Cluster Forests

Computational Statistics and Data Analysis, vol. 66 (2013), pp. 178-192

Donghui Yan
Berkeley University

Aiyou Chen
Google Inc

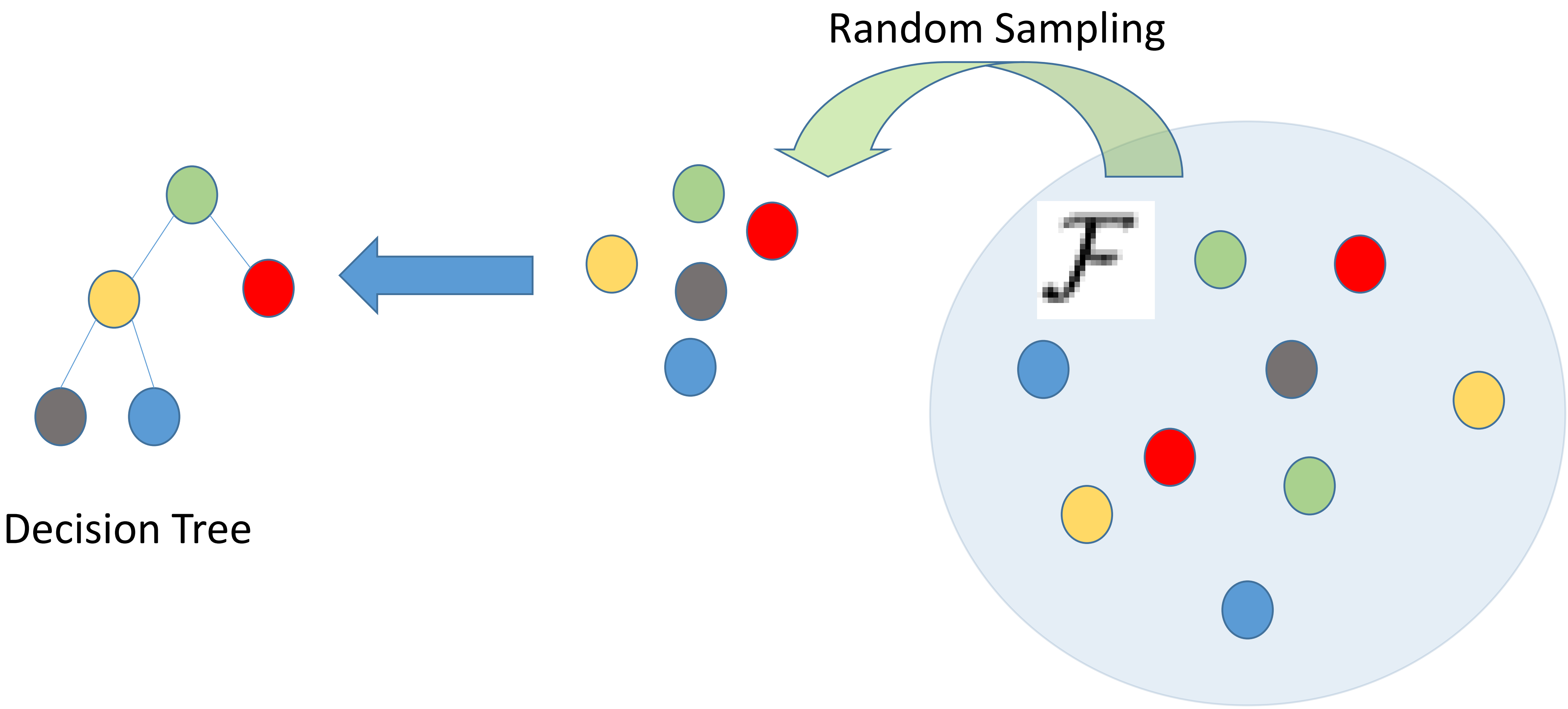
Michael I. Jordan
Berkeley University

2015.10.21 Group Meeting Report
指導教授 | 林志青
0356624 | 葉美伶

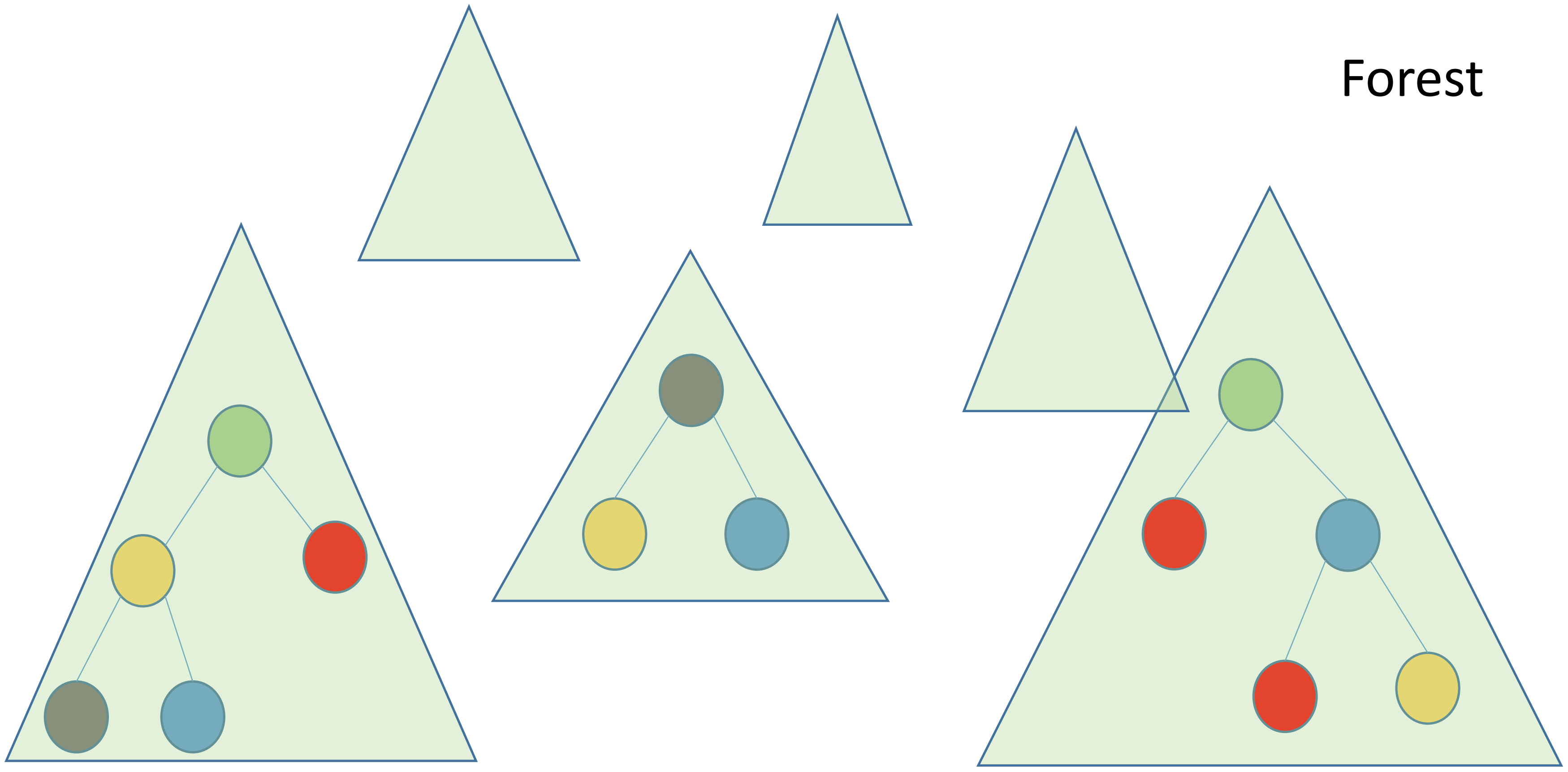
Outline

- Background
  - Before CF - Random forest
- Cluster Forest
  - The Method
- Experiments Result
  - Summary of UC Irvine datasets
  - Evaluation of the clustering result (ρr, ρc)
- Conclusion

