K-Means

lecturer: Jiří Matas, `matas@cmp.felk.cvut.cz`

authors: J. Matas, O. Drbohlav

Czech Technical University, Faculty of Electrical Engineering
Department of Cybernetics, Center for Machine Perception
121 35 Praha 2, Karlovo nám. 13, Czech Republic

http://cmp.felk.cvut.cz

4/Dec/2015

Last update: 7/Dec/2015, 11am

LECTURE PLAN

- Least squares clustering problem statement
- K-means, algorithm, properties
- Initialization by K-means++
- Related methods

Formulation of the Least-Squares Clustering Problem

Given:

$\mathcal{T} = \{\mathbf{x}_l\}_{l=1}^{L}$	the set of observations
K	the desired number of cluster prototypes

Output:

$\{\mathbf{c}_k\}_{k=1}^{K}$	the set of cluster prototypes (etalons)
$\{\mathcal{T}_k\}_{k=1}^{K}$	the clustering (partitioning) of the data
	$\cup_{k=1}^{K} \mathcal{T}_k = \mathcal{T}, \mathcal{T}_i \cap \mathcal{T}_j =$ for $i \neq j$

The result is obtained by solving the following optimization problem:

$$(\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_K; \mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_K) = \underset{\text{all } \mathbf{c}'_k, \mathcal{T}'_k}{\operatorname{argmin}} J(\mathbf{c}'_1, \mathbf{c}'_2, ..., \mathbf{c}'_K; \mathcal{T}'_1, \mathcal{T}'_2, ..., \mathcal{T}'_K) \tag{1}$$

where

$$J(\mathbf{c}'_1, \mathbf{c}'_2, ..., \mathbf{c}'_K; \mathcal{T}'_1, \mathcal{T}'_2, ..., \mathcal{T}'_K) = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{T}'_k} \|\mathbf{x} - \mathbf{c}'_k\|^2 \,. \tag{2}$$

Formulation of the Least-Squares Clustering Problem

Given:

$\mathcal{T} = \{\mathbf{x}_l\}_{l=1}^{L}$	the set of observations
K	the desired number of cluster prototypes

Output:

$\{\mathbf{c}_k\}_{k=1}^{K}$	the set of cluster prototypes (etalons)
$\{\mathcal{T}_k\}_{k=1}^{K}$	the clustering (partitioning) of the data
	$\cup_{k=1}^{K}\mathcal{T}_k = \mathcal{T}, \mathcal{T}_i \cap \mathcal{T}_j =$ for $i \neq j$

Note that this problem can be equivalently rewritten as an optimization in cluster centres only:

$$(\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_K) = \operatorname*{argmin}_{\{\mathbf{c}'_k\}_{k=1}^{K}} J(\mathbf{c}'_1, \mathbf{c}'_2, ..., \mathbf{c}'_K)\,, \tag{3}$$

$$J(\mathbf{c}'_1, \mathbf{c}'_2, ..., \mathbf{c}'_K) = \sum_{l=1}^{L} \min_{k \in \{1,2,...,K\}} \|\mathbf{x}_l - \mathbf{c}'_k\|^2\,, \tag{4}$$

with $\mathcal{T}_k$'s then computed as

$$\mathcal{T}_k = \{\mathbf{x} \in \mathcal{T} : \forall j, \|\mathbf{x} - \mathbf{c}_k\|^2 \leq \|\mathbf{x} - \mathbf{c}_j\|^2\} \qquad (\forall k = 1, 2, ..., K)\,. \tag{5}$$

K-Means: An Algorithm for the LS Clustering Problem

Given:

$\mathcal{T} = \{\mathbf{x}_l\}_{l=1}^{L}$	the set of observations
K	the desired number of cluster prototypes

Output:

$\{\mathbf{c}_k\}_{k=1}^{K}$	the set of cluster prototypes (etalons)
$\{\mathcal{T}_k\}_{k=1}^{K}$	the clustering (partitioning) of the data
	$\cup_{k=1}^{K} \mathcal{T}_k = \mathcal{T}, \mathcal{T}_i \cap \mathcal{T}_j =$ for $i \neq j$

K-Means Algorithm:

1. Initialize the cluster centres $\{\mathbf{c}_k\}_{k=1}^{K}$ (e.g. by random selection from the data points $\mathcal{T}$, without replacement)
2. Assignment optimization (assign to closest etalon):

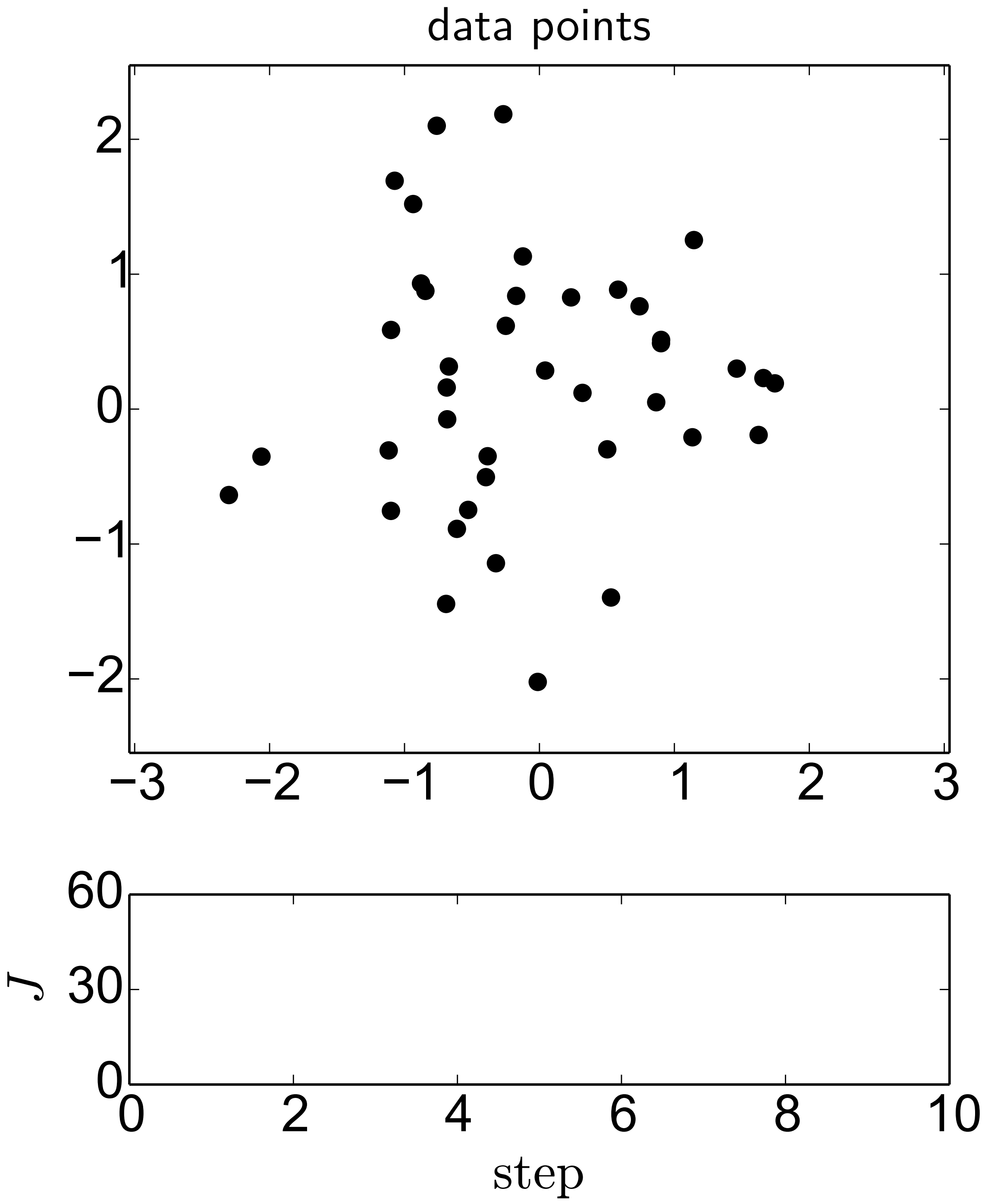
$$\mathcal{T}_k = \{\mathbf{x} \in \mathcal{T} : \forall j, \|\mathbf{x} - \mathbf{c}_k\|^2 \leq \|\mathbf{x} - \mathbf{c}_j\|^2\} \qquad (\forall k = 1, 2, ..., K) \tag{6}$$

