Affinity Propagation

Brendan Frey
University of Toronto

Where is the exemplar?

An interpretation of affinity propagation by Marc Mezard, *Laboratoire de Physique Théorique et Modeles Statistique, Paris.*

Caravaggio's "Vocazione di San Matteo"

What is cognition?

It’s what gets fixed up by recognition

Exemplar-based clustering

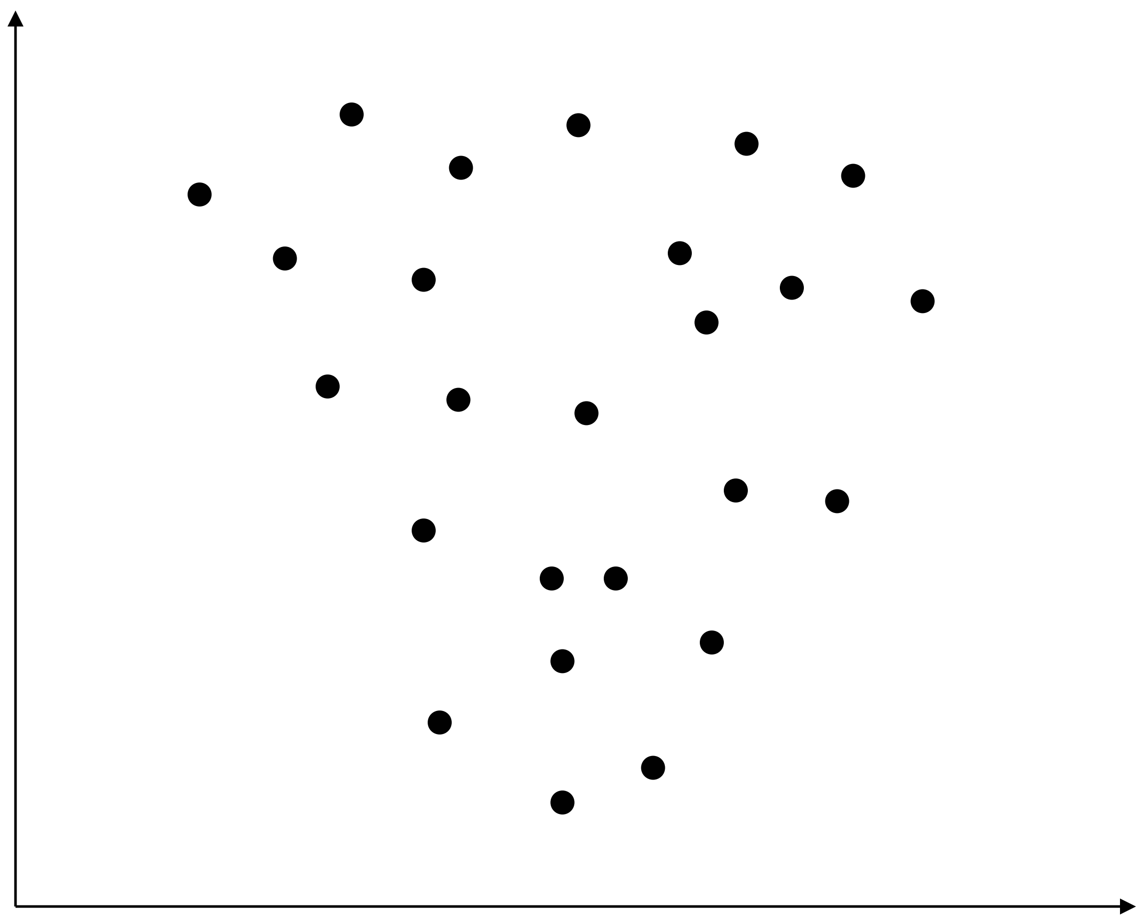
Input: A set of real-valued pair-wise similarities $\{s(i,k)\}$ between data points, plus the number of exemplars or a real-valued exemplar cost

Output: A subset of exemplar data points and an assignment of every other point to an exemplar

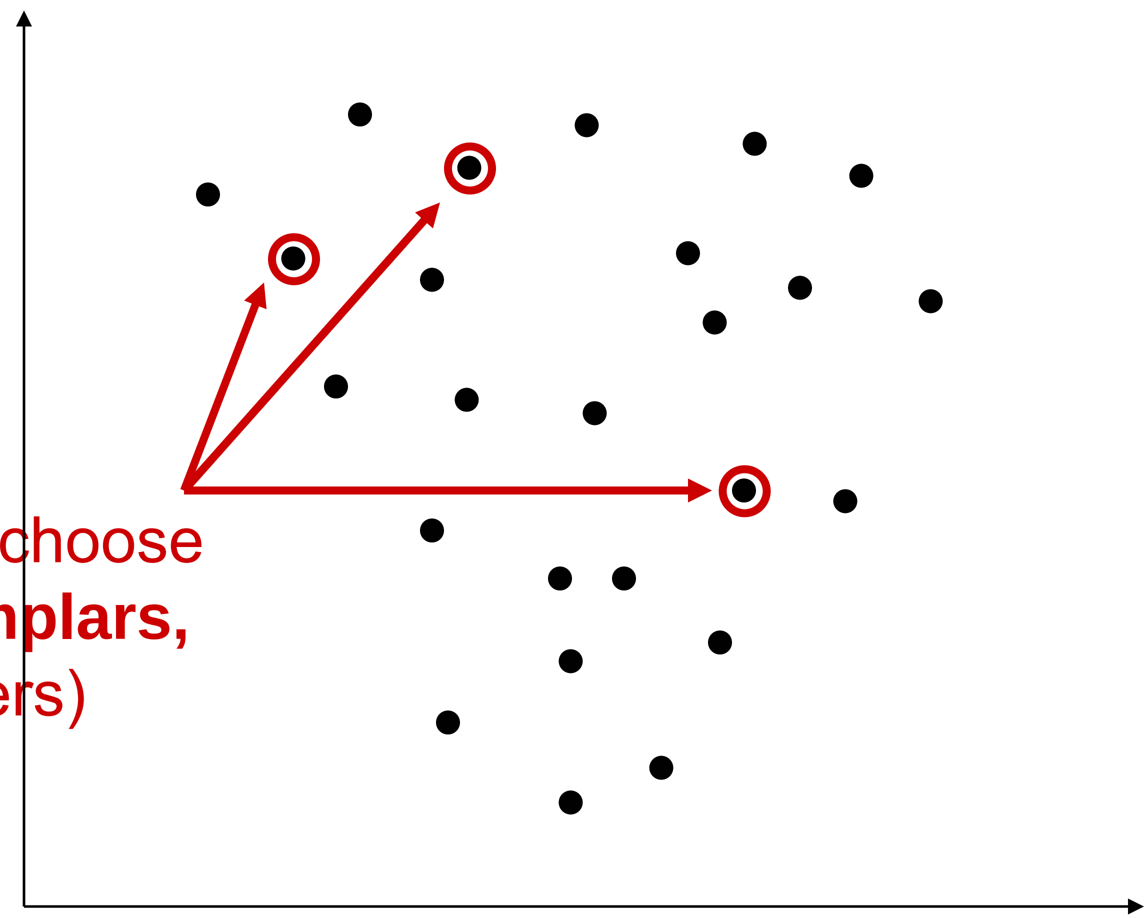
Objective: Maximize the sum of similarities between data points and their exemplars, minus the exemplar costs

k-medians clustering

(Lloyd's/LBG algorithm, facility location, p-median model)



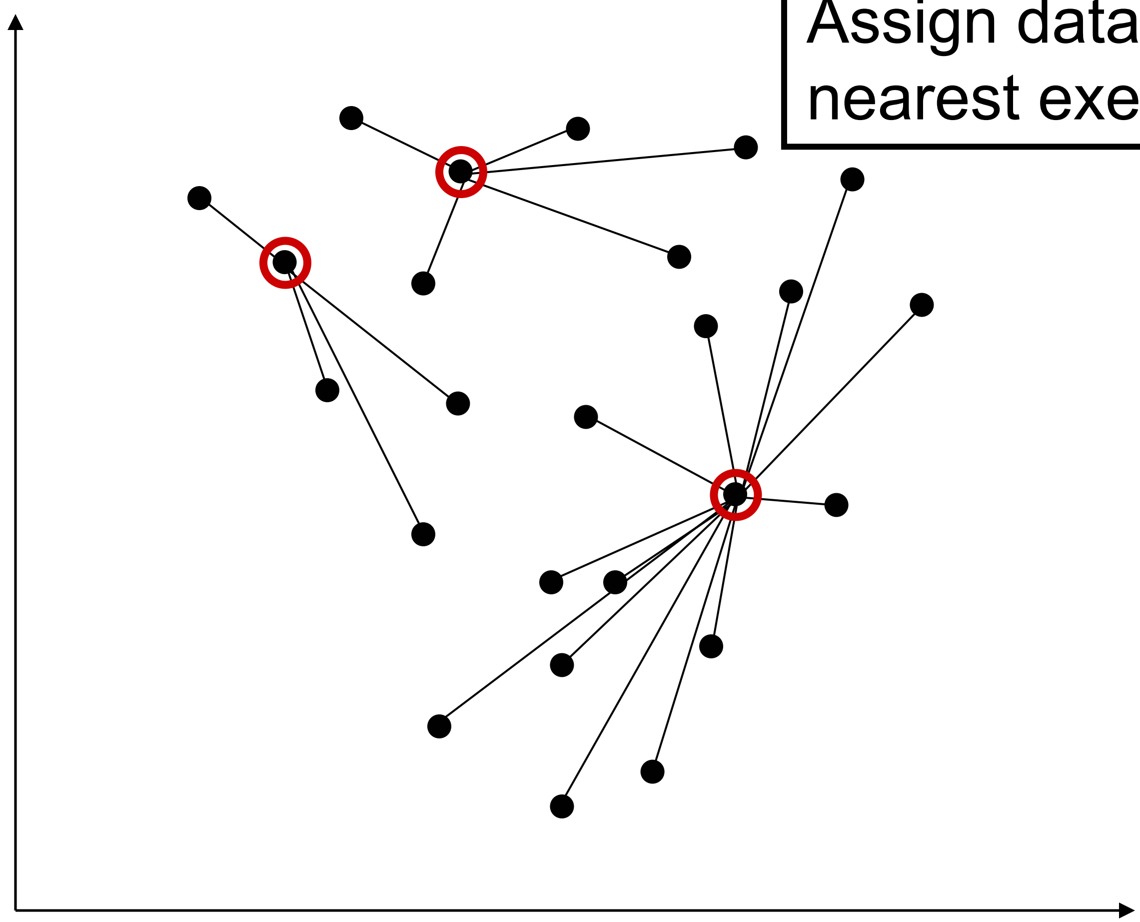
k-medians clustering



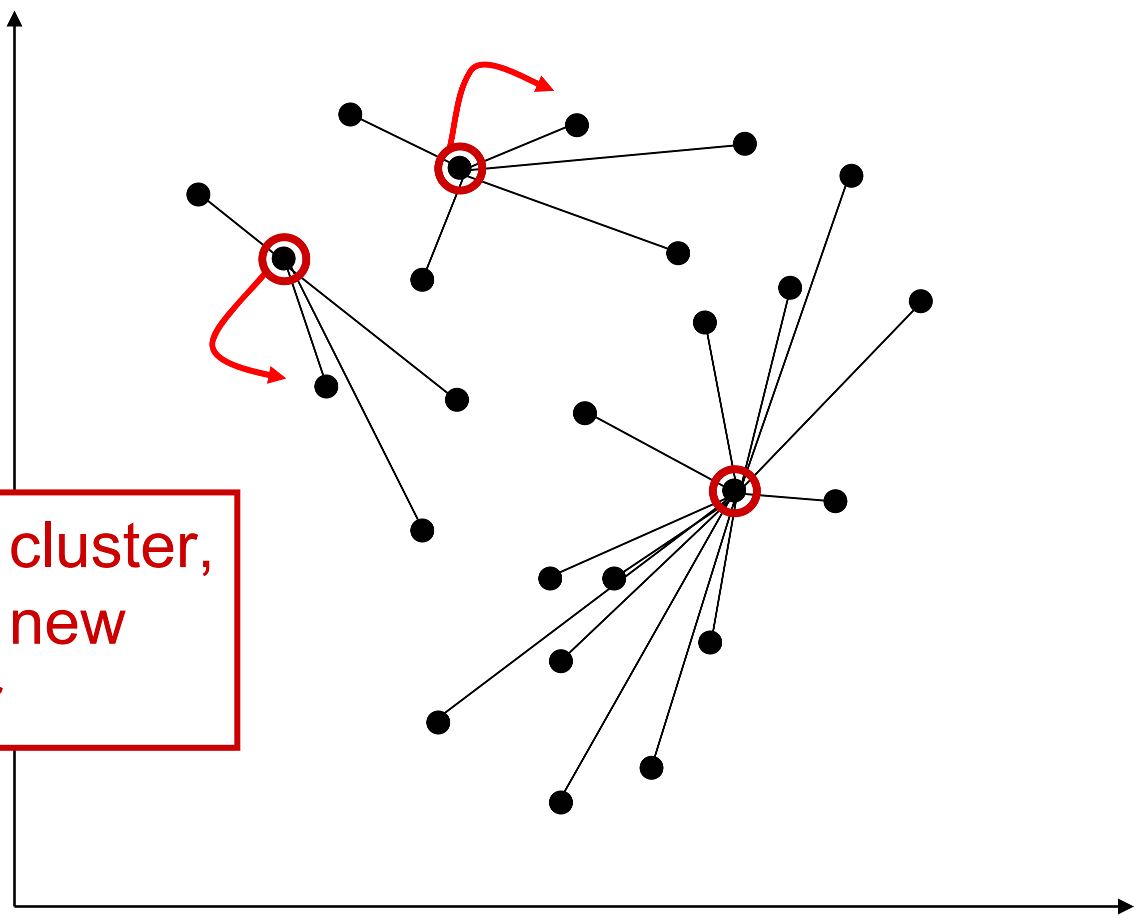
Randomly choose
initial **exemplars,**
(data centers)

