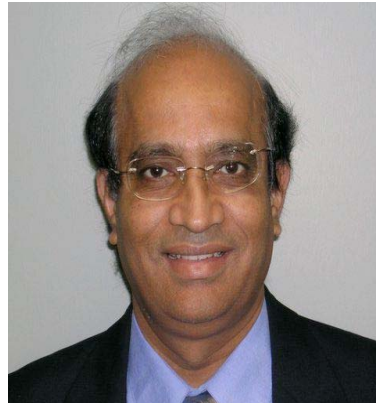


Approximate Clustering in Very Large Relational Data

eNERF



Chris Leckie



Rao Kotagiri



Jackie Huband



Jim Bezdek



Rick Hathaway

The NERF Family

RFCM

Relational Fuzzy (c-means)

NERFCM

Non Euclidean rFCM (aka NERF or LNERF)

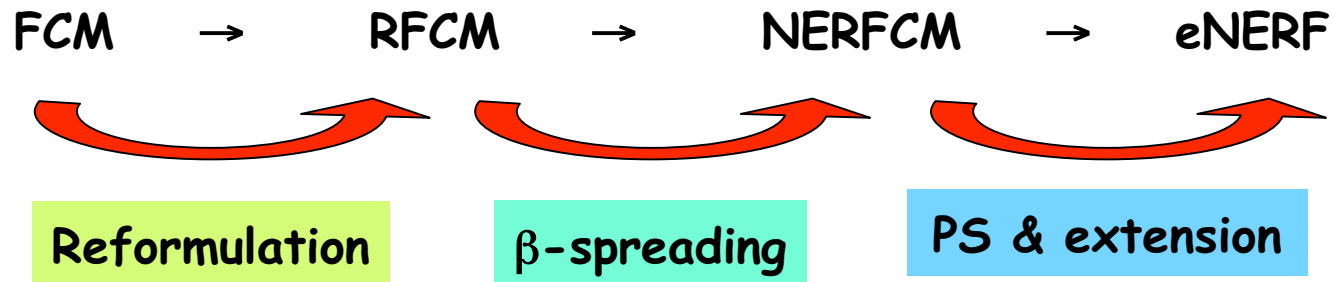
kNERF

kernelized NERF: clustering in
(badly messed up) relational data

eNERF

extended NERF for (VL=Unloadable)
relational data

The eNERF *family tree*



same basic theory for the HCM and PCM cases

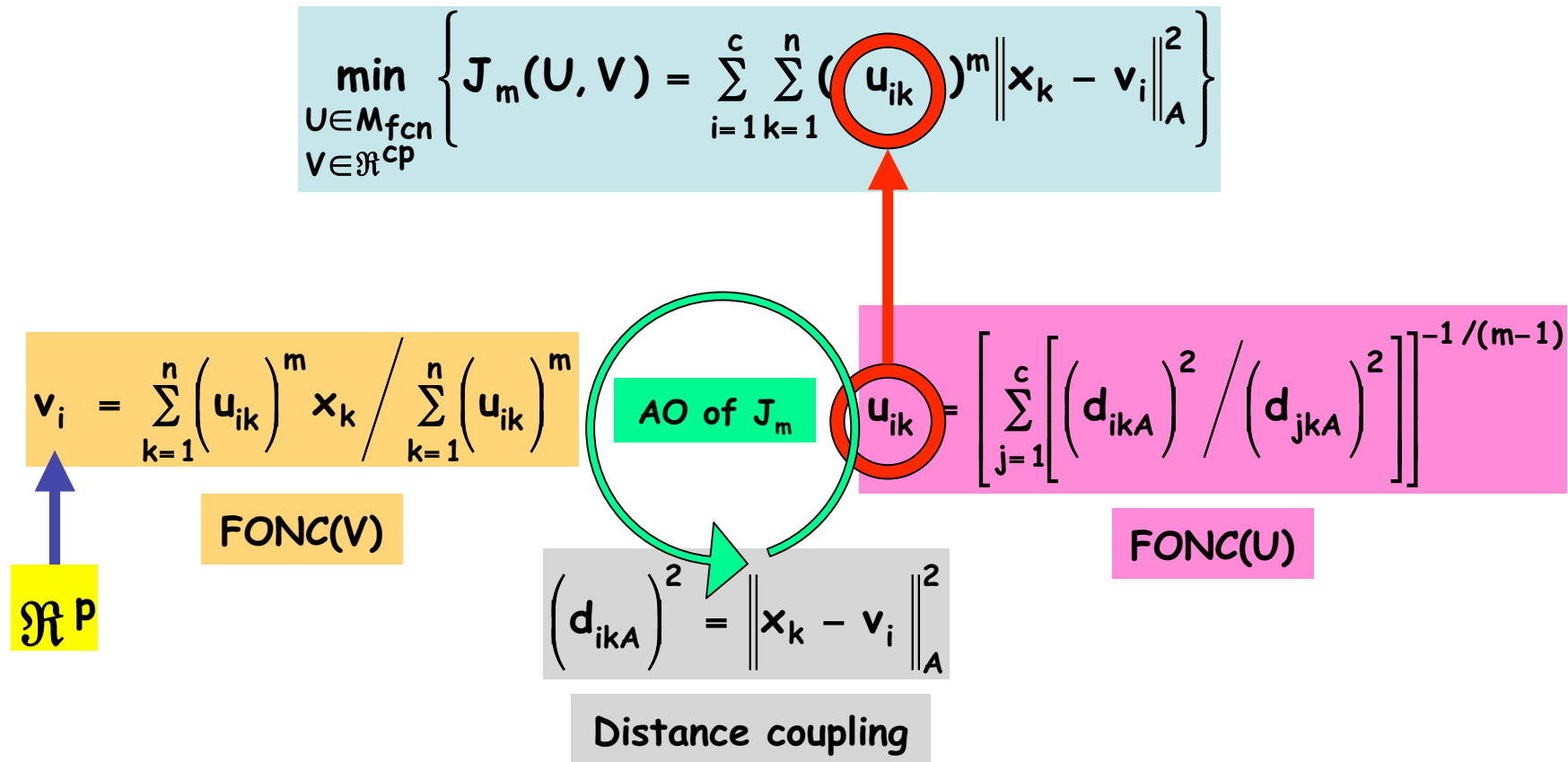
Reformulation has 2 objectives

Constrained optimization (FCM) becomes *unconstrained* (RFCM)