3. Prototype optimization (updated etalon is the mean of data assigned to it):

$$\mathbf{c}_k = \begin{cases} \dfrac{1}{|\mathcal{T}_k|} \displaystyle\sum_{\mathbf{x} \in \mathcal{T}_k} \mathbf{x} & \text{if } |\mathcal{T}_k| > 0 \\ \text{re-initialize} & \text{if } \mathcal{T}_k = \emptyset \end{cases} \qquad (\forall k = 1, 2, ..., K) \tag{7}$$

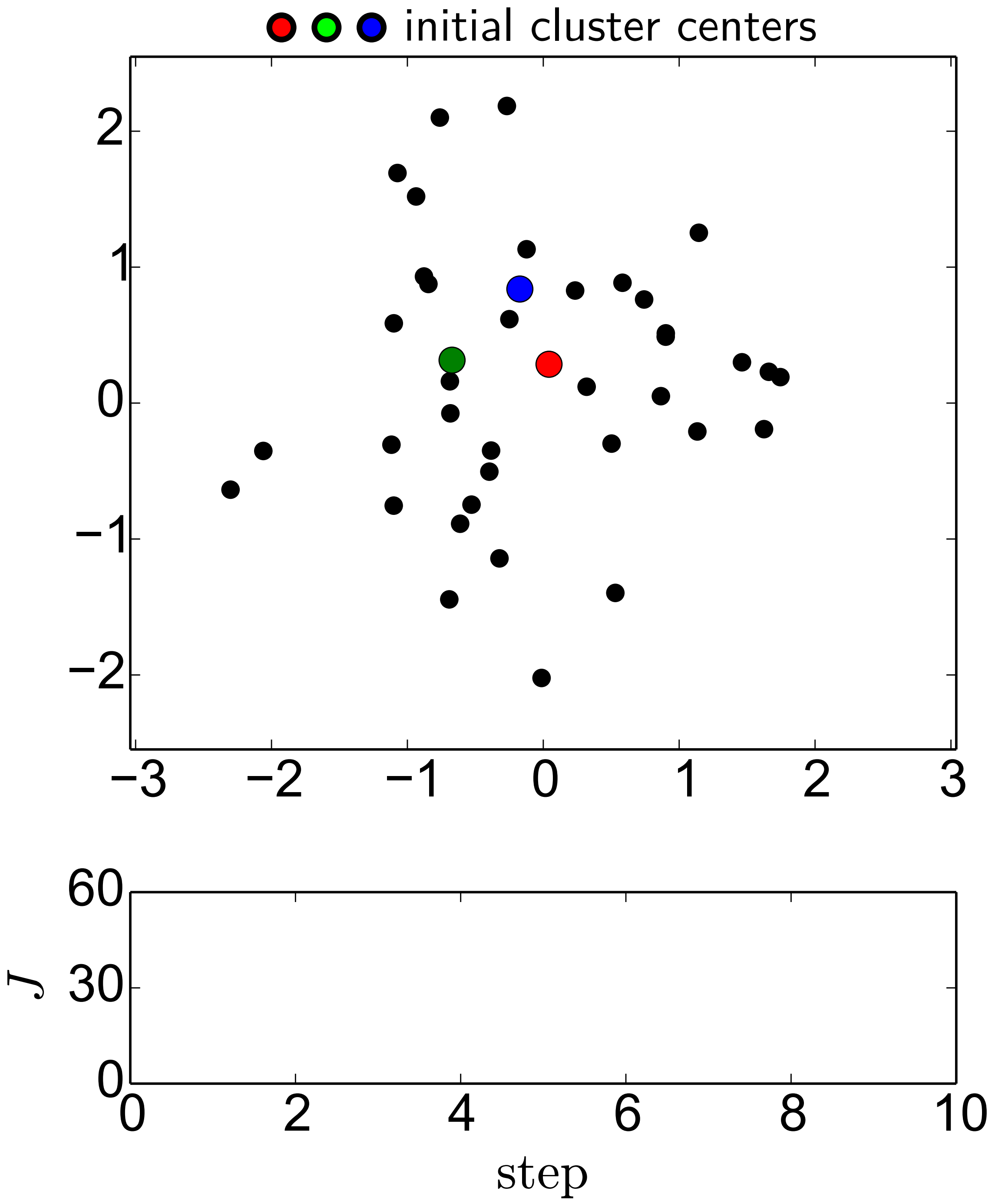
4. Terminate if $\forall k : \mathcal{T}_k^{t+1} = \mathcal{T}_k^{t}$, otherwise goto 2

