



Jim Bezdek

geFFCM

geFEM

Approximate Clustering in Very Large Object Data



Rick Hathaway

The geFFCM Algorithm

Input

X_N A VL object data set

I_A *active feature* index set

SM *selection method* in {A, FA, CA, SA}

Pick

$\{b_k\}$ # of EC histogram intervals, $k \in I_A$

$\{\alpha_k\}$ termination thresholds, $k \in I_A$

p initial *sample* % (of N)

Δp percentage increment

Compute

initial # of samples $n = \left\lfloor (pN) / 100 \right\rfloor$

n/b = initial # of samples per bin

PS1

Randomly choose (*w^o replacement*)* n vectors $X_n \subset X_N$

PS2

Choose a *selection strategy* for *active features* (define I_A)



S1 = A = Single Accept

A *specified feature* passes



S2 = FA = First Accept

Any single feature passes



S3 = CA = Cumulative Accept

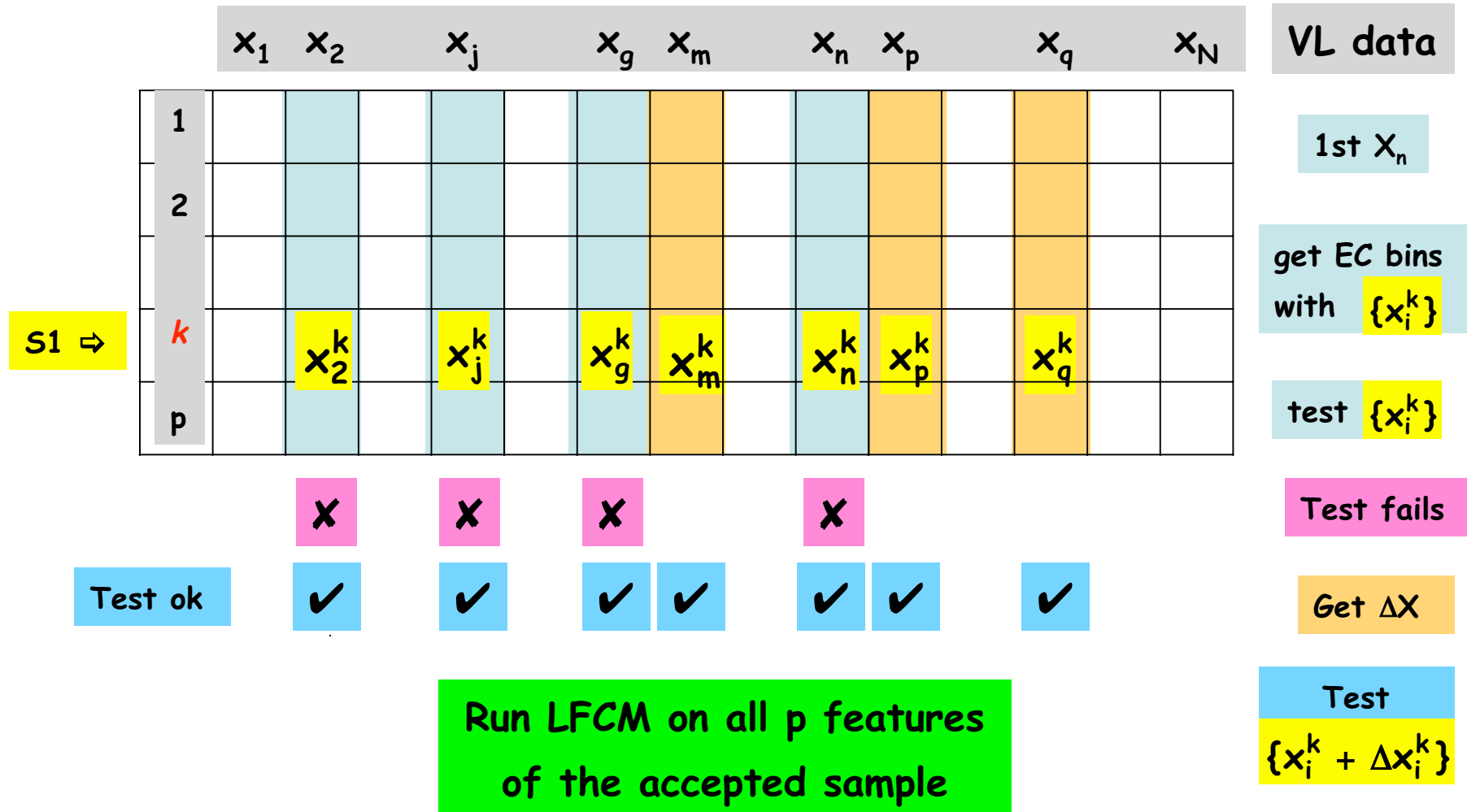
Each feature in a specified set either *did or does pass*