Reduction to the *smallest possible number* of problem variables

Reformulation (FCM Case)

Eliminate U variables (a $c \times n$ matrix) by substitution of the FONC for U into $J_m(U, V)$, thereby obtaining $K(V)$, with V a $c \times p$ matrix



The reformulated problem *is* RFCM

$$\min_{V \in \mathcal{R}^{cn}} \left\{ K_m(V) = \sum_{k=1}^n \left[\sum_{i=1}^c (d_{ikA})^{1/(1-m)} \right]^{1-m} \right\}$$

(Equiv.) FONC(V)

$$v_i = \left(u_{i1}^m, \dots, u_{in}^m \right)^T / \sum_{k=1}^n u_{ik}^m$$

\mathcal{R}^n

(Same formula) FONC(U)

$$u_{ik} = \left[\sum_{j=1}^c \left[\left(\delta_{ikA} \right)^2 / \left(\delta_{jkA} \right)^2 \right] \right]^{-1/(m-1)}$$

AO of K_m

$$\delta_{ikA}^2 = \left(Dv_i \right)_k - 0.5 * v_i^T Dv_i$$

$$D = [D_{jk}] = \left[\left\| x_j - x_k \right\|_A^2 \right]$$

(Equiv.) "distance" coupling



Un-kernelized Duality Theory for c-Means Clustering



Object Data

$$X = \{x_1, \dots, x_n\}$$

HCM



Hard



RHCM

FCM



Fuzzy



RFCM

PCM



Poss.



RPCM



Equivalent

\Leftrightarrow

$$D_{ij} = \left\| x_i - x_j \right\|_{\text{Eucl.}}^2$$



Yeow !

D is a **Euclidean Matrix**, X is a **realization** of D



$$\exists X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathcal{R}^{n-1} \text{ s.t. } D_{ij} = \sum_{k=1}^{n-1} (x_{ik} - x_{jk})^2$$



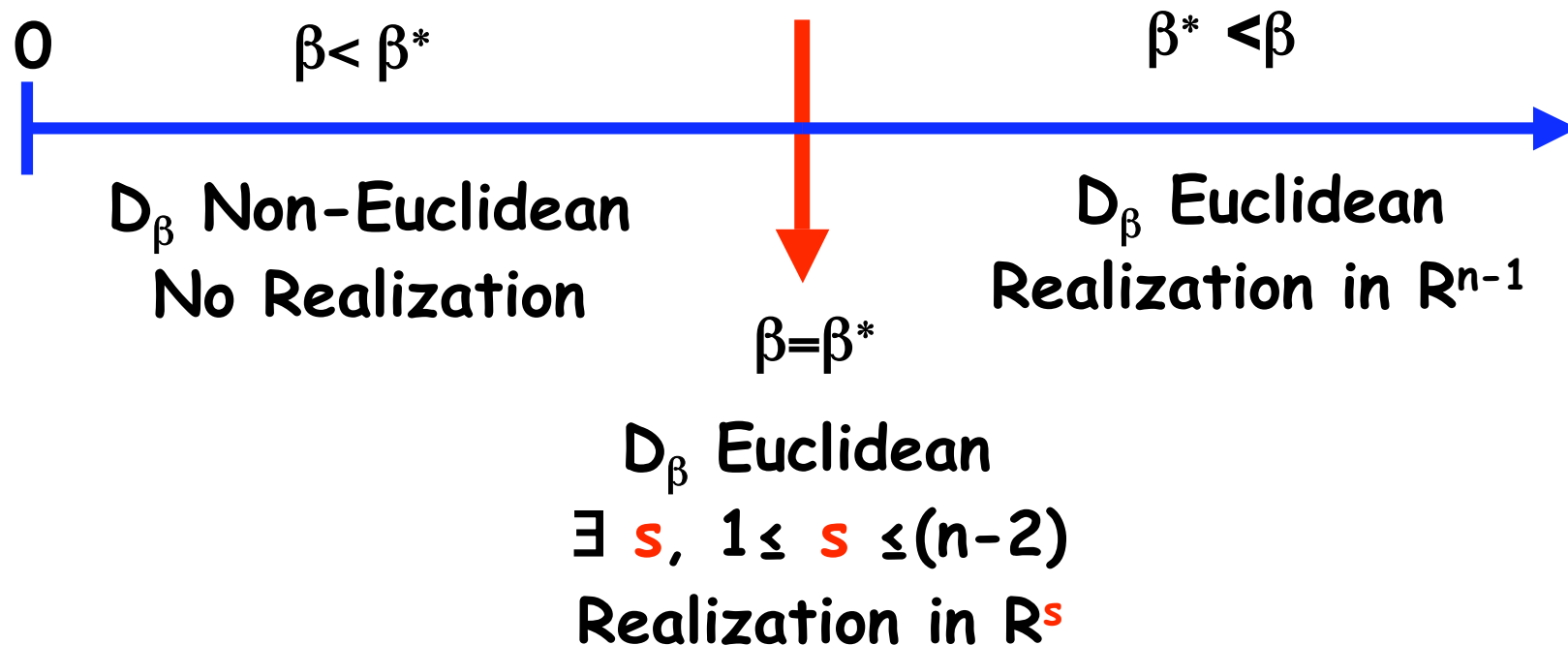
$$\mathbf{z}^T D \mathbf{z} \leq 0 \quad \forall \mathbf{z} \in \mathcal{R}^n \text{ s.t. } \sum \mathbf{z}_i = 0$$

RFCM \Leftrightarrow FCM \Leftrightarrow D is Euclidean

NERF extends RFCM for $D \neq D_{Eu}$

The of NERF c-means

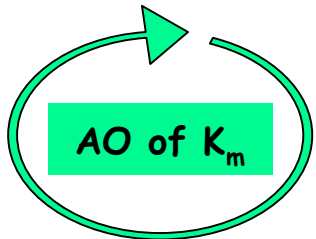
$$\beta - \text{spread} \dots D_\beta = D + \beta([1]_n - I_n)$$