K-Means: Example



Cluster the data points to $K = 3$ clusters

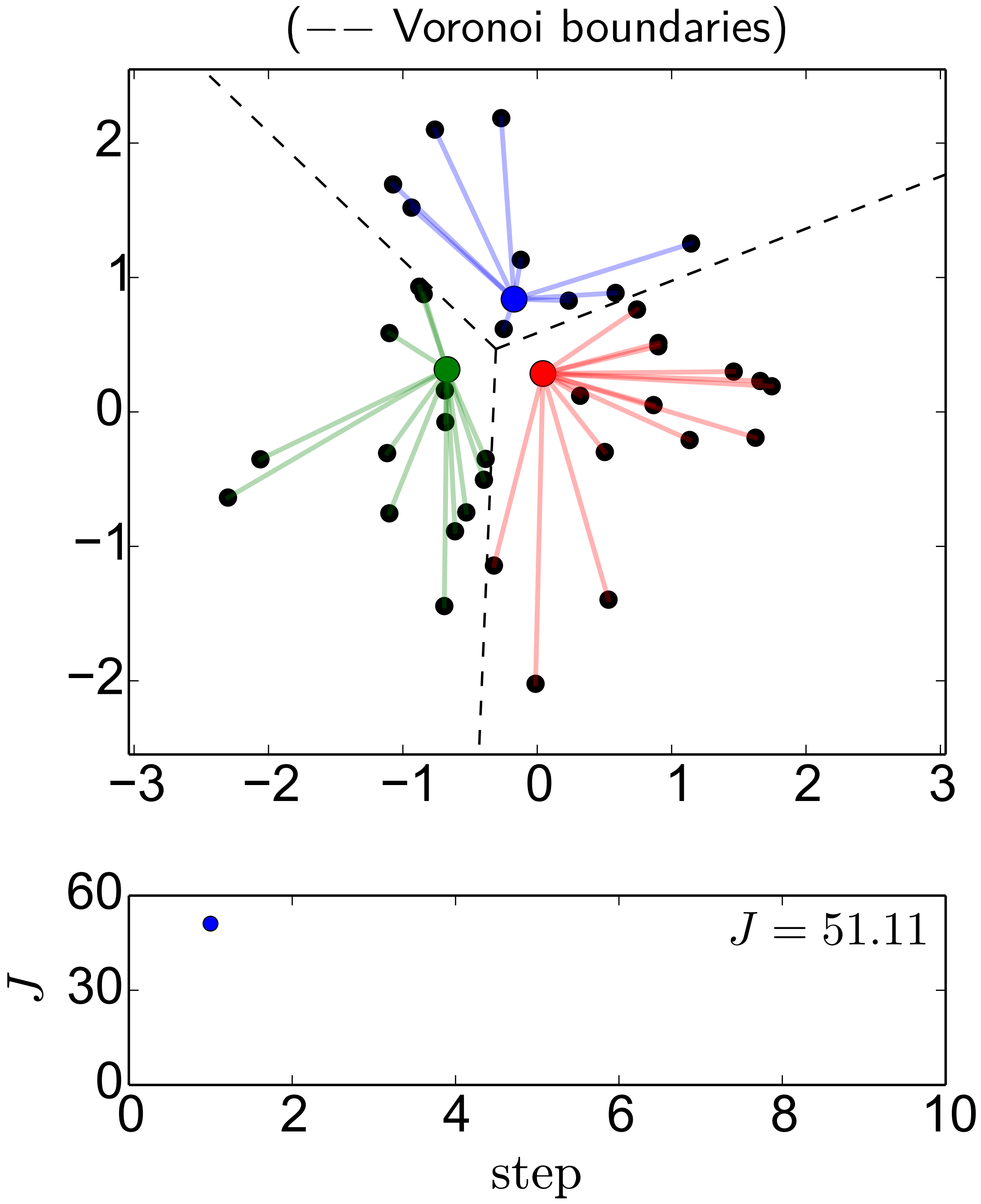
K-Means: Example



Cluster the data points to $K = 3$ clusters

Initial cluster centers (here selected randomly from data points)

K-Means: Example

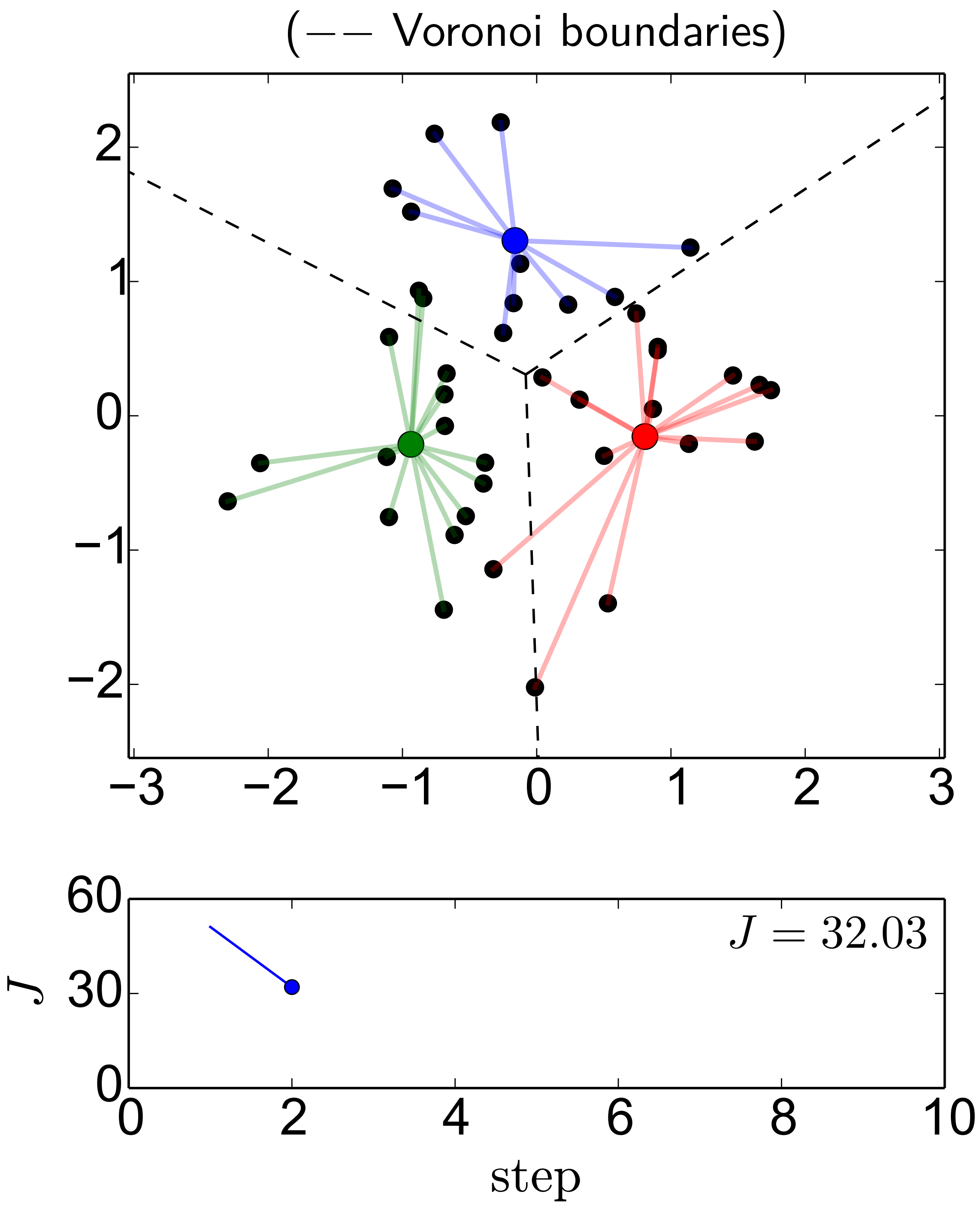


Cluster the data points to $K = 3$ clusters

Initial cluster centers (here selected randomly from data points)