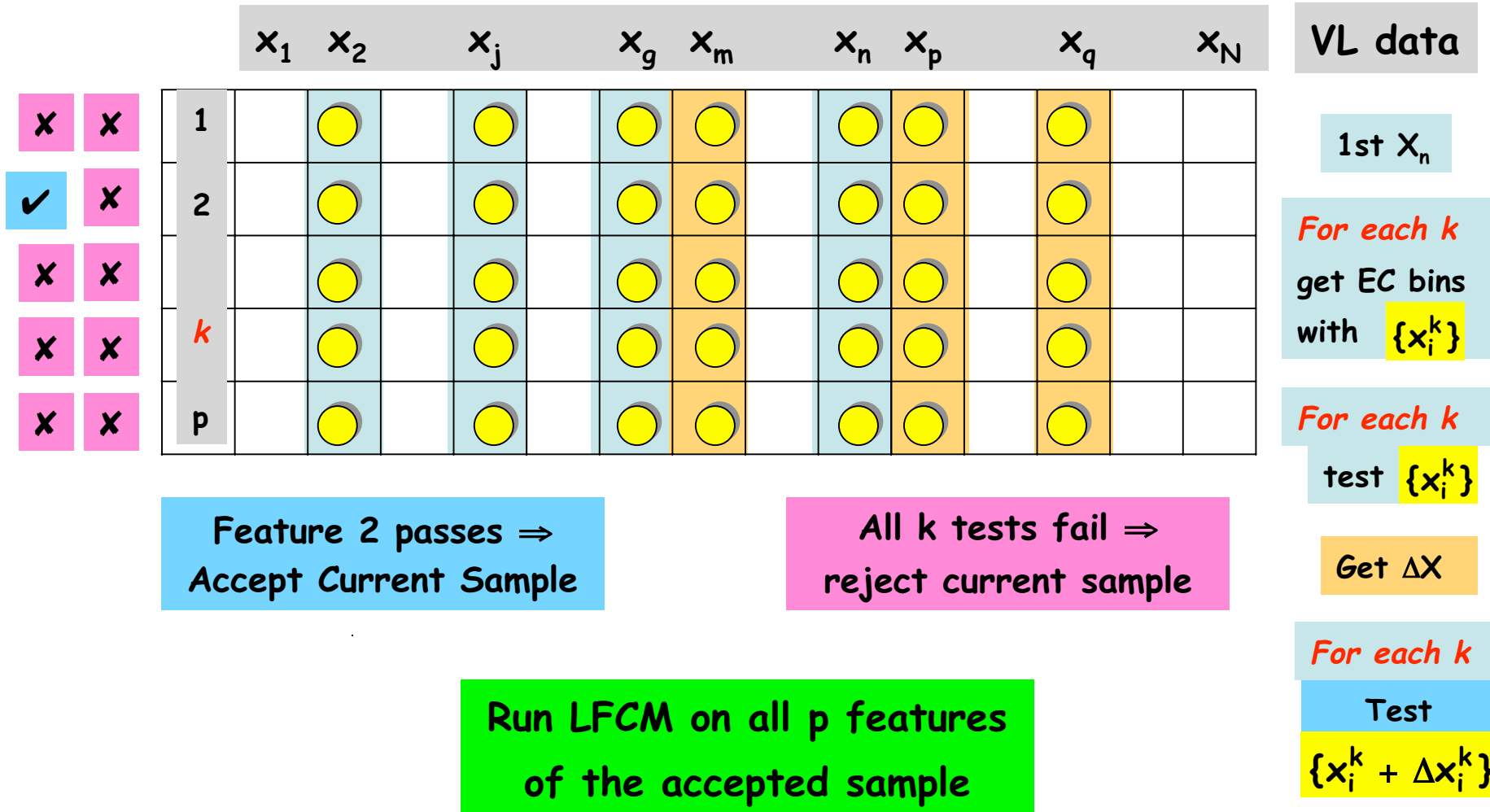
S4 = SA = Simultaneous Accept

All features in a specified set *pass simultaneously*

The *single accept* (S1=A) selection strategy for active *feature k*



The *first accept* (S2=FA) selection strategy for *any single feature*



The *cumulative accept* (S3=CA) selection strategy for *features* (j,k)

	x_1	x_2	x_j	x_g	x_m	x_n	x_p	x_q	x_N	VL data
1										1st x_n
2										
j		x_2^j	x_j^j	x_g^j	x_m^j	x_n^j	x_p^j	x_q^j		get EC bins for $\{x_i^j\}$ $\{x_i^k\}$
k		x_2^k	x_j^k	x_g^k	x_m^k	x_n^k	x_p^k	x_q^k		test $\{x_i^j\}$ $\{x_i^k\}$
p										

