# Module Master Recherche
# Apprentissage et Fouille

Michele Sebag – Balazs Kegl – Antoine Cornuéjols
http://tao.lri.fr

19 novembre 2008

# Unsupervised Learning

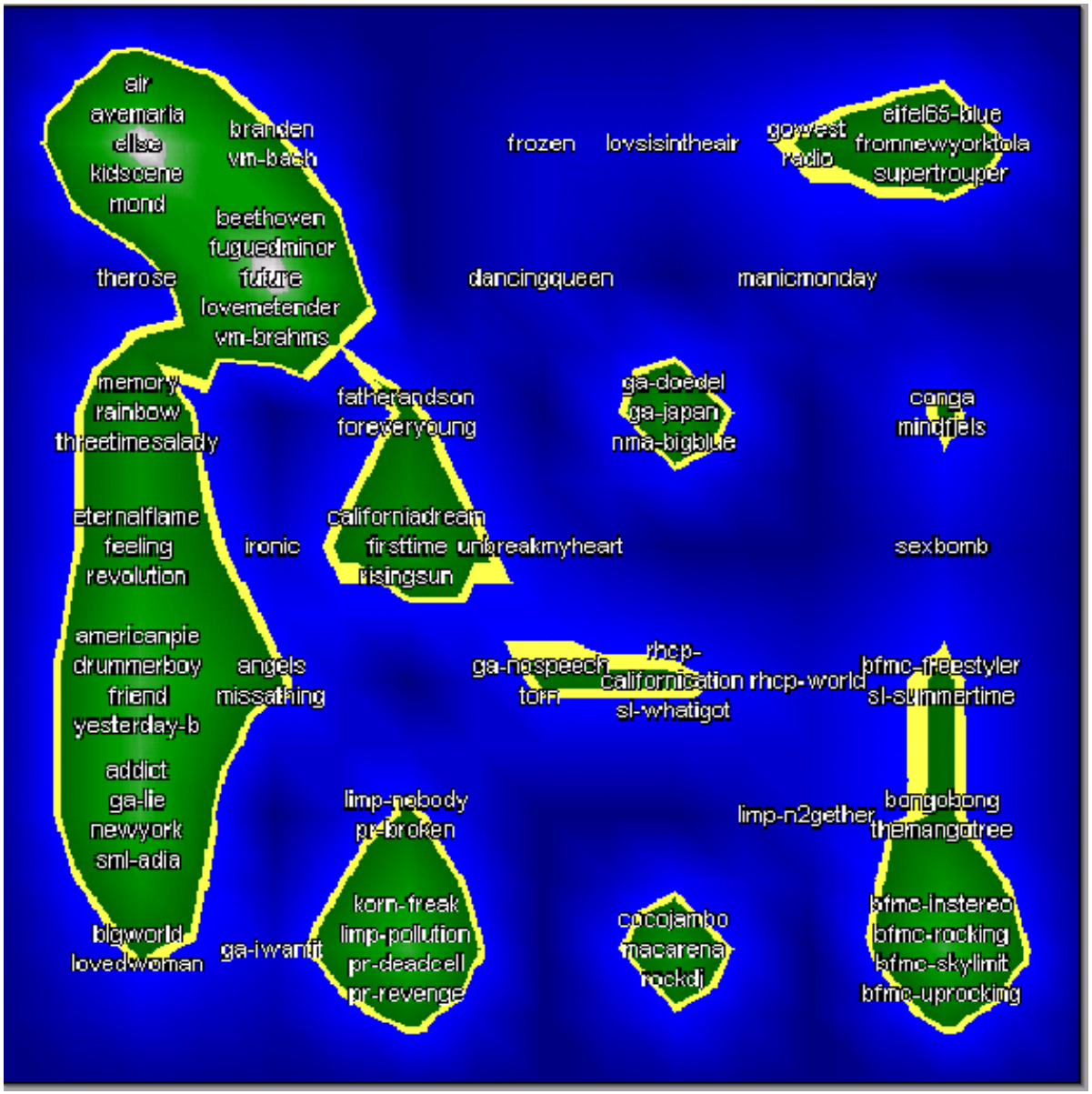
- Clustering
- Data Streaming
- Application: Clustering of EGEE Jobs, Intrusion Detection

# Clustering

- K-Means
- Expectation Maximization
- Selecting the number of clusters
- Affinity propagation
- Scalability

# Clustering

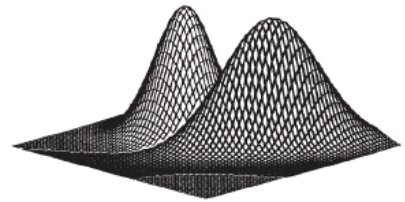
http://www.ofai.at/ elias.pampalk/music/



# Clustering Questions

## Hard or soft ?

- Hard: find a partition of the data
- Soft: estimate the distribution of the data as a mixture of components.



## Parametric *vs* non Parametric ?

- Parametric: number $K$ of clusters is known
- Non-Parametric: find $K$
  (wrapping a parametric clustering algorithm)

## Caveat:

- Complexity
- Outliers
- Validation

# Formal Background

## Notations

| | | |
|---|---|---|
| $\mathcal{E}$ | $\{\mathbf{x}_1, \ldots \mathbf{x}_N\}$ dataset | |
| $N$ | number of data points | |
| $K$ | number of clusters | given or optimized |
| $C_k$ | $k$-th cluster | Hard clustering |
| $\tau(i)$ | index of cluster containing $\mathbf{x}_i$ | |
| $f_k$ | $k$-th model | Soft clustering |
| $\gamma_k(i)$ | $Pr(\mathbf{x}_i \mid f_k)$ | |

## Solution

| | |
|---|---|
| Hard Clustering | Partition $\Delta = (C_1, \ldots C_k)$ |
| Soft Clustering | $\forall i \ \sum_k \gamma_k(i) = 1$ |

# Formal Background, 2

## Quality / Cost function

Measures how well the clusters characterize the data

- (log)likelihood — soft clustering
- dispersion — hard clustering

$$\sum_{k=1}^{K} \frac{1}{|C_k|^2} \sum_{\mathbf{x}_i, \mathbf{x}_j \ in \ C_k} d(\mathbf{x}_i, \mathbf{x}_j)^2$$

## Tradeoff

Quality increases with $K \Rightarrow$ Regularization needed

to avoid one cluster per data point

# Clustering vs Classification

Marina Meila
http://videolectures.net/

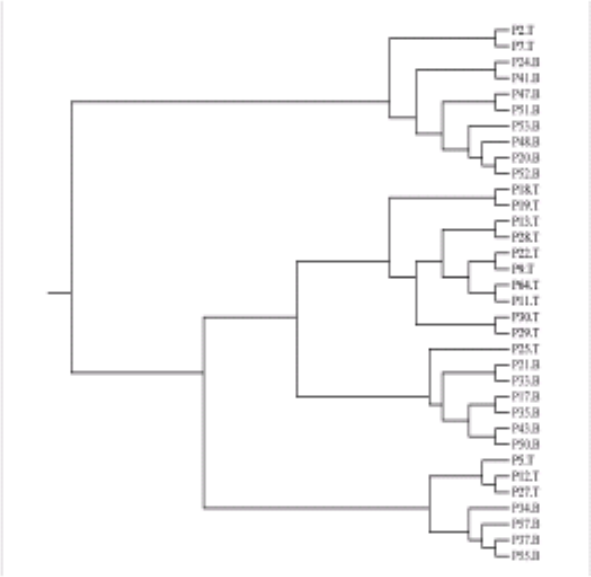
| | Classification | Clustering |
|---|---|---|
| $K$ | # classes (given) | # clusters (unknown) |
| Quality | Generalization error | many cost functions |
| Focus on | Test set | Training set |
| Goal | Prediction | Interpretation |
| Analysis | discriminant | exploratory |
| Field | mature | new |