RFCM \rightarrow **NERFCM** when $D \rightarrow D_\beta$ (**RFCM** \leftarrow NERFCM when $\beta = 0$)

$$D = [D_{jk}] = \left[\left\| x_j - x_k \right\|_A^2 \right] \longrightarrow D_\beta = D + \beta \left([1]_n - I_n \right) = D + \beta M$$

(Equiv.) FONC(V)
(Same)



(Same formula) FONC(U)
(Same)

$$\delta_{ikA}^2 = \left(D_\beta v_i \right)_k - 0.5 * v_i^T D_\beta v_i$$

D, D_β are $(n \times n)$ matrices, soon to be associated with sample $|X_n|=n$ when X_N is VL

This makes extension **more difficult** than for eFFCM and geFFCM

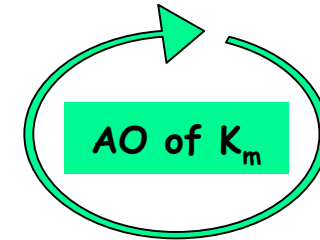
Relational Clustering with NERF aka "LNERF"

Input	$n \times n$ Dissimilarity Matrix D_n
Constraints	$d_{ij} \geq 0; d_{ij} = d_{ji}; d_{ii} = 0$
Choose	$M = [1]_n - I_n$ $c = \#$ of clusters, $2 \leq c \leq n$
Reformat this like the others effcm, etc	
	$\epsilon_L =$ termination criterion $Q_M =$ max. # of iterations termination norm $\ U^{(q)} - U^{(q-1)}\ $
Initialize	$q = 0; \beta = 0; U_{diff} = 2 * \epsilon_L; U^{(0)} \in M_{fcn}$

While $U_{\text{diff}} > \varepsilon_L$ and $q < Q_M$

$$v_i^{(q)} = ((u_{i1}^{(q)})^m, (u_{i2}^{(q)})^m, \dots, (u_{in}^{(q)})^m)^T / \sum_{j=1}^n (u_{ij}^{(q)})^m$$

$$\delta_{ik} = ((D + \beta M)v_i^{(q)})_k - (v_i^{(q)})^T (D + \beta M)v_i^{(q)} / 2$$



If any $\delta_{ik} < 0$:

$$\Delta\beta = \max_{i,k} \left\{ -2\delta_{ik} / \|v_i^{(q)} - e_k\|^2 \right\}$$

$$\delta_{ik} = \delta_{ik} + (\Delta\beta / 2) \cdot \|v_i^{(q)} - e_k\|^2$$

$$\beta = \beta + \Delta\beta$$

*Works fine ...
if D is small !*

If $\delta_{ik} > 0$ (else usual) :

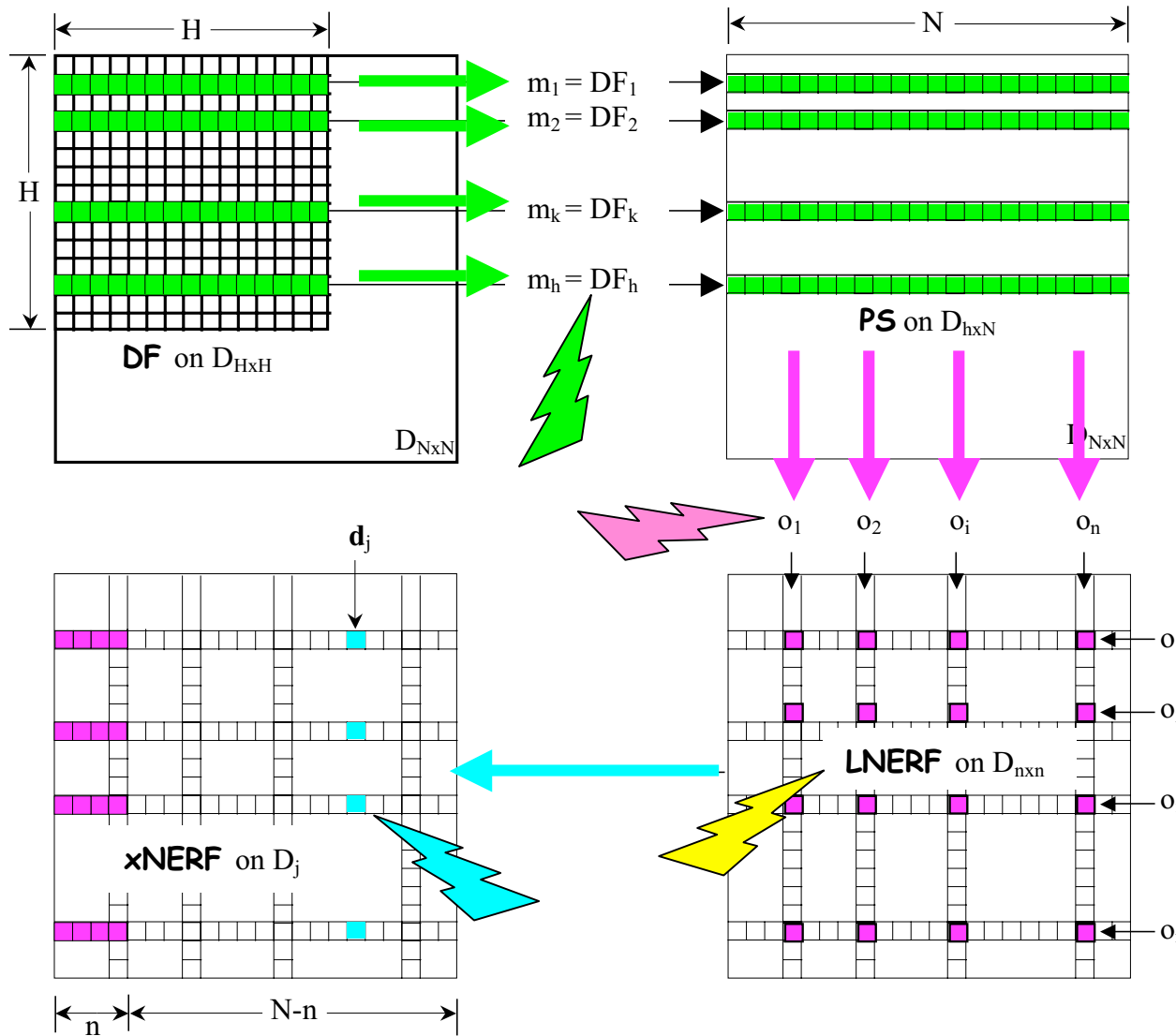
$$u_{ik}^{(q+1)} = 1 / \left[\sum_{j=1}^c (\delta_{ik} / \delta_{jk})^{1/(m-1)} \right]$$

$q = q + 1$

$$U_{\text{diff}} = \|U^{(q)} - U^{(q-1)}\|$$

but what if D is un-loadable (VL) ?