k-medians clustering

Assign data points to nearest exemplars

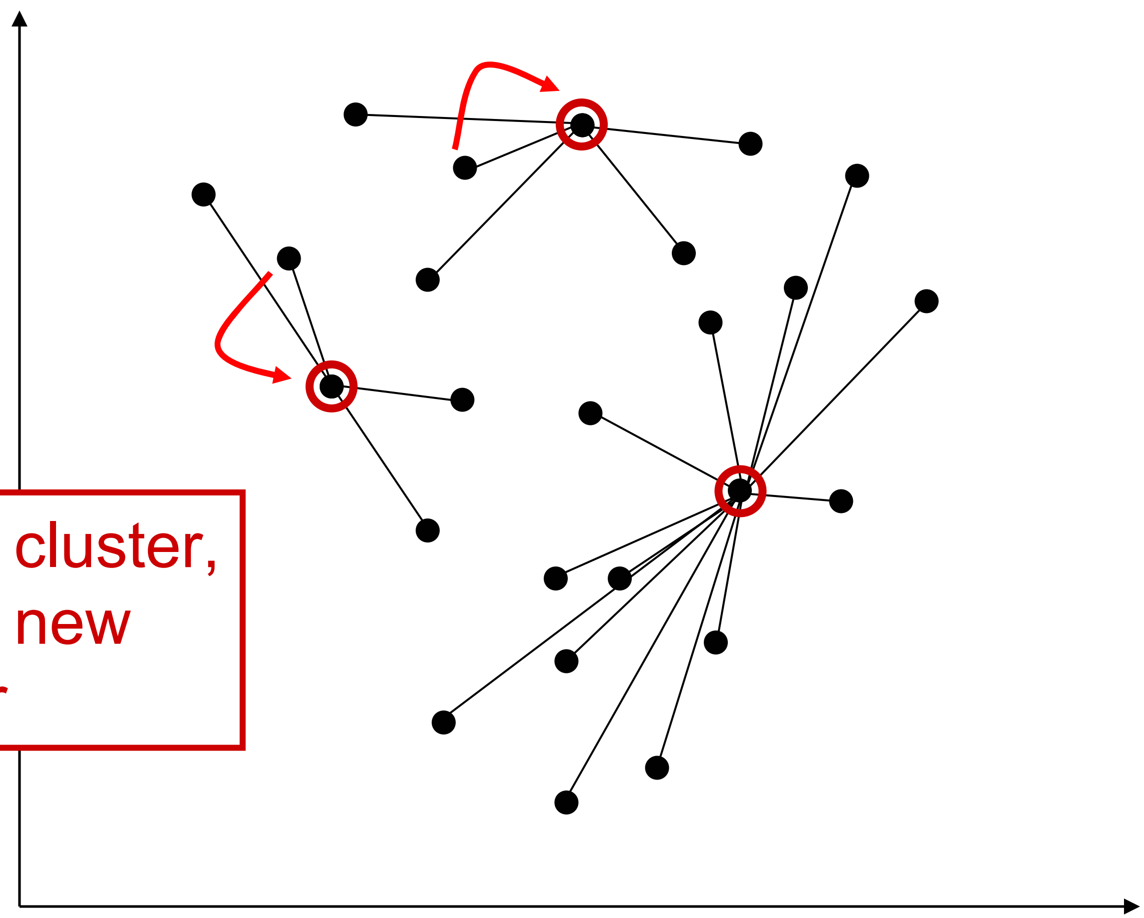


k-medians clustering



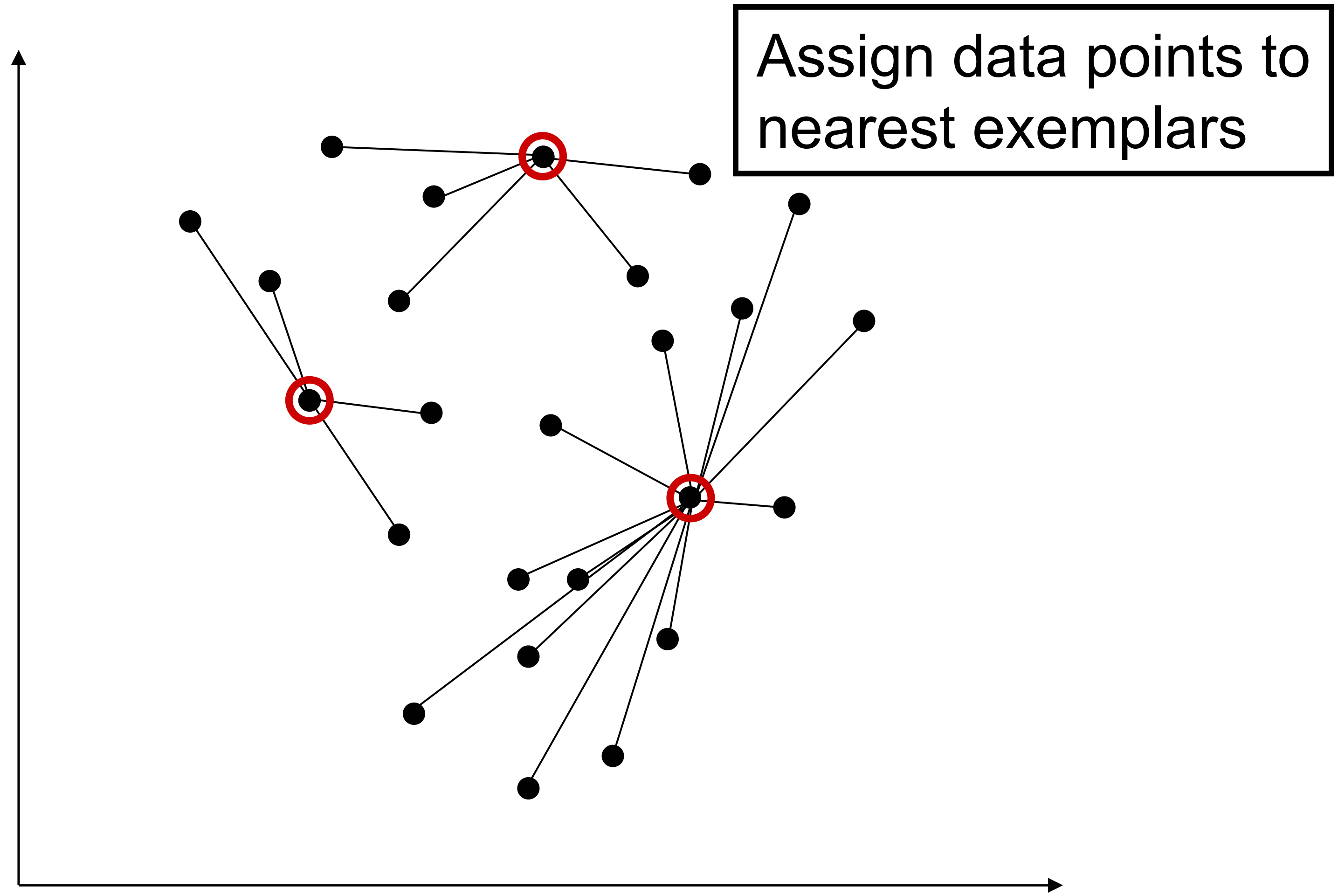
For each cluster, pick best new exemplar