step 1, compute assignments

K-Means: Example

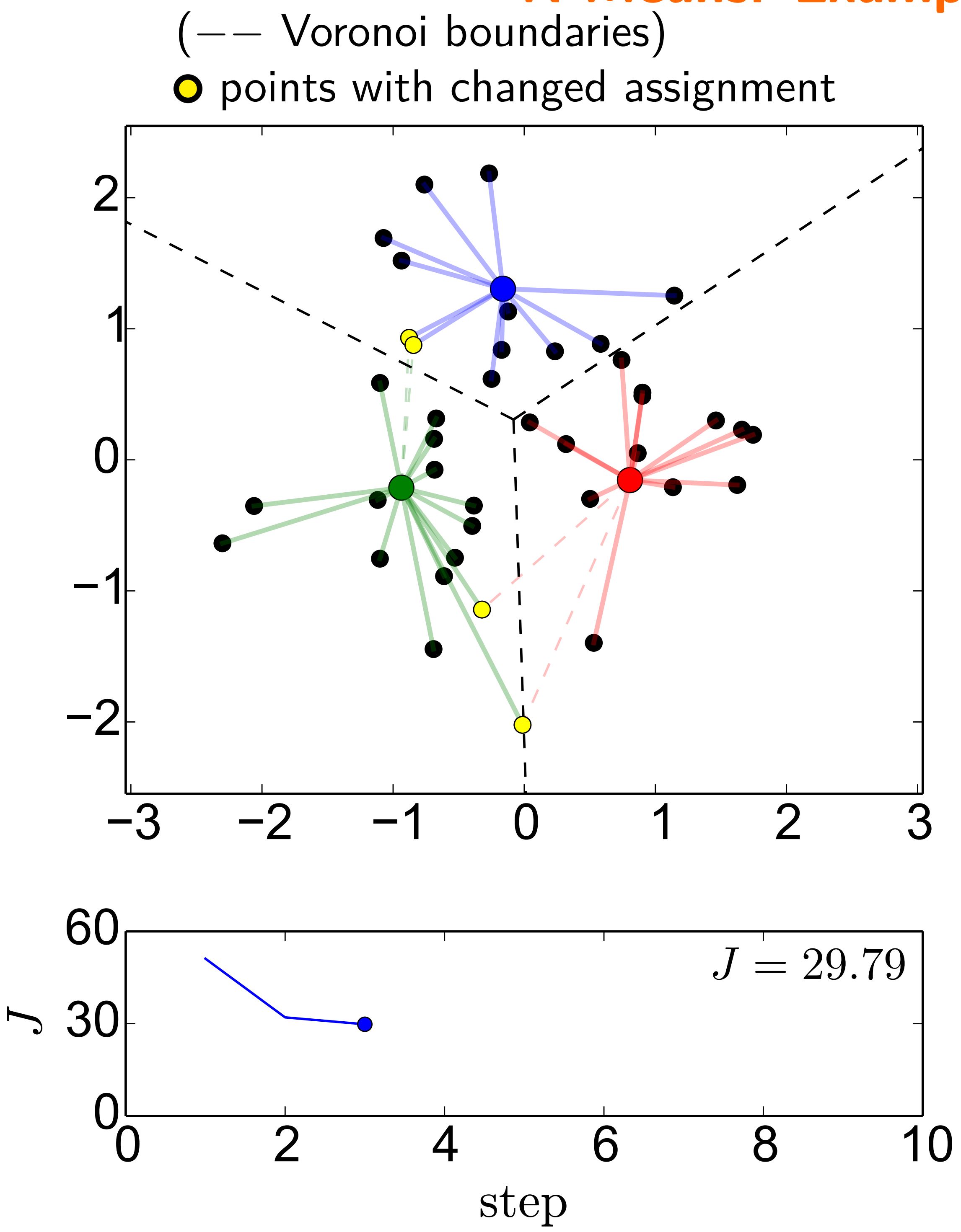


Cluster the data points to $K = 3$ clusters

Initial cluster centers (here selected randomly from data points)

step 1, compute assignments
step 2, recompute centers

K-Means: Example



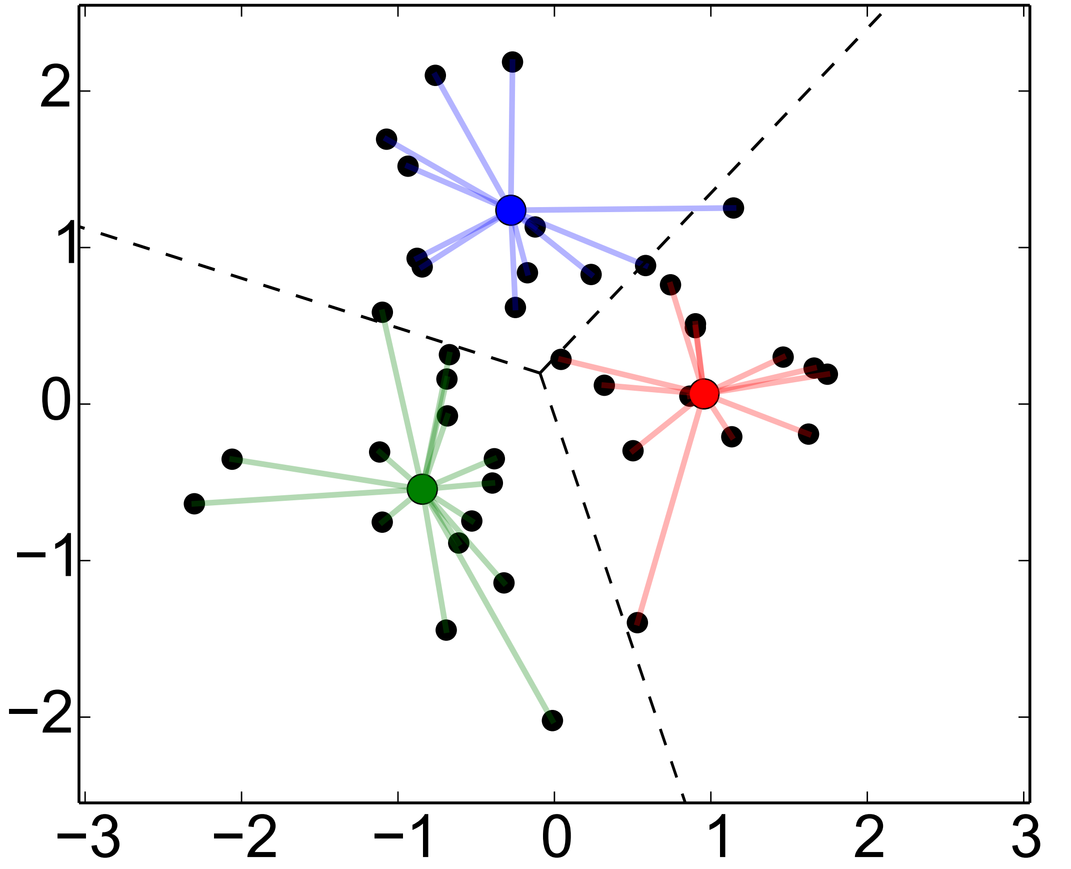
Cluster the data points to $K = 3$ clusters

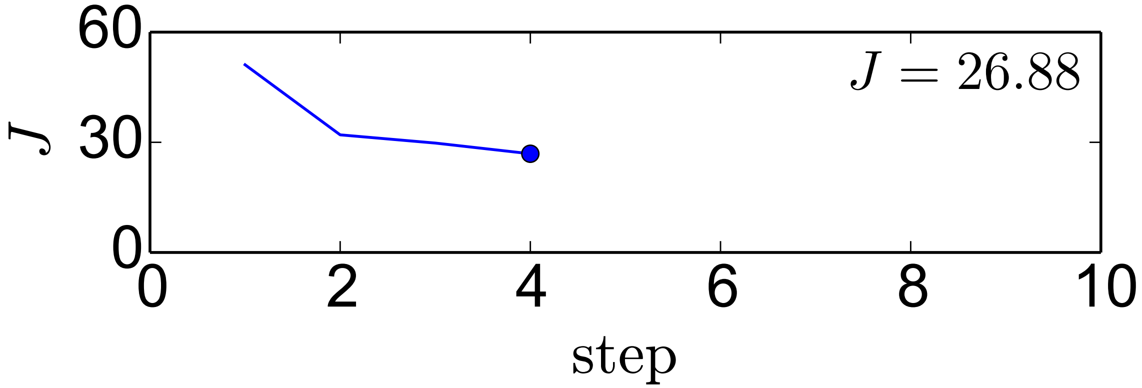
Initial cluster centers (here selected randomly from data points)

step 1, compute assignments
step 2, recompute centers
step 3, recompute assignments