Get ΔX

x	x	x	x	x						j fails, k passes
✓	✓	✓	✓	✓						

Test

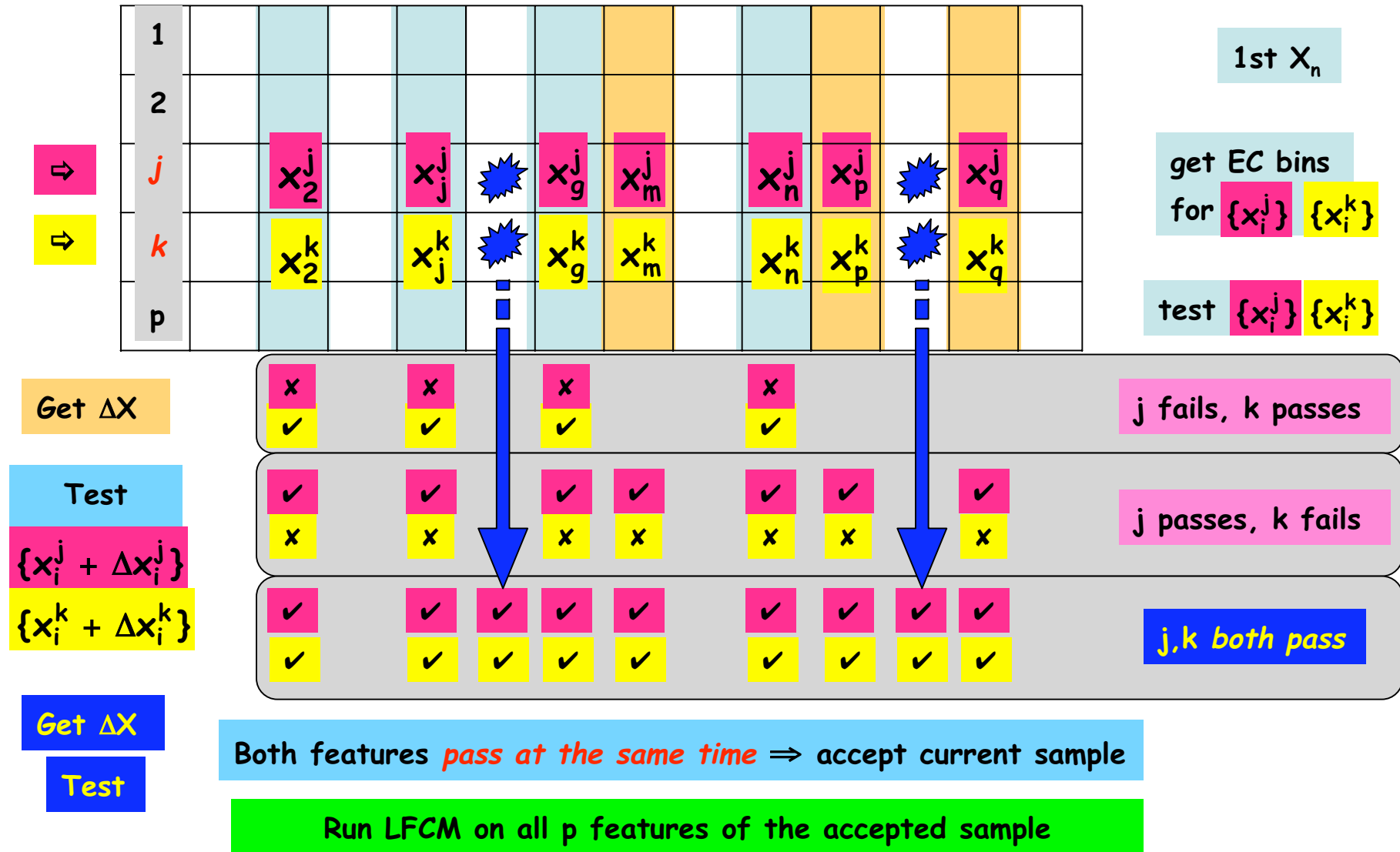
$\{x_i^j + \Delta x_i^j\}$
 $\{x_i^k + \Delta x_i^k\}$

✓	✓	✓	✓	✓	✓	✓	✓	✓		j passes, k fails
x	x	x	x	x	x	x	x	x		

Each feature *did pass*, or *does pass* \Rightarrow accept current sample

Run LFCM on all p features of the accepted sample

The *simultaneous accept* (S4=SA) selection strategy for *features* (j,k)



PS3

For each *active feature k*

- Sort *initial sample* $\{x_1^k, \dots, x_n^k\}$
- Construct b *EC bins* with order statistics

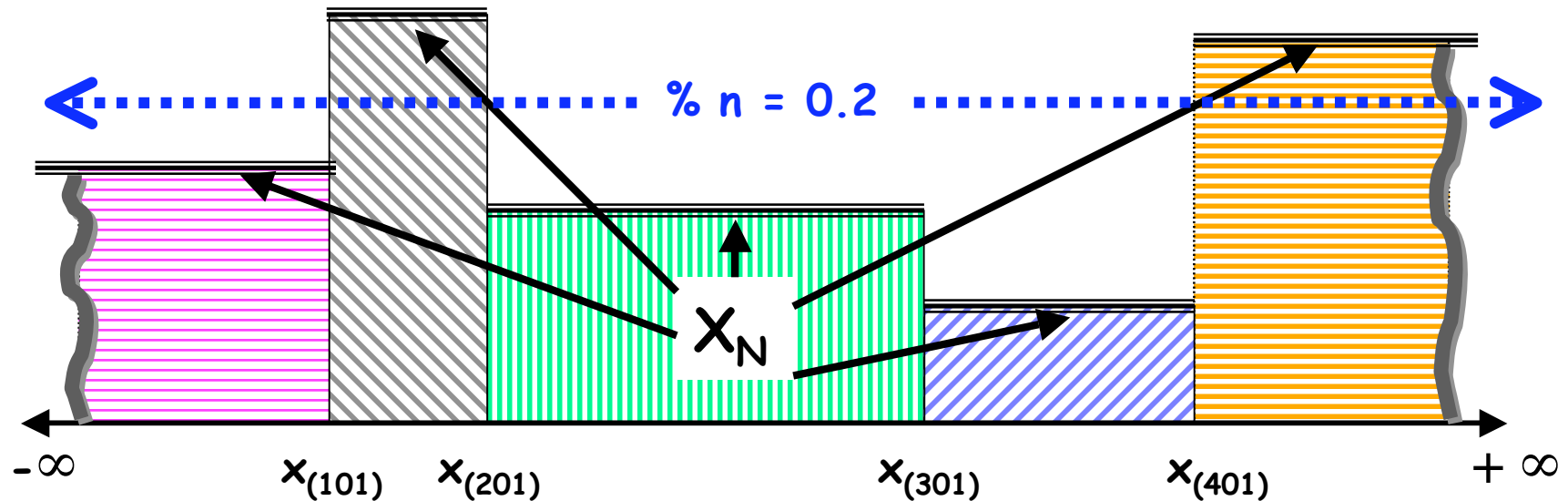
$$\left(-\infty, x_{(1/b)}^k \right] \quad \left(x_{(1/b)}^k, x_{(2/b)}^k \right] \quad \dots \quad \left(x_{(b-1/b)}^k, \infty \right)$$

- 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

What do (EC) histogram bins look like ?