k-medians clustering

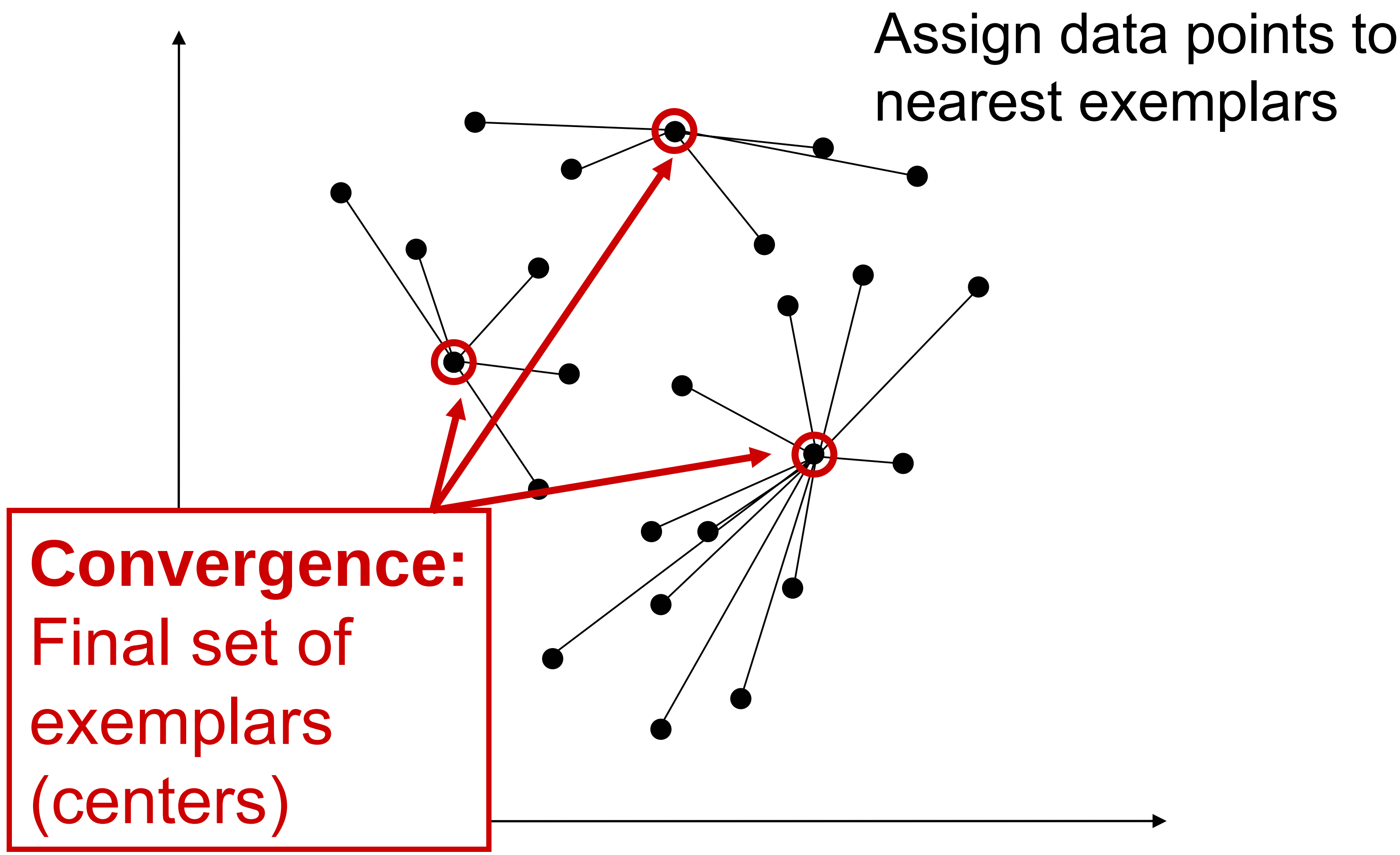


For each cluster, pick best new exemplar

k-medians clustering

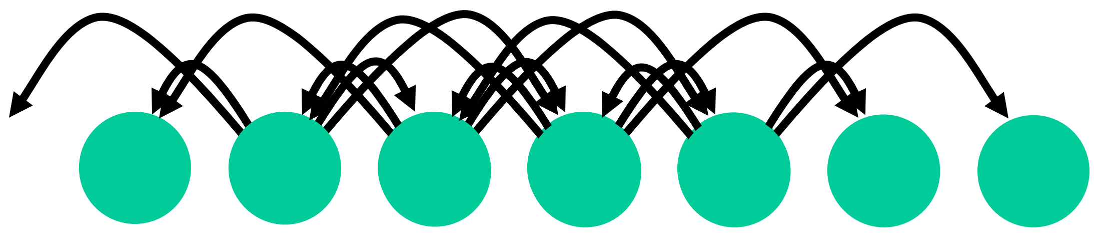


k-medians clustering

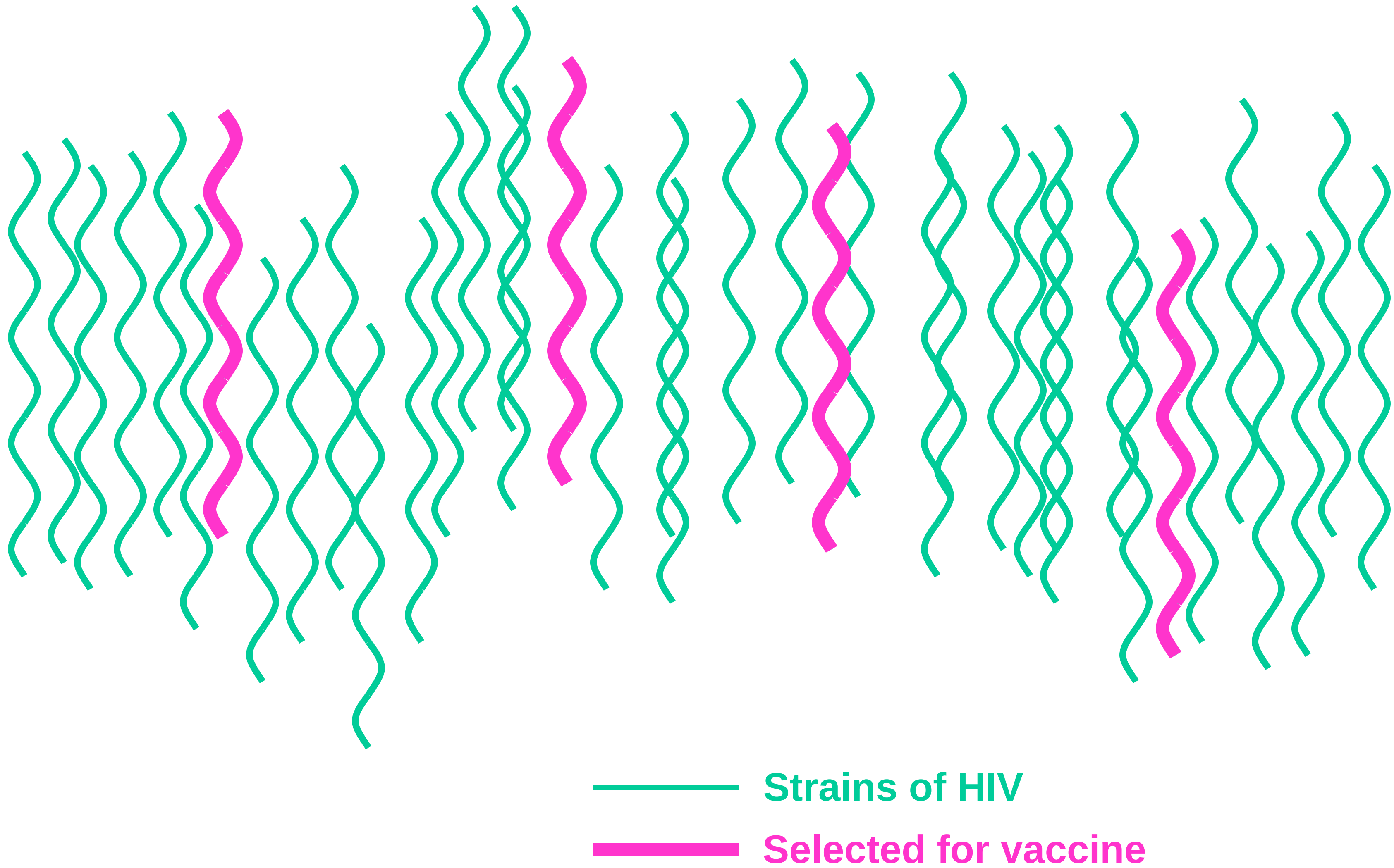


Example: Vision

Example: Winner-take-all activation

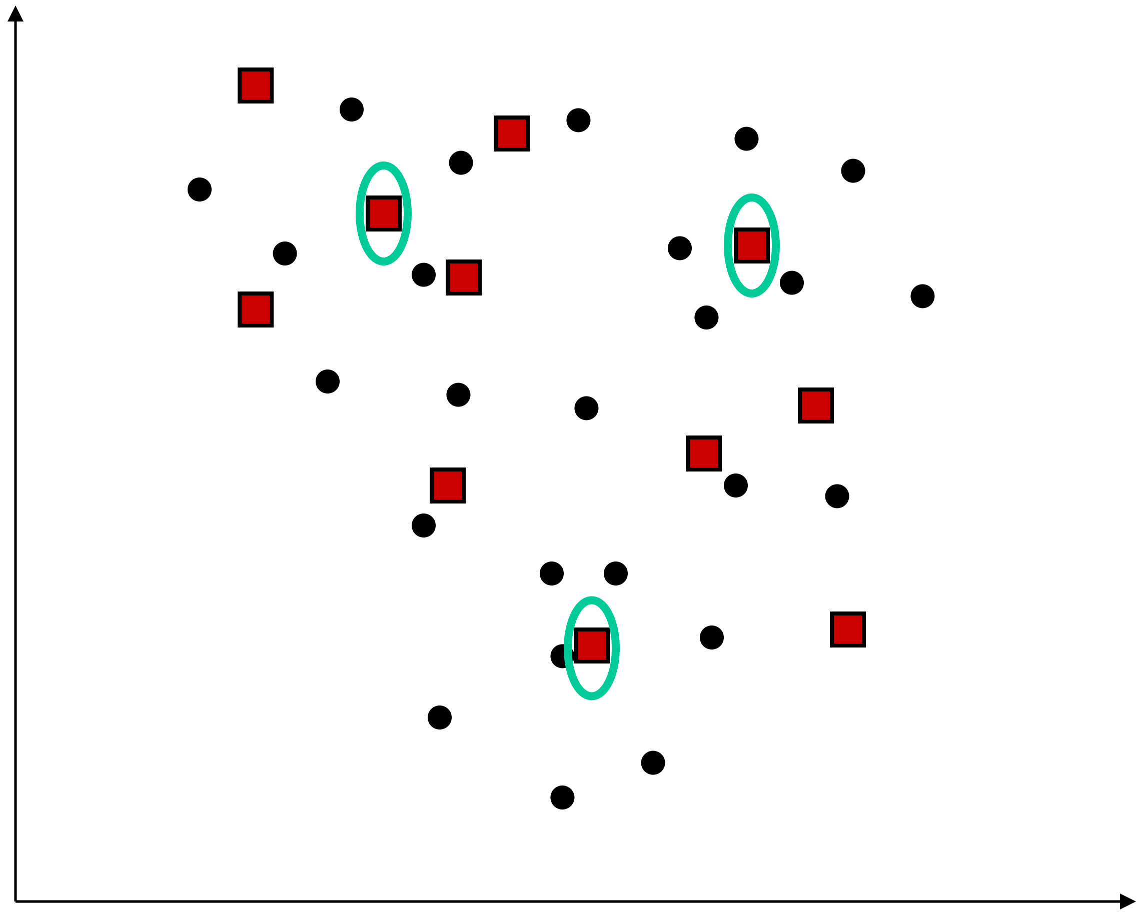


Example: Genomics, HIV vaccine design

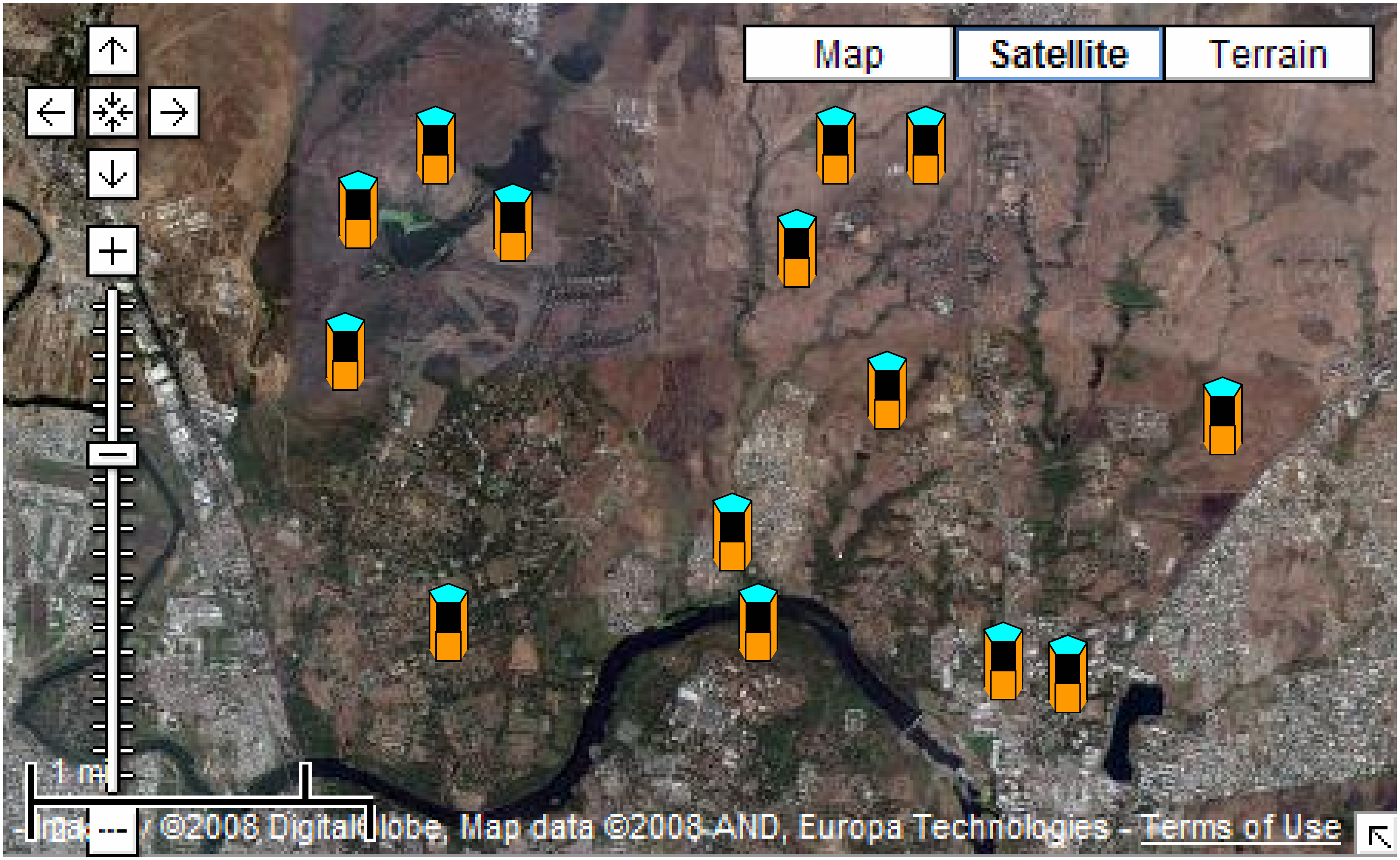


The facility location generalization

Identify a subset of potential facilities and assign users to facilities



Example: Optimal kiosk location

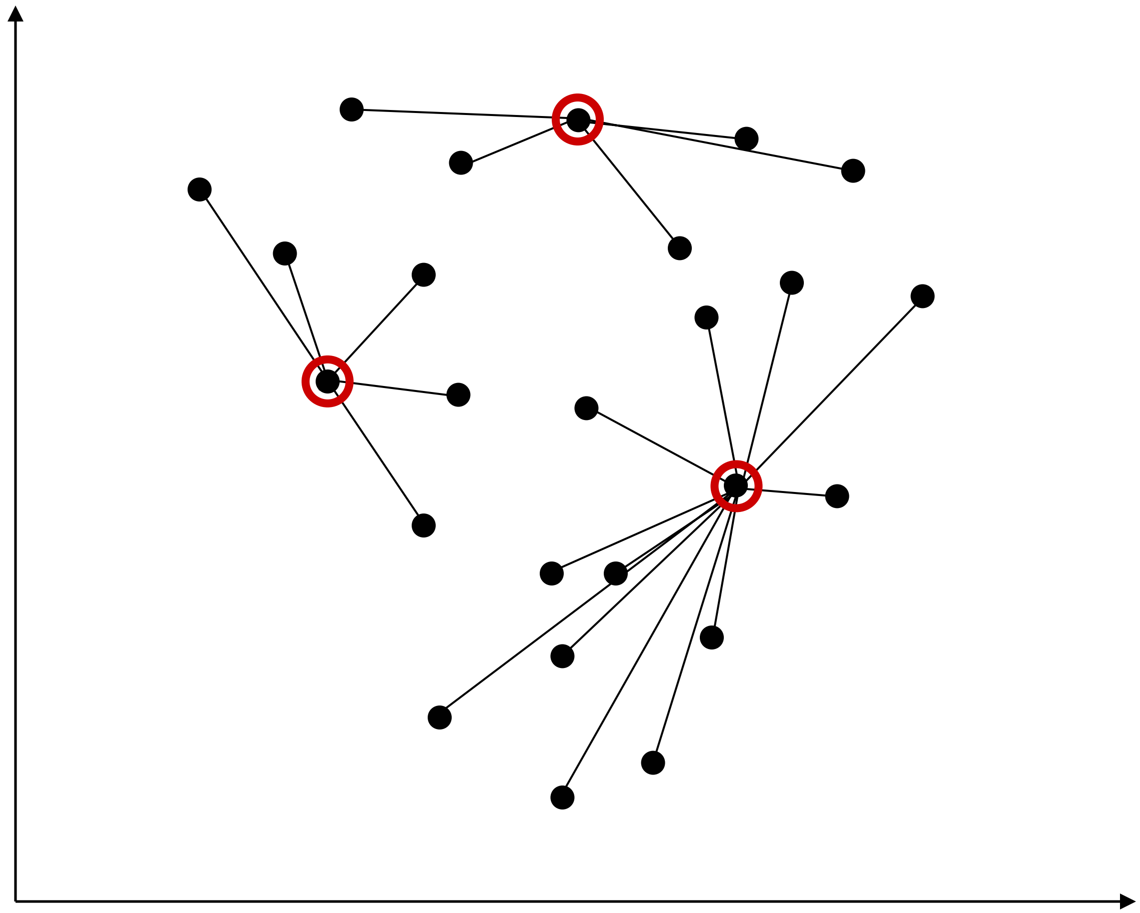


Why exemplar-based clustering is an important problem

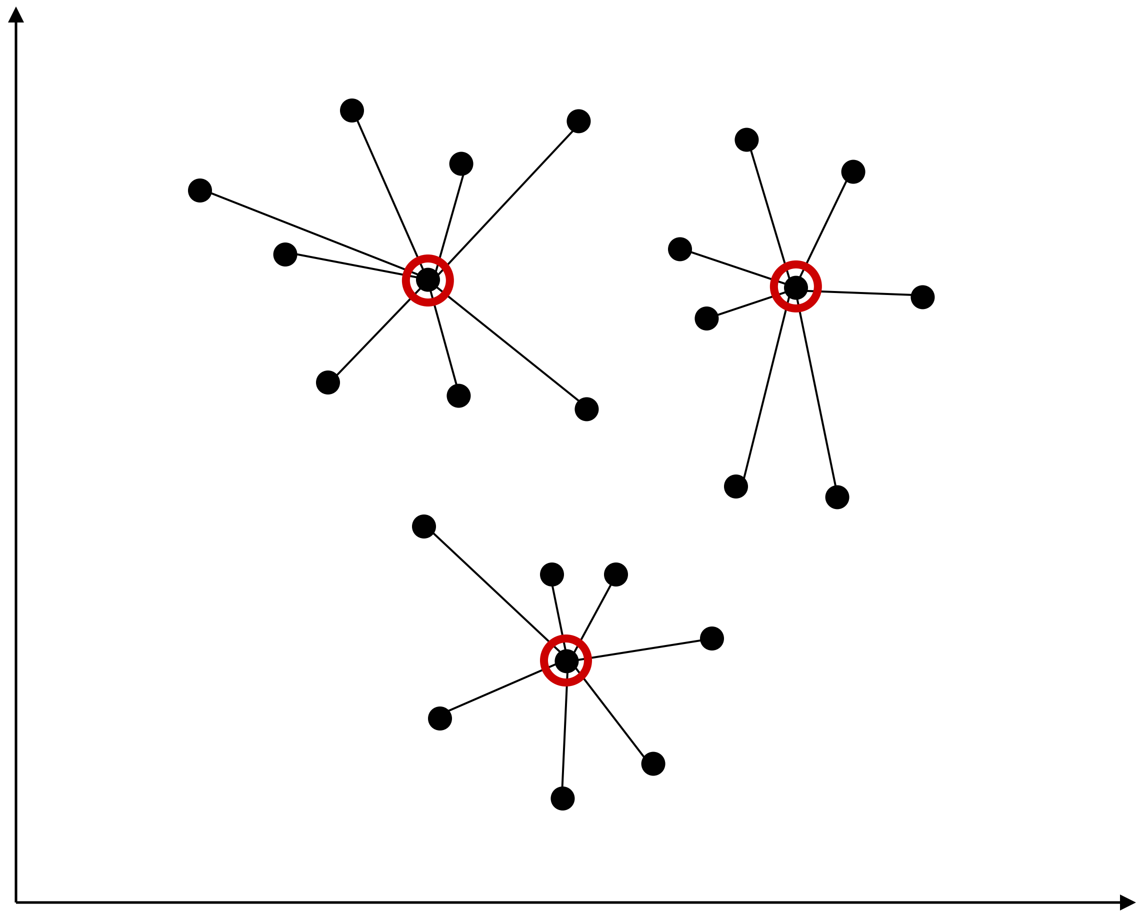
- Everybody (almost) needs clustering
- User-specified similarities offer increased flexibility over statistical models
- The clustering algorithm can be uncoupled from the details of how similarities are computed
- There ~~is~~ was potential for significant improvement on existing algorithms

How well does k-medians clustering work?

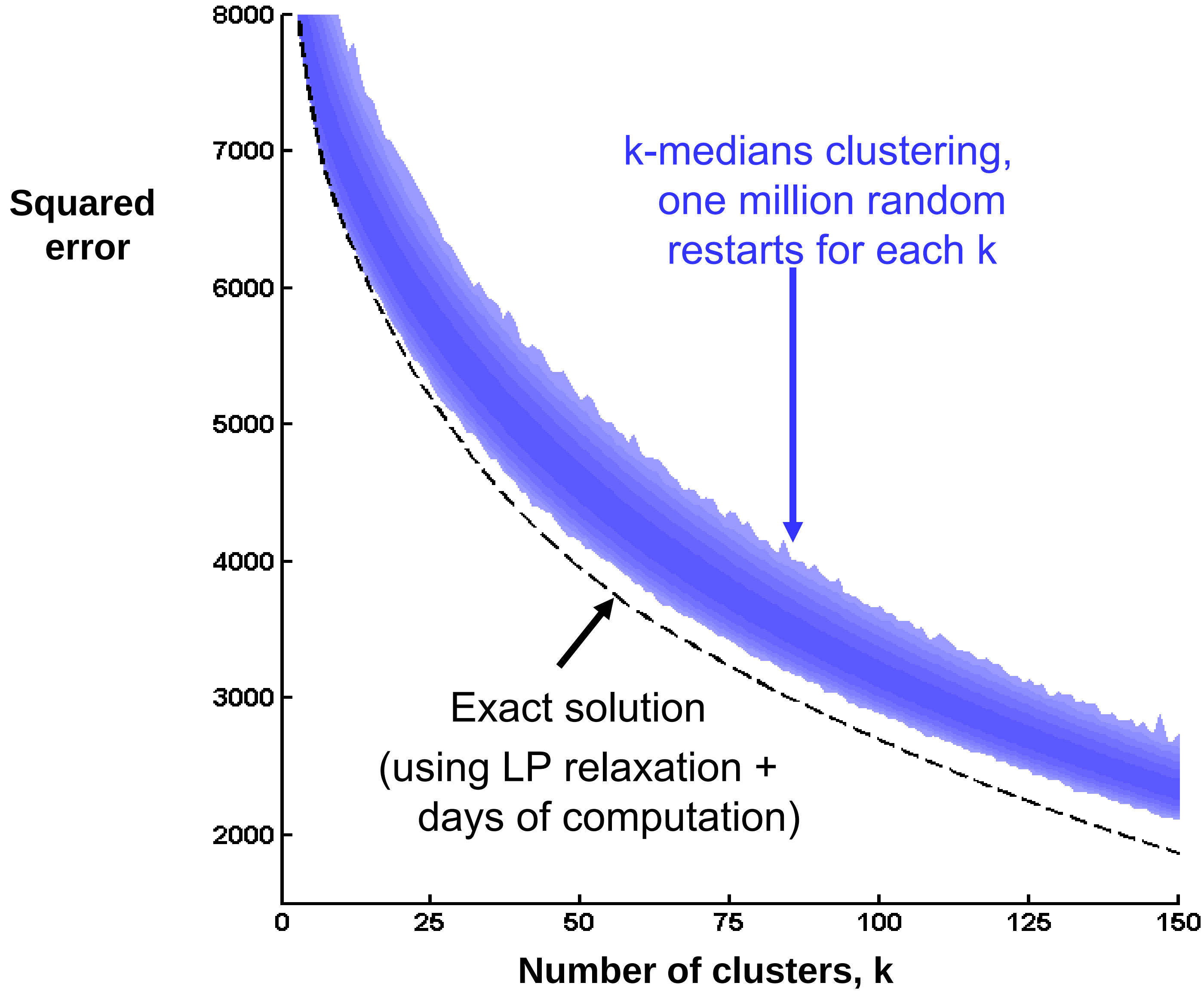
Recall solution for toy problem



Optimal solution
(minimizes sum of squared errors)



Squared error achieved by 1 million runs of *k*-medians clustering on 400 Olivetti face images



Let’s close the gap!

Source: MSNBC

Affinity Propagation