# Non-Parametric Clustering

## Hierarchical Clustering

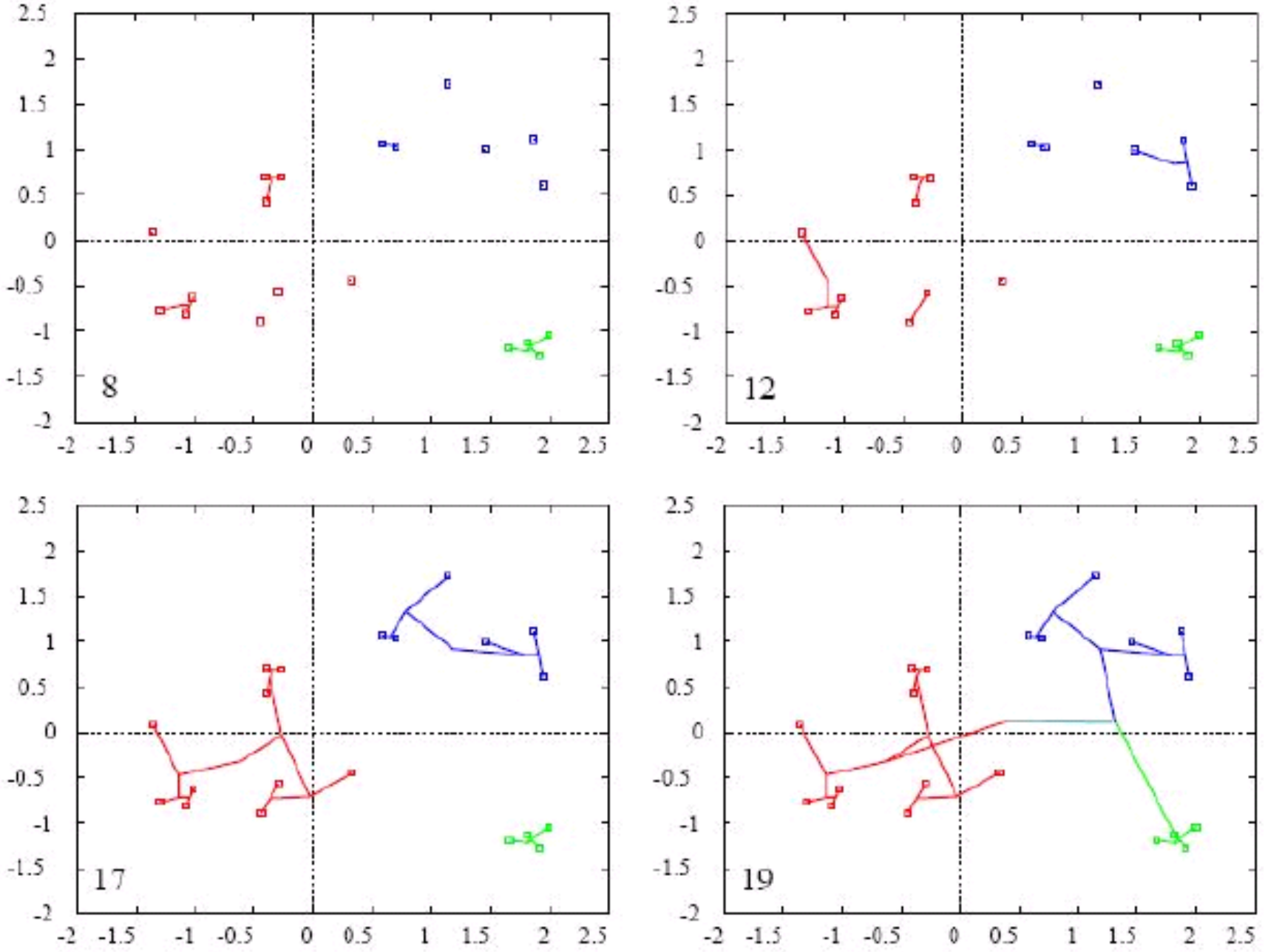
### Principle

- agglomerative (join nearest clusters)
- divisive (split most dispersed cluster)

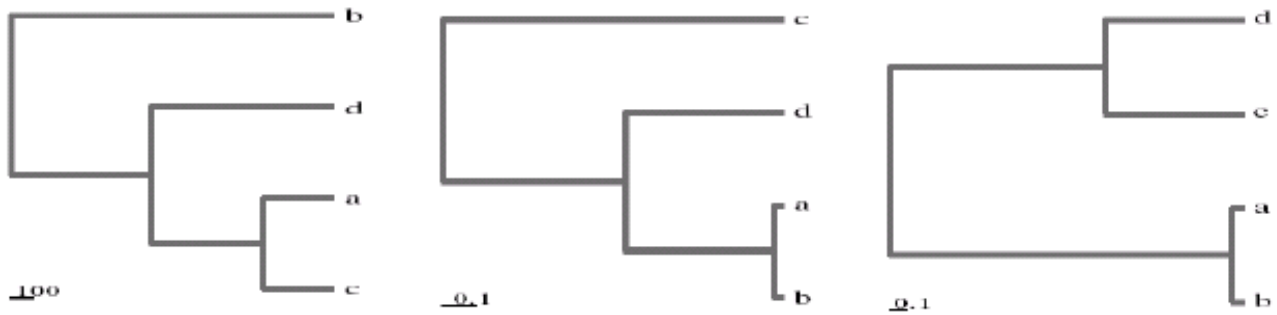


CONS: Complexity $\mathcal{O}(N^3)$

# Hierarchical Clustering, example



# Influence of distance/similarity




Euclidean Vector angle Pearson

$$d(x, x') = \begin{cases} \sqrt{\sum_i (x_i - x'_i)^2} & \text{Euclidean distance} \\ 1 - \frac{\sum_i x_i x'_i}{||x||.||x'||} & \text{Cosine angle} \\ 1 - \frac{\sum_i (x_i - \bar{x})(x'_i - \bar{x}')}{||x - \bar{x}||.||x' - \bar{x}'||} & \text{Pearson} \end{cases}$$

# Parametric Clustering

$K$ is known

## Algorithms based on distances

- $K$-means
- graph / cut

## Algorithms based on models

- Mixture of models: EM algorithm

# Clustering

- K-Means
- Expectation Maximization
- Selecting the number of clusters
- Affinity propagation
- Scalability

# $K$-Means

## Algorithm

1. Init:
   Uniformly draw $K$ points $\mathbf{x}_{i_j}$ in $\mathcal{E}$
   Set $C_j = \{\mathbf{x}_{i_j}\}$
2. Repeat
3. Draw without replacement $\mathbf{x}_i$ from $\mathcal{E}$
4. $\tau(i) = argmin_{k=1\ldots K}\{d(\mathbf{x}_i, C_k)\}$ find best cluster for $\mathbf{x}_i$
5. $C_{\tau(i)} = C_{\tau(i)} \bigcup \mathbf{x}_i$ add $\mathbf{x}_i$ to $C_{\tau(i)}$
6. Until all points have been drawn
7. If partition $C_1 \ldots C_K$ has changed Stabilize
   Define $\mathbf{x}_{i_k} =$ best point in $C_k$, $C_k = \{x_{i_k}\}$, goto 2.

Algorithm terminates

# $K$-Means, Knobs

## Knob 1 : define $d(\mathbf{x}_i, C_k)$ — favors

- $min\{d(\mathbf{x}_i, \mathbf{x}_j), \mathbf{x}_j \in C_k\}$ — long clusters
- $*$ $average\{d(\mathbf{x}_i, \mathbf{x}_j), \mathbf{x}_j \in C_k\}$ — compact clusters
- $max\{d(\mathbf{x}_i, \mathbf{x}_j), \mathbf{x}_j \in C_k\}$ — spheric clusters

## Knob 2 : define "best" in $C_k$

- Medoid — $argmin_i\{\sum_{\mathbf{x}_j \in C_k} d(\mathbf{x}_i, \mathbf{x}_j)\}$
- $*$ Average (does not belong to $\mathcal{E}$) — $\frac{1}{|C_k|} \sum_{\mathbf{x}_j \in C_k} \mathbf{x}_j$

# No single best choice

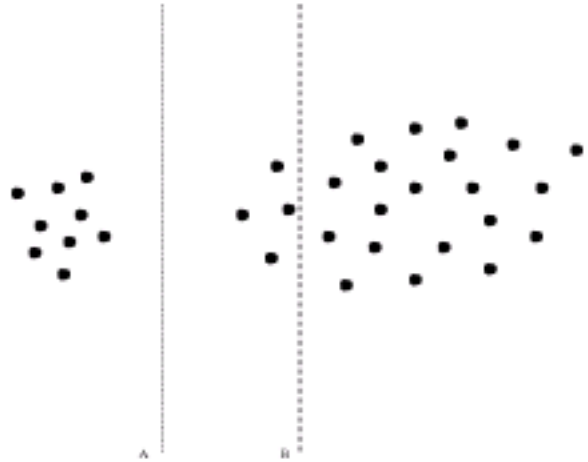



FIG. 1. Optimizing the diameter produces B while A is clearly more desirable.

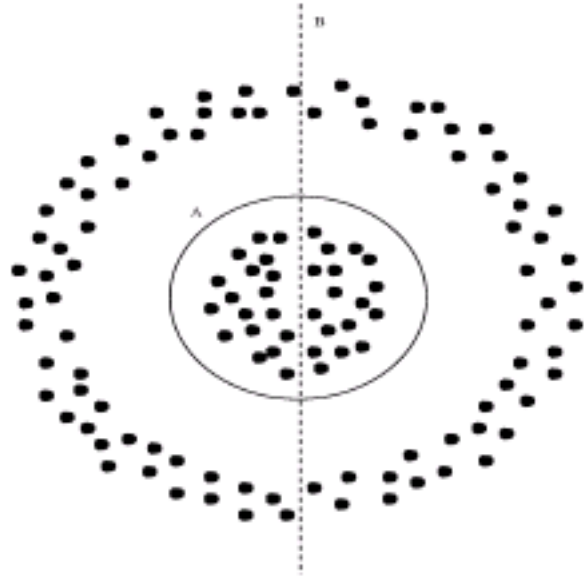



FIG. 2. The inferior clustering B is found by optimizing the 2-median measure.

# $K$-Means, Discussion

## PROS

- Complexity $\mathcal{O}(K \times N)$
- Can incorporate prior knowledge initialization

## CONS

- Sensitive to initialization
- Sensitive to outliers
- Sensitive to irrelevant attributes

# $K$-Means, Convergence

- For cost function

$$\mathcal{L}(\Delta) = \sum_{k} \sum_{i,j \ / \ \tau(i)=\tau(j)=k} d(\mathbf{x}_i, \mathbf{x}_j)$$

- for $d(\mathbf{x}_i, C_k) =$ average $\{d(\mathbf{x}_i, \mathbf{x}_j), \mathbf{x}_j \in C_k\}$
- for "best" in $C_k$ = average of $\mathbf{x}_j \in C_k$

$K$-means converges toward a (local) minimum of $\mathcal{L}$.

