



P3: c=

Ex. 2: "best case" - ALL CS co-clusters



○ = row object

□ = col. object

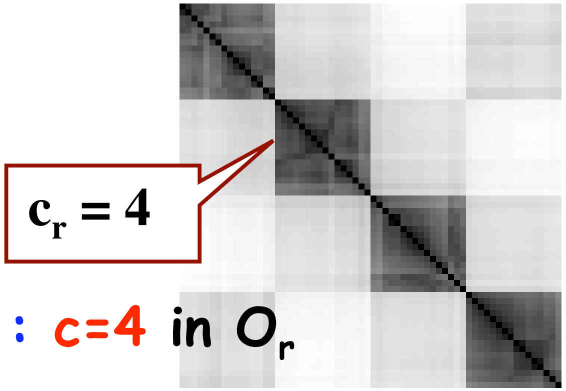
$c_r = 4$

$c_c = 4$

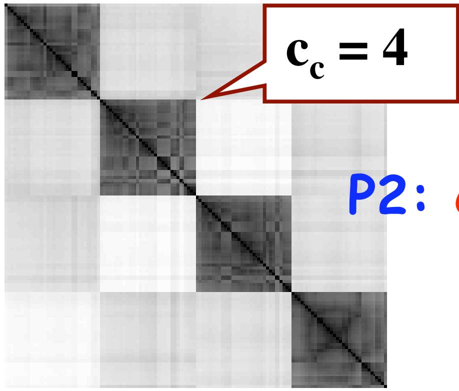
$c_{r\cup c} = 4$

$c_{co} = 4$

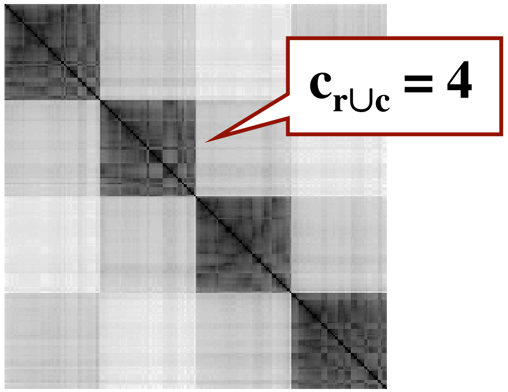
Ex. 2: "best case" - ALL CS co-clusters



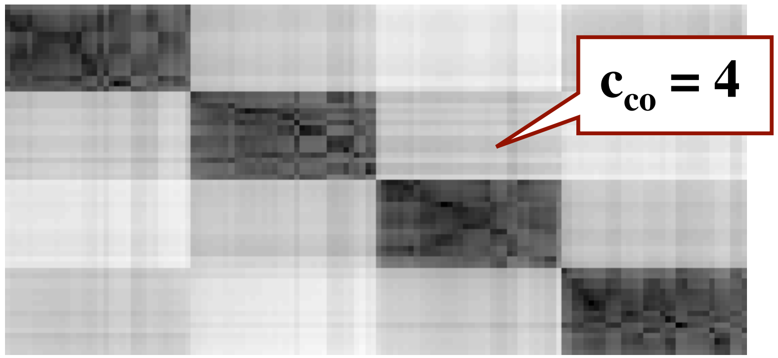
P1: c=4 in O_r



P2: c=4 in O_c

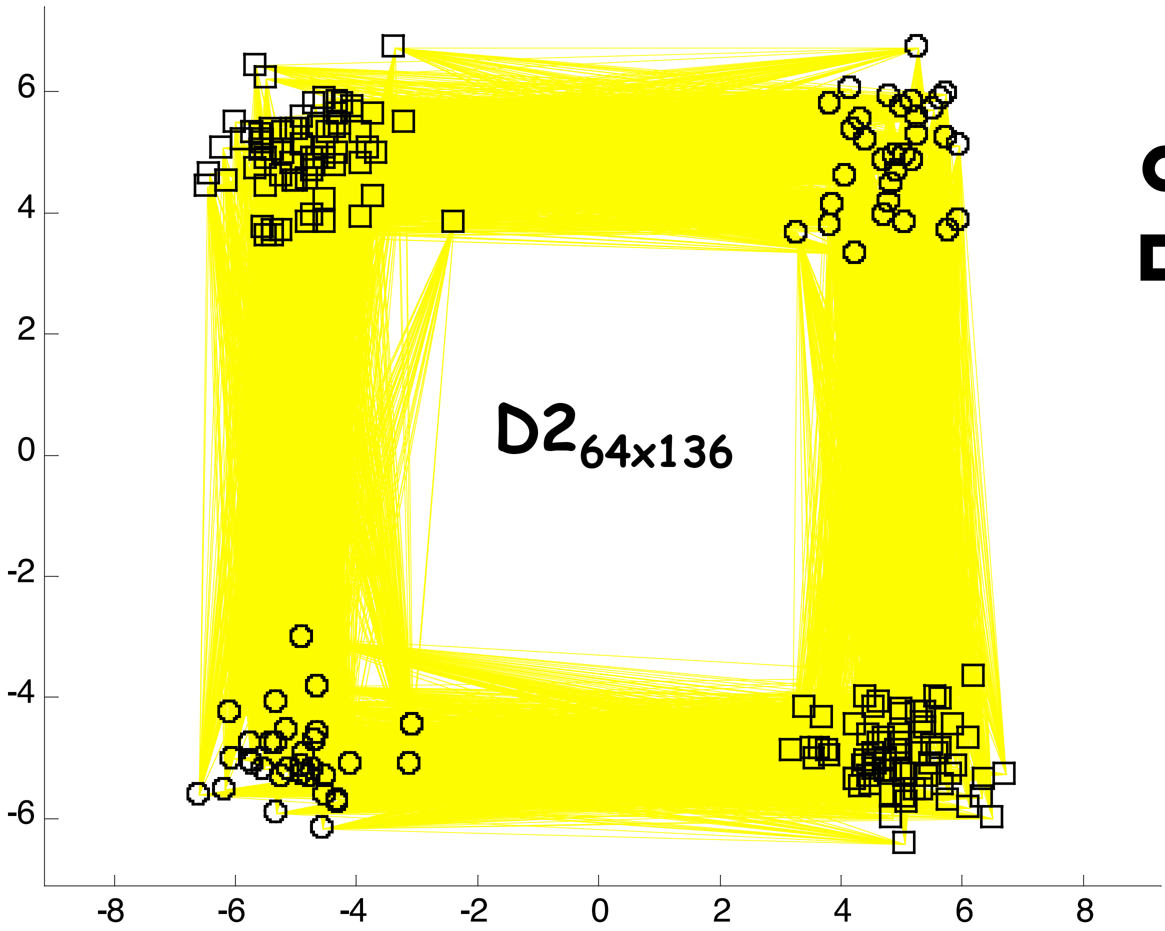


P3: c=4 in $O_r \cup O_c$



P4: c=4 (mixed) in $O_r \cup O_c$

Ex. 3: “worst case” - No co-clusters



○ = row object

□ = col. object

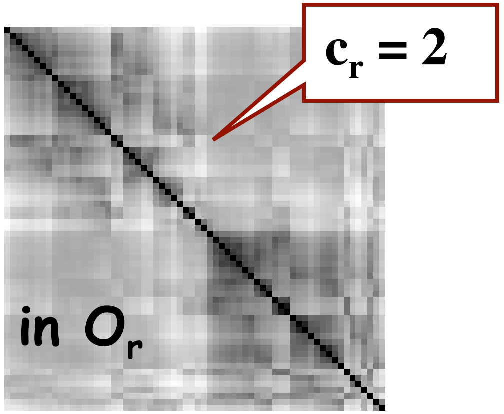
$c_r = 2$