Science, Feb 16, 2007 and Feb 26, 2008

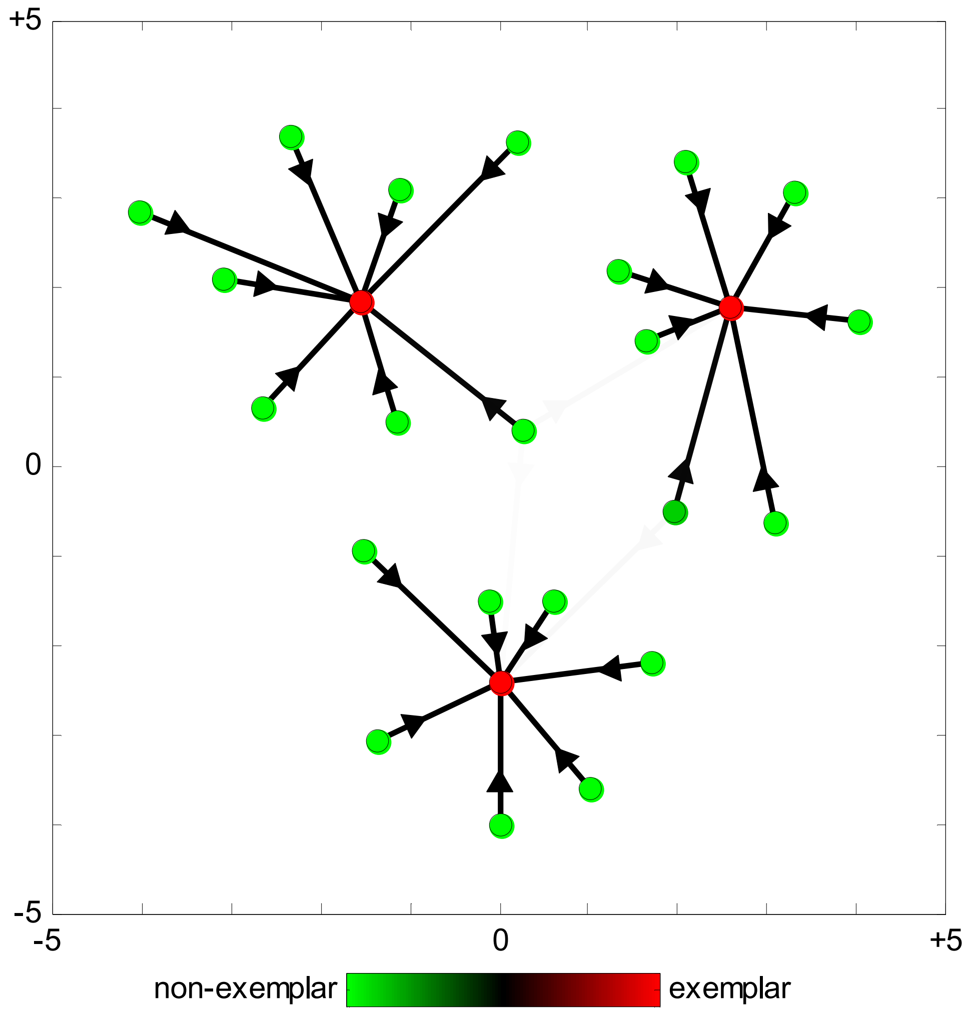
Joint work with Delbert Dueck

One-sentence summary:

All data points are simultaneously considered as exemplars, but exchange deterministic messages until a good set of exemplars gradually emerges

Demonstration of affinity propagation

ITERATION #15



Input to affinity propagation

- A set of pair-wise **similarities** $\{ s(i,k) \}$:
  $s(i,k)$ is a real number indicating how well-suited data point k is as an exemplar for point i
  - Example: $s(i,k) = - \| \mathbf{x}_i - \mathbf{x}_k \|^2, i \neq k$

Need not be metric

- For each data point k, a real number $s(k,k)$ indicating the a priori **preference** that it be chosen as an exemplar
  - Example: $s(k,k) = \text{median } \{s(i,j)\}$

An objective function for clustering

- Variables: For data points indexed $1, \ldots, N$, c_{ik} indicates whether ($c_{ik} = 1$) or not ($c_{ik} = 0$) point k is the exemplar of point i
- $c_{\boldsymbol{kk}} = 1$ indicates that data point $\boldsymbol{k}$ is an exemplar
- Exact clustering: Find a "valid" configuration of $\mathbf{c}$ that maximizes $\sum_{ik} c_{ik}\, s(i,k)$
  - Note that the number of clusters emerges automatically, due to the preferences, $s(k,k)$
  - This problem is NP hard (Megiddo & Supowit, 1984)