$n = 500$ observations (e.g., active feature values of 500 columns of X_N)

$b = 5$ bins \Rightarrow each bin contains 100 values = 20% of samples



Determine bin endpoints by order statistics

PS4

For each *active feature* ($k=1$ to I_A)

For *each bin* ($i = 1$ to b_{ik})

Get *full count* for b_{ik}

N_k^i

Get *sample count* for b_{ik}

n_k^i

Compute *divergence**

$$\text{div}_k = n \sum_{i=1}^{b_{ik}} \left(\frac{N_k^i}{N} - \frac{n_k^i}{n} \right) \log \left(\frac{n N_k^i}{N n_k^i} \right)$$

* *w^o replacement sampling* \Rightarrow div is *not* approx. χ^2 , but here div measures *goodness of fit* (i.e., is *not* a hypothesis test statistic)

PS5

WHILE $\left(\exists k \in \mathbf{I}_A \mid \text{div}_k > F^{-1}(1 - \alpha_k) \right)$

($F = \text{cdf for } \chi^2(b_k - 1)$)



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



Get $\Delta X_{\Delta n}$ from $X_N - X_n$



$X_n = X_n + \Delta X_{\Delta n}$



Return to PS4

PS6

Run LFCM on X_n to get (U_n, V_n)

PS7

Extend fuzzy partition : $U_n \rightarrow [U_n \mid U_{N-n}]$

Ex. Set 2D

Data X_L (loadable), $N=100,000$ draws from a mixture of $c=2$ 2D normals

priors	means	covariances
$p_1 = p_2 = 0.5$	$\mu_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mu_2 = \begin{pmatrix} 10 \\ 0 \end{pmatrix}$	$\Sigma_1 = \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Algorithm Parameters

LFCM & geFFCM

$c = m = 2$
 $\varepsilon = .00001$
MaxIt = 1000
2-Norm for J_m

geFFCM

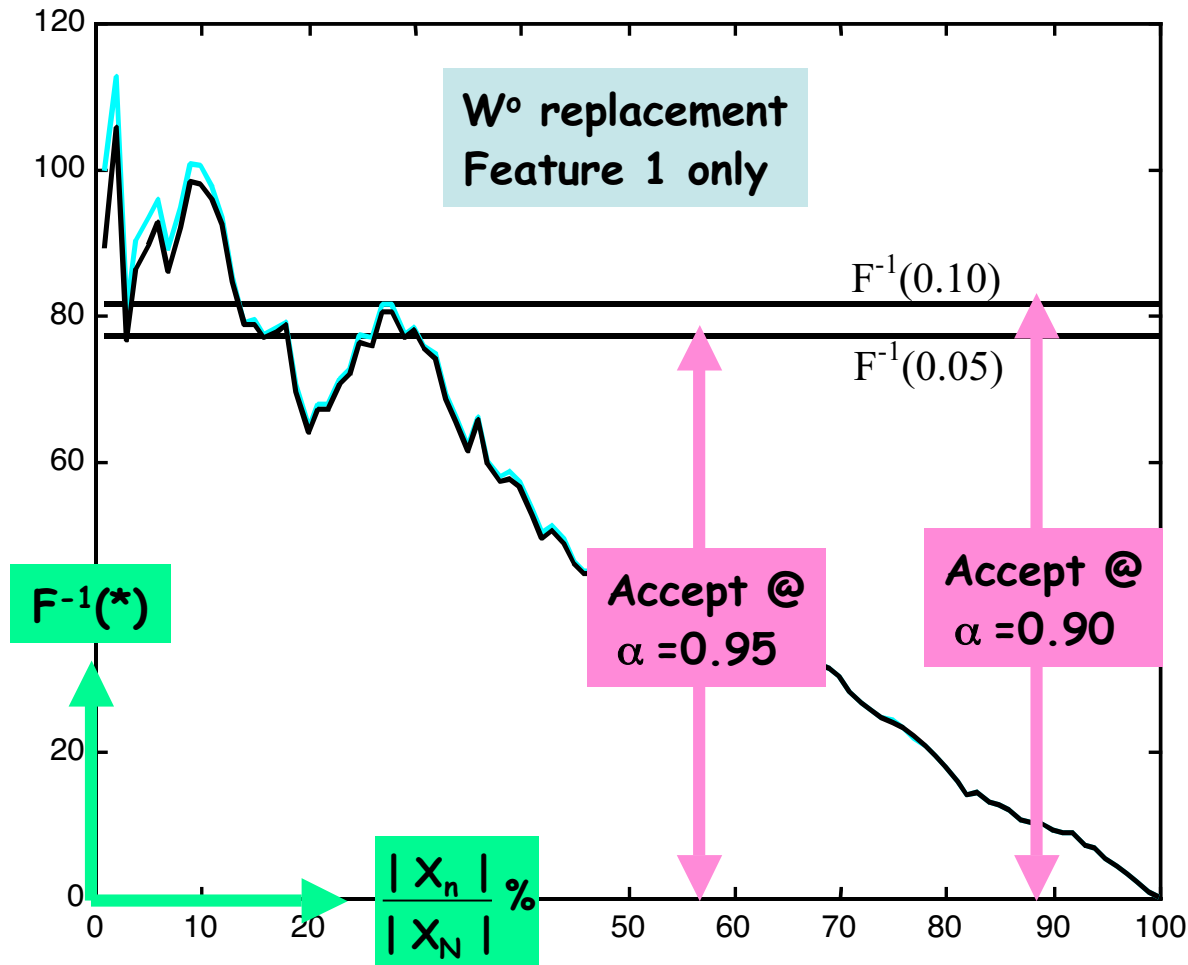
$n = 1000$ samples
 $\Delta n = 1000$ samples
 $b_k = b = 100$ bins
 $\alpha_k = \alpha \in \{0.90, 0.95\}$