$c_c = 2$

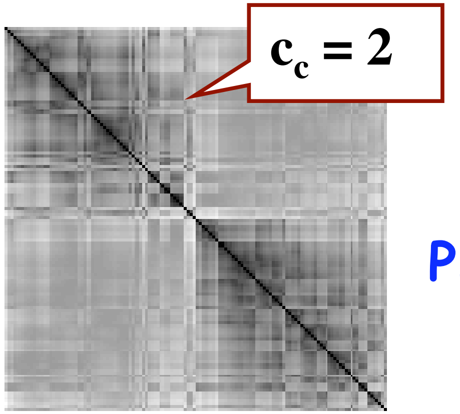
$c_{r\cup c} = 4$

$c_{co} = 0$

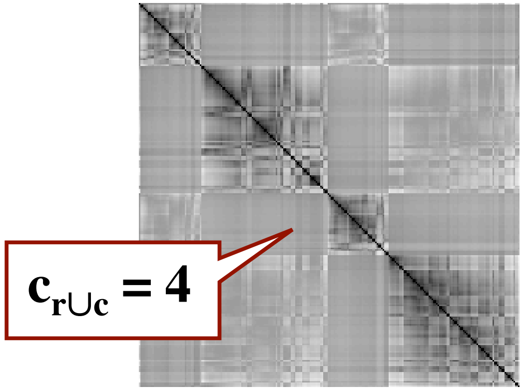
Ex. 3: "worst case" - No co-clusters



P1: $c=2$ in O_r



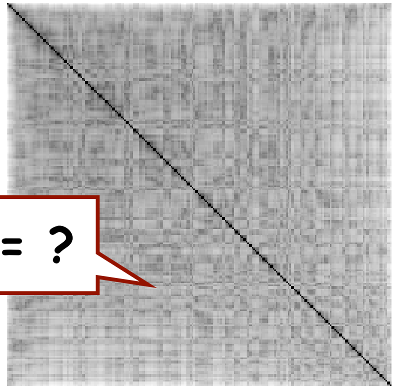
P2: $c=2$ in O_c

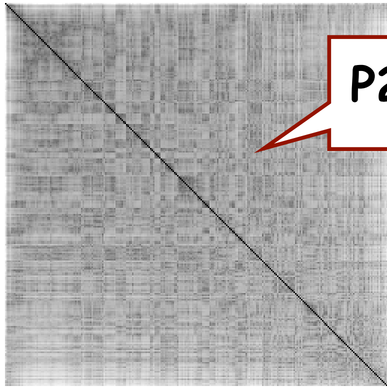


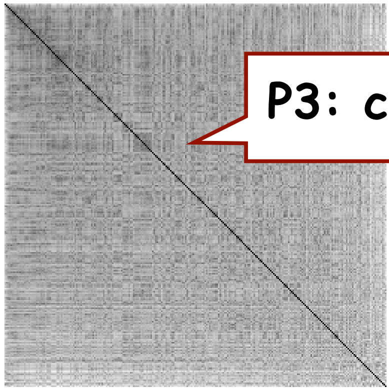
P3: $c=4$ in $O_r \cup O_c$

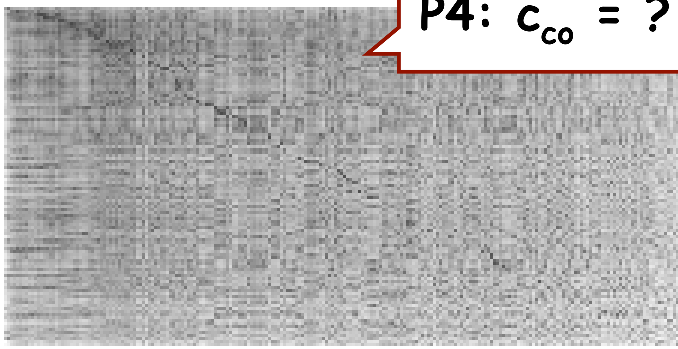
P4: $c=0$ (mixed) in $O_r \cup O_c$

Ex. 4: You be the judge !

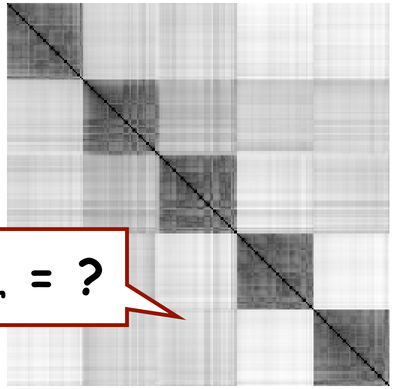


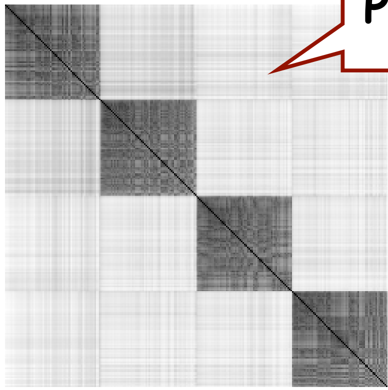


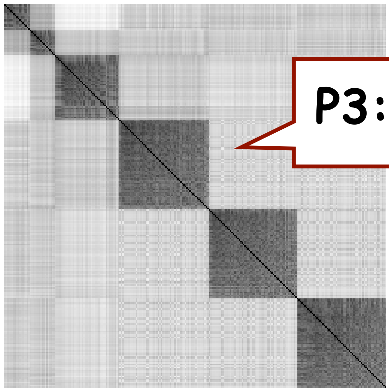