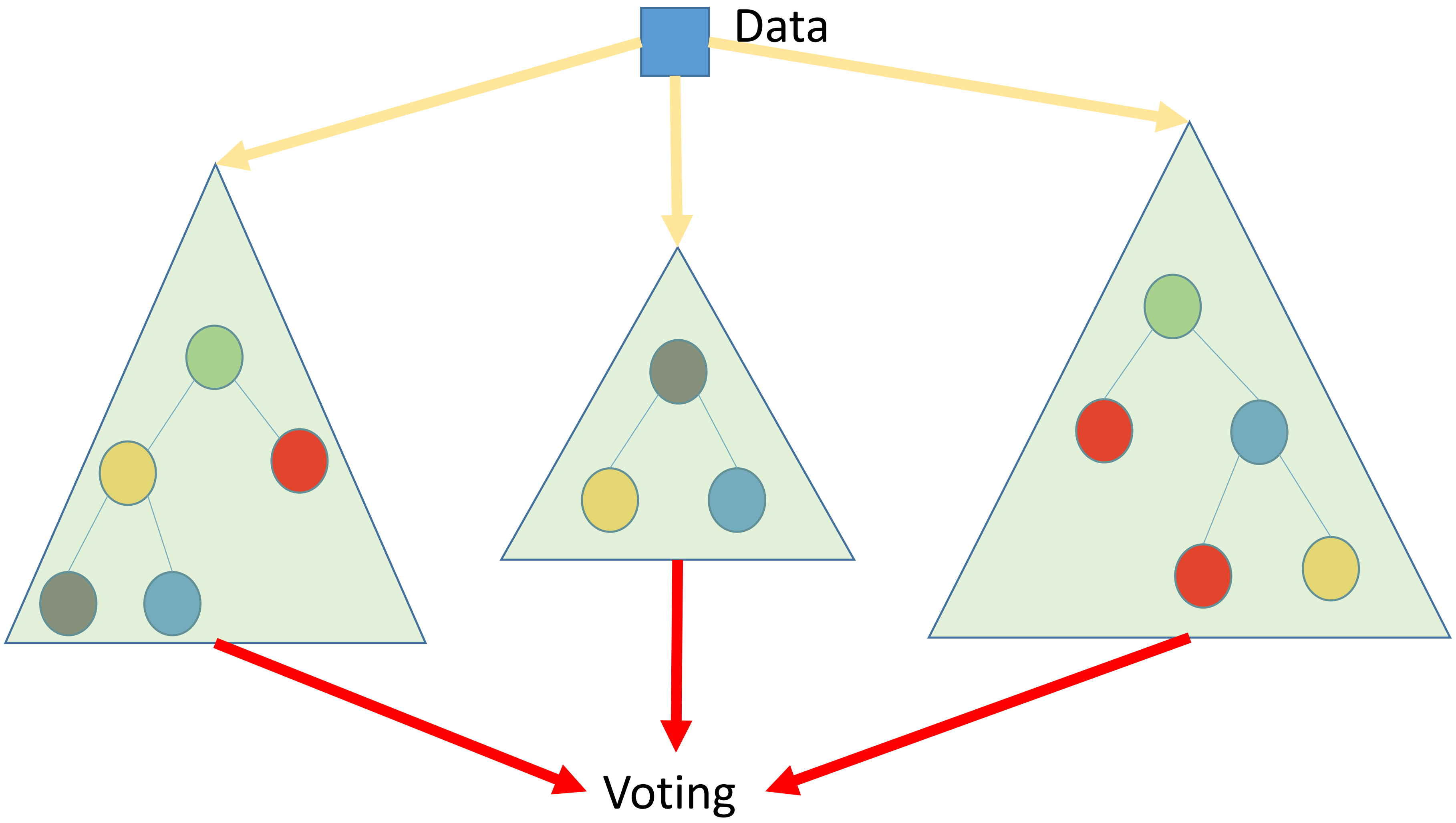
Before CF - Random Forest



Before CF - Random Forest



Before CF - Random Forest



Cluster Forest - Method

- Growth of clustering vectors
  - [alg 1] Feature competition
    - It aims to provide a good initialization for the growth of a clustering vector.
    - It prevent noisy or “weak” features from entering the clustering vector at the initialization
  - [alg 2] The growth of a clustering vector

- The CF algorithm

Method – Feature competition

Aims to,

- Provide a good initialization for the growth of a clustering vector.
- Prevent noisy or “weak” features from entering the clustering vector at the initialization.

clustering vector: data will projected on it

F: feature space

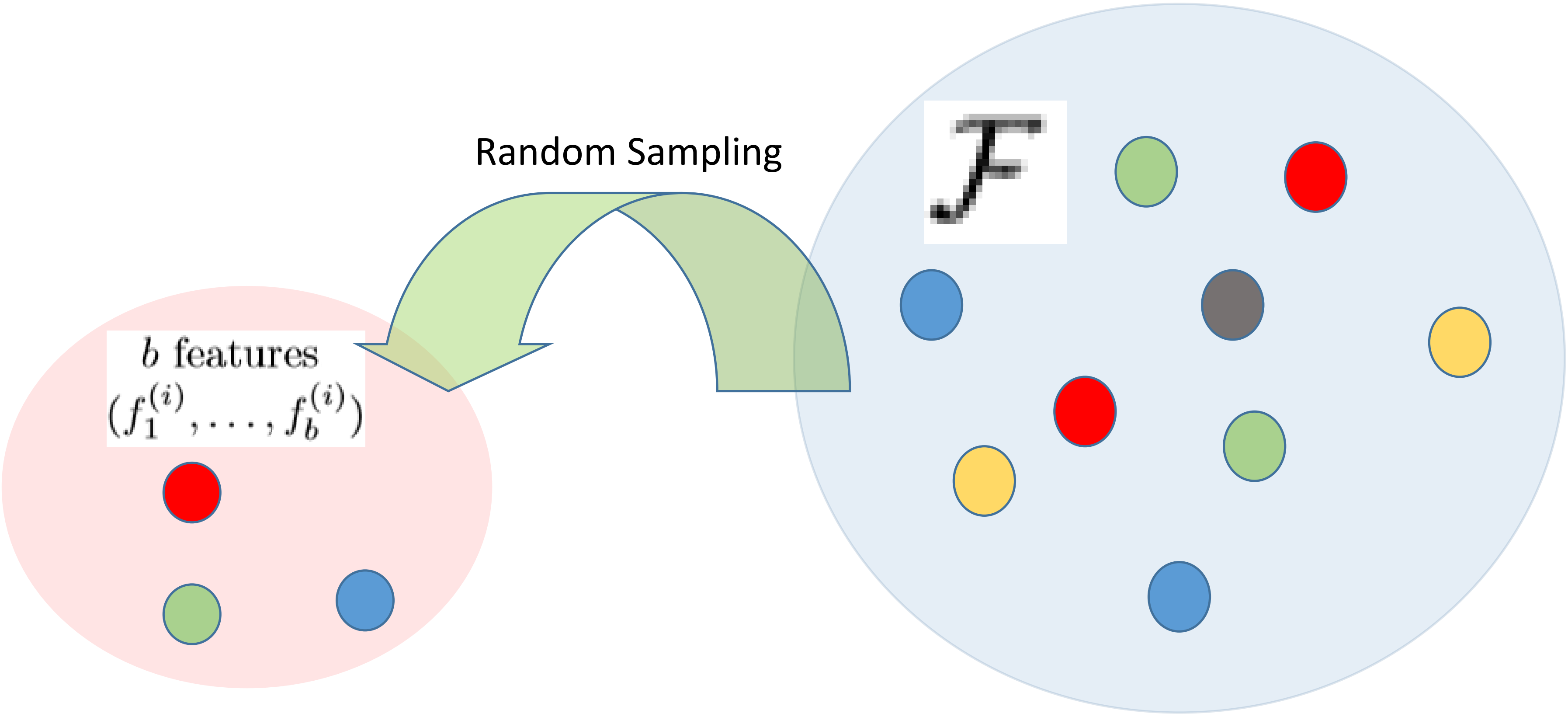
$\tilde{f}$: set of feature, current in used

SSw: within-cluster squared error distance

SSb: between-cluster squared error distance

Method - Feature competition

1: Sample b features, $f_1^{(i)}, \ldots, f_b^{(i)}$, from the feature space $\mathcal{F}$