# $K$-Means, Practicalities

## Initialization

- Uniform sampling
- Average of $\mathcal{E}$ + random perturbations
- Average of $\mathcal{E}$ + orthogonal perturbations
- Extreme points: select $\mathbf{x}_{i_1}$ uniformly in $\mathcal{E}$, then

$$\text{Select } x_{i_j} = argmax\{\sum_{k=1}^{j} d(\mathbf{x}_i, x_{i_k})\}$$

## Pre-processing

- Mean-centering the dataset

# Clustering

- K-Means
- Expectation Maximization
- Selecting the number of clusters
- Affinity propagation
- Scalability

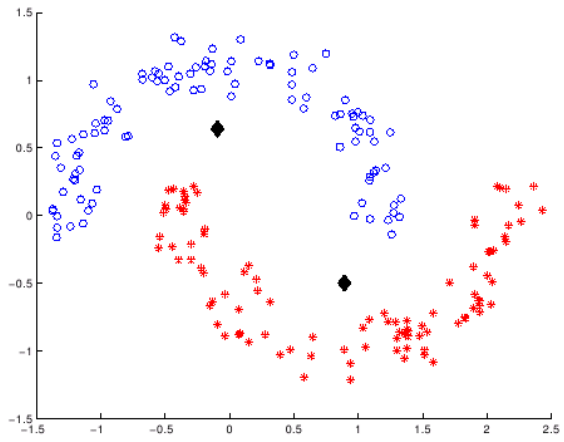
# Model-based clustering

## Mixture of components

- Density $f = \sum_{k=1}^{K} \pi_k f_k$
- $f_k$: the $k$-th component of the mixture
- $\gamma_k(i) = \frac{\pi_k f_k(x)}{f(x)}$
- induces $C_k = \{\mathbf{x}_j \ / \ k = argmax\{\gamma_k(j)\}\}$

## Nature of components: prior knowledge

- Most often Gaussian: $f_k = (\mu_k, \Sigma_k)$
- Beware: clusters are not always Gaussian...



# Model-based clustering, 2

## Search space

- Solution : $(\pi_k, \mu_k, \Sigma_k)_{k=1}^K = \theta$

## Criterion: log-likelihood of dataset

$$\ell(\theta) = \log(Pr(\mathcal{E})) = \sum_{i=1}^{N} \log Pr(\mathbf{x}_i) \propto \sum_{i=1}^{N} \sum_{k=1}^{K} \log(\pi_k f_k(\mathbf{x}_i))$$

to be maximized.

# Model-based clustering with EM

## Formalization

- Define $z_{i,k} = 1$ iff $\mathbf{x}_i$ belongs to $C_k$.
- $E[z_{i,k}] = \gamma_k(i)$ prob. $\mathbf{x}_i$ generated by $\pi_k f_k$
- Expectation of log likelihood

$$E[\ell(\theta)] \propto \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_i(k) \log(\pi_k f_k(\mathbf{x}_i))$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_i(k) \log \pi_k + \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_i(k) \log f_k(\mathbf{x}_i)$$

## EM optimization

E step Given $\theta$, compute

$$\gamma_k(i) = \frac{\pi_k f_k(\mathbf{x}_i)}{f(x)}$$

M step Given $\gamma_k(i)$, compute

$$\theta^* = (\pi_k, \mu_k, \Sigma_k)^* = \mathit{argmin} E[\ell(\theta)]$$

# Maximization step

$\pi_k$: Fraction of points in $C_k$

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_k(i)$$

$\mu_k$: Mean of $C_k$

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma_k(i) \mathbf{x}_i}{\sum_{i=1}^{N} \gamma_k(i)}$$

$\Sigma_k$: Covariance

$$\Sigma_k = \frac{\sum_{i=1}^{N} \gamma_k(i)(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)'}{\sum_{i=1}^{N} \gamma_k(i)}$$

# Clustering

- K-Means
- Expectation Maximization
- Selecting the number of clusters
- Affinity propagation
- Scalability

# Choosing the number of clusters

$K$-means constructs a partition whatever the $K$ value is.

## Selection of K

- Bayesian approaches
  Tradeoff between accuracy / richness of the model
- Stability
  Varying the data should not change the result
- Gap statistics
  Compare with null hypothesis: all data in same cluster.

# Bayesian approaches

## Bayesian Information Criterion

$$BIC(\theta) = \ell(\theta) - \frac{\#\theta}{2} \log N$$

Select $K = \text{argmax } BIC(\theta)$
where $\#\theta =$ number of free parameters in $\theta$:

- if all components have same scalar variance $\sigma$

$$\#\theta = K - 1 + 1 + Kd$$

- if each component has a scalar variance $\sigma_k$

$$\#\theta = K - 1 + K(d + 1)$$

- if each component has a full covariance matrix $\Sigma_k$

$$\#\theta = K - 1 + K(d + d(d - 1)/2)$$

# Gap statistics

## Principle: hypothesis testing

1. Consider hypothesis $H_0$: there is no cluster in the data. $\mathcal{E}$ is generated from a no-cluster distribution $\pi$.
2. Estimate the distribution $f_{0,K}$ of $\mathcal{L}(C_1, \ldots C_K)$ for data generated after $\pi$. Analytically if $\pi$ is simple
   Use Monte-Carlo methods otherwise
3. Reject $H_0$ with confidence $\alpha$ if the probability of generating the true value $\mathcal{L}(C_1, \ldots C_K)$ under $f_{0,K}$ is less than $\alpha$.

Beware: the test is done for all $K$ values...

# Gap statistics, 2

## Algorithm

Assume $\mathcal{E}$ extracted from a no-cluster distribution,
e.g. a single Gaussian.

1. Sample $\mathcal{E}$ according to this distribution
2. Apply $K$-means on this sample
3. Measure the associated loss function

Repeat : compute the average $\bar{\mathcal{L}}_0(K)$ and variance $\sigma_0(K)$
Define the gap:

$$Gap(K) = \bar{\mathcal{L}}_0(K) - \mathcal{L}(C_1, \dots C_K)$$

Rule Select min $K$ s.t.

$$Gap(K) \geq Gap(K+1) - \sigma_0(K+1)$$

What is nice: also tells if there are no clusters in the data...

# Stability

## Principle

- Consider $\mathcal{E}'$ perturbed from $\mathcal{E}$
- Construct $C'_1, \ldots C'_K$ from $\mathcal{E}'$
- Evaluate the "distance" between $(C_1, \ldots C_K)$ and $(C'_1, \ldots C'_K)$
- If small distance (stability), $K$ is OK

## Distortion $D(\Delta)$

Define $S$ $S_{ij} = < \mathbf{x}_i, \mathbf{x}_j >$

$(\lambda_i, v_i)$ i-th (eigenvalue, eigenvector) of $S$

$X$ $X_{i,j} =$ 1 iff $\mathbf{x}_i \in C_j$

$$D(\Delta) = \sum_i ||\mathbf{x}_i - \mu_{\tau(i)}||^2 = tr(S) - tr(X'SX)$$

Minimal distortion $D^* = tr(S) - \sum_{k=1}^{K-1} \lambda_k$

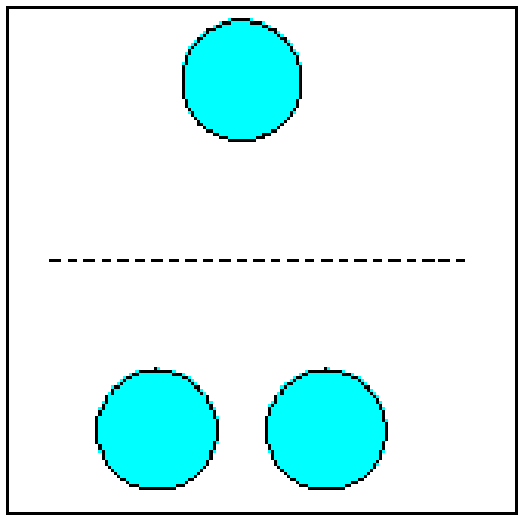
# Stability, 2

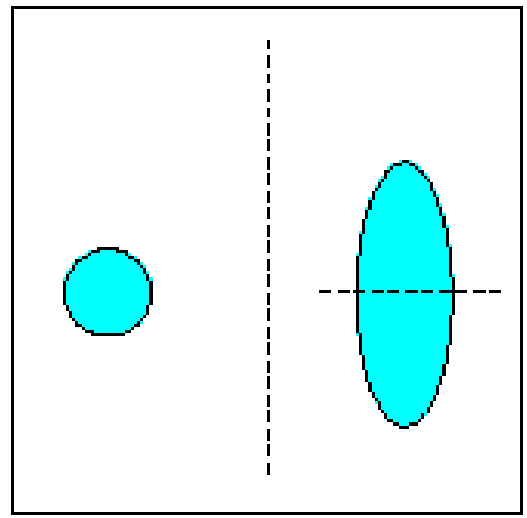
## Results

- $\Delta$ has low distortion $\Rightarrow$ $(\mu_1, \ldots \mu_K)$ close to space $(v_1, \ldots v_K)$.
- $\Delta_1$, and $\Delta_2$ have low distortion $\Rightarrow$ "close"
- (and close to "optimal" clustering)