K-Means: Example

(−− Voronoi boundaries)
● points with changed assignment



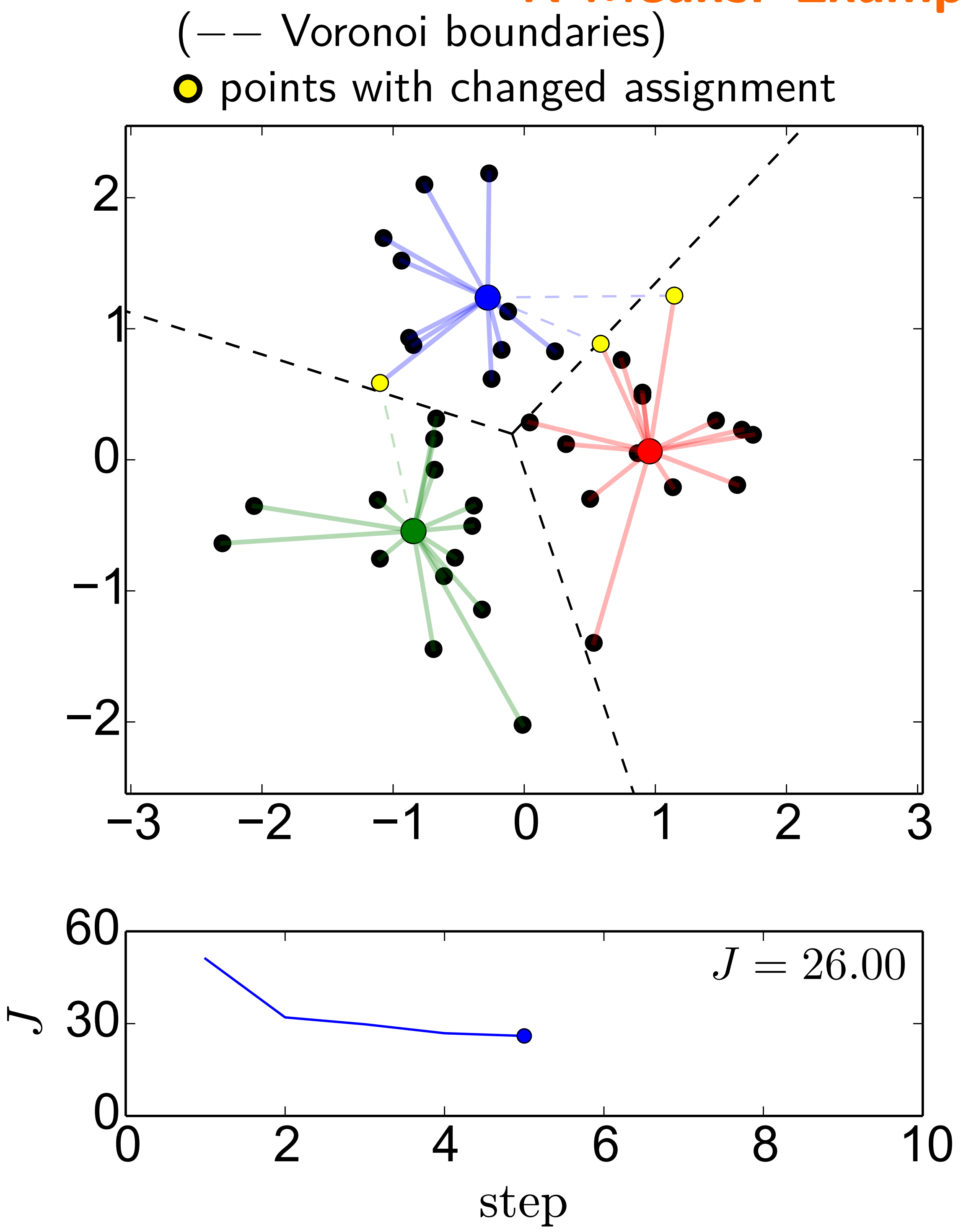


Cluster the data points to $K = 3$ clusters

Initial cluster centers (here selected randomly from data points)

step 1, compute assignments
step 2, recompute centers
step 3, recompute assignments
step 4, recompute centers

K-Means: Example



Cluster the data points to $K = 3$ clusters

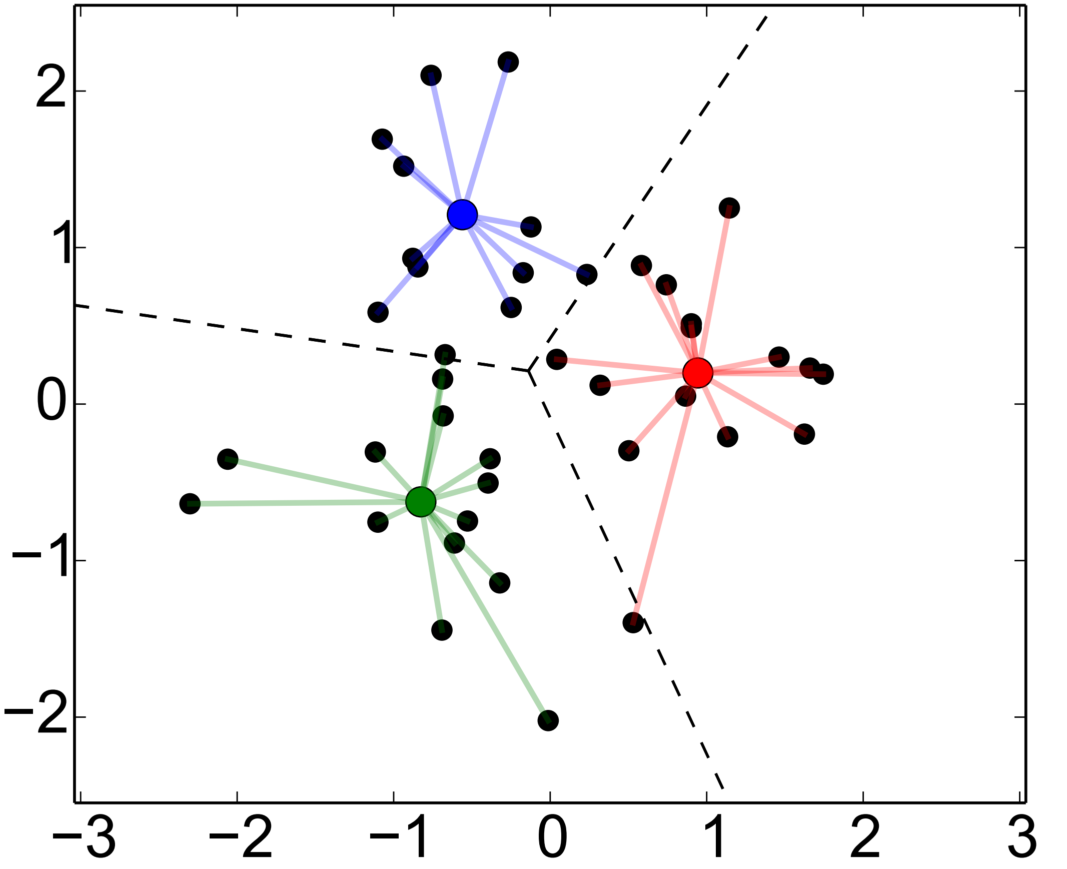
Initial cluster centers (here selected randomly from data points)