Bird's eye view of eNERF Architecture



1. DF
get h features
(indices)

2. PS
get n samples
(indices)

3. LNERF
literal clusters
in D_n

4. xNERF
fuzzy labels
in D_{N-n}

1. Algorithm DF : find distinguished features (DFs)

Input

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

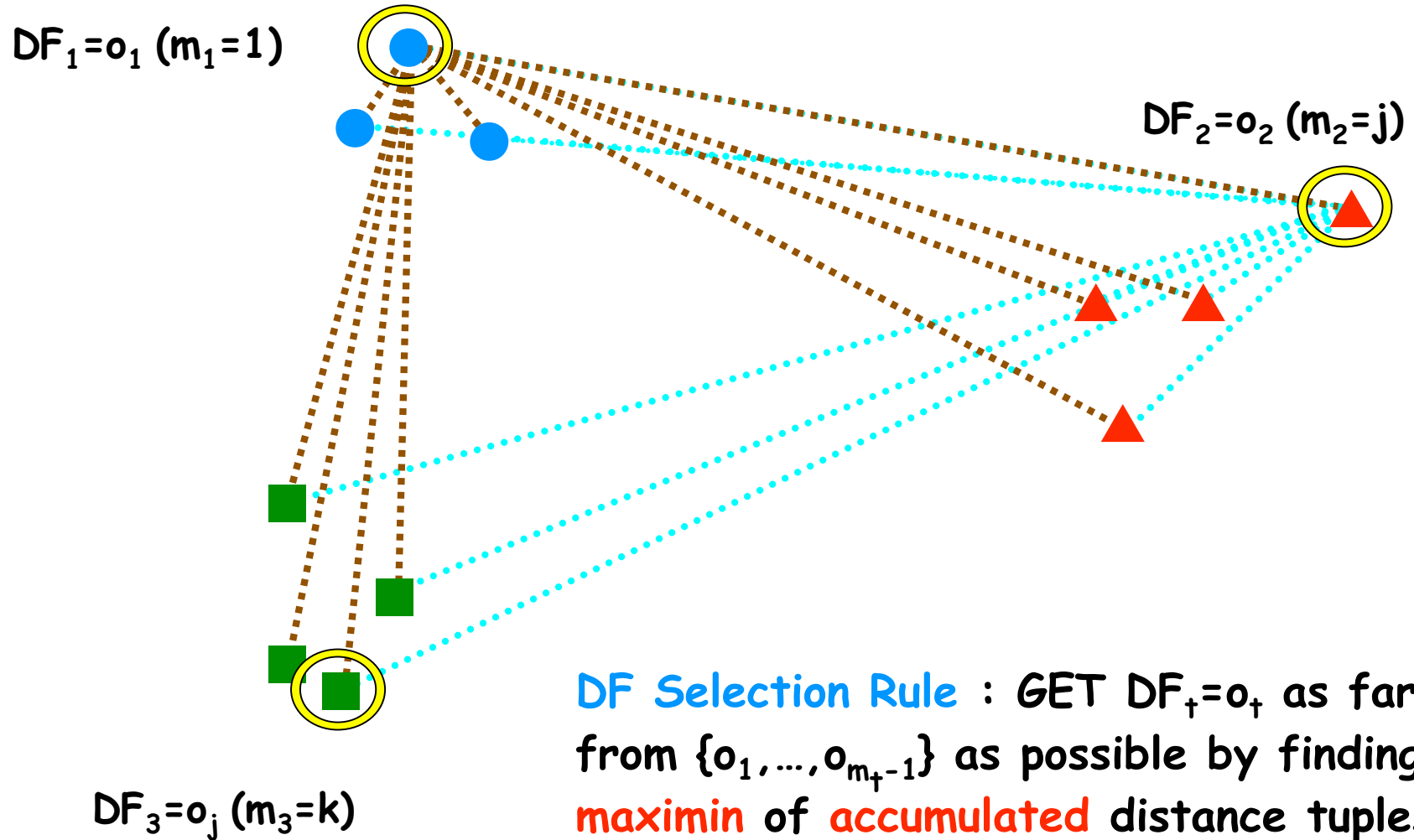
Choose

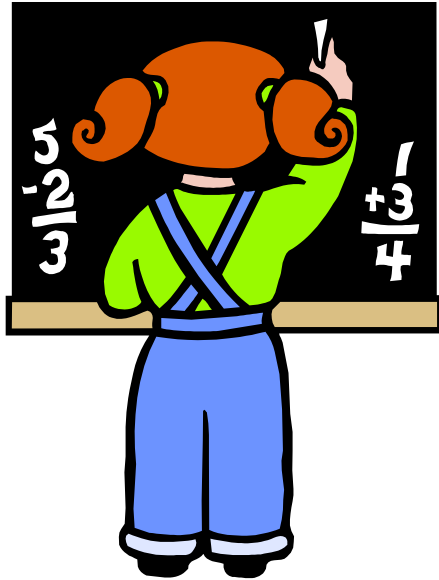
H = # of candidate rows in D_N

h = # of DFs to select : $h \leq H$

$m_1 = 1$ (object) $o_1 = 1^{st}$ DF

Get indices $\{m_i\}$ of (h) DFs





Small Result

D_H contains (c) CS (Dunn) clusters

$\{U_j : 1 \leq j \leq c \leq h\}$,

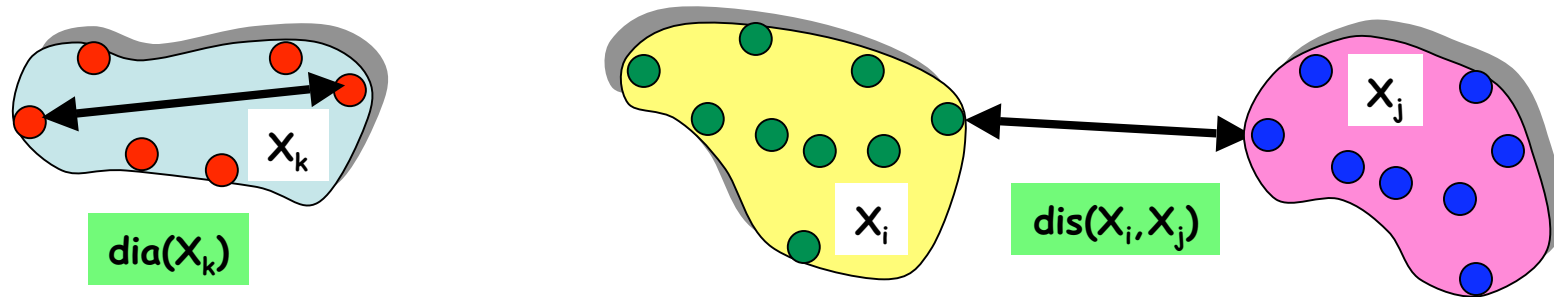


DF selects $o_j \in U_j$ for $j = 1, 2, \dots, c$

aside

What *are* CS clusters ?

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



$$V_{\text{Dunn}}(U) = \min_{1 \leq i \leq c} \left\{ \min_{\substack{1 \leq j \leq c \\ j \neq i}} \left\{ \frac{\text{dis}(X_i, X_j)}{\max_{1 \leq k \leq c} \text{dia}(X_k)} \right\} \right\}$$

X has CS clusters

$$\Leftrightarrow \max_{U \in M_{hcn}} \{ V_{\text{Dunn}}(U) \} > 1$$

2. Algorithm PS : Relational Progressive Sampling

Input $D_{h \times N}$: The DFs identify h rows of D_N

Pick b = # of EC histogram intervals

p = initial *sample* % (of N)

Δp = percentage increment

ϵ_{PS} = termination threshold

Compute initial # of samples $n = \left\lceil (pN) / 100 \right\rceil$

n/b = initial # of samples per bin

PS1. Randomly choose (w^o repl.) n columns $\{c_1, \dots, c_n\}$ of $D_{h \times N}$

PS2. For each DF ($k=1$ to h)

Define **EC histogram bins of unequal widths** for DF m_k using $\{d_{m_k c_j}\}$

- Sort initial sample $\{d_{m_k c_j}\}$
- Construct b $[*, *)$ bins with order statistics

$$\left[0, d_{\left(1 + \left\lceil \frac{n}{b} \right\rceil\right)} \right) \quad \left[d_{\left(1 + \left\lceil \frac{n}{b} \right\rceil\right)}, d_{\left(1 + \left\lceil \frac{2n}{b} \right\rceil\right)} \right) \quad \dots \quad \left[d_{\left(1 + \left\lceil \frac{(b-1)n}{b} \right\rceil\right)}, \infty \right)$$

- Many narrow bins in dense data areas
- Fewer (wider) bins in sparse data areas
- Endpoints vary from feature to feature
- Endpoints depend only on original n -sample

PS3. For each bin ($i = 1$ to b)

For each DF ($k=1$ to h)

Get **full count** for bin (i, m_k) using $\{d_{m_k}\}$

N_k^i

Get **sample count** for bin (i, m_k) using $\{d_{m_k c_j}\}$

n_k^i

PS4. For each DF ($k=1$ to h) compute **divergence**

$$\text{div}_k = n \sum_{i=1}^b \left(\frac{N_i^k}{N} - \frac{n_i^k}{n} \right) \log \left(\frac{n N_i^k}{N n_i^k} \right)$$