Meila ICML 06

## Counter-example





# From K-Means to K-Centers

## Assumptions for K-Means

- A distance or dissimilarity
- Possibility to create artefacts — barycenters
- Not applicable in some domains — *average molecule?* *average sentence?*

## K-Centers, position of the problem

- A combinatorial optimization problem.
  Find $\sigma : \{1, \ldots, N\} \mapsto \{1, \ldots, N\}$ minimizing:

$$E[\sigma] = \sum_{i=1}^{N} d(\mathbf{x}_i, \mathbf{x}_{\sigma(i)})$$

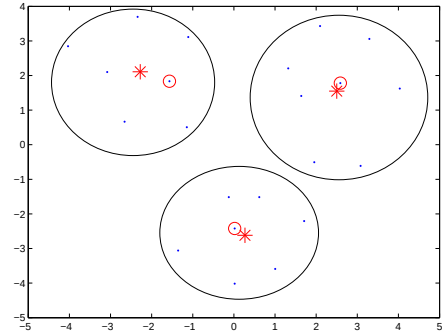
*(What is missing here ?)*

# Clustering

- K-Means
- Expectation Maximization
- Selecting the number of clusters
- Affinity propagation
- Scalability

# Motivations

## Clustering: Unsupervised learning



## Affinity Propagation and State of the art

| | **K-means** | **K-centers** | **AP** |
|---|---|---|---|
| exemplar | artefact | actual point | actual point |
| parameter | K | K | $s^*$ (penalty) |
| algorithm | greedy search | greedy search | message passing |
| performance | not stable | not stable | stable |
| complexity | $N \times K$ | $N \times K$ | $N^2 log(N)$ |

Clustering by Passing Messages Between Data Points. B.J. Frey, D. Dueck. Science 2007

# Affinity Propagation

Given

$\mathcal{E} = \{e_1, e_2, ..., e_N\}$ *elements*

$d(e_i, e_j)$ *their dissimilarity*

Find $\sigma : \mathcal{E} \mapsto \mathcal{E}$ $\sigma(e_i)$, exemplar representing $e_i$

such that:

$$\sigma = argmax \sum_{i=1}^{N} S(e_i, \sigma(e_i))$$

where $\begin{cases} S(e_i, e_j) = -d^2(e_i, e_j) & \text{if } i \neq j \\ S(e_i, e_i) = -s^* \end{cases}$ $s^*$: penalty parameter

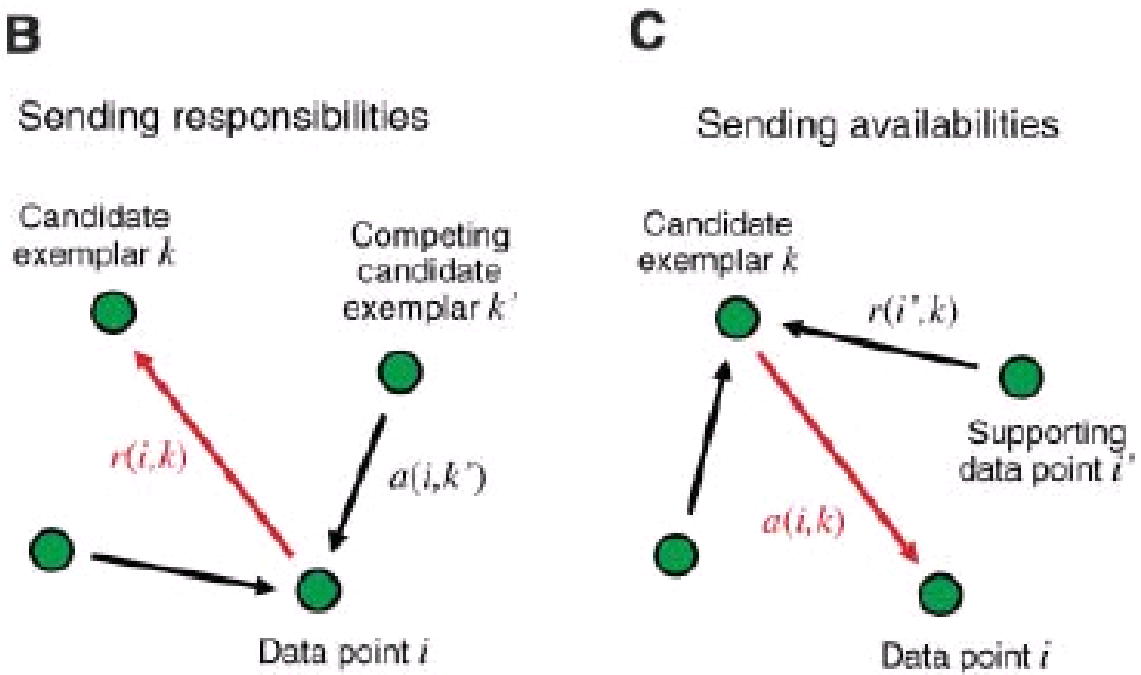
Particular cases

- $s^* = \infty$, only one exemplar — 1 cluster
- $s^* = 0$, every point is an exemplar — N clusters

# Affinity Propagation, Principle

## Algorithm: Message propagation

- Responsibility $r(i, k)$ could $\mathbf{x}_k$ be examplar for $\mathbf{x}_i$
- Availability $a(i, k)$.



# Affinity Propagation, 2

Two types of messages

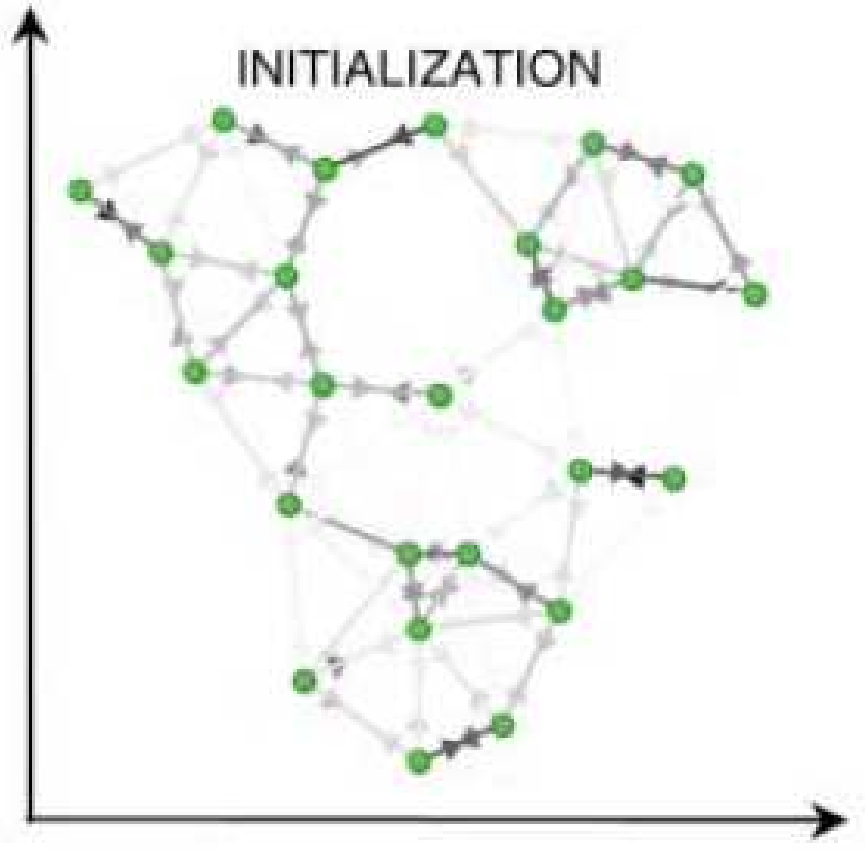
- $r(i,k)$ : Responsibility of $i$ to $k$
- $a(i,k)$ : Availability of $i$ as examplar for $k$