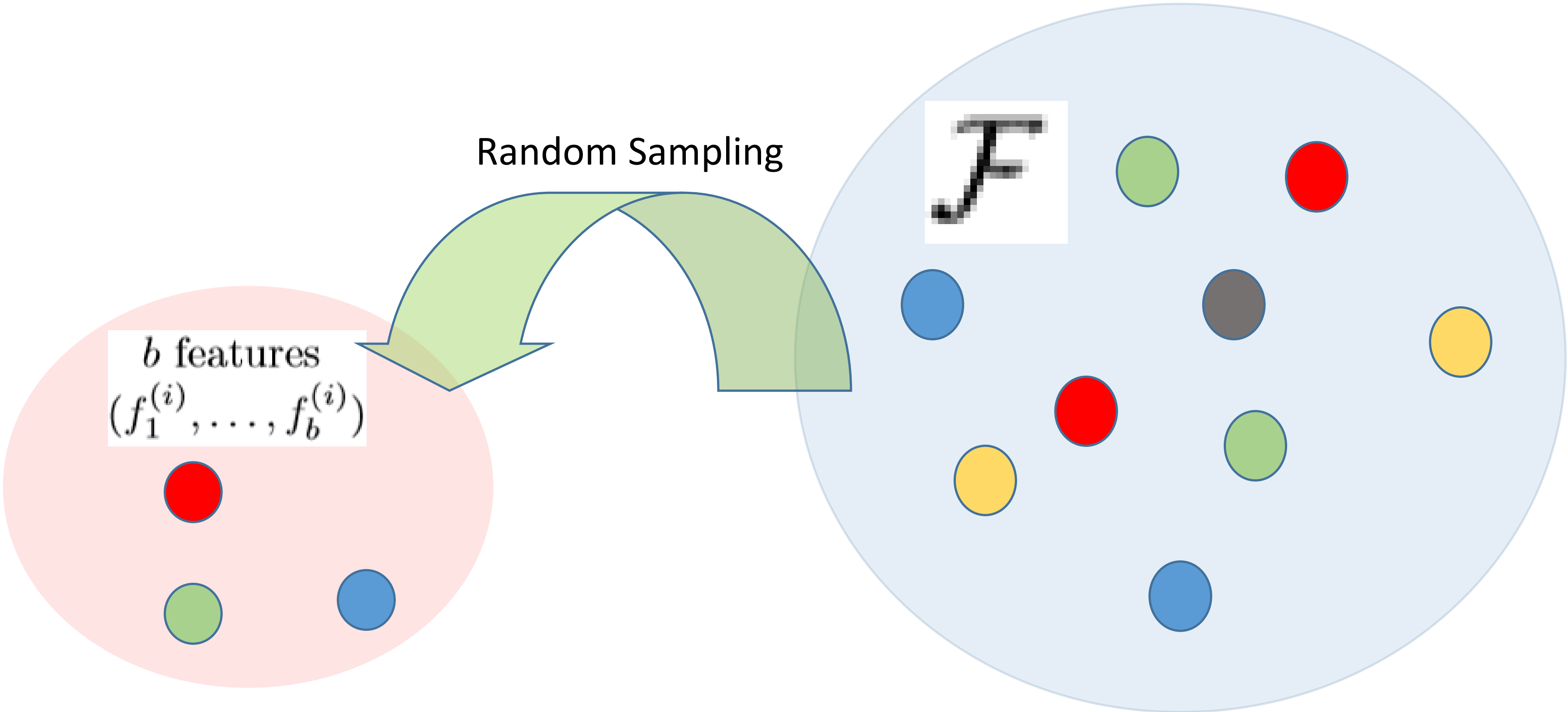


Method - Feature competition

$$\kappa(\tilde{\boldsymbol{f}}) = \frac{SS_W(\tilde{\boldsymbol{f}})}{SS_B(\tilde{\boldsymbol{f}})}$$

1: Sample b features, $f_1^{(i)}, \ldots, f_b^{(i)}$, from the feature space $\mathcal{F}$

2: Apply $\boxed{K\text{-means}}$ to the data projected on $(f_1^{(i)}, \ldots, f_b^{(i)})$ to get $\kappa(f_1^{(i)}, \ldots, f_b^{(i)})$



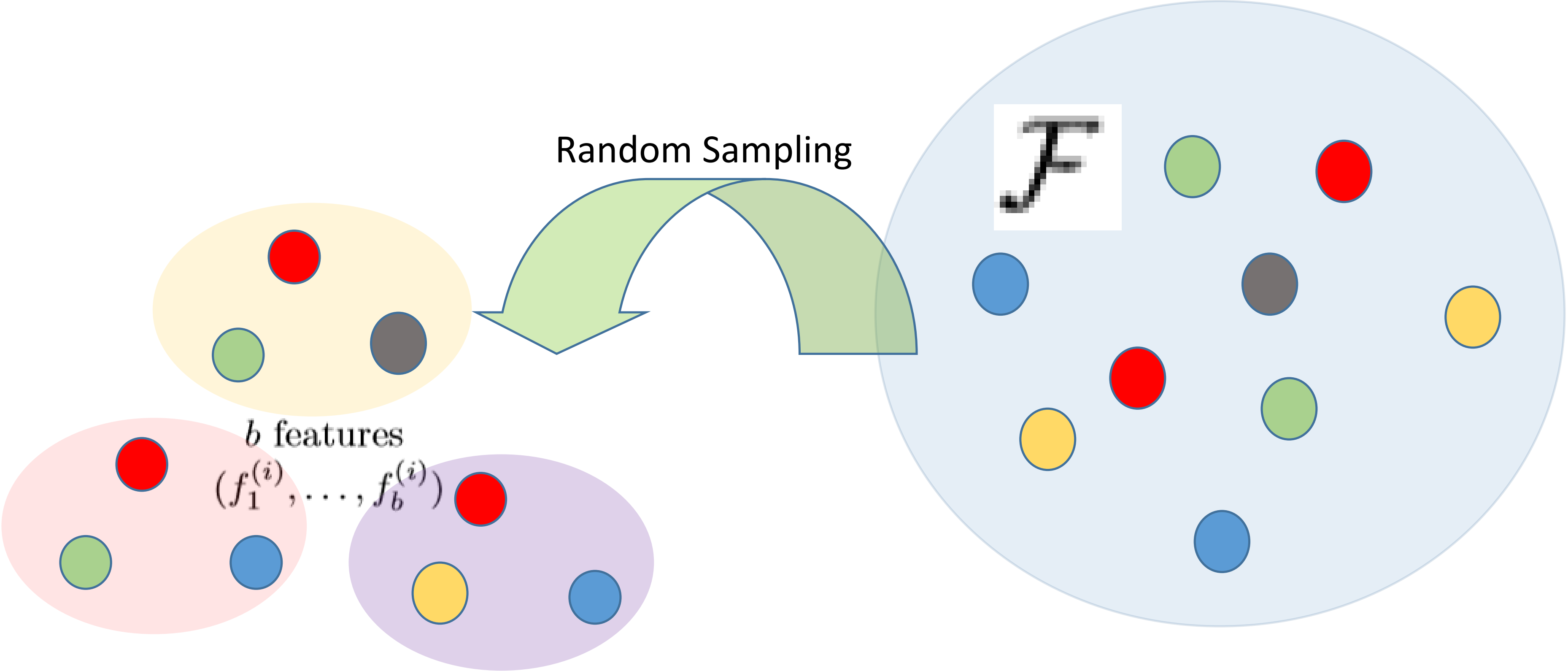
Method - Feature competition

$$\kappa(\tilde{\boldsymbol{f}}) = \frac{SS_W(\tilde{\boldsymbol{f}})}{SS_B(\tilde{\boldsymbol{f}})}$$

q times

1: Sample b features, $f_1^{(i)}, \dots, f_b^{(i)}$, from the feature space $\mathcal{F}$

2: Apply $\boxed{K\text{-means}}$ to the data projected on $(f_1^{(i)}, \dots, f_b^{(i)})$ to get $\kappa(f_1^{(i)}, \dots, f_b^{(i)})$



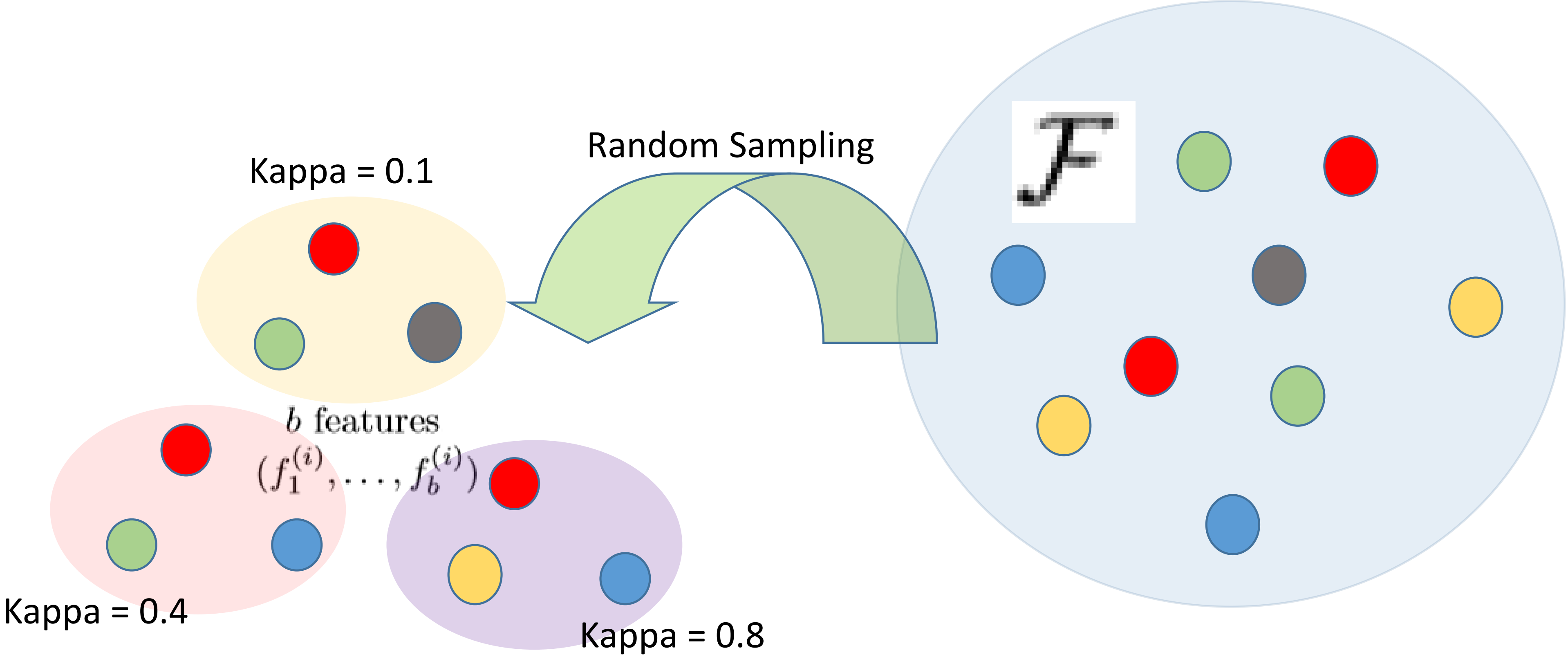
Method - Feature competition

$$\kappa(\tilde{\boldsymbol{f}}) = \frac{SS_W(\tilde{\boldsymbol{f}})}{SS_B(\tilde{\boldsymbol{f}})}$$

q times

1: Sample b features, $f_1^{(i)}, \ldots, f_b^{(i)}$, from the feature space $\mathcal{F}$