PS5. WHILE $\exists k [\text{div}_k > F^{-1}(1 - \epsilon_{PS})]$

$$\Delta n = \min \{ N - n, (\Delta p N / 100) \}$$

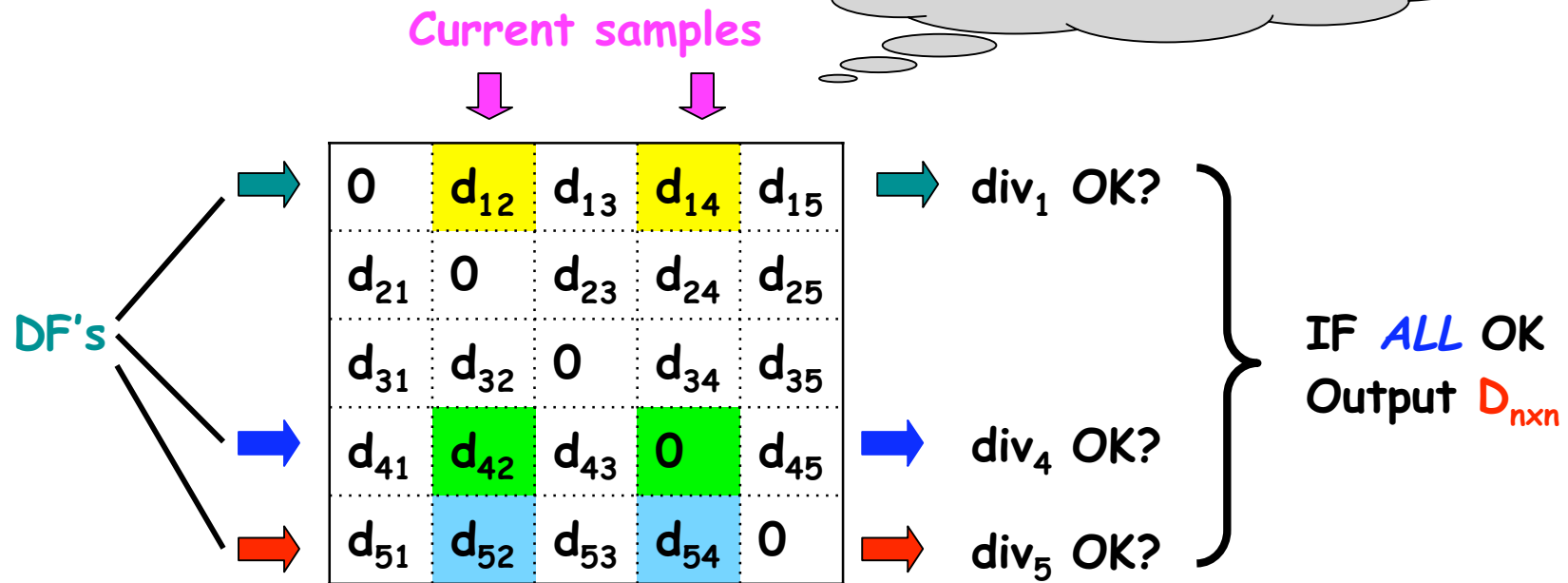
$$n = n + \Delta n$$

Randomly choose $\Delta D = \Delta n$ unused columns of $D_{h \times N}$

$$D_n = D_n + \Delta D$$

Return to PS3

Illustration of the Test



4. XNERF : Why extension is more difficult

eFFCM and geFFCM

Run LFCM on Sample X_n
to get prototypes V_n

	x_1	x_2	x_j	x_g	x_m	x_n	x_p	x_q	x_N
1	●	●	●	●	●	●	●	●	●
2	●	●	●	●	●	●	●	●	●
	●	●	●	●	●	●	●	●	●
	●	●	●	●	●	●	●	●	●
p	●	●	●	●	●	●	●	●	●



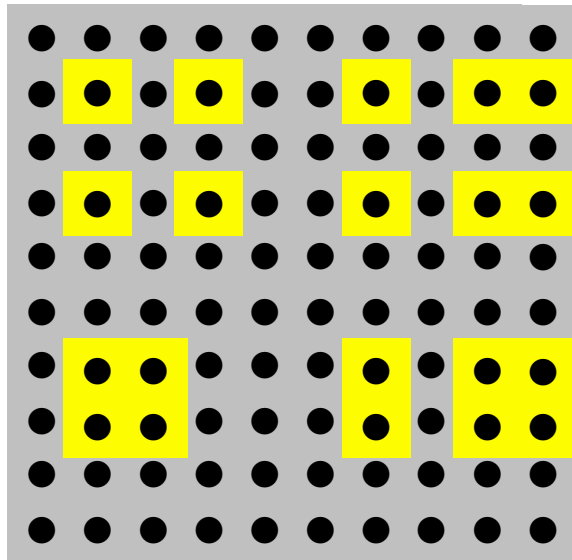
Use FONC for U in FCM
to label remaining points


$$U(\bullet) = G(V_n, x_k)$$


4. XNERF : Why extension is more difficult

eNERF

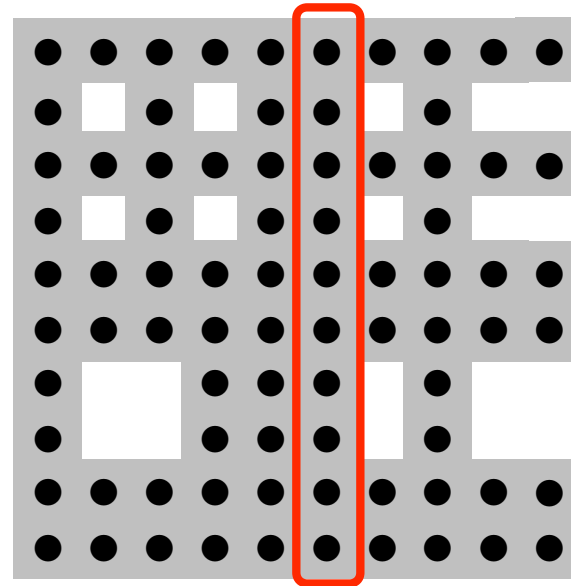
Run LNERF on Sample D_n
to get "prototypes" V_n



$D_N =$ 

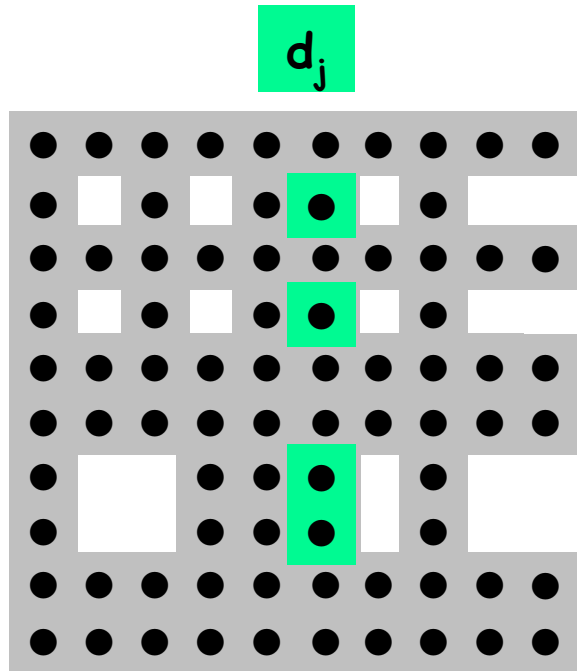
$D_n =$ 

U_n and V_n are **fixed**, but here
is what's left - now what?



Which $(n+1)$ of the N
values for o_j to use,
and how?

4. XNERF : The general idea



$$d_j = (d_{1j}, d_{2j}, \dots, d_{nj})^T$$

β_n = final value of β

$$\beta_n = (\beta_n, \dots, \beta_n)^T$$

$$D_{\beta_n} = D_n + \beta_n M$$

$$z_j = (d_j + \beta_n)$$

$$D_j = \begin{bmatrix} D_{\beta_n} & z_j \\ z_j^T & 0 \end{bmatrix}_{(n+1) \times (n+1)}$$

Get new V_{n+1} , new $\{\delta_{ik}\}$, and labels for o_j . I can't explain all the math left in the time I have today so



or



?

4. XNERF in brief

Inputs

From PS : D_n

From LNERF on D_n : β_n, U_n, V_n

Choose

$c ; m ; \epsilon_x ; \left\| U^{(q)} - U^{(q-1)} \right\|$

Do

xNERF iteration for $j=n+1$ to N

Inefficient - but works!