Rules of propagation

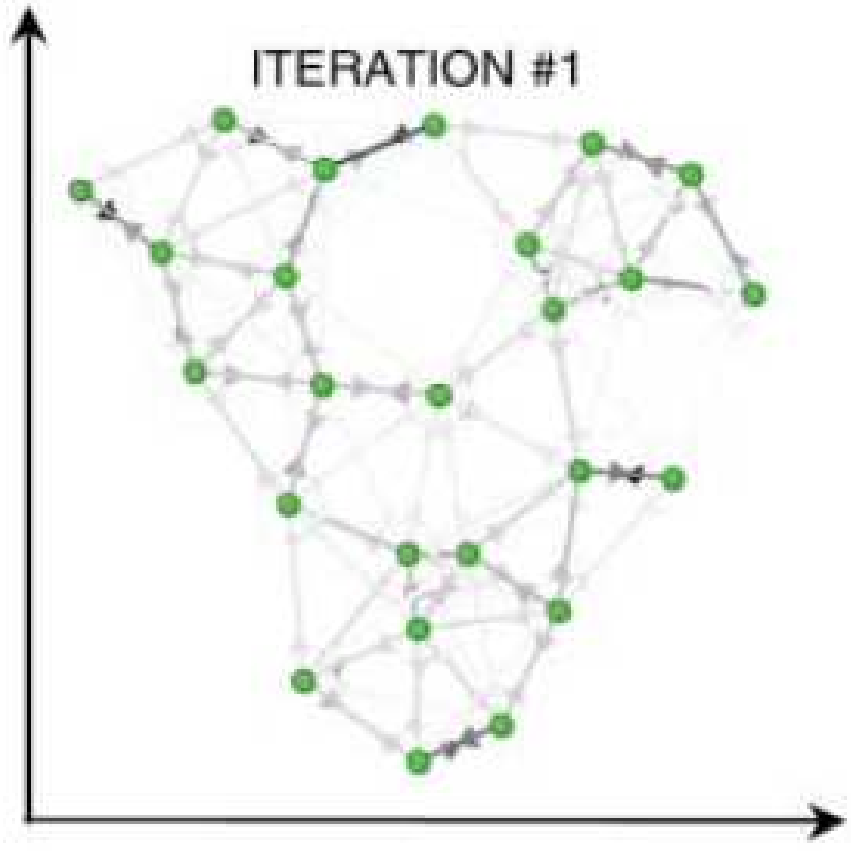
$$\begin{aligned} r(i,k) &= S(e_i,e_k) - \max_{k',k'\neq k}\{a(i,k') + S(e_i,e'_k)\} \\ r(k,k) &= S(e_k,e_k) - \max_{k',k'\neq k}\{S(e_k,e'_k)\} \end{aligned}$$

$$\begin{aligned} a(i,k) &= min\{0, r(k,k) + \sum_{i',i'\neq i,k} \max\{0, r(i',k)\}\} \\ a(k,k) &= \sum_{i',i'\neq k} \max\{0, r(i',k)\} \end{aligned}$$