2: Apply $\boxed{K\text{-means}}$ to the data projected on $(f_1^{(i)}, \ldots, f_b^{(i)})$ to get $\kappa(f_1^{(i)}, \ldots, f_b^{(i)})$



Method - Feature competition

$$\kappa(\tilde{\boldsymbol{f}}) = \frac{SS_W(\tilde{\boldsymbol{f}})}{SS_B(\tilde{\boldsymbol{f}})}$$

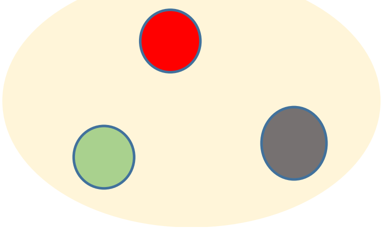
q times

1: Sample b features, $f_1^{(i)}, \ldots, f_b^{(i)}$, from the feature space $\mathcal{F}$

2: Apply $\boxed{K\text{-means}}$ to the data projected on $(f_1^{(i)}, \ldots, f_b^{(i)})$ to get $\kappa(f_1^{(i)}, \ldots, f_b^{(i)})$

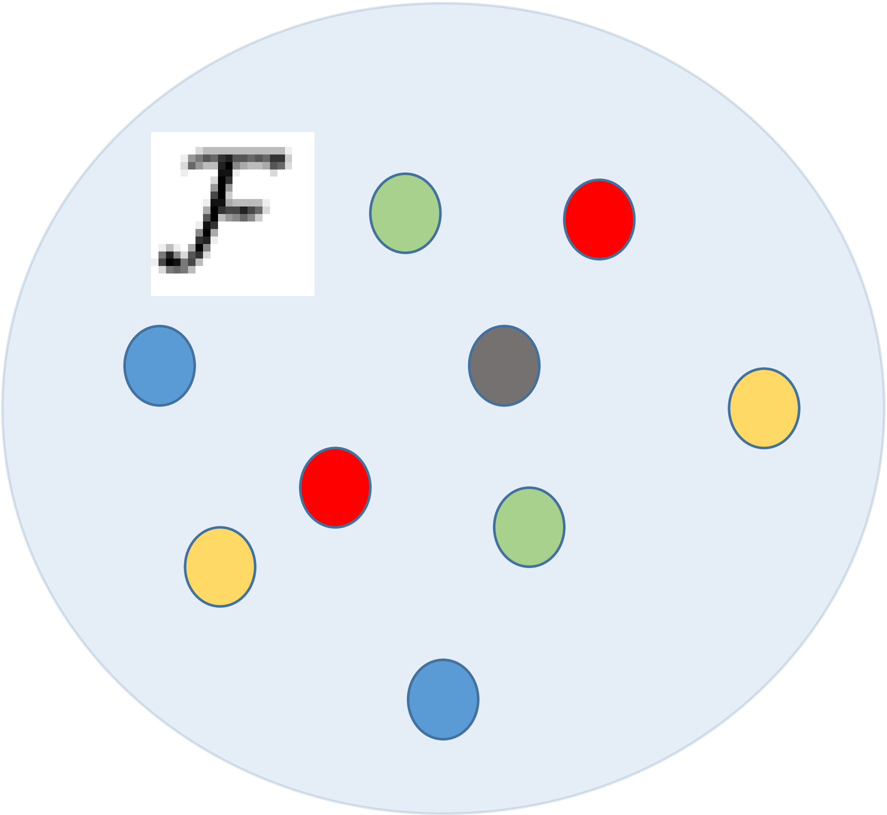
3: Set $(f_1^{(0)}, \ldots, f_b^{(0)}) \leftarrow \arg\min_{i=1}^{q} \kappa(f_1^{(i)}, \ldots, f_b^{(i)})$

Kappa = 0.1



b features

$(f_1^{(i)}, \ldots, f_b^{(i)})$



Method - Feature competition

$$\kappa(\tilde{\boldsymbol{f}}) = \frac{SS_W(\tilde{\boldsymbol{f}})}{SS_B(\tilde{\boldsymbol{f}})}$$

q times

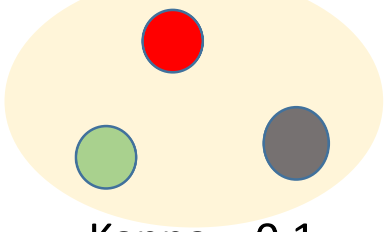
1: Sample b features, $f_1^{(i)}, \ldots, f_b^{(i)}$, from the feature space $\mathcal{F}$

2: Apply $\boxed{K\text{-means}}$ to the data projected on $(f_1^{(i)}, \ldots, f_b^{(i)})$ to get $\kappa(f_1^{(i)}, \ldots, f_b^{(i)})$

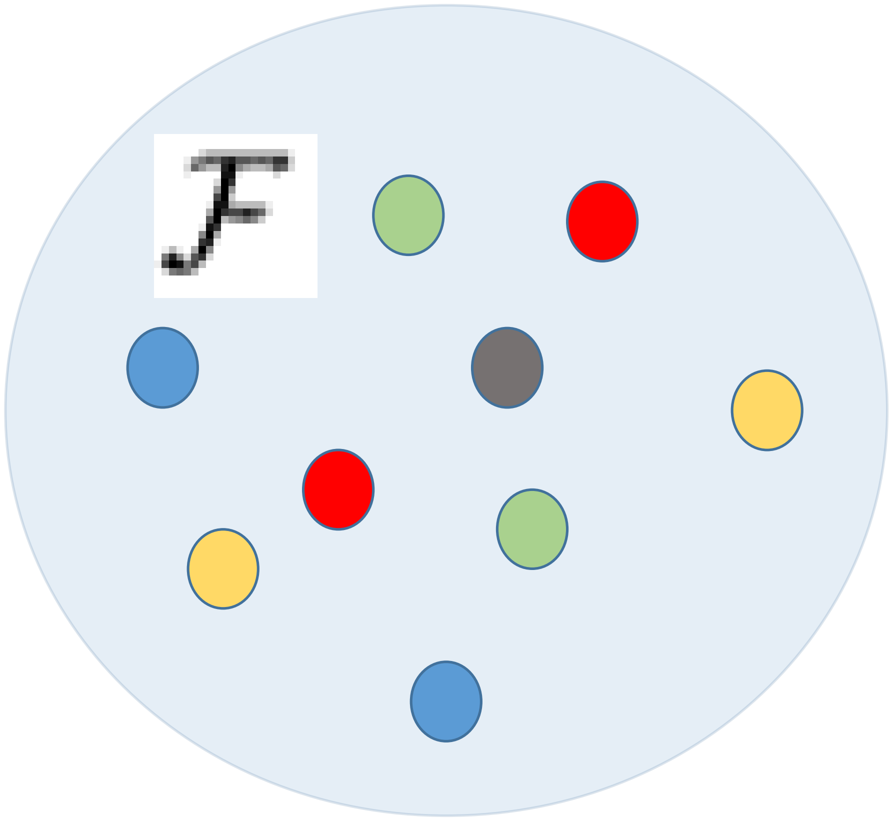
3: Set $(f_1^{(0)}, \ldots, f_b^{(0)}) \leftarrow \arg\min_{i=1}^{q} \kappa(f_1^{(i)}, \ldots, f_b^{(i)})$