Output

$U_{\text{xNERF}} \mapsto U_{\text{app}} = [U_{\text{LNERF}} \mid U_{\text{xNERF}}]$ for D_N

Ex.1 Gene Product Data (GPD194)

GPD194 extracted from "ENSEMBL"

ENSEMBL has gene product Clusters discovered by clustering **sequence similarity data** with a Markov clustering algorithm (MCL)

Enright, A.J., Van Dongen, S., and Ouzounis, C.A. (2002). An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*. 30(7), 1575-1584. [<http://www.ensembl.org/>]

The GPD194_{12.10.03} data set of $c=3$ MCL Clusters

Ensembl ID	Protein Family	Biological Functionality	# of genes In Family	# of Individual Human GPs
ENSF00000000339	myotubularin	dephosphorylation	7	21
ENSF00000000073	receptor precursor	cell division and cell differentiation	7	87
ENSF00000000042	collagen alpha chain	strength and structure of connective tissue	13	86

The extraction date (12.10.03) is important because the GO is updated every 30 minutes !

"Benchmark" partition : Crisp U_{MCL} for GPD194

$$U_{MCL} = \begin{bmatrix}
 \text{Protein} & 1 & \dots & 21 & & 22 & \dots & 108 & & 109 & \dots & 194 \\
 \hline
 & \boxed{\begin{matrix} 1 & \dots & 1 \\ 0 & \dots & 0 \\ 0 & \dots & 0 \end{matrix}} & & & \boxed{\begin{matrix} 0 & \dots & 0 \\ 1 & \dots & 1 \\ 0 & \dots & 0 \end{matrix}} & & & \boxed{\begin{matrix} 0 & \dots & 0 \\ 0 & \dots & 0 \\ 1 & \dots & 1 \end{matrix}} & & & & \\
 & F_1 & & & F_2 & & & F_3 & & & &
 \end{bmatrix}$$

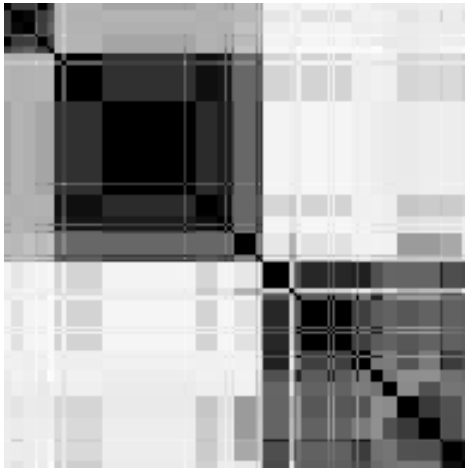
F_1 = myotubularin (muscle tissue)

F_2 = receptor precursor (cell division)

F_3 = collagen alpha chain (connective tissue)