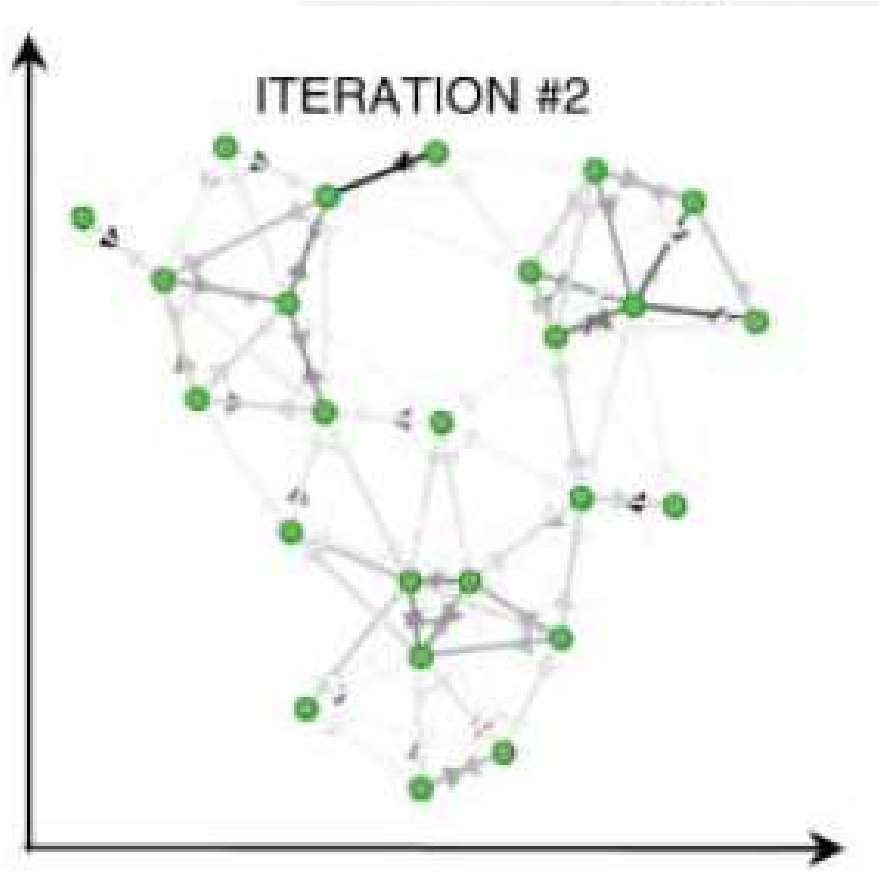
# Iterations of Message passing



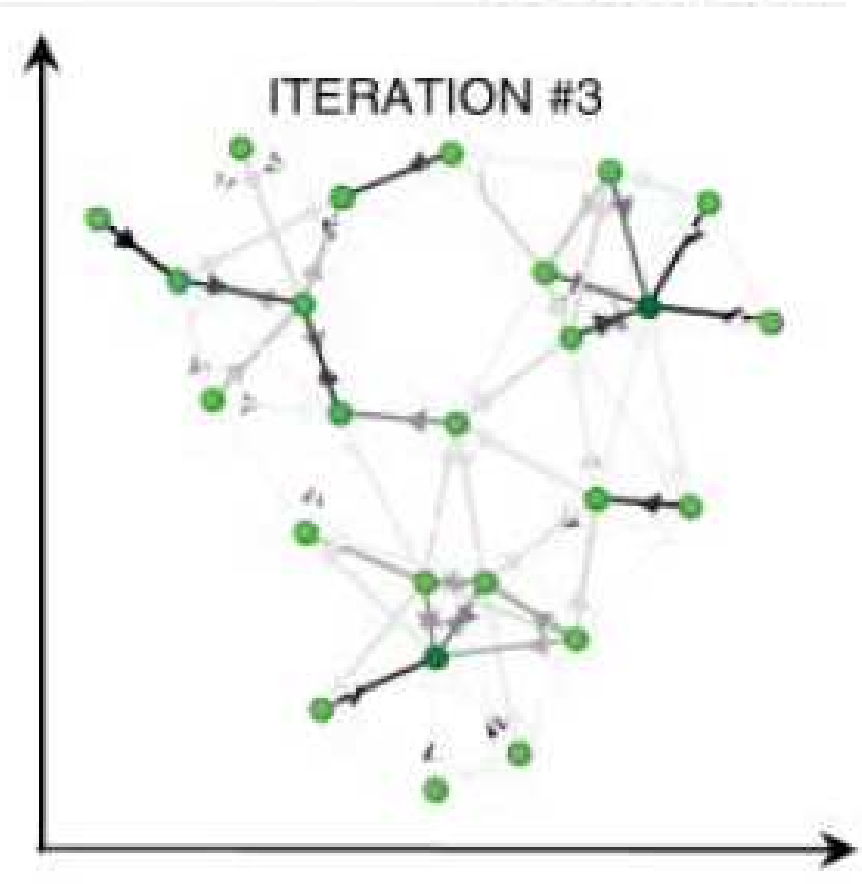
# Iterations of Message passing



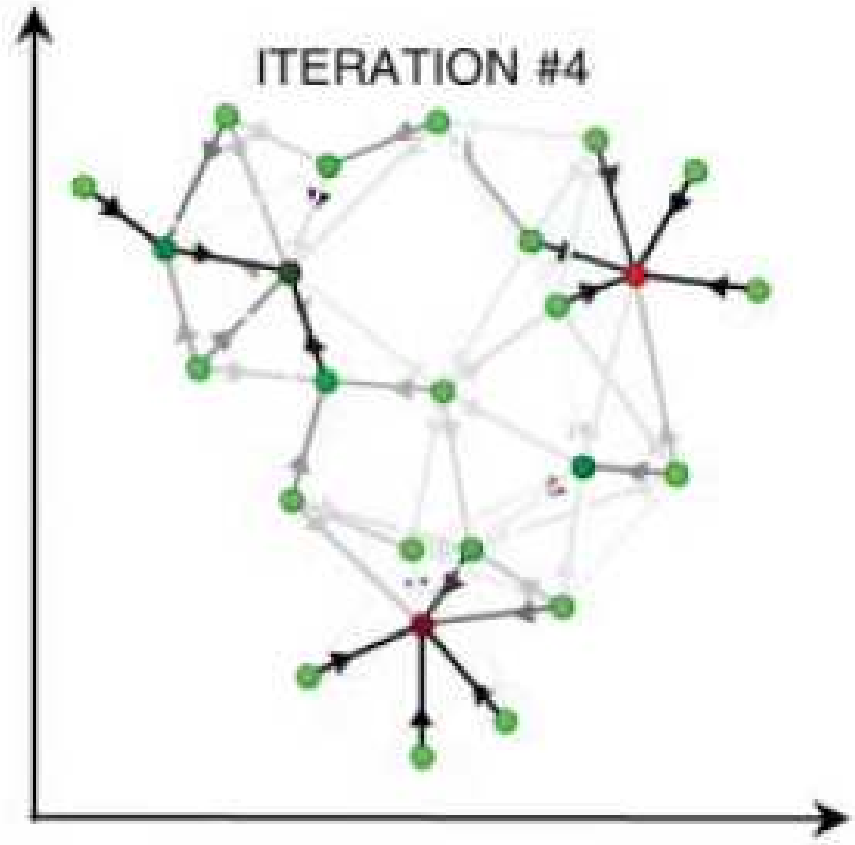
# Iterations of Message passing



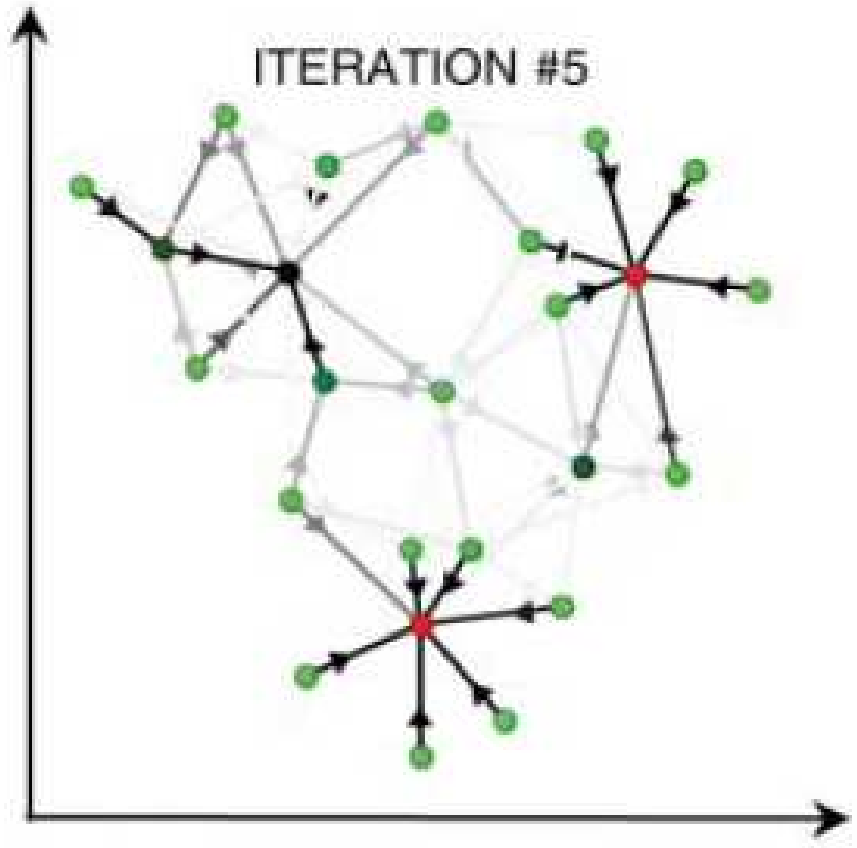
# Iterations of Message passing



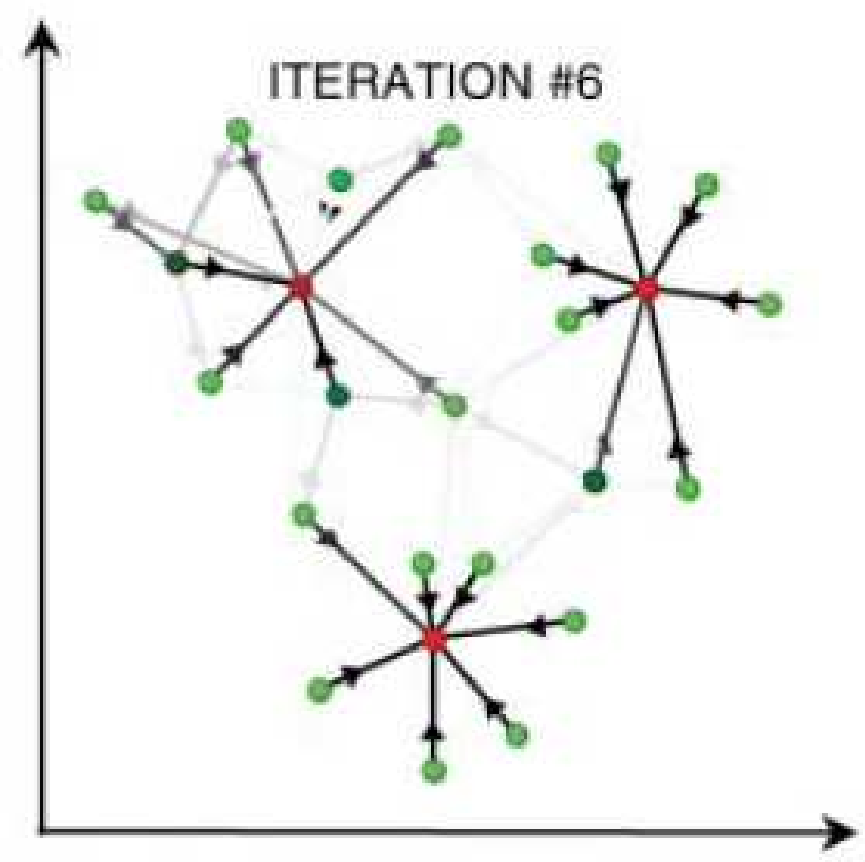
# Iterations of Message passing



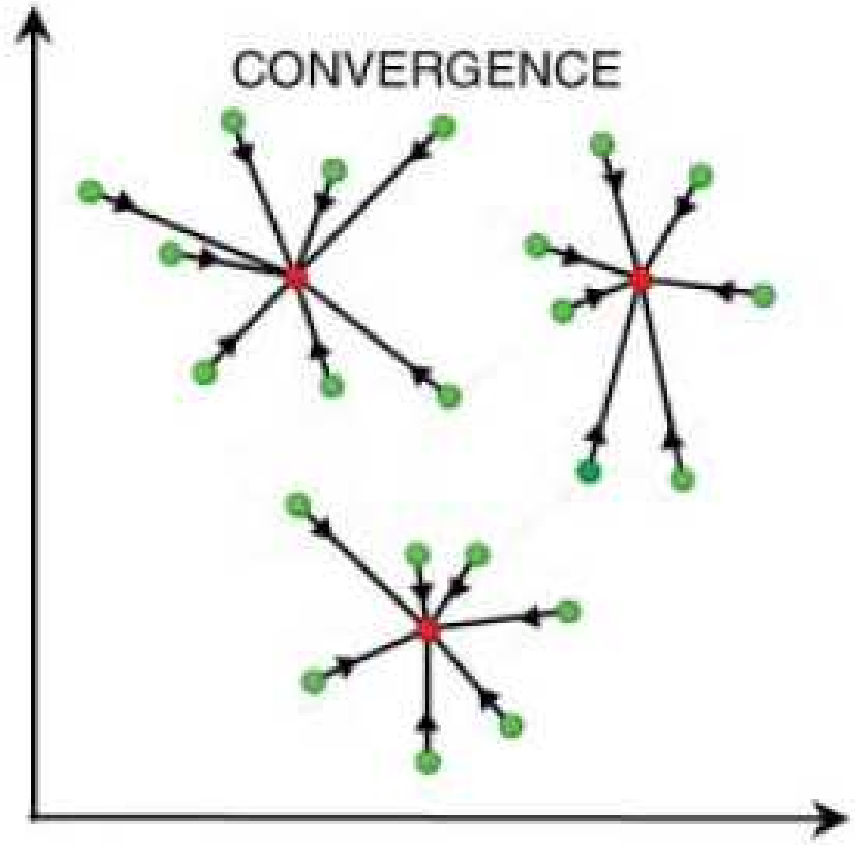
# Iterations of Message passing



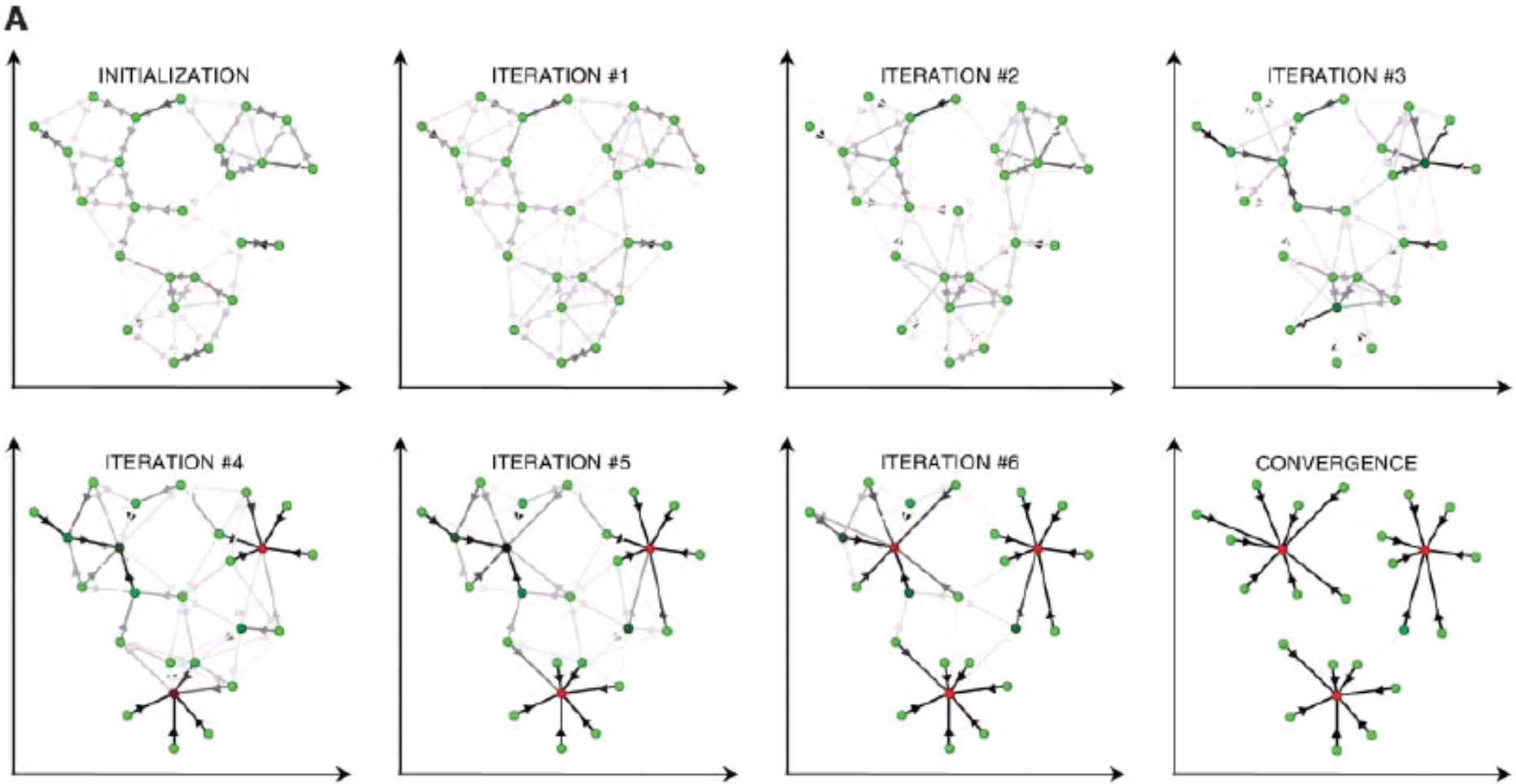