An objective function for clustering

Iverson's notation:
[True]=1, [False]=0

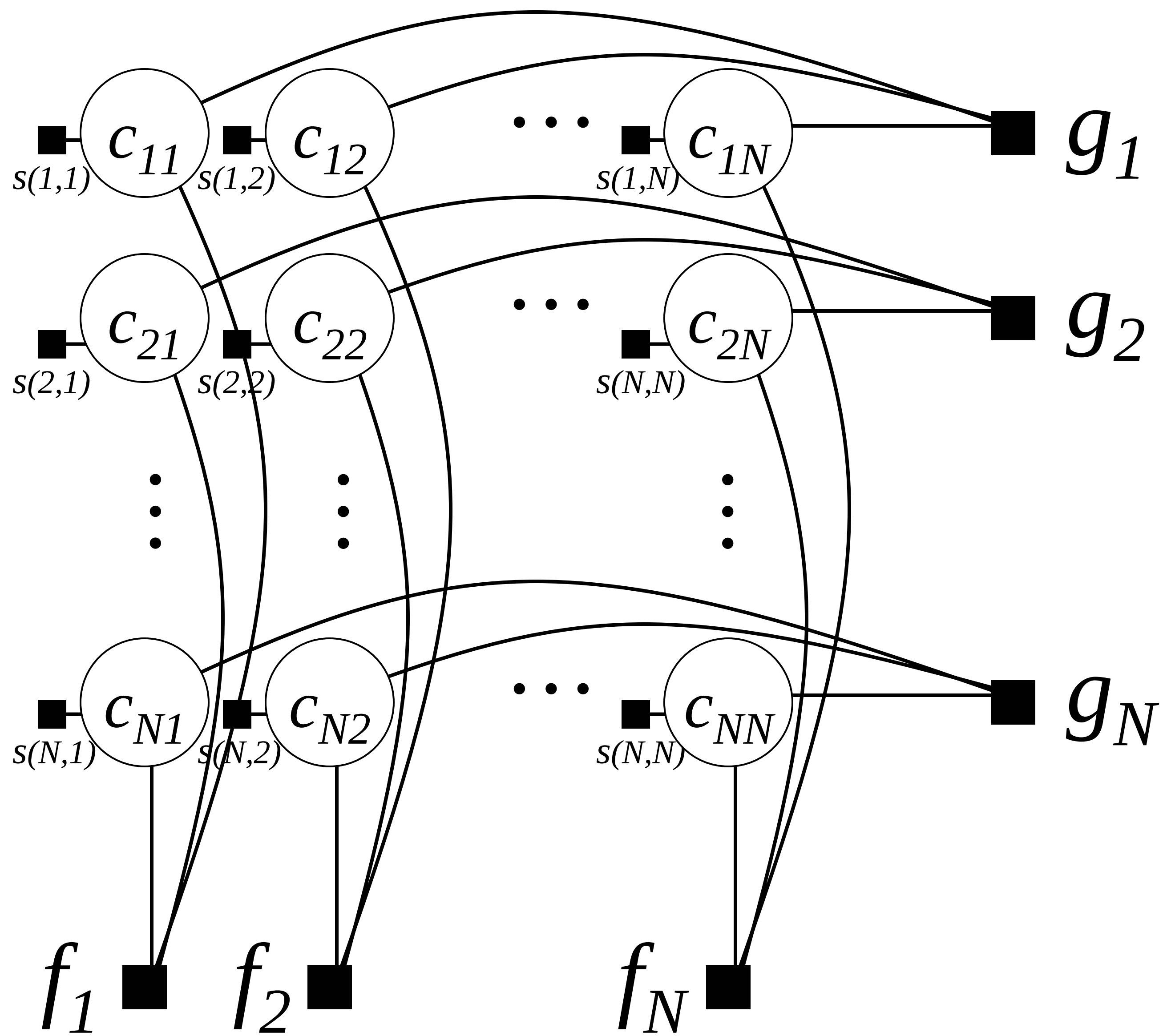
$$\text{NetSimilarity}(\mathbf{c}) =$$

$$\Sigma_{ik}\, c_{ik}\, s(i,k)\ -\ \underset{\alpha \to \infty}{\alpha}\Sigma_k\, \underbrace{(1\text{-}c_{kk})[\Sigma_i c_{ik} > 0])}_{f_k(c_{1k},\ldots,\, c_{Nk})}\ -\ \alpha\ \Sigma_i\, \underbrace{[\Sigma_j c_{ij} \neq 1]}_{g_i(c_{i1},\ldots,\, c_{iN})}$$

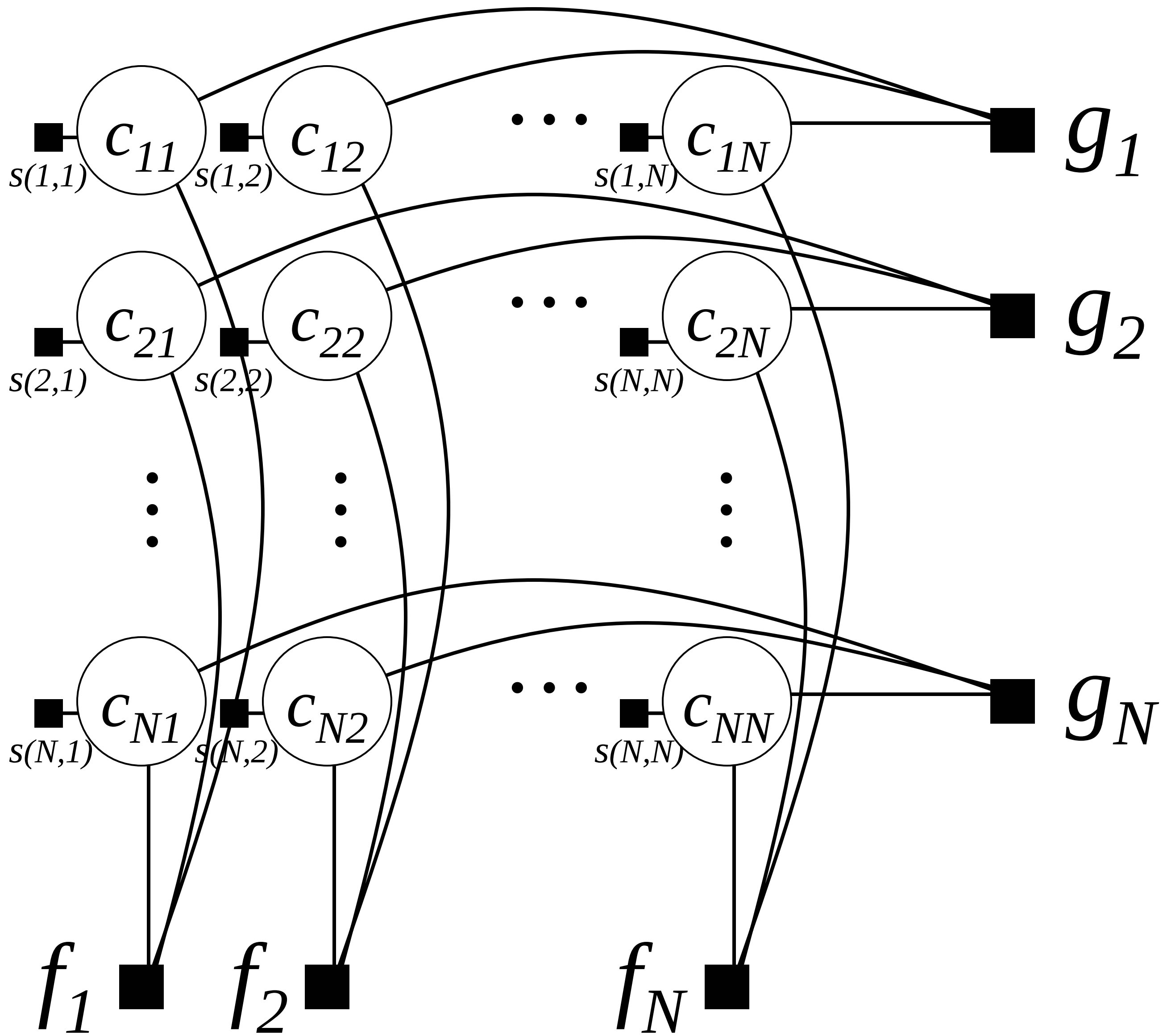
Penalty for having a cluster without an exemplar

1-of-N: Penalty for a point being assigned to more than one cluster

A factor graph describing NetSimilarity(**c**)



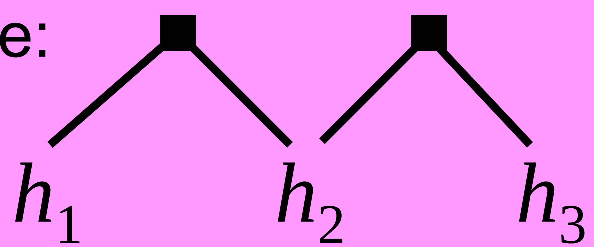
Affinity propagation: The loopy max-sum algorithm is used to approximately maximize the objective function



Mini-Tutorial:
Factor Graphs and the sum-product algorithm

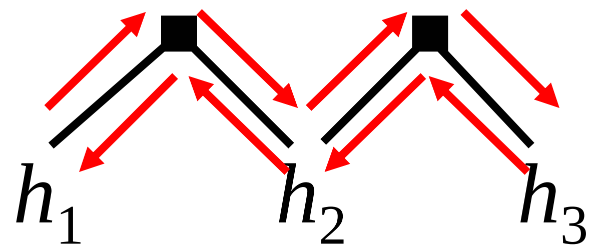
Representing problems using factor graphs

- Many problems require finding the values of variables $\mathbf{h}$ that maximize an objective function of the form $\mathrm{F}(\mathbf{h}) = \Sigma_s f_s(\mathbf{h}_s)$, or $\mathrm{P}(\mathbf{h}) = \Pi_s p_s(\mathbf{h}_s)$
  - Example: $\mathrm{F}(h_1,h_2,h_3) = -(h_1-h_2)^2-(h_2-h_3)^2$
    $f_1(h_1,h_2) = -(h_1-h_2)^2, \quad f_2(h_2,h_3) = -(h_2-h_3)^2$
- A factor graph is a graph with two types of node:
  - Each variable node corresponds to a variable in $\mathbf{h}$
  - Each function node corresponds to a local function $f_s()$ or $p_s()$, and is connected to all variables in the function's argument
  - Example:



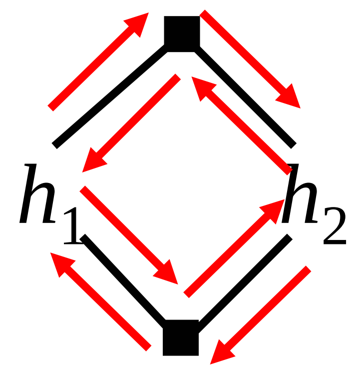
Solving problems using the max-product or sum-product algorithm

- Kalman filtering, the Viterbi algorithm and dynamic programming can be thought of as <u>message-passing</u> in a factor graph



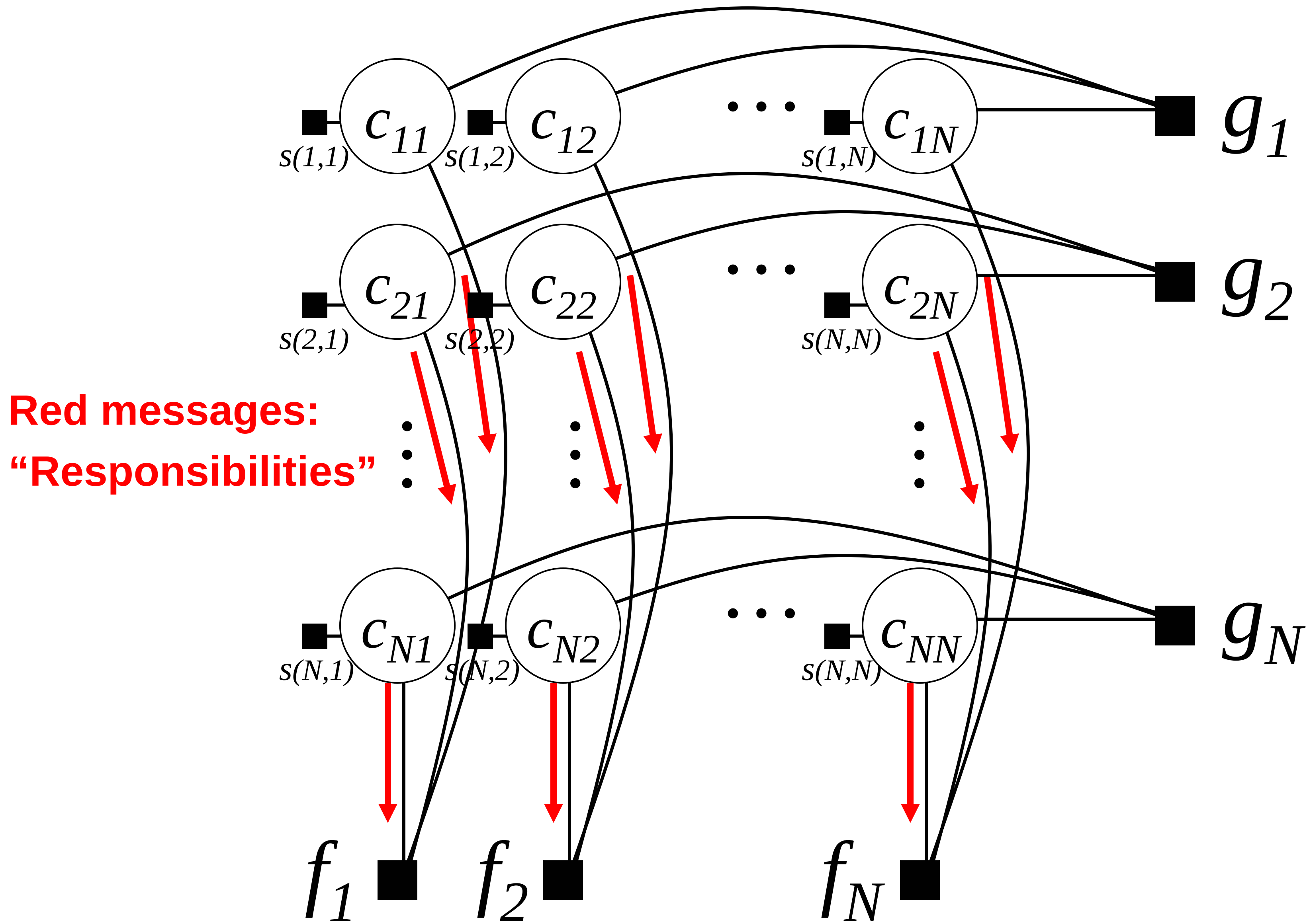
- The <u>max-product (or sum-product) algorithm</u> exactly maximizes $F(\mathbf{h})$ (or marginalizes $P(\mathbf{h})$) if the factor graph is a tree
  - Other names: Belief revision or propagation