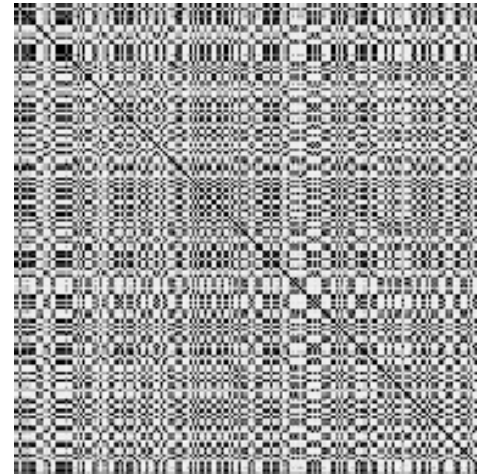
Dissimilarity Images for GPD

LOS4 Input Data



$c = 3$ clusters
visually apparent

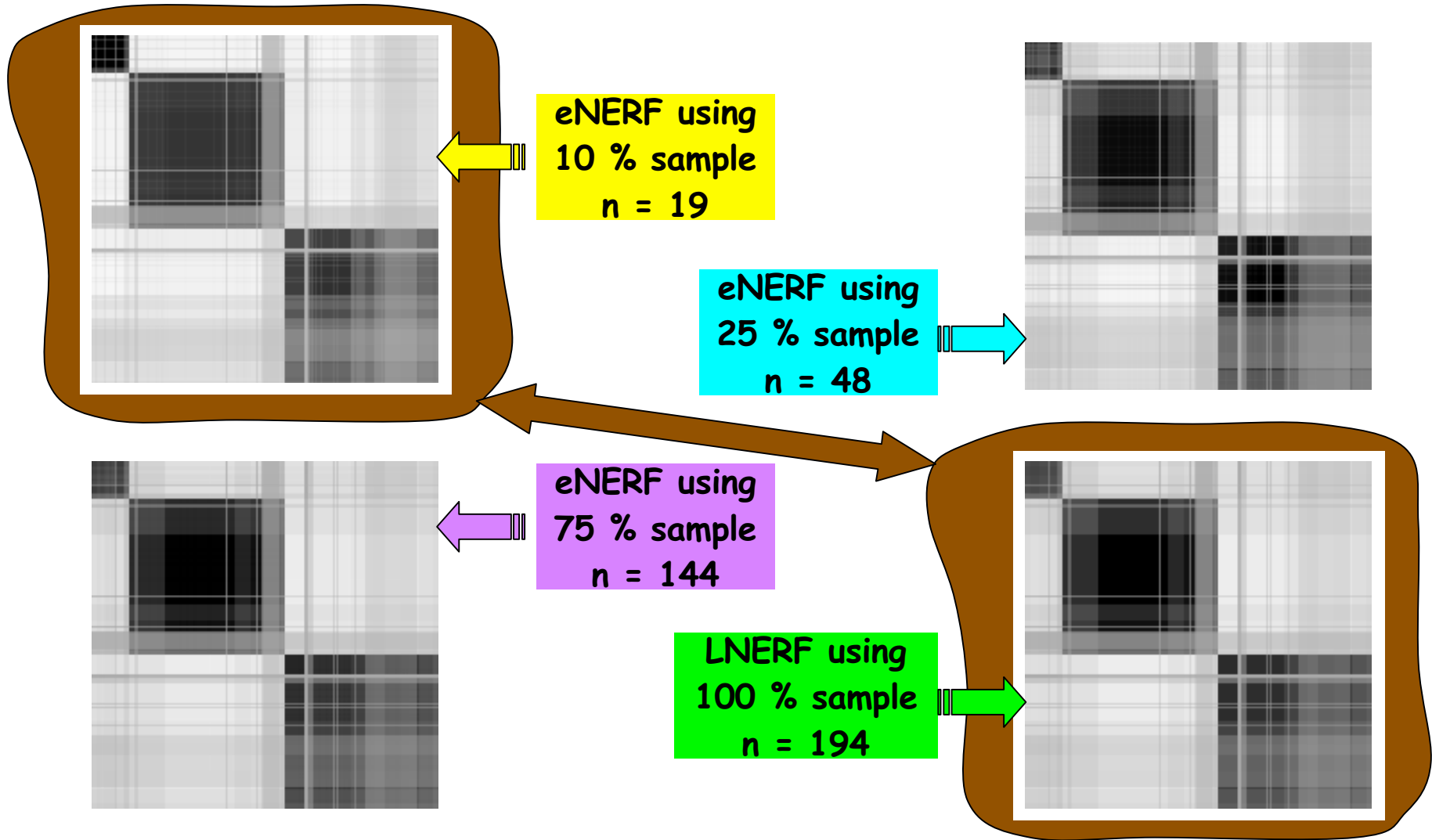
Scrambled indices



Randomly sampled
to test xNERF
(omit PS here)

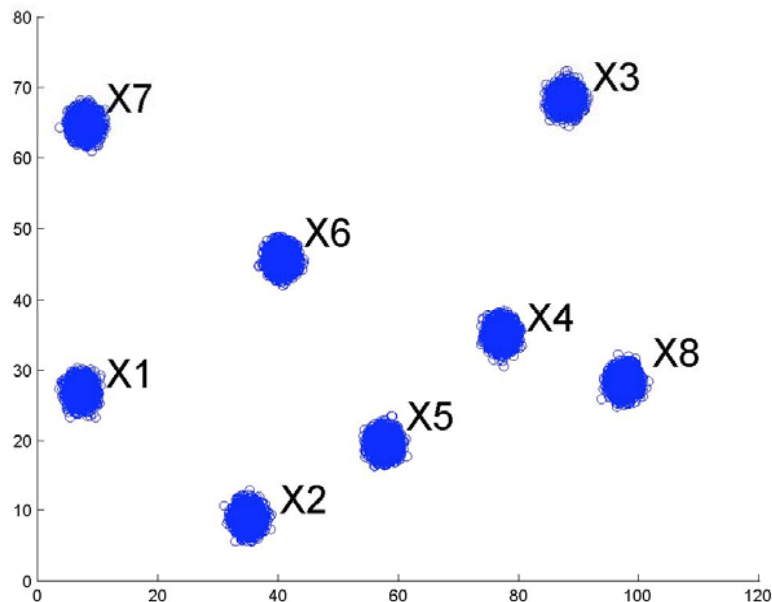
Fuzzy partition Images for GPD

$$e\text{NERF}(D_{\text{scram}}) = U_{\text{scram}} \rightarrow U_{\text{unscram}} \rightarrow I(U) = [1] - (U^T U / \max\{U^T U\})$$



Ex.2 : Normal Mixture Data X

X = 40,000 vectors from a mixture of $c=8$ bivariate normals



FCM can be used on X, so the LNERF partition U_{lit} is found using duality, with FCM; $m = 2, c = 8, \varepsilon = 10^{-5}$

These clusters are very compact and well-separated

$D = D_{40,000 \times 40,000}$ is the (Euclidean) distance matrix for $X \times X$

Storage of $D = 12.2$ GB vs MATLAB limit 250 MB

... so... D is unloadable (VL) for this platform

Progressive sampling

$\left\{ \begin{array}{l} H=10 \text{ DF candidates} \\ h=1 \text{ DF} \Rightarrow m_1 = 1 \forall H \\ b=10 \text{ EC bins} \\ p=\Delta p=1\% \Rightarrow n=\Delta n=400 \\ \varepsilon_{PS}=0.80 \end{array} \right\}$

Terminated after
1 increment
 $\Rightarrow n_{\text{final}}=800$

LNERF on $D_{800 \times 800}$

$\left\{ \begin{array}{l} m=2 \\ c=8 \\ \varepsilon_L=10^{-5} \end{array} \right\}$

LNERF terminated
after 3 iterations

xNERF on $(D_N - D_n)$

$\left\{ \begin{array}{l} m=2 \\ c=8 \\ \varepsilon_x=10^{-3} \end{array} \right\}$