# Iterations of Message passing



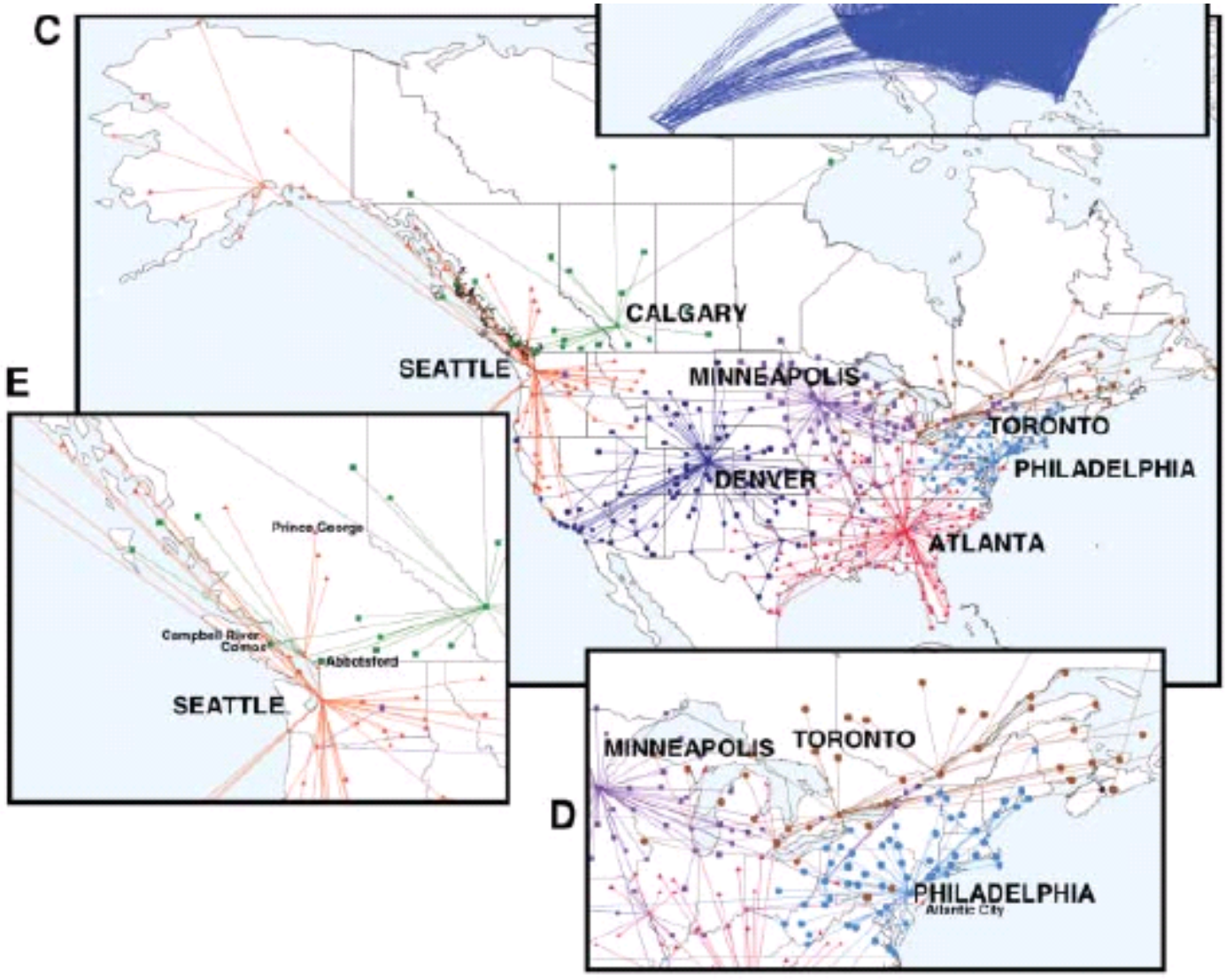
# Iterations of Message passing



# Affinity Propagation, cont'd



# Affinity Propagation, cont'd



# Affinity Propagation in a Nutshell

WHEN to use it ?

When averages don't make sense e.g., molecules; documents

PROS vs $K$-centers

Lower distortion $D([\sigma]) = \sum_{i=1}^{N} d^2(e_i, \sigma(e_i))$

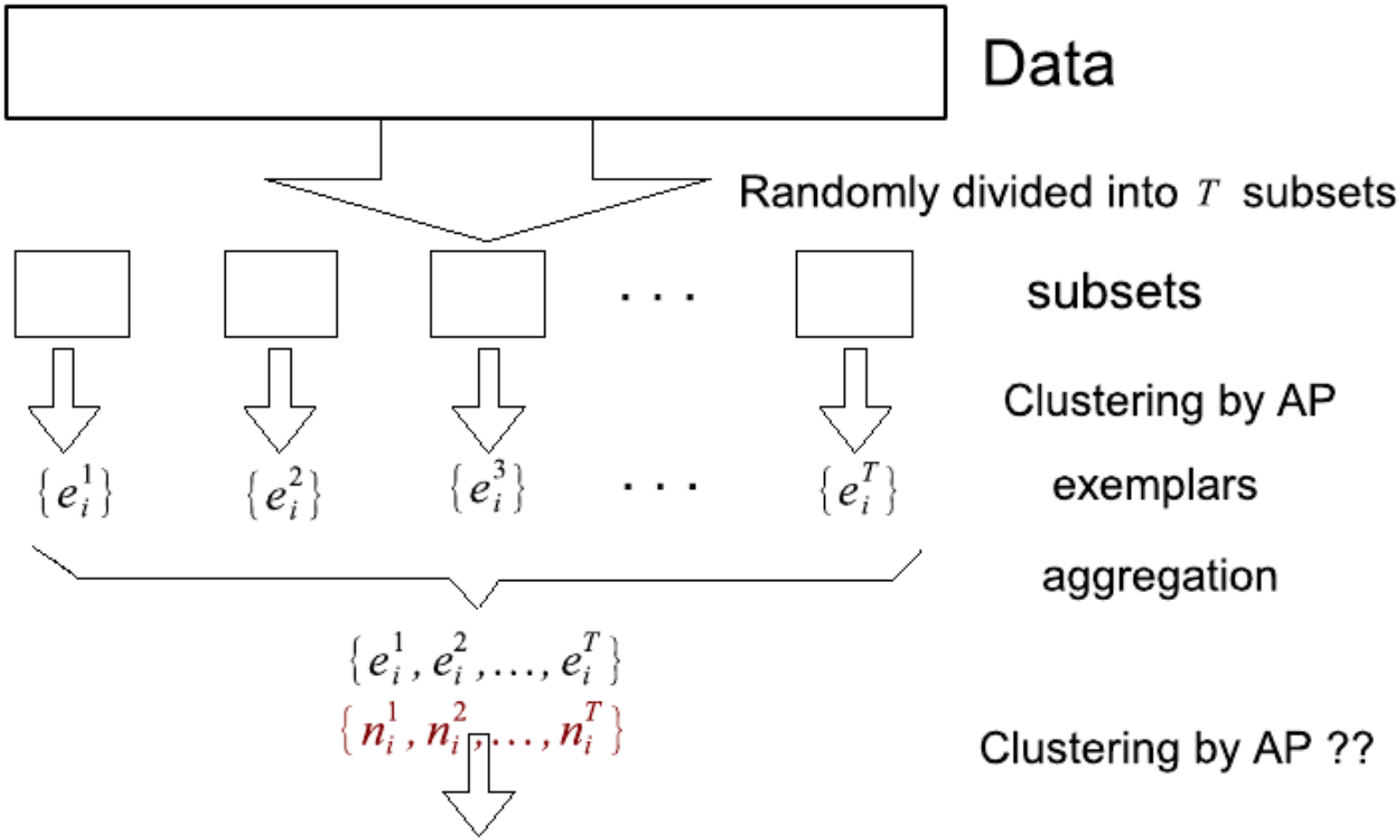
CONS: Computational complexity

- Similarity computation: $\mathcal{O}(N^2)$
- Message passing: $\mathcal{O}(N^2 \log N)$