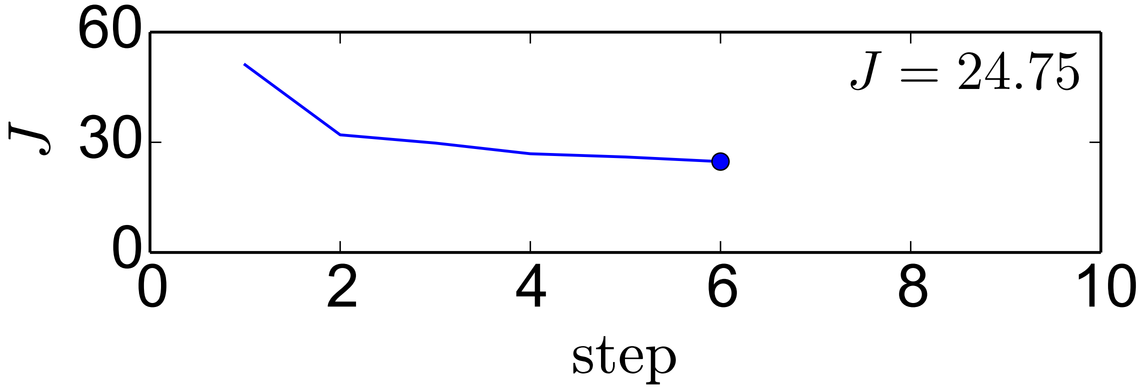
step 1, compute assignments
step 2, recompute centers
step 3, recompute assignments
step 4, recompute centers
step 5, recompute assignments

K-Means: Example

(−− Voronoi boundaries)

● points with changed assignment



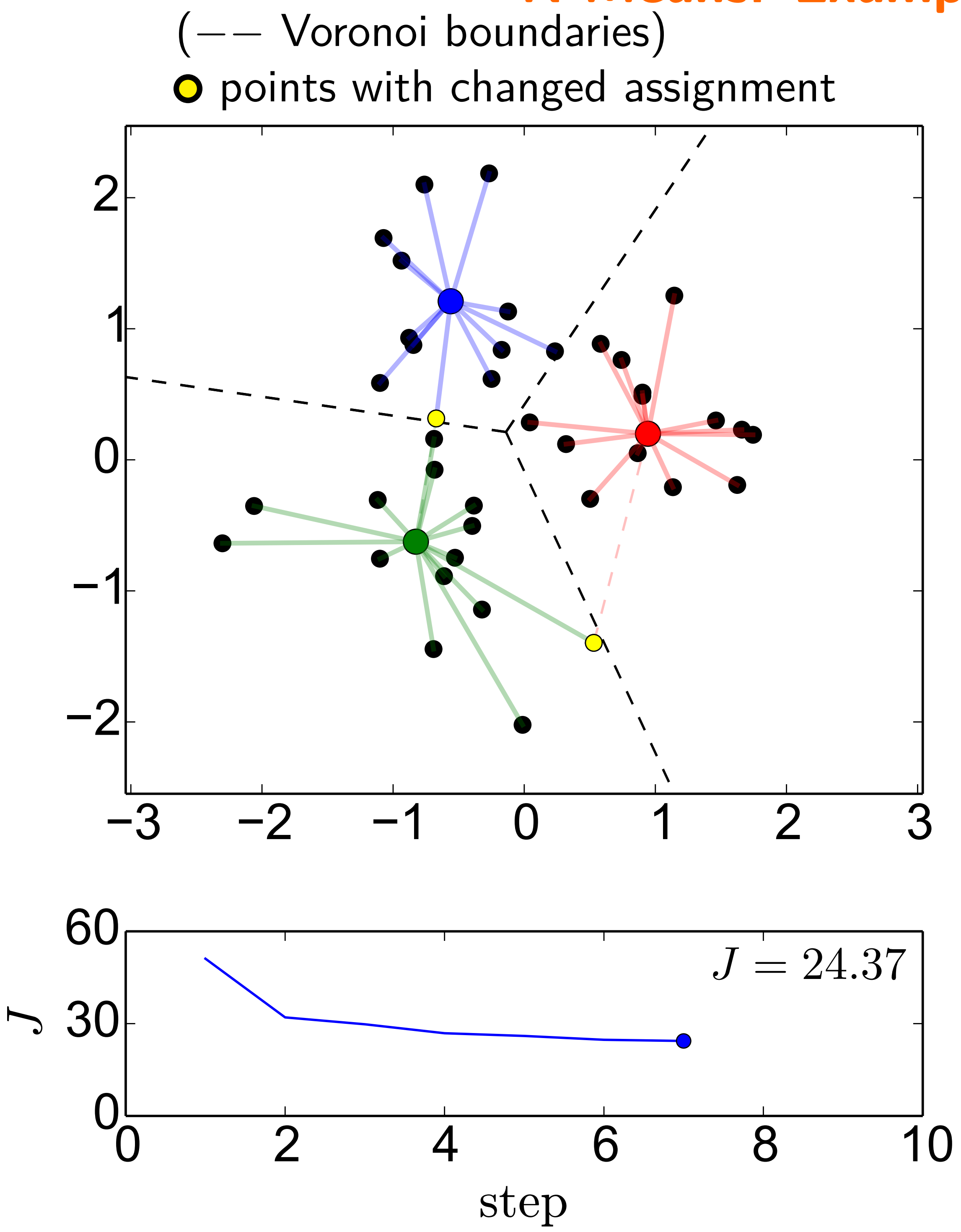


Cluster the data points to $K = 3$ clusters

Initial cluster centers (here selected randomly from data points)

step 1, compute assignments
step 2, recompute centers
step 3, recompute assignments
step 4, recompute centers
step 5, recompute assignments
step 6, recompute centers