- Both algorithms are "only" approximate if the graph has cycles and messages circulate around the graph until convergence

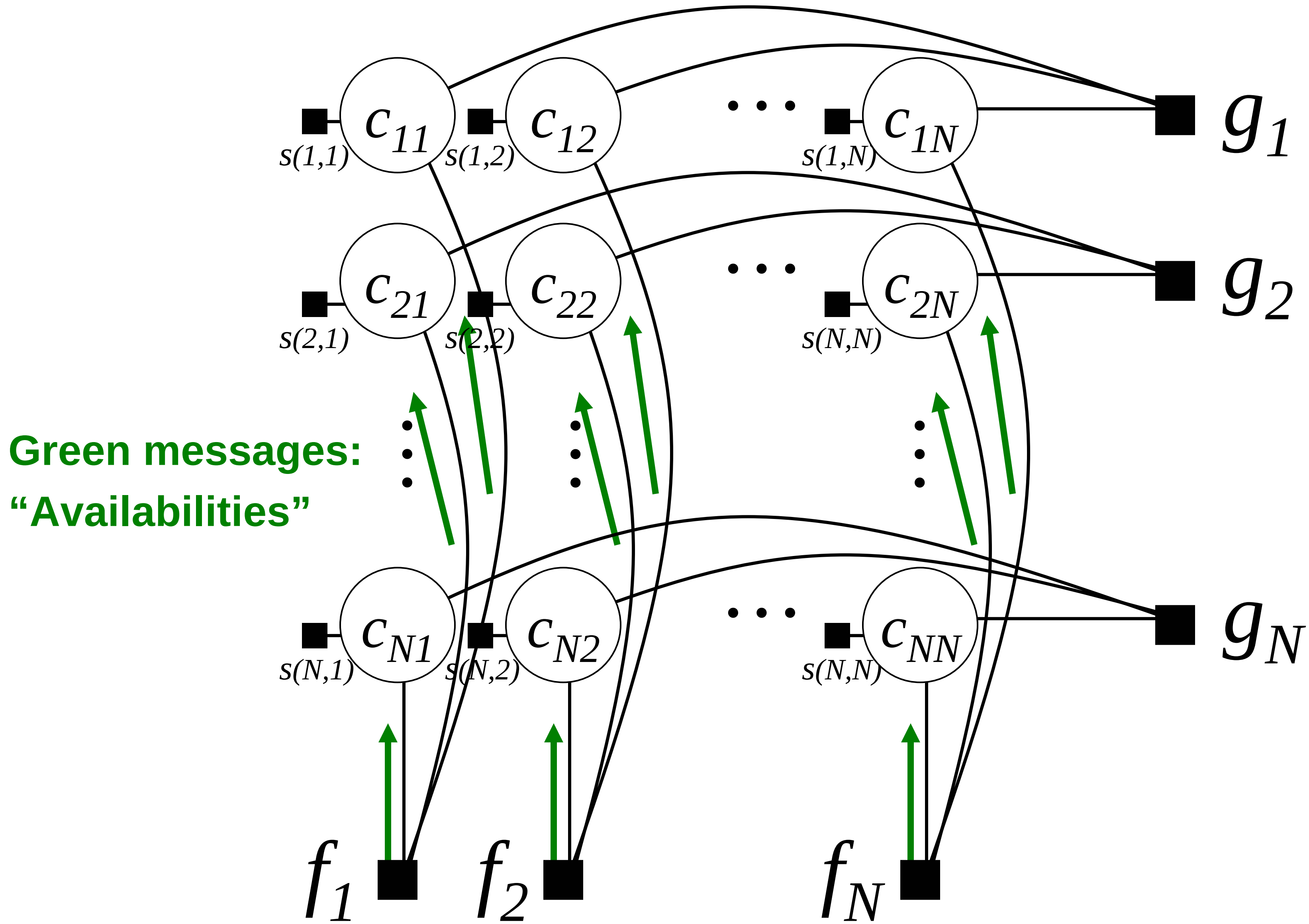


Now, back to affinity propagation…

Affinity propagation: The loopy max-sum algorithm is used to approximately maximize the objective function



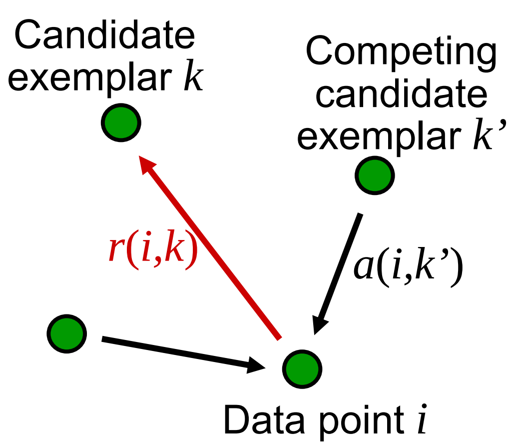
Affinity propagation: The loopy max-sum algorithm is used to approximately maximize the objective function



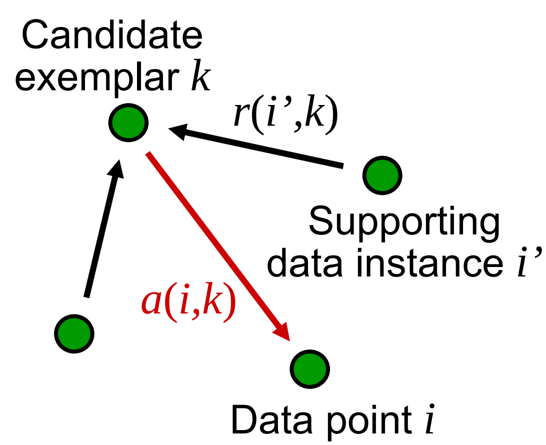
The simple picture:

Affinity propagation can be viewed as exchanging messages between the data points themselves

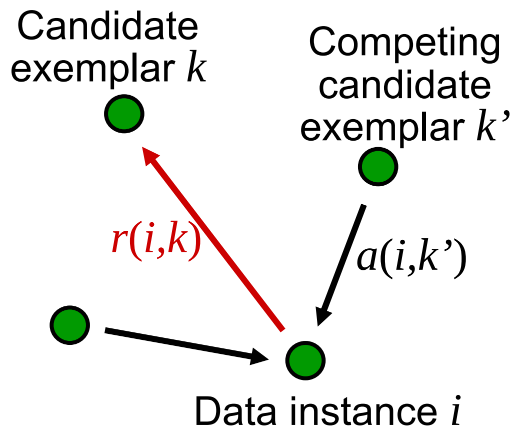
Sending responsibilities



Sending availabilities

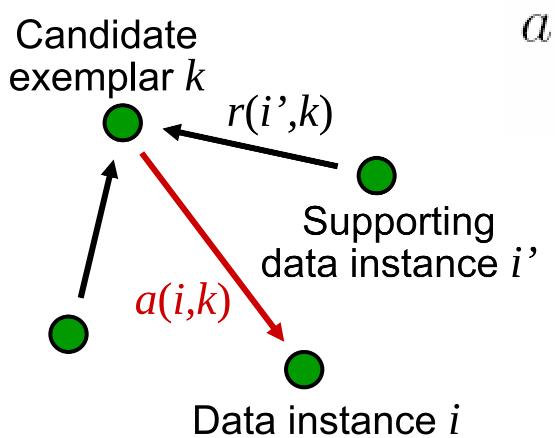


Sending responsibilities



$$r(i,k) \leftarrow s(i,k) - \max_{k' \neq k} \{a(i,k') + s(i,k')\}$$

Sending availabilities



$$a(i,k) \leftarrow \min\Big\{0\ ,\ r(k,k) + \sum_{i' \neq i,k} \max\{0, r(i',k)\}\Big\}$$

$$a(k,k) \leftarrow \sum_{i' \neq k} \max\{0, r(i',k)\}$$

Making decisions:

$$\mathrm{argmax}_k \{a(i,k) + r(i,k)\}$$

Message damping

- Unstable dynamics are always avoided in practice by damping messages:

$$r(i,k)^* = \lambda\, r(i,k) + (1- \lambda)\, r(i,k)^{old}$$

$$a(i,k)^* = \lambda\, a(i,k) + (1- \lambda)\, a(i,k)_{old}$$

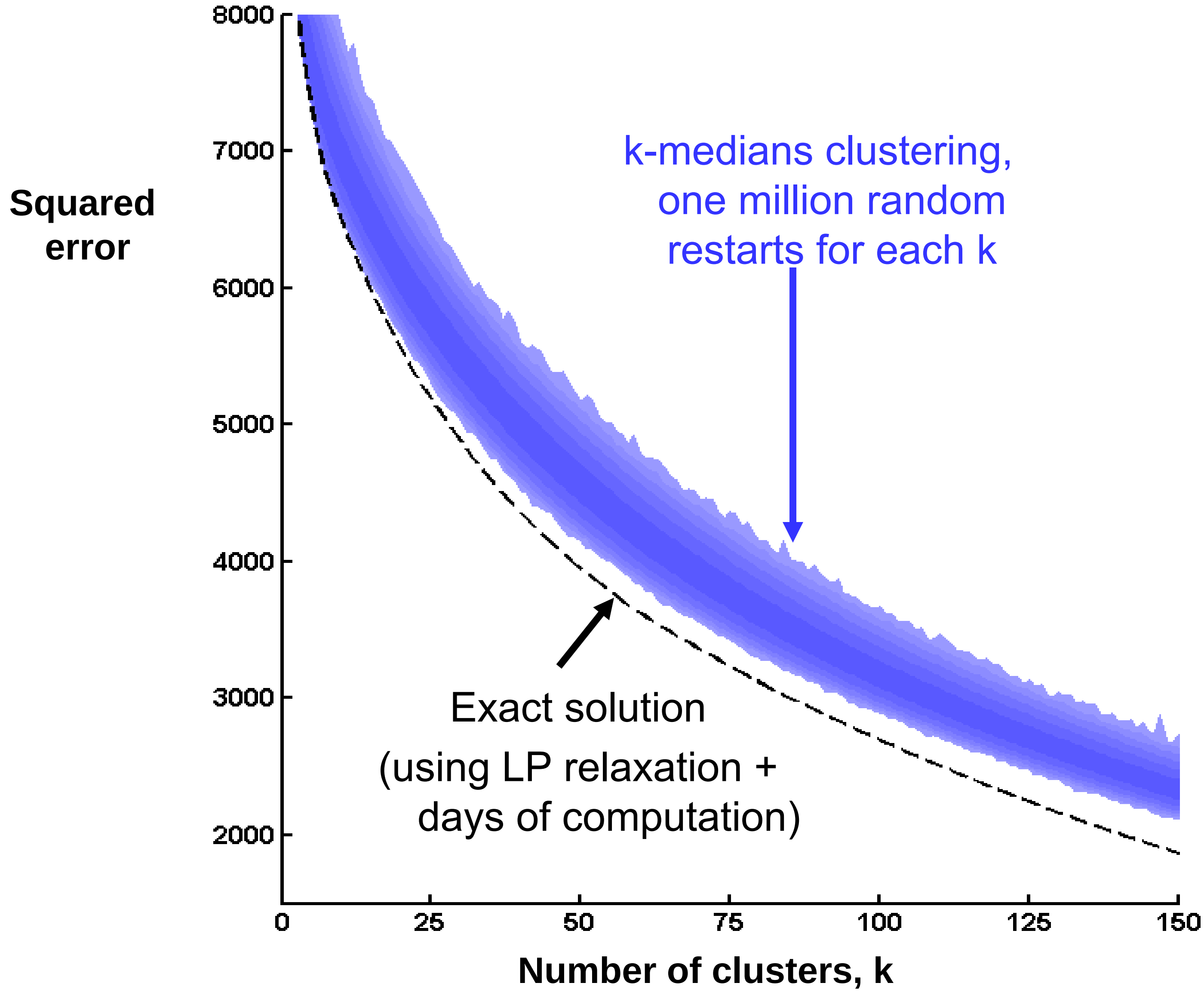
- Default: $\lambda = 0.9$

MATLAB implementation

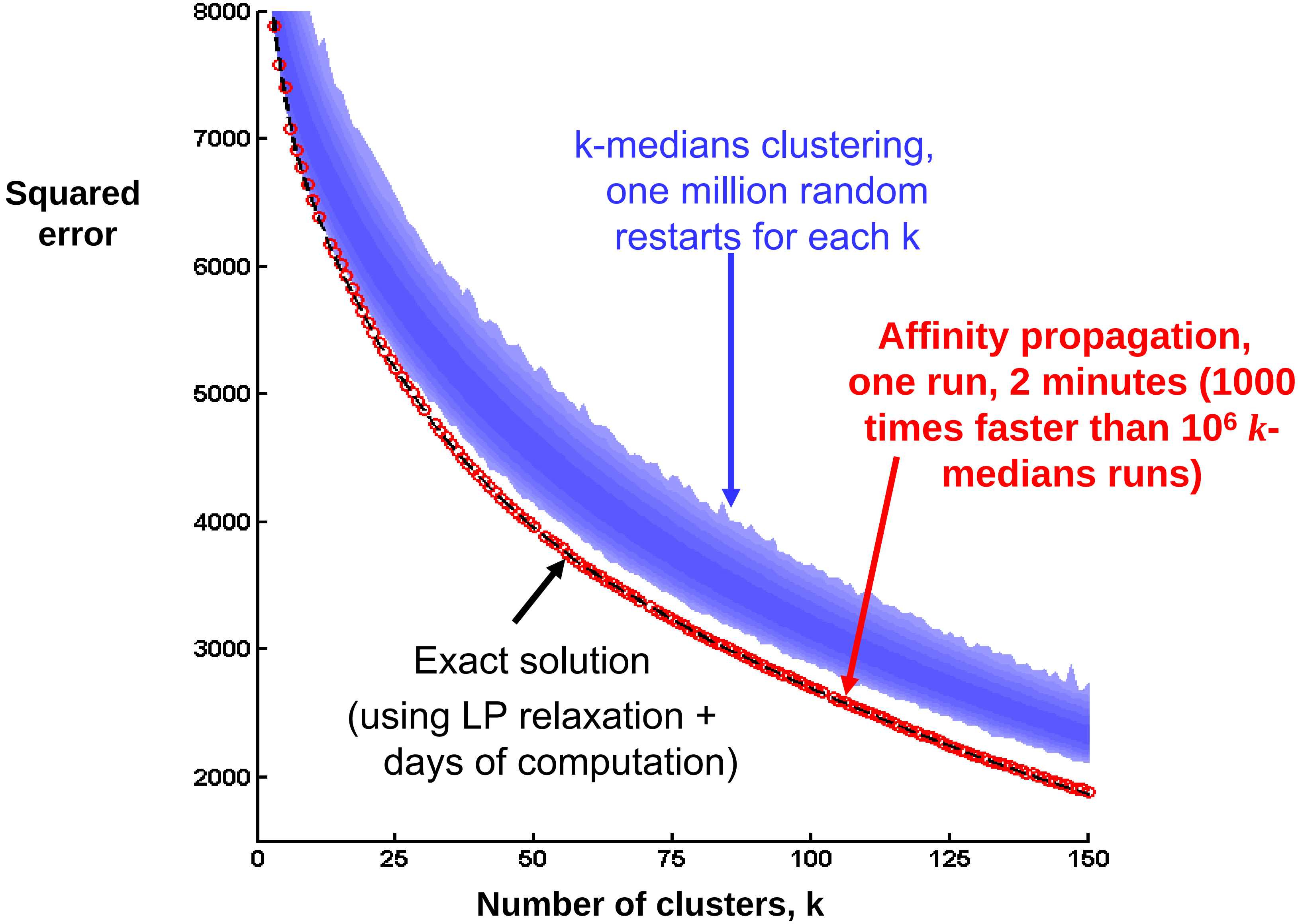
```
n=size(S,1);
A=zeros(n,n); % Initialize availabilities to 0
lambda=0.5; % Dampening factor
for iter=1:100
    % Compute responsibilities
    Rold=R;
    AS=A+S; [Y,I]=max(AS,[],2);
    for i=1:n AS(i,I(i))=-realmax; end;
    [Y2,I2]=max(AS,[],2);
    R=S-repmat(Y,[1,n]);
    for i=1:n R(i,I(i))=S(i,I(i))-Y2(i); end;
    R=(1-lambda)*R+lambda*Rold;  % Dampen responsibilities

    % Compute availabilities
    Aold=A;
    Rp=max(R,0);
    for k=1:n Rp(k,k)=R(k,k); end;
    A=repmat(sum(Rp,1),[n,1])-Rp;
    dA=diag(A); A=min(A,0); for k=1:n A(k,k)=dA(k); end;
    A=(1-lambda)*A+lambda*Aold;  % Dampen availabilities
end;
E=R+A; % Pseudomarginals
idx=find(diag(E)>0); % Indices of exemplars
K=length(idx); % Number of detected exemplars
[tmp ass]=max(S(:,idx),[],2); ass(idx)=1:K; % Assignments
```

Squared error achieved by 1 million runs of k-medians clustering on 400 Olivetti face images

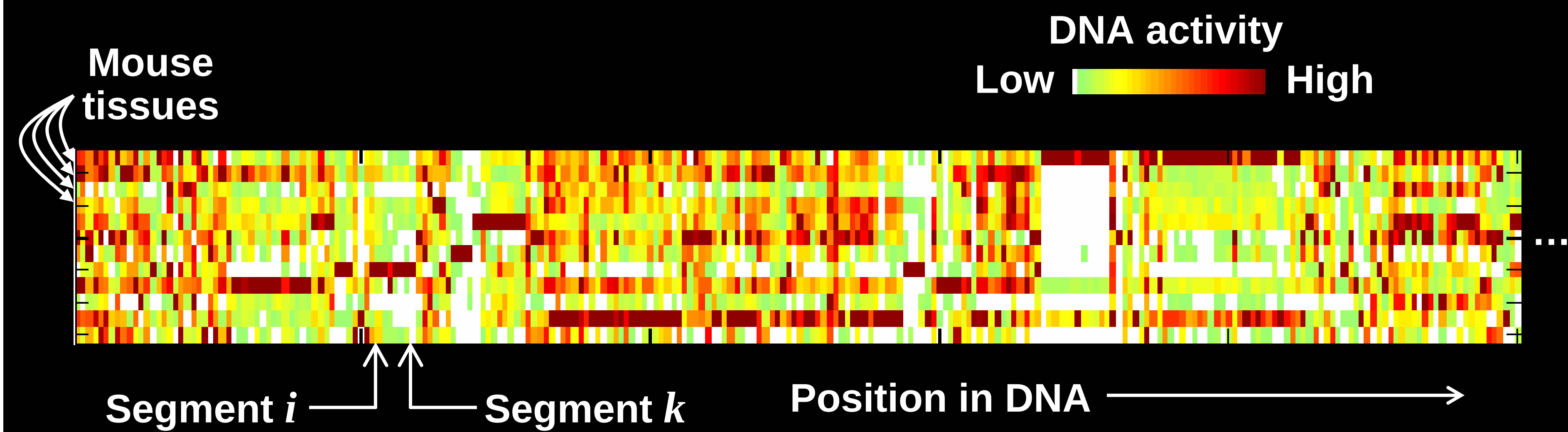


Squared error achieved by affinity propagation on 400 Olivetti face images



Detecting transcripts (genes) using microarray data

(Data from Frey et al, Nature Genetics 2005, Science 2006)

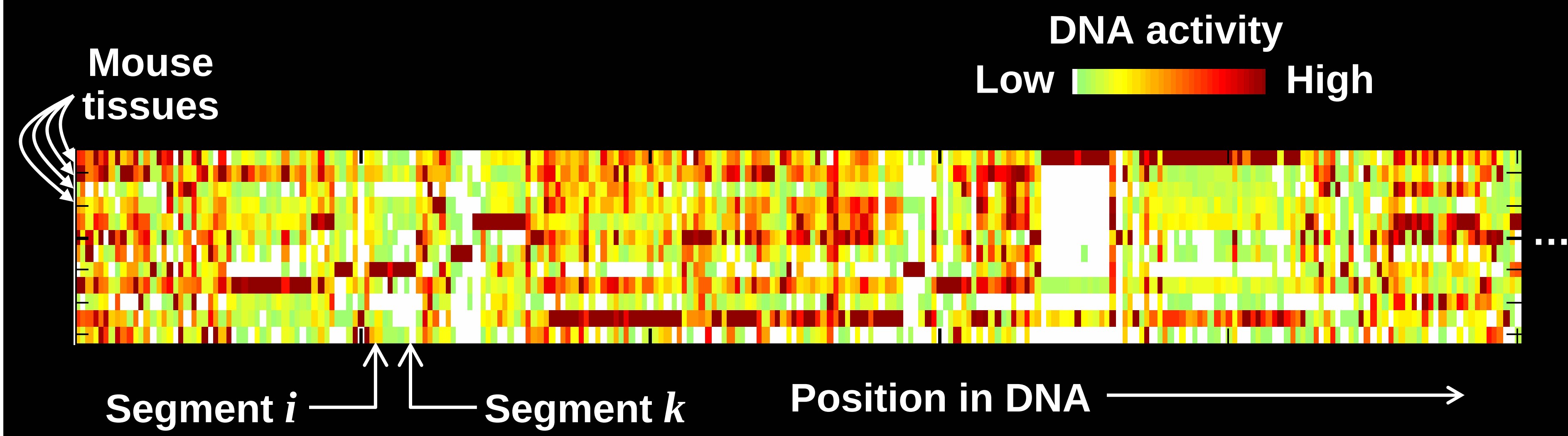


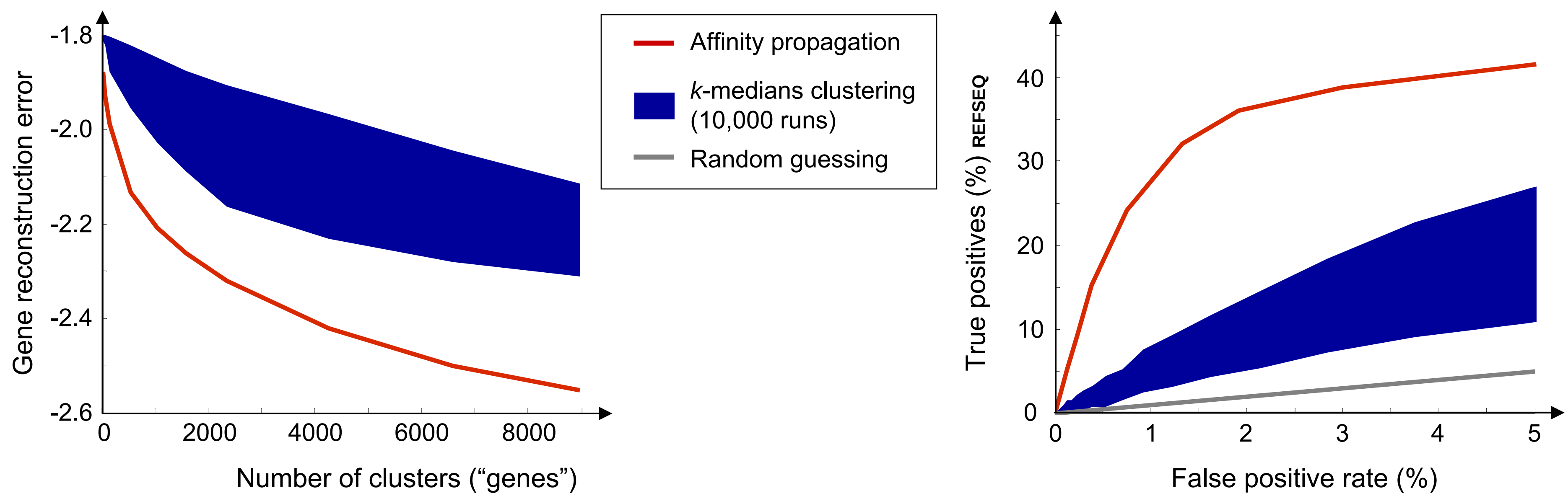
s(segment i, segment k) = Similarity of expression patterns (columns) minus distance between segments in the DNA/genome

s(segment i, garbage) = tunable constant

segments = 76,000 for chromosome 1

Detecting transcripts (genes) using microarray data
(Data from Frey et al, Nature Genetics 2005, Science 2006)





A survey of applications investigated by other researchers and developers

- VQ codebook design, Jiang et al, 2007
- Image segmentation, Xiao et al, 2007
- Object classification, Fu et al, 2007
- Finding light sources using images, An et al, 2007
- Microarray analysis, Leone et al, 2007
- Computer network analysis, Code et al, 2007
- Audio-visual data analysis, Zhang et al, 2007
- Protein sequence analysis, Wittkop et al, 2007
- Protein clustering, Lees et al, 2007
- Analysis of cuticular hydrocarbons, Kent et al, 2007
- …

How competitive is affinity propagation?