setting q = 1 reduces to the usual mode

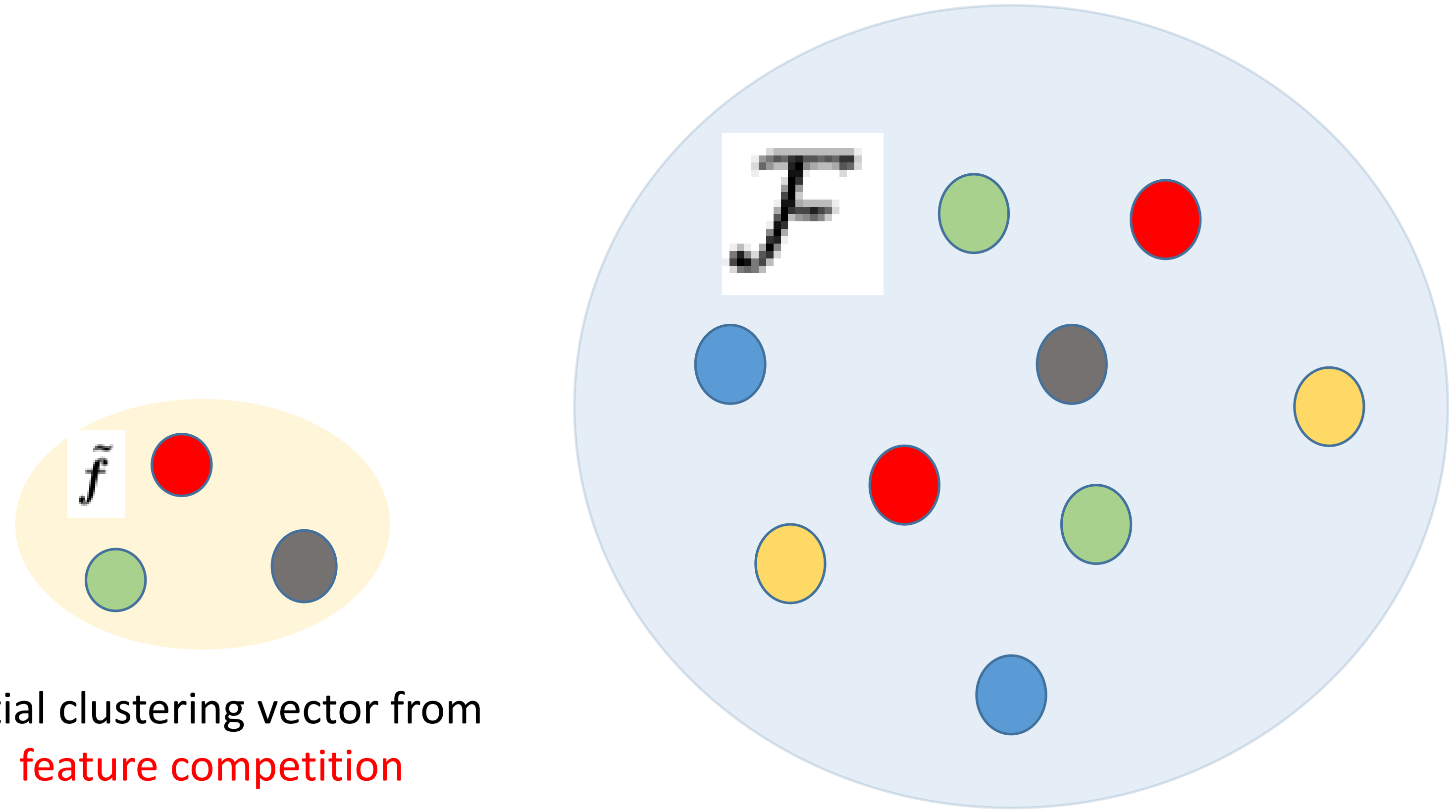
$\tilde{\boldsymbol{f}} \leftarrow (f_1^{(0)}, \ldots, f_b^{(0)})$



Kappa = 0.1



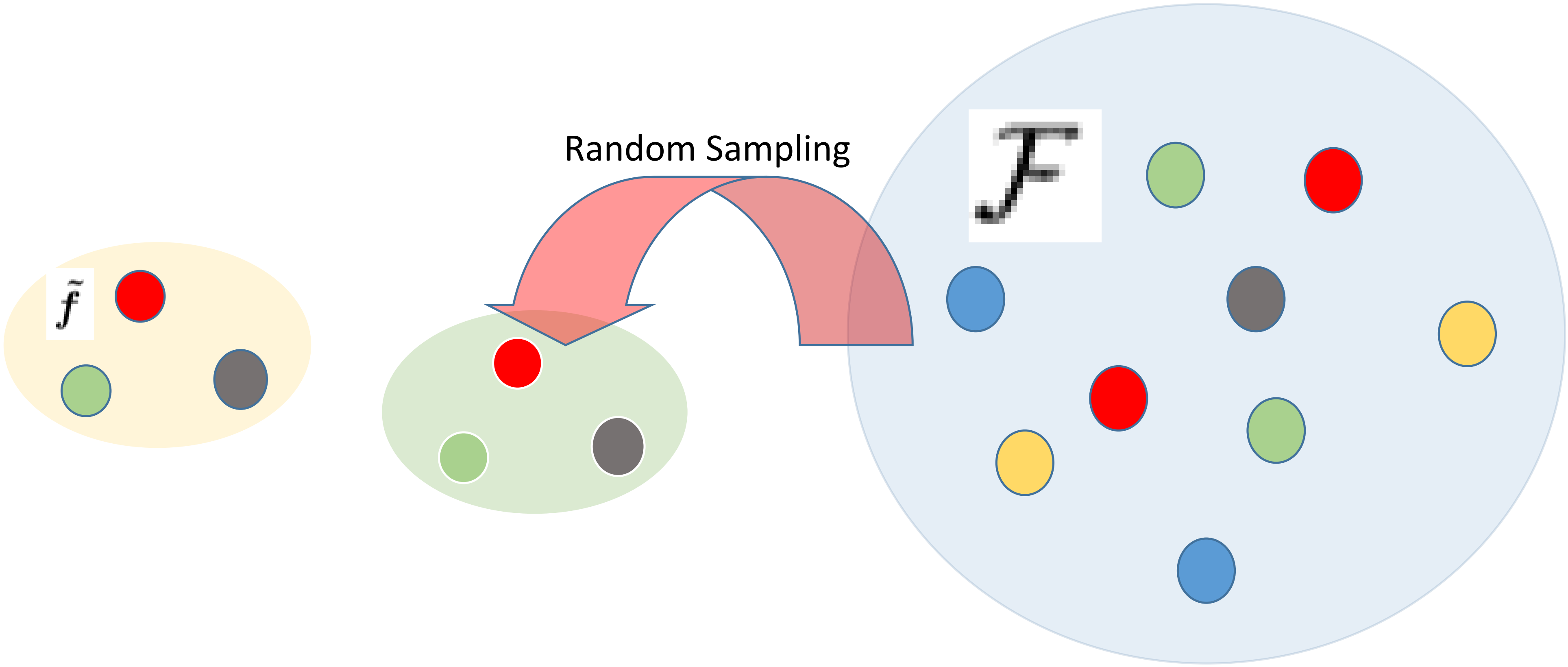
Method - Growth of clustering vectors



initial clustering vector from
feature competition

Method - Growth of clustering vectors

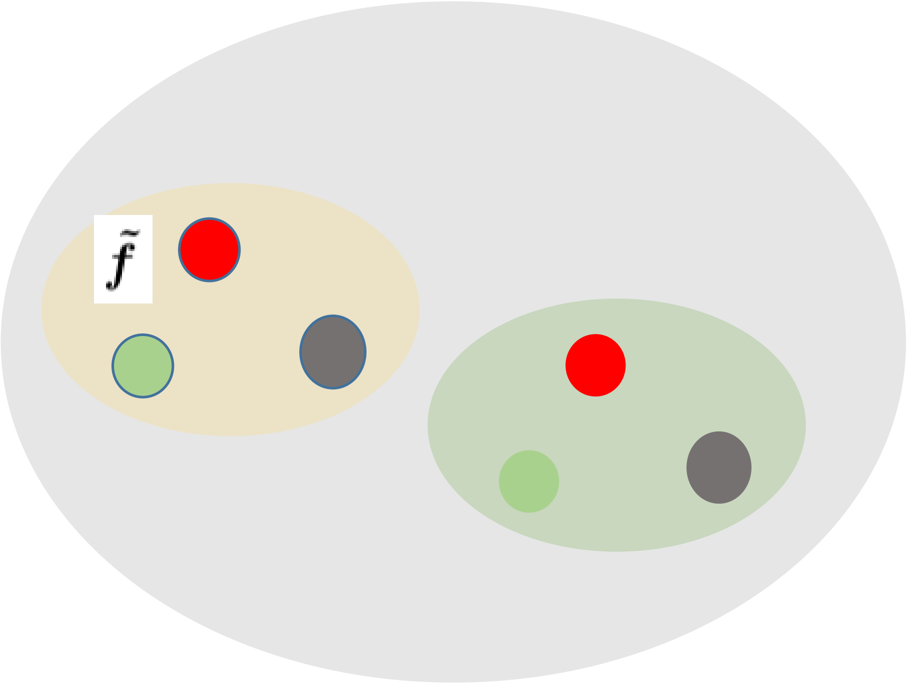
1: Sample b features, denoted as $f_1, \ldots, f_b$, from the feature space $\mathcal{F}$