K-Means: Example



Cluster the data points to $K = 3$ clusters

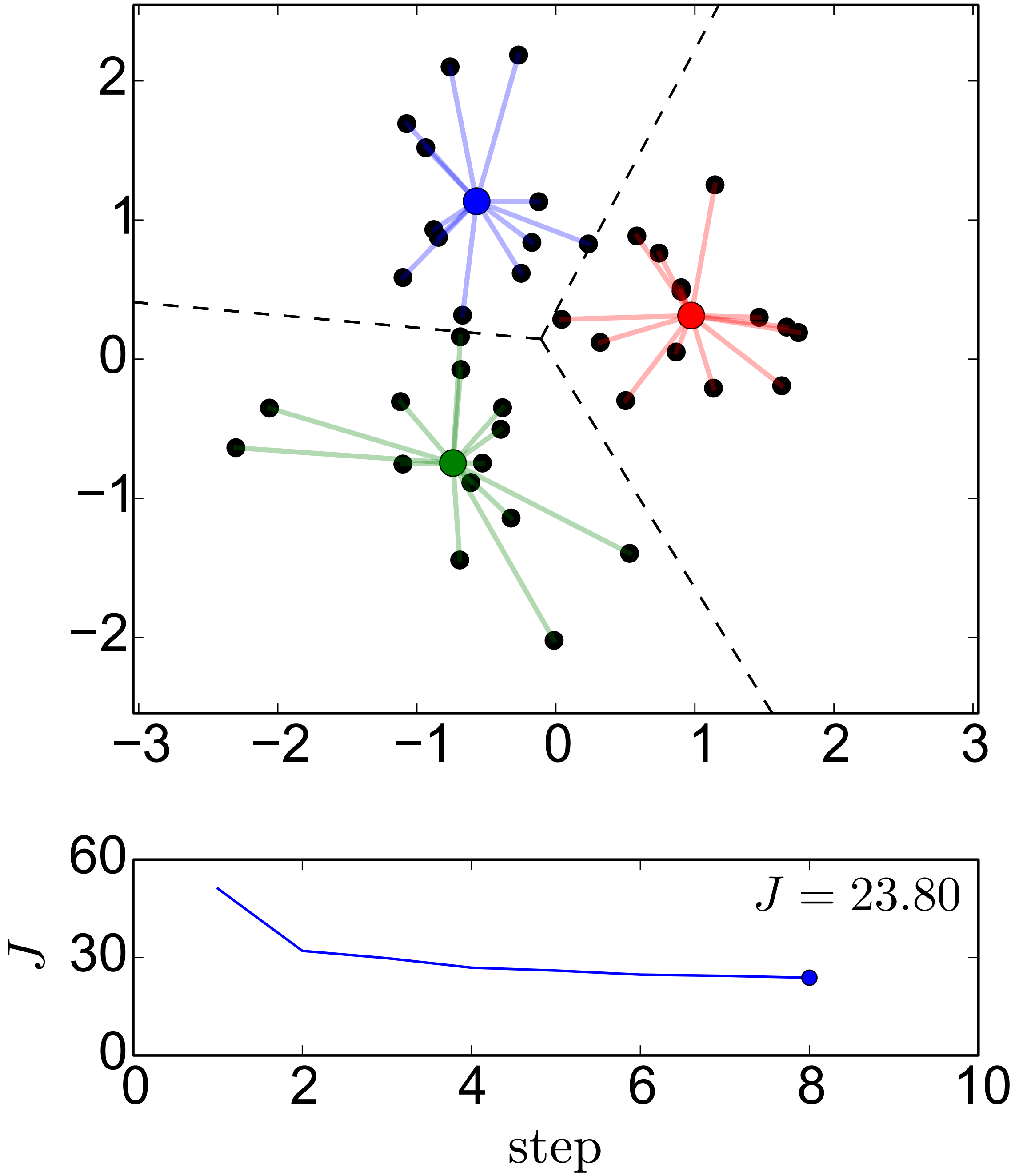
Initial cluster centers (here selected randomly from data points)

step 1, compute assignments
step 2, recompute centers
step 3, recompute assignments
step 4, recompute centers
step 5, recompute assignments
step 6, recompute centers
step 7, recompute assignments

K-Means: Example

(−− Voronoi boundaries)

● points with changed assignment



Cluster the data points to $K = 3$ clusters

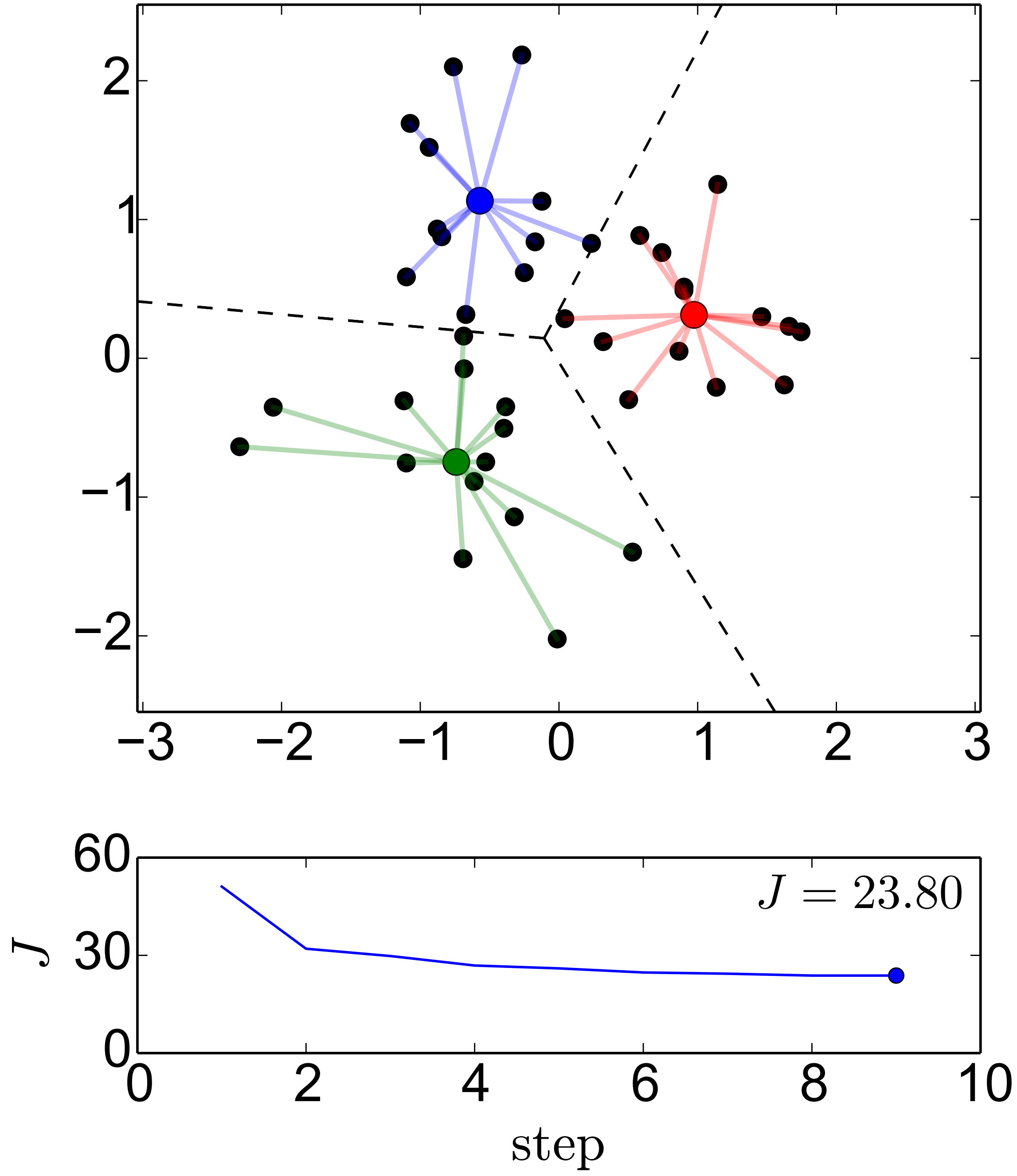
Initial cluster centers (here selected randomly from data points)

step 1, compute assignments
step 2, recompute centers
step 3, recompute assignments
step 4, recompute centers
step 5, recompute assignments
step 6, recompute centers
step 7, recompute assignments
step 8, recompute centers

K-Means: Example

(−− Voronoi boundaries)