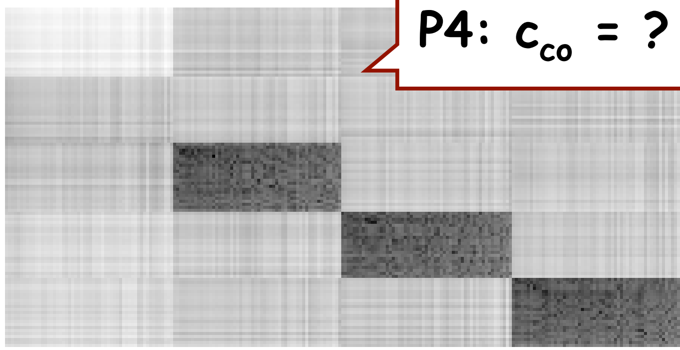


Ex. 5: You be the judge !









Summary and Conclusions : VAT Family

VAT →

Input	nxn Dissimilarity Matrix D
Output	VAT image I(D*)
Limits	Storage & Resolution, $n \leq 5000$

sVAT →

Input	NxN Dissimilarity Matrix D_N
Output	VAT sample image $I(D_n{}^*)$
Limit	No size limit !

coVAT →

Input	mxn Dissimilarity Matrix D
Output	(3) VAT, (1) coVAT Images
Limit	No size limit ! (use sVAT for (3))

$I(D_r)$ for c_r in row objects

$I(D_c)$ for c_c in column objects

$I(D_{r\cup c})$ for $c_{r\cup c}$, pure coclusters

$I(D_{co})$ for c_{co}, mixed coclusters


Gracias !
Adios !