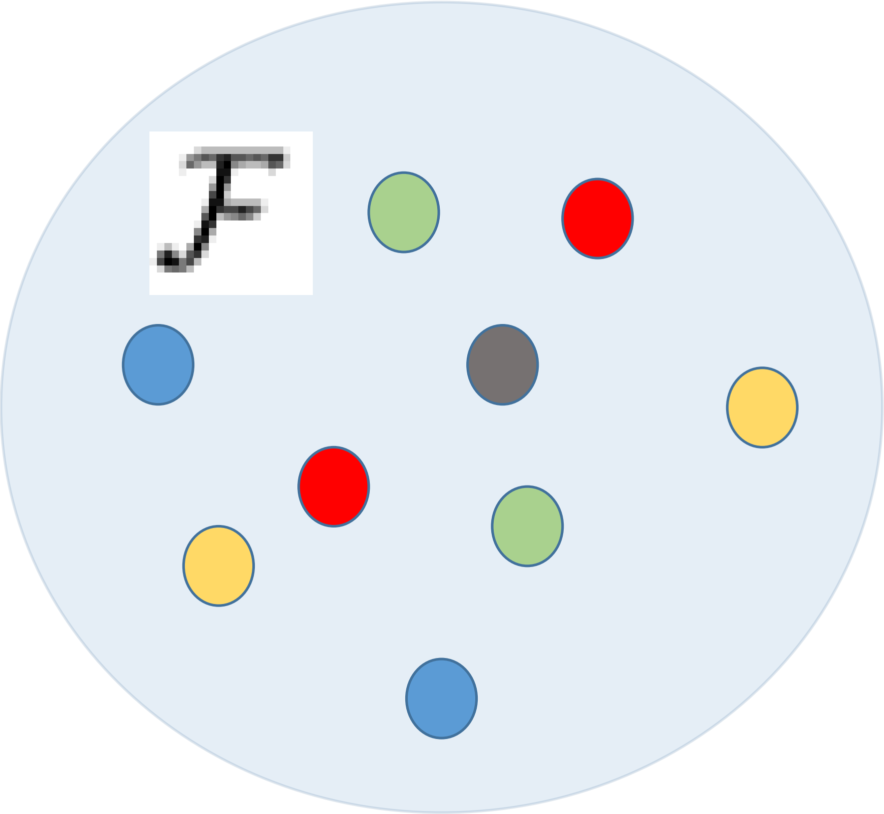


Method - Growth of clustering vectors

1: Sample b features, denoted as $f_1, \ldots, f_b$, from the feature space $\mathcal{F}$

2: Apply $\boxed{K\text{-means}}$ (the *base clustering algorithm*) to the data induced by the feature vector $(\tilde{f}, f_1, \ldots, f_b)$



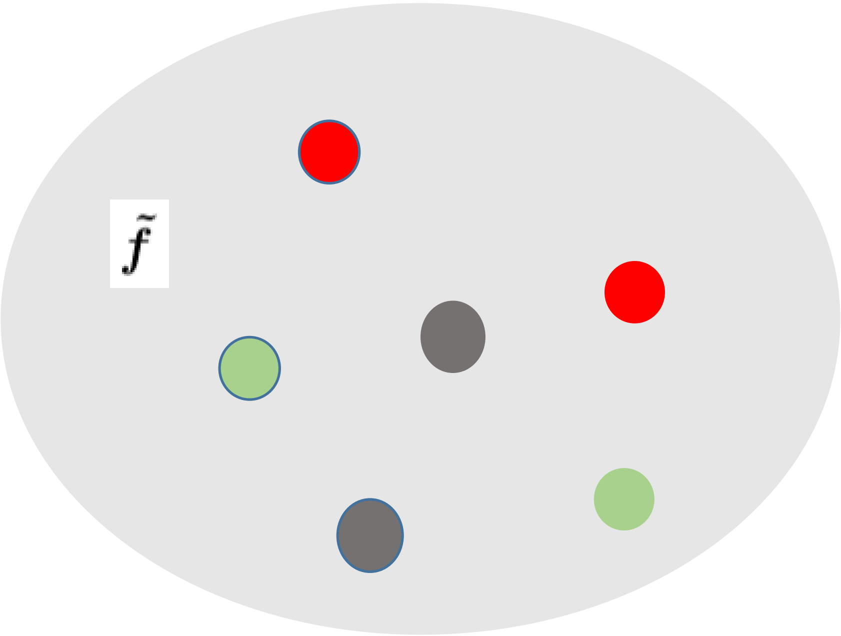


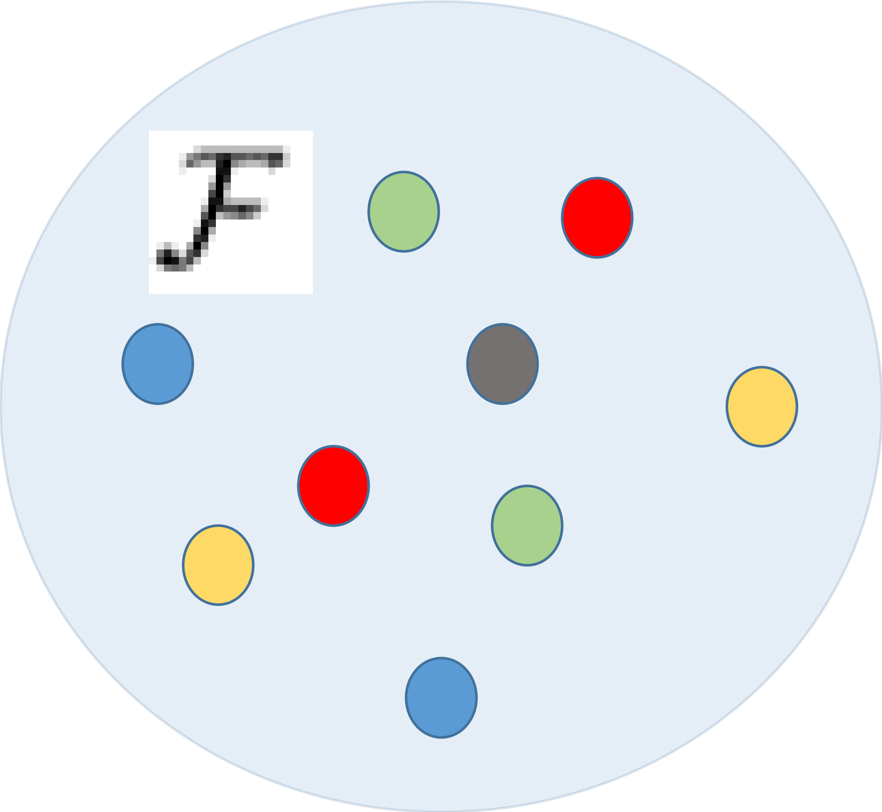
Method - Growth of clustering vectors

1: Sample b features, denoted as $f_1, \ldots, f_b$, from the feature space $\mathcal{F}$

2: Apply K-means (the *base clustering algorithm*) to the data induced by the feature vector $(\tilde{\boldsymbol{f}}, f_1, \ldots, f_b)$

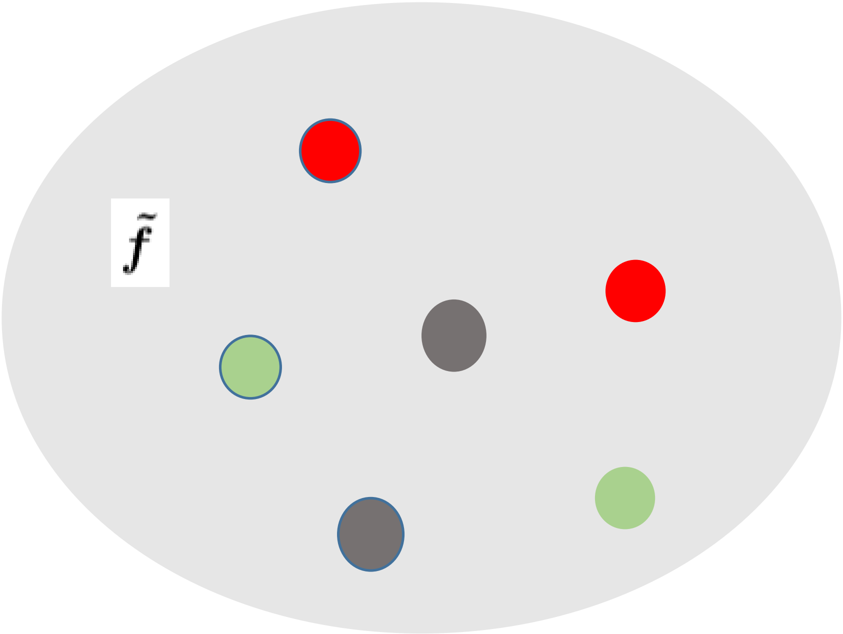
3: **if** $\kappa(\tilde{\tilde{\boldsymbol{f}}}, f_1 \ldots, f_b) < \kappa(\tilde{\tilde{\boldsymbol{f}}})$ **then**
expand $\tilde{\boldsymbol{f}}$ by $\tilde{\boldsymbol{f}} \leftarrow (\tilde{\boldsymbol{f}}, f_1, \ldots, f_b)$ and set $\tau \leftarrow 0$





Method - The CF algorithm

1: Grow a clustering vector, $\tilde{f}^{(l)}$, according to Algorithm 2.