Termination

$$\|U^{k+1} - U^k\| = \max_{i,j} \left\{ \left| U_{i,j}^{k+1} - U_{i,j}^k \right| \right\} < \varepsilon$$

2D

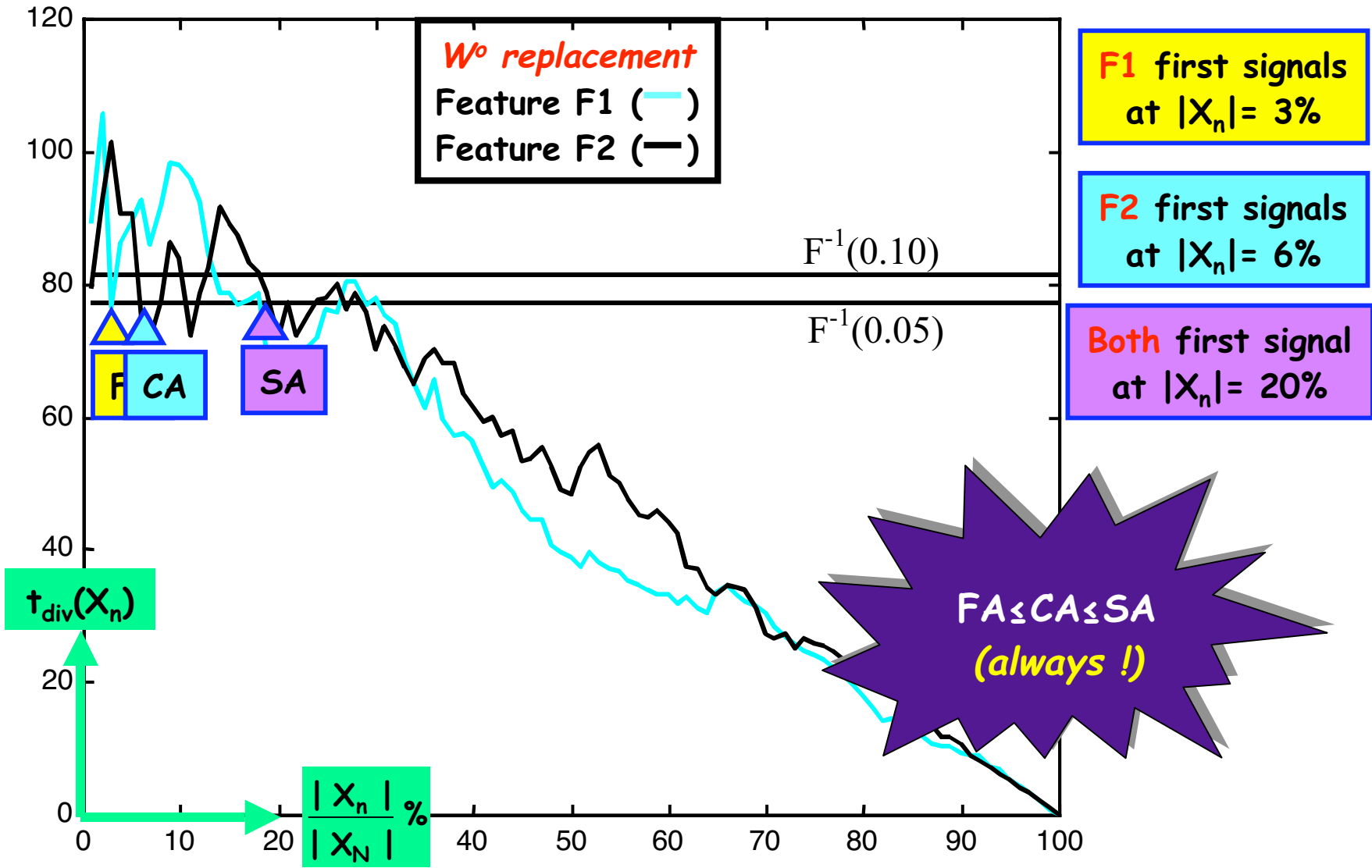
Divergence vs χ^2 : Why we use either one



Because they are basically identical !