49 successive calls
on 800×800 chunks

Approximation error in (derived) crisp labels

$$E_{\text{app}} = 0.5 \cdot \sum_{i=1}^8 \sum_{k=1}^{40000} |H(U_{\text{lit}})_{ik} - H(U_{\text{app}})_{ik}| / 40000 = 0$$

Training error in (derived) crisp labels

$$E_{\text{tr}} = 0.5 \cdot \sum_{i=1}^8 \sum_{k=1}^{40000} |(U_{\text{true}})_{ik} - H(U_{\text{app}})_{ik}| / 40000 = 0$$

Error of eNERF fuzzy labels as approximation to LNERF fuzzy labels

$$\|U_{\text{lit}} - U_{\text{app}}\|_F = 0.2548$$

This is the TOTAL error for
8x40,000=320,000 memberships

Summary and Conclusions : NERF Family

NERF



Input $n \times n$ Dissimilarity Matrix D_n
(neither **transitive** nor **Euclidean**)
Output Fuzzy Clusters in D_n
Limit Storage required for D_n

Input/Output/Limit (SAME as NERF)

kNERF



Does “feature extraction” in relational data
✓ VAT \Rightarrow good value for [Gaussian K] (σ)
✓ VAT \Rightarrow good guess about c (in X or D)
kNERF \rightarrow NERF as $\sigma \rightarrow \infty$

eNERF



Input VL $N \times N$ Dissimilarity Matrix D_N
Output Approx NERF Clusters in D_N
Limit No size limit
✓ Easily kernelized to get VL kNERF

Yet to do ...

①

CLUSTER VALIDITY (how many clusters to seek) in VL data?

②

Need a more efficient **EXTENSION** procedure than xNERF

③

NERF algorithm for **RECTANGULAR** (MxN) dissimilarity data

④

Perform eNERF testing on **REAL VL data** (useful applications)

Thanks mates !



G'Day



Wake up!

It's over!