In the past year…

- Researchers have compared affinity propagation to dozens of other clustering algorithms

Two contenders have emerged:

- Linear program relaxation of the binary integer program (Charikar et al, 2002):

$$\max_{\mathbf{c}} \textstyle\sum_{ik} c_{ik}\, s(i,k)$$

$$0 \le c_{ik} \le 1,\ \textstyle\sum_k c_{ik} \le 1,\ c_{ik} \le c_{kk}$$

Limitation:

Practical for only < 500 data points

- The vertex substitution heuristic, VSH (Hansen & Mladenovic, 1997)

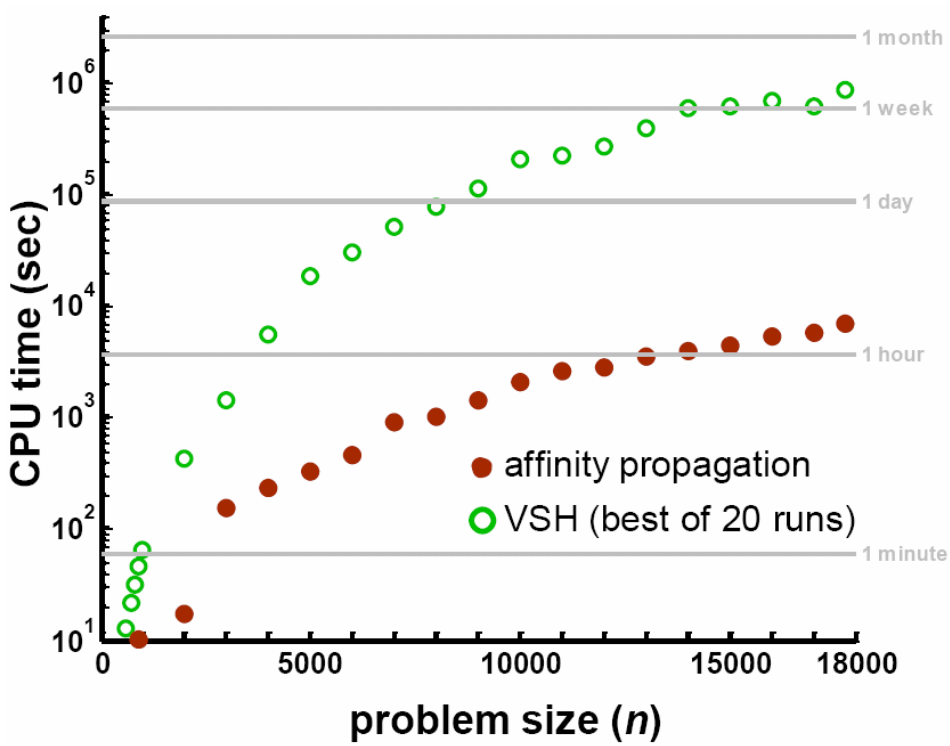
Error and timing comparison of affinity propagation and the VSH

(Results from Brusco & Kohn and Frey & Dueck)

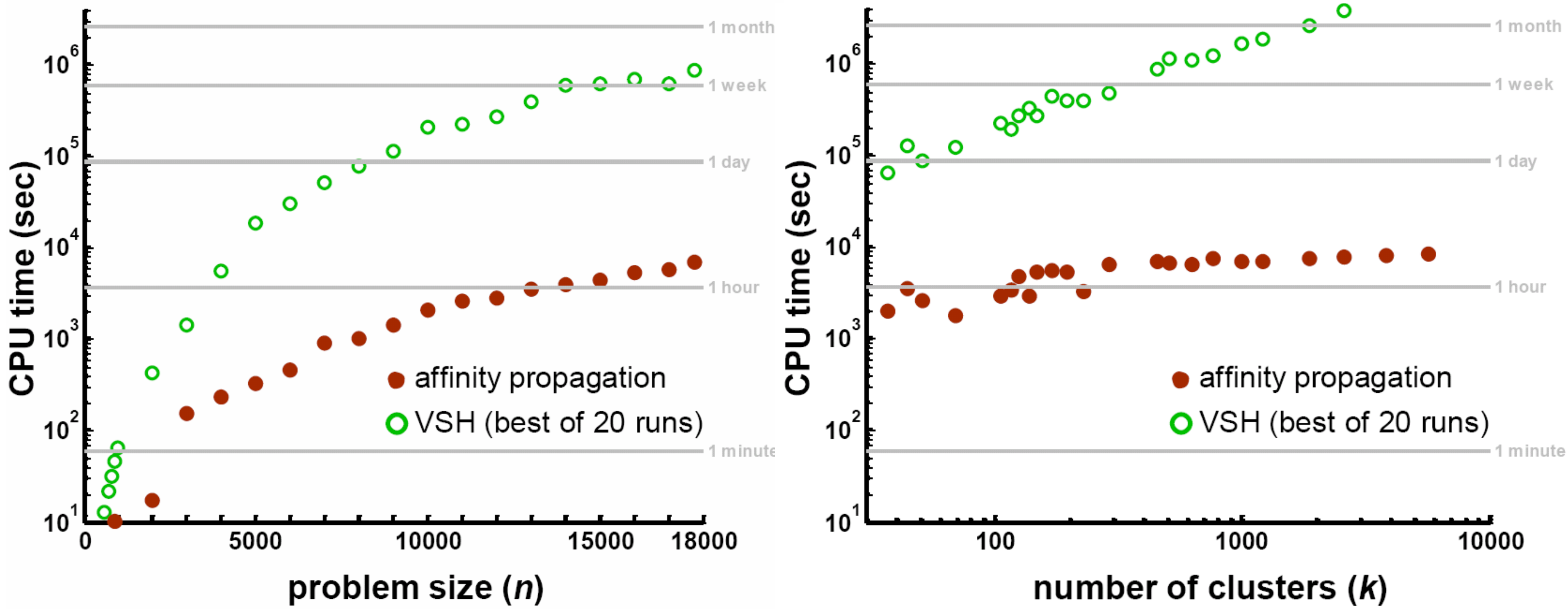
Problem	n	k	Relative Error		CPU Time VSH:AP
			AP (%)	VSH (%)	
Birth/death rates	70	6	0	0	0.46:1
Fisher's iris data	150	6	4.50	0	0.59:1
Circuit board	318	11	0.22	0	1.20:1
Random S	400	34	0	1.110	0.57:1
City coordinates	666	17	3.32	0	1.20:1
Olivetti images	900	62	0.44	0	2.79:1
Ext. circuit board	1,272	103	0	0.18	5.19:1
Face video	1,965	239	0	0.0021	10.84:1
Putative exons	10,000	542	0.0003	0	29.42:1
Gene expression	10,000	566	0	0	86.07:1
Netflix movies	17,770	454	0	0.019	123.09:1

>900 data points

Timing comparison of affinity propagation and the VSH on 17,770 Netflix movies



Timing comparison of affinity propagation and the VSH on 17,770 Netflix movies



Closing remarks:
Open problems

Relationship to Dirichlet process mixture models? (Blei & Jordan, Jain & Neal, Teh & Welling)

- Dirichlet process mixture models use a Dirichlet prior on the mixing weights of an infinite number of clusters

- Affinity propagation can be viewed as MAP inference of a Dirichlet mixture model where the means are constrained to be on data points and variances are fixed

  (Tarlow, Zemel and Frey, to appear at UAI)

Open problem:
Guarantees on solutions

Separability Theorem (Weak)

If the data set can be partitioned into classes so that the minimum and maximum within-class similarities ($s_{\min}^{\mathrm{wc}}$ and $s_{\max}^{\mathrm{wc}}$) and maximum between-class similarity ($s_{\max}^{\mathrm{bc}}$) satisfy

$$s_{\min}^{\mathrm{wc}} - s_{\max}^{\mathrm{bc}} > 2(s_{\max}^{\mathrm{wc}} - s_{\min}^{\mathrm{wc}}), \tag{7}$$

then running affinity propagation on the entire set of similarities using a preference p satisfying

$$p \geq p_{\mathrm{sep}} = \frac{1}{2}(s_{\max}^{\mathrm{bc}} + s_{\min}^{\mathrm{wc}}) \tag{8}$$

is equivalent to running affinity propagation separately on each class of data, regardless of whether or not damping is used.

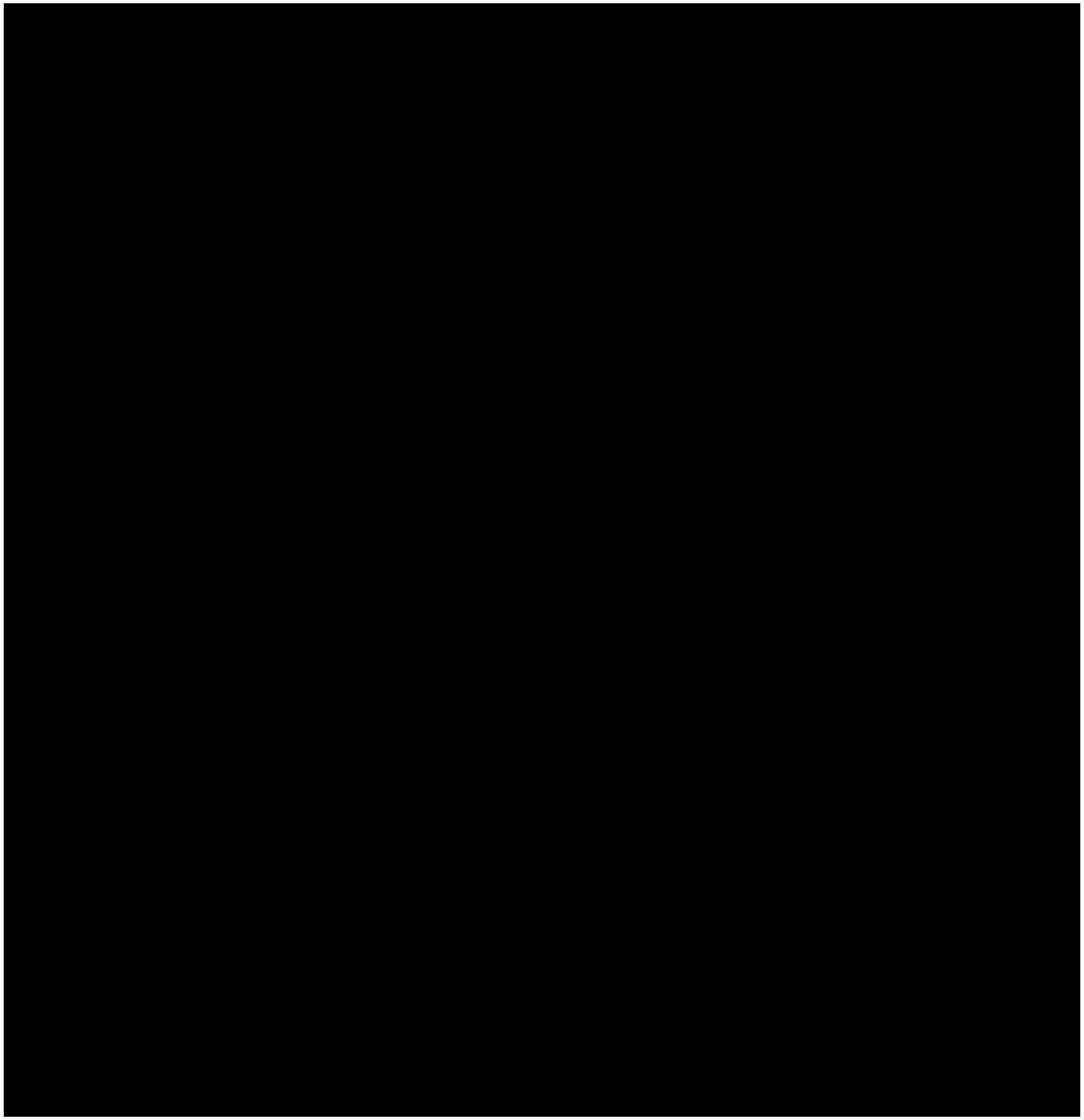
A relevant result?

On the problem of weighted matching on graphs, “if the LP relaxation is tight, ie, if the unique solution is integral, then the max-sum algorithm converges and the resulting estimate is the optimal matching”

– Sujay Sanghavi, Dimitry Malioutov, Alan Willsky, NIPS 2007

Open problem: Extensions

Facility location
(Dueck et al, RECOMB 2008)



- Here, potential exemplars are laid out on a fine grid

- Does “generalized belief propagation” (Yedidia et al; Yuille) produce different results?
- How about tree-reweighted belief propagation?
- Extensions to multiple hidden-variables?

Software, data, and comparisons available at

http://www.psi.toronto.edu/affinitypropagation