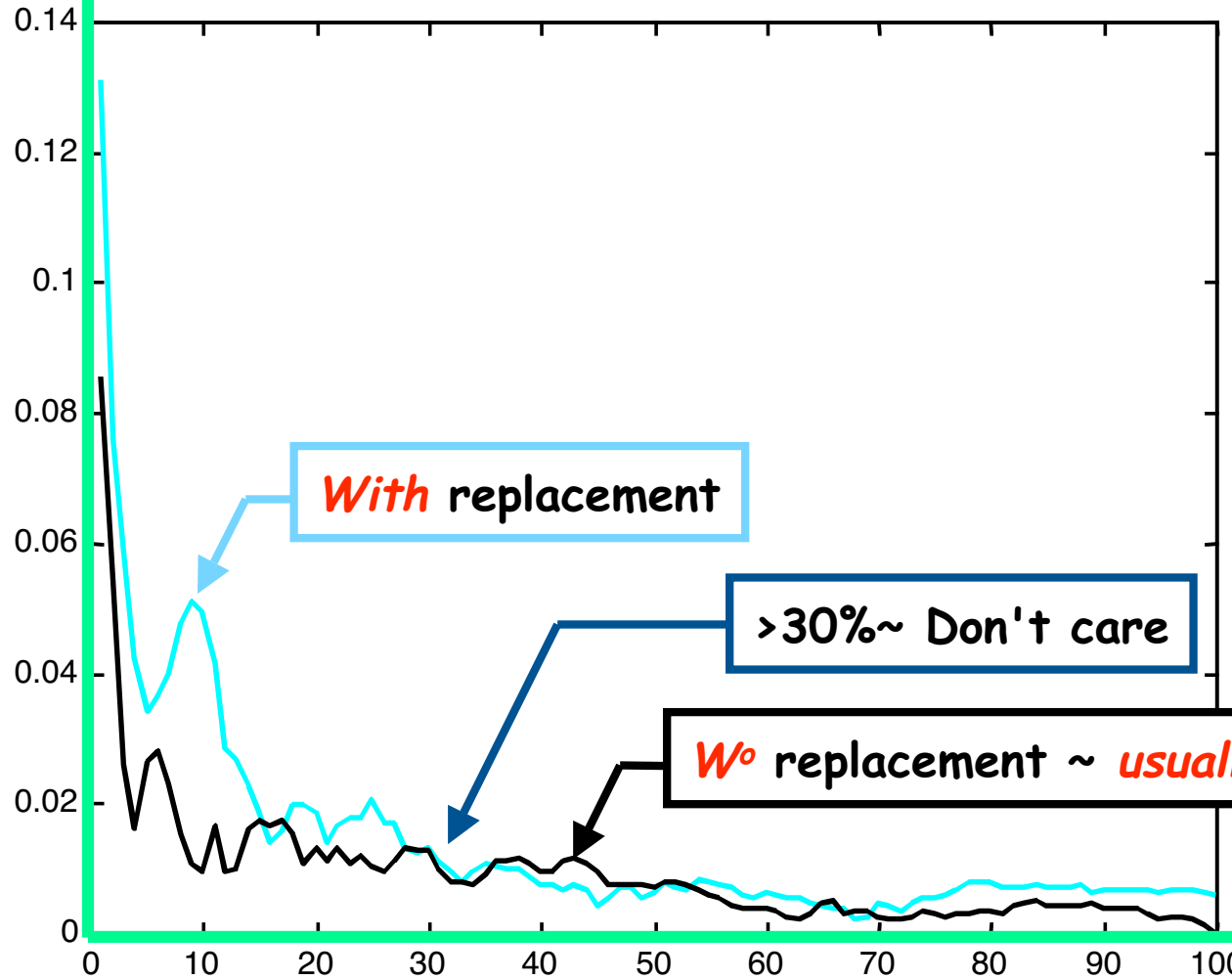
2D

Acceptance Strategies



2D

$$\|V_{X_N} - V_{X_n}\|$$



Terminal Prototypes

$$\text{LFCM}(X_N) \Rightarrow V_{X_N}$$

$$\text{LFCM}(X_n) \Rightarrow V_{X_n}$$

$$\frac{|X_n|}{|X_N|} \%$$

Ex.Set 5D

Data X_L (loadable), $N=100,000$ draws from a mixture of $c=4$ 5D normals

priors	means	covariances
$p_k = 0.25$ $k = 1, \dots, 4$	$\mu_1 = \begin{pmatrix} 0 \\ 1 \\ 3 \\ 0 \\ 0 \end{pmatrix}, \mu_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \mu_3 = \begin{pmatrix} 2 \\ 3 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \mu_4 = \begin{pmatrix} 3 \\ 3 \\ 3 \\ 0 \\ 0 \end{pmatrix}$	$\Sigma_k = \sigma^2 \mathbf{I}_5$ $k = 1, \dots, 4$ $\sigma^2 \in \{.1, .5, 1\}$

LFCM & geFFCM

geFFCM

Algorithm Parameters

Same (as 2D) except $c = 4$

Same (as 2D) Except

$b_k = b \in \{25, 50, 75, 200\}$ bins

Termination

Same

Study parameters

5D

b =		25		50		100		200	
α =		.90	.95	.90	.95	.90	.95	.90	.95
X ₁ X ₂ X ₃ X ₄ X ₅	X _{S1}	19	21	15	19	12	14	7	10
	X _{S1}	17	27	14	19	9	14	8	12
	X _{S1}	17	26	16	21	13	15	9	13
	X _{S1}	13	22	12	17	13	19	10	12
	X _{S1}	19	27	18	23	13	18	11	13
FA	X _{S2}	3	6	4	6	3	5	3	4
time, secs		.14	.15	.16	.17	.18	.19	.20	.21
CA	X _{S3}	35	47	30	35	24	29	17	22
time, secs		.17	.19	.20	.21	.22	.25	.26	.29
SA	X _{S4}	44	54	35	40	27	33	21	26
time, secs		.21	.23	.22	.24	.25	.26	.29	.31

"good" separation

$$\sigma^2 = 0.1$$

Typical Result

25 trials ave.