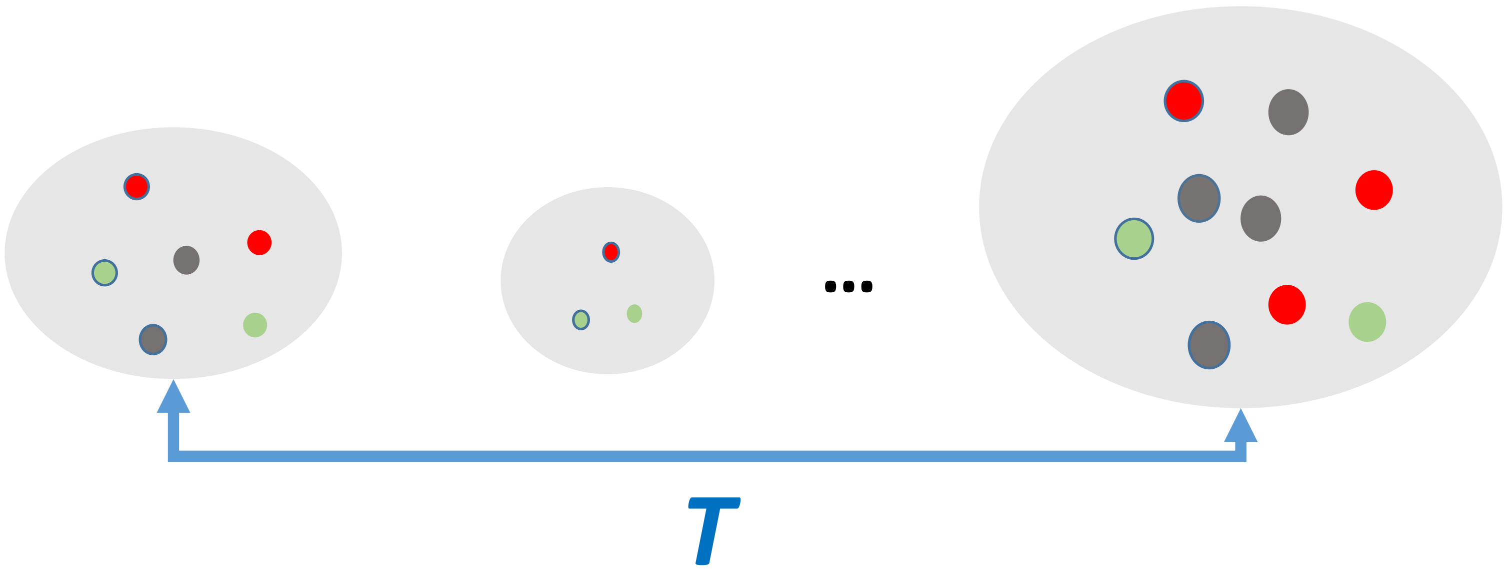


Method - The CF algorithm

***T* times**

1: Grow a clustering vector, $\tilde{\boldsymbol{f}}^{(l)}$, according to Algorithm 2.

2: Apply the base clustering algorithm to the data induced by clustering vector $\tilde{\boldsymbol{f}}^{(l)}$ to get a partition of the data



Method - The CF algorithm

1: Grow a clustering vector, $\tilde{\boldsymbol{f}}^{(l)}$, according to Algorithm 2.

2: Apply the base clustering algorithm to the data induced by clustering vector $\tilde{\boldsymbol{f}}^{(l)}$ to get a partition of the data

3: Construct $n \times n$ co-cluster indicator matrix (or affinity matrix) $P^{(l)}$

$$P_{ij}^{(l)} = \begin{cases} 1, & \text{if } X_i \text{ and } X_j \text{ are in the same cluster} \\ 0, & \text{otherwise} \end{cases}$$

	X1	X2	X3	X4	X5
X1	1	1	0	1	0
X2	1	1	0	1	1
X3	0	0	1	0	1
X4	1	1	0	1	0
X5	0	1	1	0	1

Method - The CF algorithm

1: Grow a clustering vector, $\tilde{\boldsymbol{f}}^{(l)}$, according to Algorithm 2.

2: Apply the base clustering algorithm to the data induced by clustering vector $\tilde{\boldsymbol{f}}^{(l)}$ to get a partition of the data

3: Construct $n \times n$ co-cluster indicator matrix (or affinity matrix) $P^{(l)}$

$$P_{ij}^{(l)} = \begin{cases} 1, & \text{if } X_i \text{ and } X_j \text{ are in the same cluster} \\ 0, & \text{otherwise} \end{cases}$$

4: Average the indicator matrices to get $P \leftarrow \frac{1}{T} \sum_{l=1}^{T} P^{(l)}$

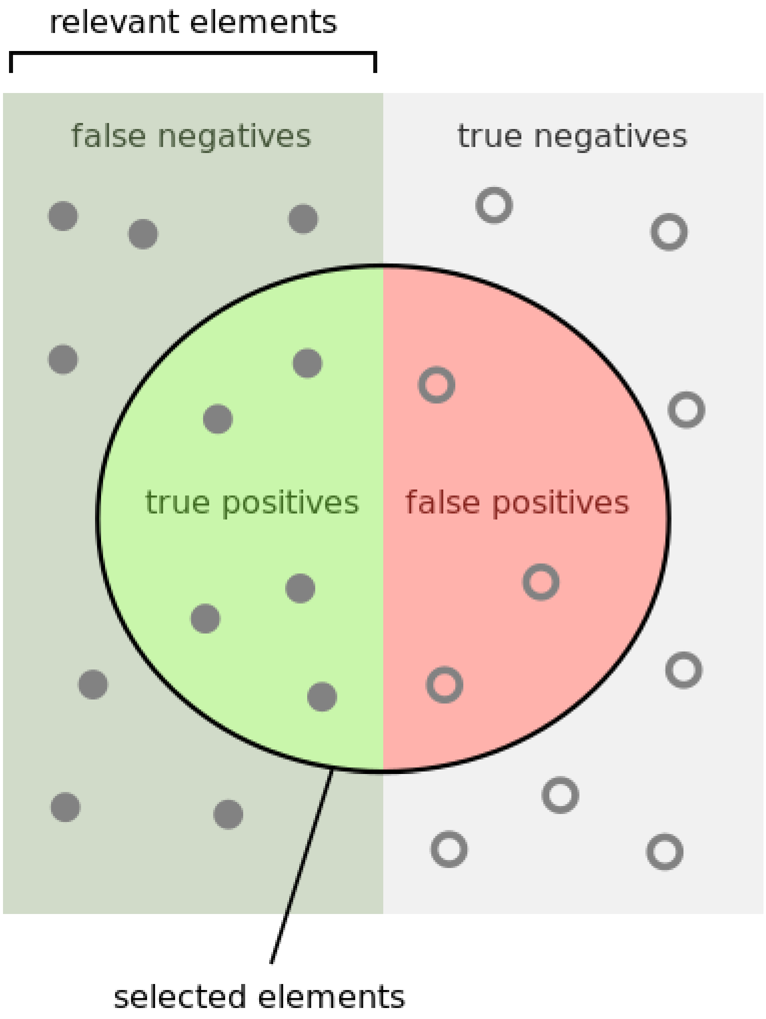
➡ Apply spectral clustering to P to get the final clustering

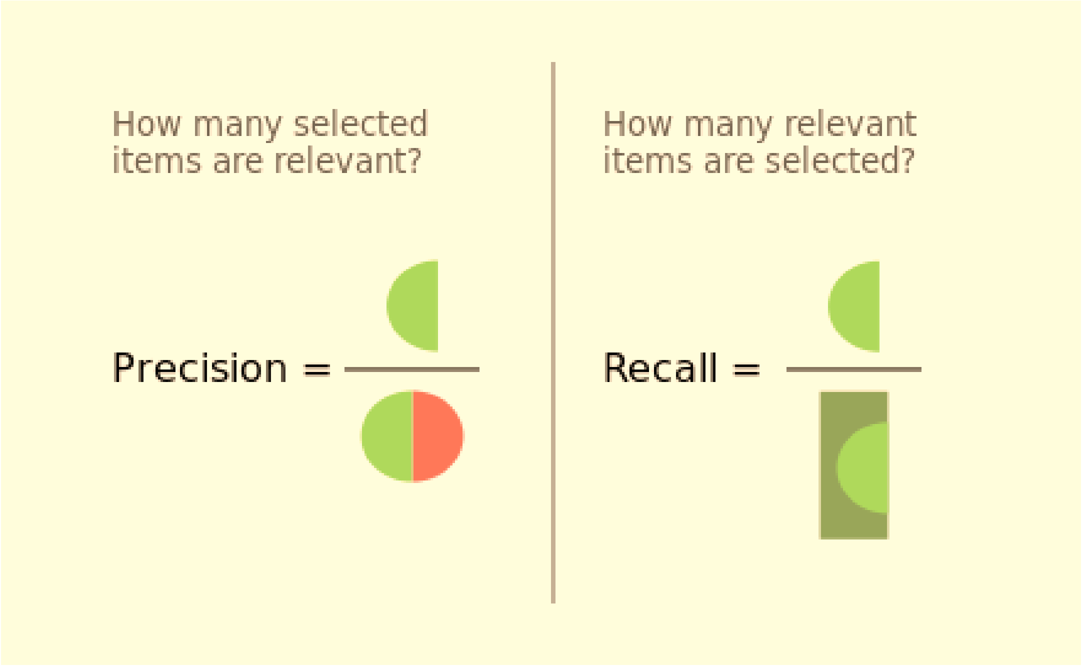
Experiments Result – Summary of UC Irvine datasets

Dataset	Features	Classes	#Instances
Soybean	35	4	47
SPECT	22	2	267
ImgSeg	19	7	2100
Heart	13	2	270
Wine	13	3	178
WDBC	30	2	569
Robot	90	5	164
Madelon	500	2	2000