# Clustering

- K-Means
- Expectation Maximization
- Selecting the number of clusters
- Affinity propagation
- Scalability

# Hierarchical AP




**Clustering data streams: Theory and practice. S. Guha, A. Meyerson, N. Mishra, R. Motwani. TKDE 2003.**

# Weighted AP

| AP | | WAP |
|---|---|---|
| $e_i$ | | $(e_i, n_i)$ |
| $S(e_i, e_j)$ | $\rightarrow$ | $n_i \times S(e_i, e_j)$ |
| $S(e_i, e_i)$ | $\rightarrow$ | $S(e_i, e_i) + (n_i - 1) \times \epsilon$ |

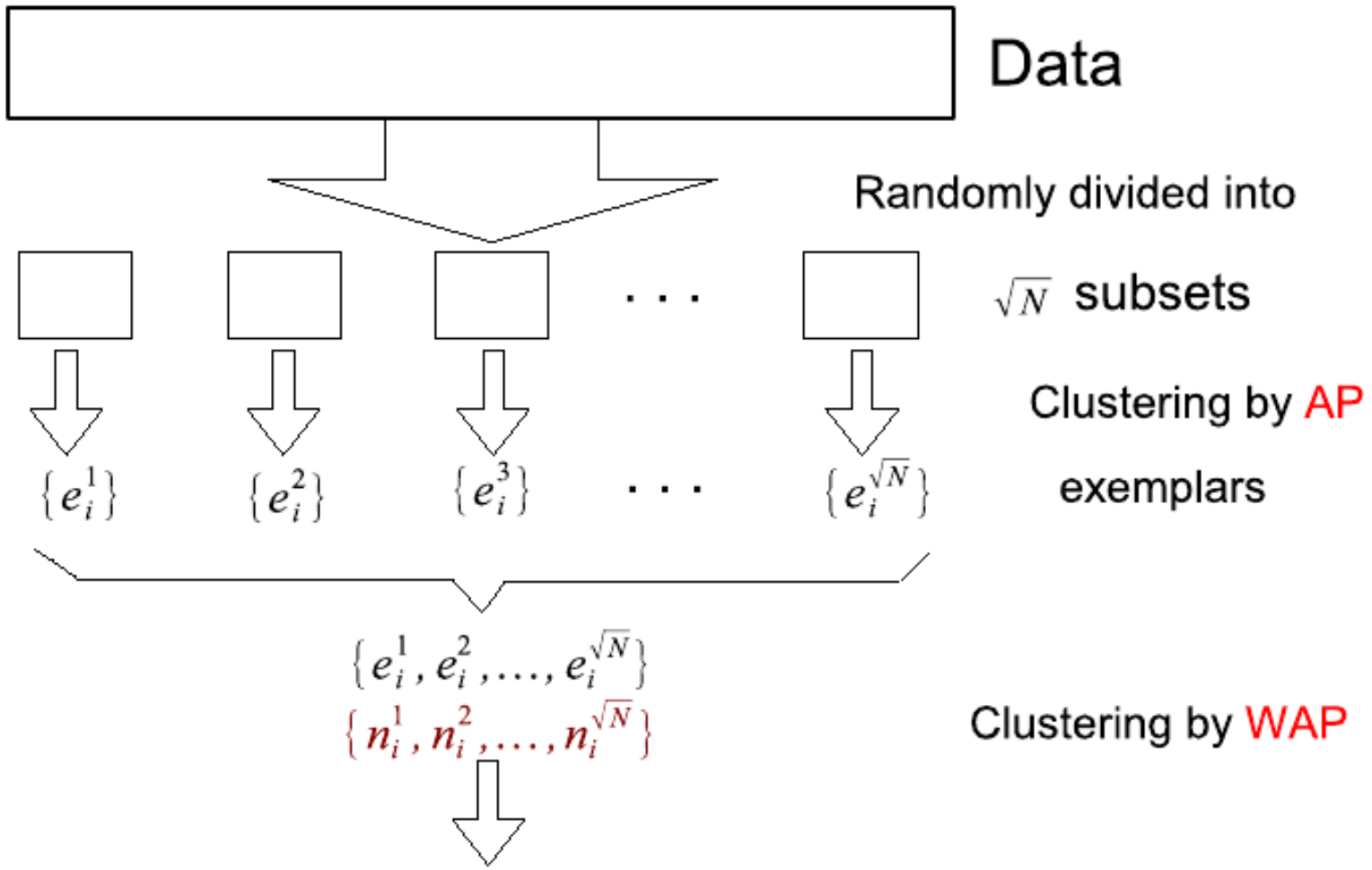
With $S(e_i, e_j)$ price for $e_i$ to select $e_j$ as an exemplar

$\epsilon$ variance of $n_i$ points

Proposition

$WAP \equiv AP$ with duplicated elements

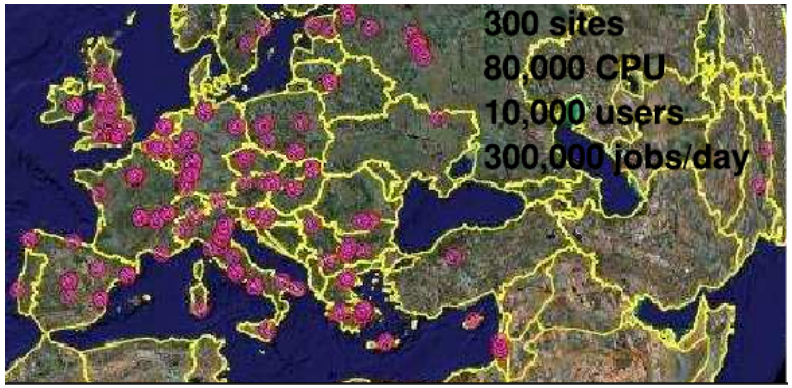
# Hierarchical WAP




- Complexity of HiWAP is $\mathcal{O}(N^{3/2})$
- $\longrightarrow$ can be iteratively reduced to $\mathcal{O}(N^{1+\gamma})$

# Validation of Hi-WAP on EGEE jobs

- EGEE (Enabling Grids for E-sciencE) http://public.eu-egee.org/
- 300 sites in 50 countries
- 10,000 user access to 80,000 CPUs
- 300,000 jobs per day



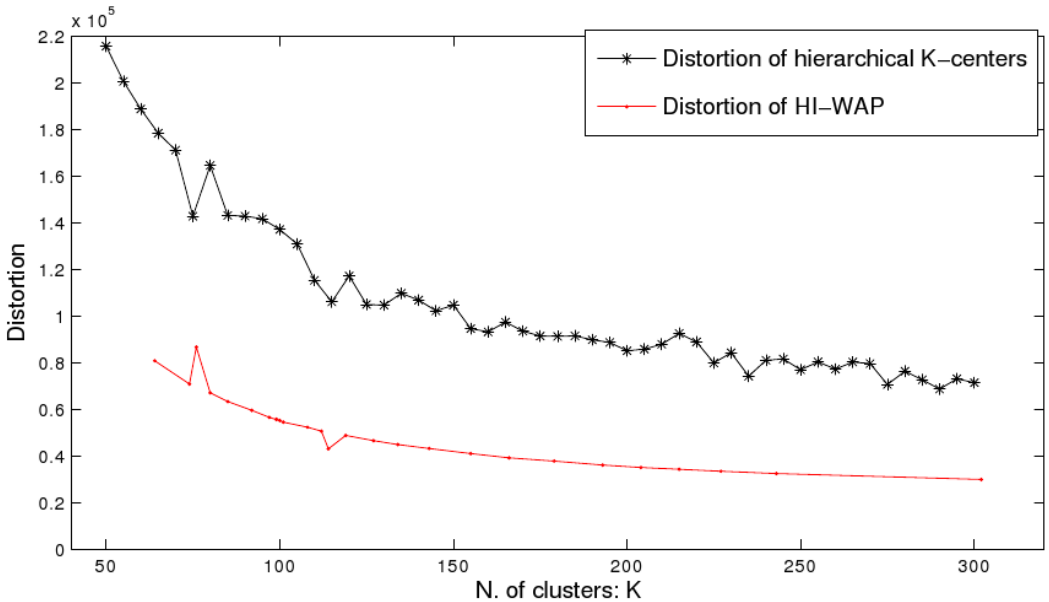

Description of jobs (237,087)

- 4 numeric features — e.g. duration of execution
- 1 symbolic feature — name of queue

# Validation of Hi-WAP on EGEE jobs

## Evaluation: Distortion

$$D([\sigma]) = \sum_{i=1}^{N} d^2(e_i, \sigma(e_i))$$




- 237,087 jobs
- each job $\in \mathbb{R}^5$
- CPU 10' (Intel 2.66GHz)

Baseline: $K$-centers

best out of ++100 runs
same overall computational cost