|X_n| of %|X_N|

Trend Studies

%|X_N| vs b

%|X_N| vs SS

%|X_N| vs α

%|X_N| vs cpu time

SS vs σ²

5D

Average Trends

Sample size vs b

25	50	100	200
25.5	19.8	16.4	12.3

↑ b 25 → 200 ↓ % |X_N|
by more than 1/2

Sample size vs α

↑ α .90 → .95 ↑ % |X_N|
about 5 %

Sample size vs cpu time

cpu time ↑ almost linearly
with b but *not* with % |X_N|

5D

Average Trends

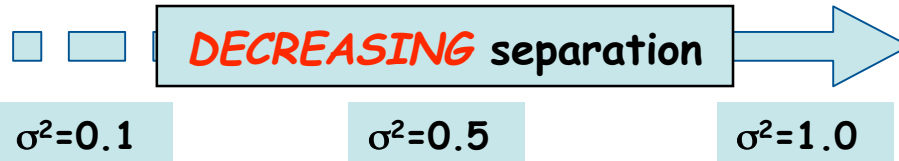
Sample size vs [SS vs σ^2]

Strategy	$\sigma^2 = 0.1$	$\sigma^2 = 0.5$	$\sigma^2 = 1.0$
A	15.7	15.7	15.5
FA	4.3	3.8	3.4
CA	29.9	30.6	30.8
SA	35.0	36.0	36.0
col. ave.	21.4	21.5	21.2

Separation (σ^2) has negligible effect on % $|X_N|$ for **each** (SS)

Separation (σ^2) has negligible effect on % $|X_N|$ over **all** SS

Approximation and Acceleration Measures



		geFFCM	LFCM			geFFCM	LFCM			geFFCM	LFCM
SS2 FA=1	t_{acc}	5.21	-	12.44	-	28.41	-				
	$\%E_{tr}$	0.03	0.03	8.90	8.85	22.40	22.18				
	$\%E_{app}$	0.00	-	0.95	-	3.15	-				
	E_v	0.00	-	0.01	-	0.03	-				
SS3 CA=5	t_{acc}	2.61	-	3.59	-	4.34	-				
	$\%E_{tr}$	0.03	0.03	8.86	8.85	22.19	22.18				
	$\%E_{app}$	0.00	-	0.23	-	0.78	-				
	E_v	0.00	-	0.00	-	0.00	-				
SS4 SA=5	t_{acc}	2.30	-	2.97	-	3.41	-				
	$\%E_{tr}$	0.03	0.03	8.85	8.85	22.19	22.18				
	$\%E_{app}$	0.00	-	0.20	-	0.69	-				
	E_v	0.00	-	0.00	-	0.00	-				

Average Trends in *Acceleration*

Acceleration vs σ^2

↓ separation (increasing σ^2) ↑ T_{acc}

FA : T_{acc} ↑ 545% from $\sigma^2 = 0.10$ to 1.00

SA : T_{acc} ↑ 48% from $\sigma^2 = 0.10$ to 1.00

Acceleration vs SS

T_{acc} ↓ As (FA → CA → SA)

833% AT $\sigma^2 = 1$!!!

5D

Average Trends in *Approximation*

Approximation Error vs [SS and σ^2]

E_{app} ↓ As (FA → CA → SA)

E_{app} ↑ as separation ↓

Training Error vs [SS and σ^2]

E_{tr} ↑ as separation ↓

Prototype Error E_v vs [SS and σ^2]

E_v is ~ 0 for all cases !

**Probabilistic Clustering
with geFEM**

5D

Typical Result (10 trials ave.)
for "good" separation ($\sigma^2=0.1$)

		FCM		EM		
		geFFCM	LFCM	geFEM	LEM	
SS2 FA=1	Acceleration	t _{acc}	5.09	1	8.71	1
	Both accelerate their literal counterparts very well					
	Prototype Error	E _v	0.01	0	0.01	0
SS3 CA=5	Acceleration	t _{acc}	2.56	1	2.98	1
	% Training Error	E _{tr}	0.03	0.03	0.03	0.03
	Both estimate the true labels with high accuracy					
SS4 SA=5	Acceleration	t _{acc}	2.29	1	2.61	1
	% Training Error	E _{tr}	0.03	0.03	0.03	0.03
	% Approx. Error	E _{app}	0.20	0	0.20	0
Both approximate their literal counterparts very well						

Does $|X_S|$ % of $|X|$ \uparrow with $|X|$? **NO** - it \downarrow !!!

$|X_n|=100,000$

5D*

$|X_N|=1,600,000$

25 trials ave.

$\alpha = 0.95$

$\sigma^2 = 0.50$

b =		25		50		100		200	
X =		X_n	X_N	X_n	X_N	X_n	X_N	X_n	X_N
x_1	$ X_{S1} $ %	27	15	25	12	20	8	13	9
x_2	$ X_{S1} $ %	24	12	17	10	14	9	11	5
x_3	$ X_{S1} $ %	25	17	18	11	16	9	12	6
x_4	$ X_{S1} $ %	25	15	21	13	16	8	9	8
x_5	$ X_{S1} $ %	27	13	20	14	15	11	11	9
FA	$ X_{S2} $ %	5	1	5	1	5	1	3	1
	time, secs	.15	1.9	.18	2.2	.19	2.4	.21	2.7
CA	$ X_{S3} $ %	48	35	37	31	29	21	21	17
	time, secs	.19	2.4	.27	2.8	.25	3.1	.28	3.5
SA	$ X_{S4} $ %	54	52	42	40	33	29	24	23
	time, secs	.22	3.3	.26	3.5	.26	3.6	.28	4.1