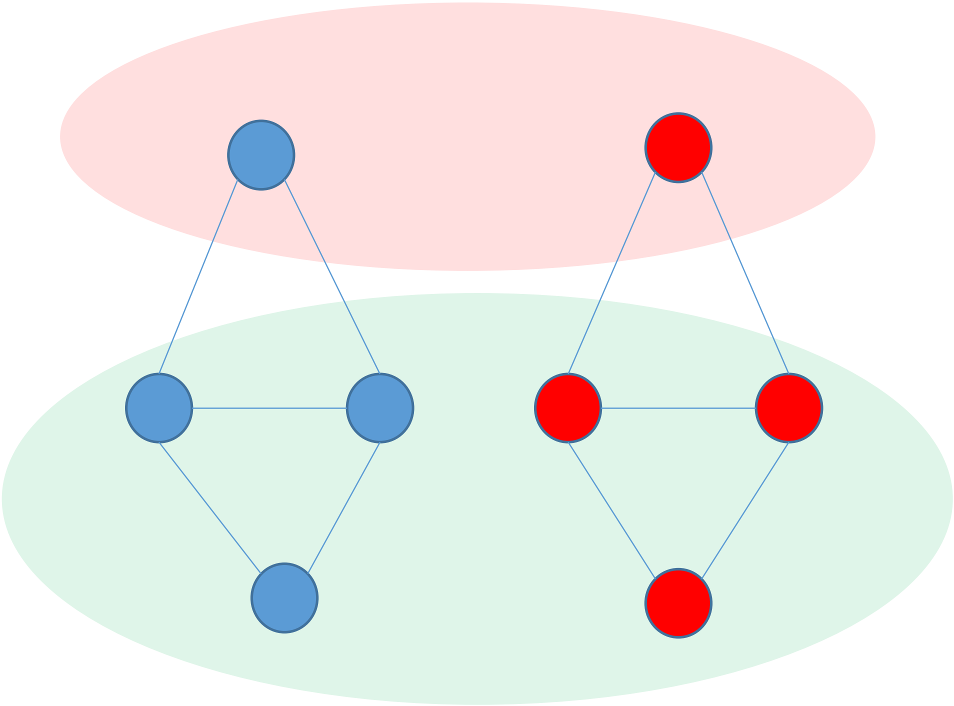
Table 1: A summary of datasets.

Evaluation of the clustering result – Precision and Recall





Evaluation of the clustering result – ρr and ρc



ρr = 6 / 10 (recall)

ρc = 6 / 16 (precision)

Experiments Result - on UC Irvine datasets

Dataset	CF	RP	bC2	EA
Soybean	**92.36**	87.04	83.16	86.48
SPECT	**56.78**	49.89	50.61	51.04
ImgSeg	79.71	**85.88**	82.19	85.75
Heart	**56.90**	52.41	51.50	53.20
Wine	**79.70**	71.94	71.97	71.86
WDBC	**79.66**	74.89	74.87	75.04
Robot	**63.42**	41.52	39.76	58.31
Madelon	50.76	**50.82**	49.98	49.98

Table 2: $\boxed{\rho_r}$ for different datasets and methods (CF calculated when $q = 1$)

(bC2) bagged clustering
(RP) random projection
(EA) evidence accumulation
(NJW) NJW spectral clustering algorithm
K-means-1 (itr = 200)
K-means-2 (itr = 1000)

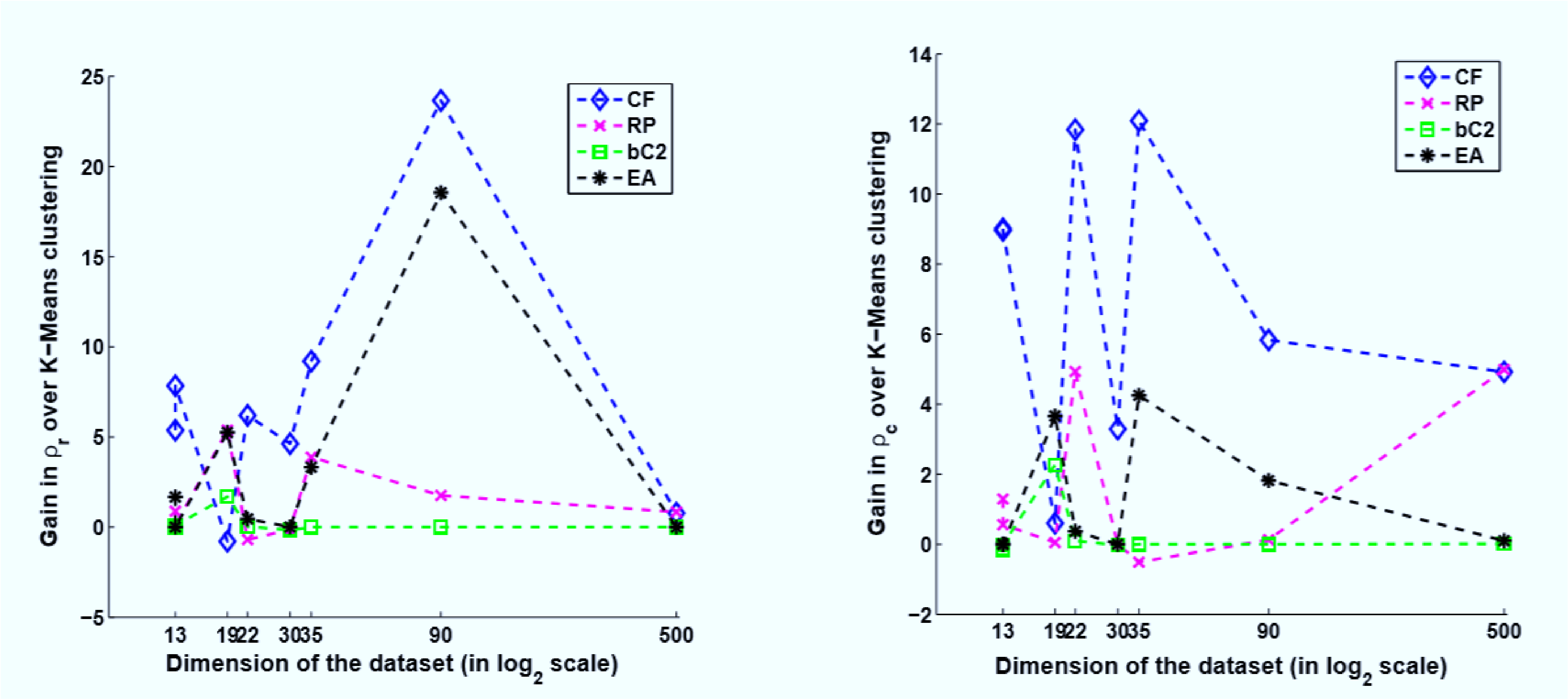
Experiments Result - on UC Irvine datasets

Dataset	CF	RP	bC2	EA
Soybean	**84.43**	71.83	72.34	76.59
SPECT	**68.02**	61.11	56.28	56.55
ImgSeg	48.24	47.71	49.91	**51.30**
Heart	**68.26**	60.54	59.10	59.26
Wine	**79.19**	70.79	70.22	70.22
WDBC	**88.70**	85.41	85.38	85.41
Robot	**41.20**	35.50	35.37	37.19
Madelon	55.12	**55.19**	50.20	50.30

Table 3: $\boxed{\rho_c}$ for different datasets and methods (CF calculated when $q = 1$).

(bC2) bagged clustering
(RP) random projection
(EA) evidence accumulation
(NJW) NJW spectral clustering algorithm
K-means-1 (itr = 200)
K-means-2 (itr = 1000)

Experiments Result - on UC Irvine datasets



Experiments Result – Comparison between CF, NJW, and K-means

Dataset	CF	NJW	K-means-1	K-means-2
Soybean	**92.36**	83.72	83.16	83.16
	84.43	76.60	72.34	72.34
SPECT	**56.78**	53.77	50.58	50.58
	68.02	64.04	56.18	56.18
ImgSeg	79.71	**82.48**	81.04	80.97
	48.24	**53.38**	48.06	47.21
Heart	**56.90**	51.82	51.53	51.53
	68.26	60.00	59.25	59.25
Wine	**79.70**	71.91	71.86	71.86
	79.19	70.78	70.23	70.22
WDBC	79.93	**81.10**	75.03	75.03
	88.70	**89.45**	85.41	85.41
Robot	63.60	**69.70**	39.76	39.76
	41.20	**42.68**	35.37	35.37
Madelon	**50.76**	49.98	49.98	49.98
	55.12	50.55	50.20	50.20

(bC2) bagged clustering
(RP) random projection
(EA) evidence accumulation
(NJW) NJW spectral clustering algorithm
K-means-1 (itr = 200)
K-means-2 (itr = 1000)

Conclusion

- This paper provides a clustering alg. framework, the base clustering alg. are pluggable.
- *CF favors strong features and is noise-resistant*
- Strong feature generates smaller tree, a set of weak features may generate larger tree.

-> Not only consider strong features.