$\%|X_S|$ for **BIG** X_N
 \downarrow in **ALL** cells

ave cpu time

0.23 for X_n

2.96 for X_N

grows $O(c)$ with $|X|$

5D* Elastic control of $n = |X_n|$ to reduce sample size

Recall that for each active feature k , we compute

$$D_k^* = \min \{n, n^*\} \left[\sum_{i=1}^{b_{ik}} \left(\frac{N_i^k}{N} - \frac{n_i^k}{n} \right) \log \left(\frac{n N_i^k}{N n_i^k} \right) \right]$$

$$D_k = n \left[\sum_{i=1}^{b_{ik}} \left(\frac{N_i^k}{N} - \frac{n_i^k}{n} \right) \log \left(\frac{n N_i^k}{N n_i^k} \right) \right]$$

Choose a target sample size n^* and define

2 Cases

$\min \{n, n^*\} = n^* \Rightarrow$ termination at $n^* < n$
 D^* prevents oversampling of X_N

$\min \{n, n^*\} = n \Rightarrow$ termination at $n < n^*$
 D satisfied by sample smaller than n^*

5D*

$|X_N|=1,600,000$

25 trials ave.

$\alpha = 0.95$

$\sigma^2 = 0.50$

$n^* = 20,000 = 1.25\% |X_N|$

52% of N = 832,000 samples

3% of N = 48,000 samples

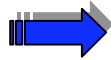
Elastic when $D^* > 1.25\%$

b =		25		50		100		200	
X =		D	D*	D	D*	D	D*	D	D*
x_1	$ X_{S1} %$	15	2	12	2	8	1	9	1
x_2	$ X_{S1} %$	12	2	10	1	9	1	5	1
x_3	$ X_{S1} %$	17	2	11	1	9	1	6	1
x_4	$ X_{S1} %$	15	2	13	1	8	1	8	1
x_5	$ X_{S1} %$	13	2	14	1	11	1	9	1
FA	$ X_{S2} %$	1	1	1	0	1	1	1	1
time, secs		1.95	1.94	2.19	2.18	2.42	2.40	2.67	2.63
CA	$ X_{S3} %$	35	3	31	2	21	2	17	2
time, secs		2.44	1.97	2.79	2.22	3.10	2.45	3.55	2.72
SA	$ X_{S4} %$	52	3	40	2	29	2	23	2
time, secs		3.28	1.99	3.53	2.23	3.63	2.44	4.07	2.65

5D*

832,000 samples

48,000 samples



5 times faster same accuracy

Same error rates LFCM vs Approx.

		b=25 bins		b=200 bins		LFCM		
		D	D*	D	D*	D	D*	
SS2 FA=1	Acceleration	t_{acc}	13.09	12.55	10.00	9.03	1	1
	% Training Error	E_{tr}	8.89	8.89	8.87	8.87	8.83	8.83
	% Approx. Error	E_{app}	0.83	0.85	0.68	0.70	0	0
	Prototype Error	E_v	0.01	0.01	0.01	0.01	0	0
SS3 CA=5	Acceleration	t_{acc}	2.80	9.70	3.89	8.24	1	1
	% Training Error	E_{tr}	8.83	8.83	8.83	8.83	8.83	8.83
	% Approx. Error	E_{app}	0.04	0.21	0.07	0.28	0	0
	Prototype Error	E_v	0.00	0.00	0.00	0.00	0	0
SS4 SA=5	Acceleration	t_{acc}	1.84	9.51	3.29	8.10	1	1
	% Training Error	E_{tr}	8.83	8.83	8.83	8.83	8.83	8.83
	% Approx. Error	E_{app}	0.03	0.21	0.06	0.28	0	0
	Prototype Error	E_v	0.00	0.00	0.00	0.00	0	0

5D*

b=100 bins
SS3=CA

25 trials aves.
n* = 20,000

$\alpha = 0.95$
 $\sigma^2 = 0.50$

$ X_n \Rightarrow$	100,000		200,000		400,000		800,000	
	D	D*	D	D*	D	D*	D	D*
$ X_n $ as % of $ X_N $	28	23	26	13	24	7	21	4
geFFCM time, secs	3.28	2.81	6.55	3.79	12.75	5.19	23.29	7.93
LFCM time, secs	10.87	10.84	21.58	21.54	42.93	42.72	85.91	85.98
Acceleration, t_{acc}	3.40	3.89	3.58	5.69	3.62	8.24	4.02	10.84

t_{acc} with D : 3.40 \Rightarrow 3.58 \Rightarrow 3.62 \Rightarrow 4.02

t_{acc} with D* : 3.89 \Rightarrow 5.69 \Rightarrow 8.24 \Rightarrow 10.84

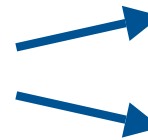
\therefore advantage of using D* \uparrow as $|X_n|$ \uparrow

Empirical Conclusions

\dagger_{acc}



Acceleration highest when



Separation is highest

SS is minimal (FA)

E_{tr}



Training errors comparable to LFCM for all 4 SS's

E_{app}



Smallest approx. errors for stringent SS's (CA and SA)

PS



Is a very general scheme which is easily adaptable for extension to VL data with *many* algorithms

Yet to do : geFFCM

- geFFCM was designed for VL data, but so far, no real tests have been made
 - Process VL data
 - Should work ok !
- Try simpler termination measures (e.g. Euclidean $\|*\|$)
- For real VL data, compare to simple random sampling

Thanks mates !



G'Day



Wake up!

It's over!