● points with changed assignment



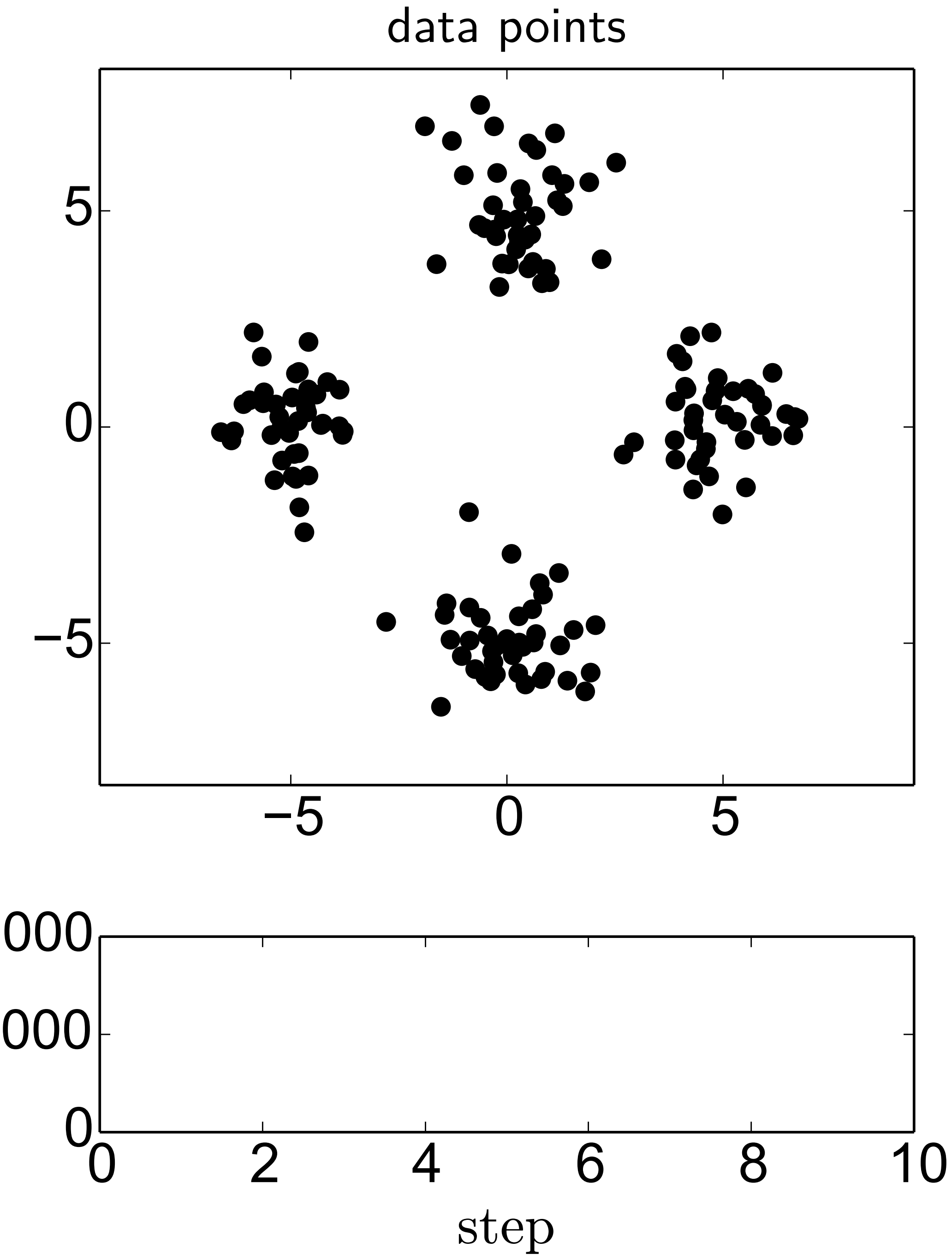
Cluster the data points to $K = 3$ clusters

Initial cluster centers (here selected randomly from data points)

step 1, compute assignments
step 2, recompute centers
step 3, recompute assignments
step 4, recompute centers
step 5, recompute assignments
step 6, recompute centers
step 7, recompute assignments
step 8, recompute centers
step 9, recompute assignments

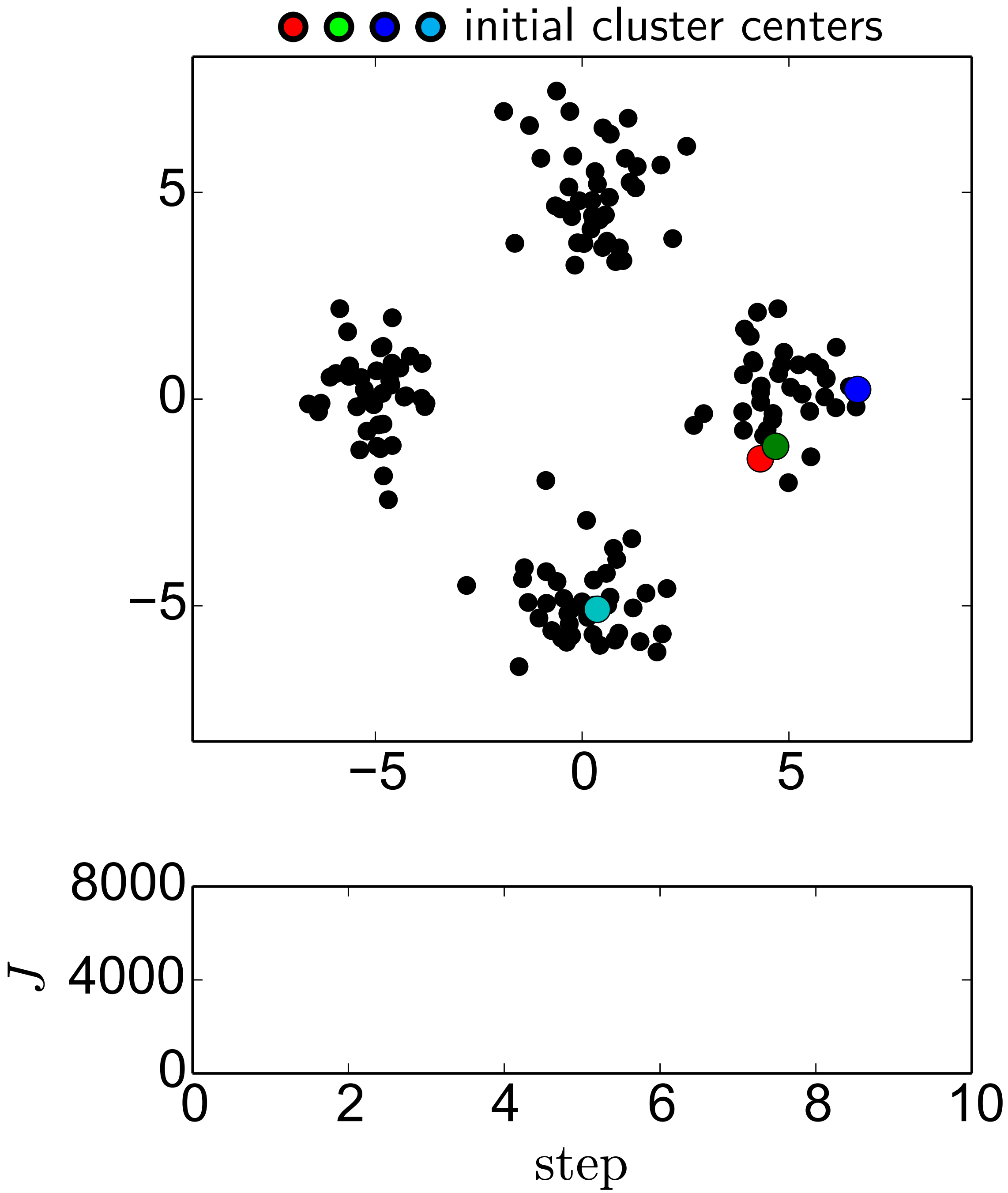
The assignments have not changed. Done.

K-Means: Example with Reinitialization



Cluster the data points to $K = 4$ clusters

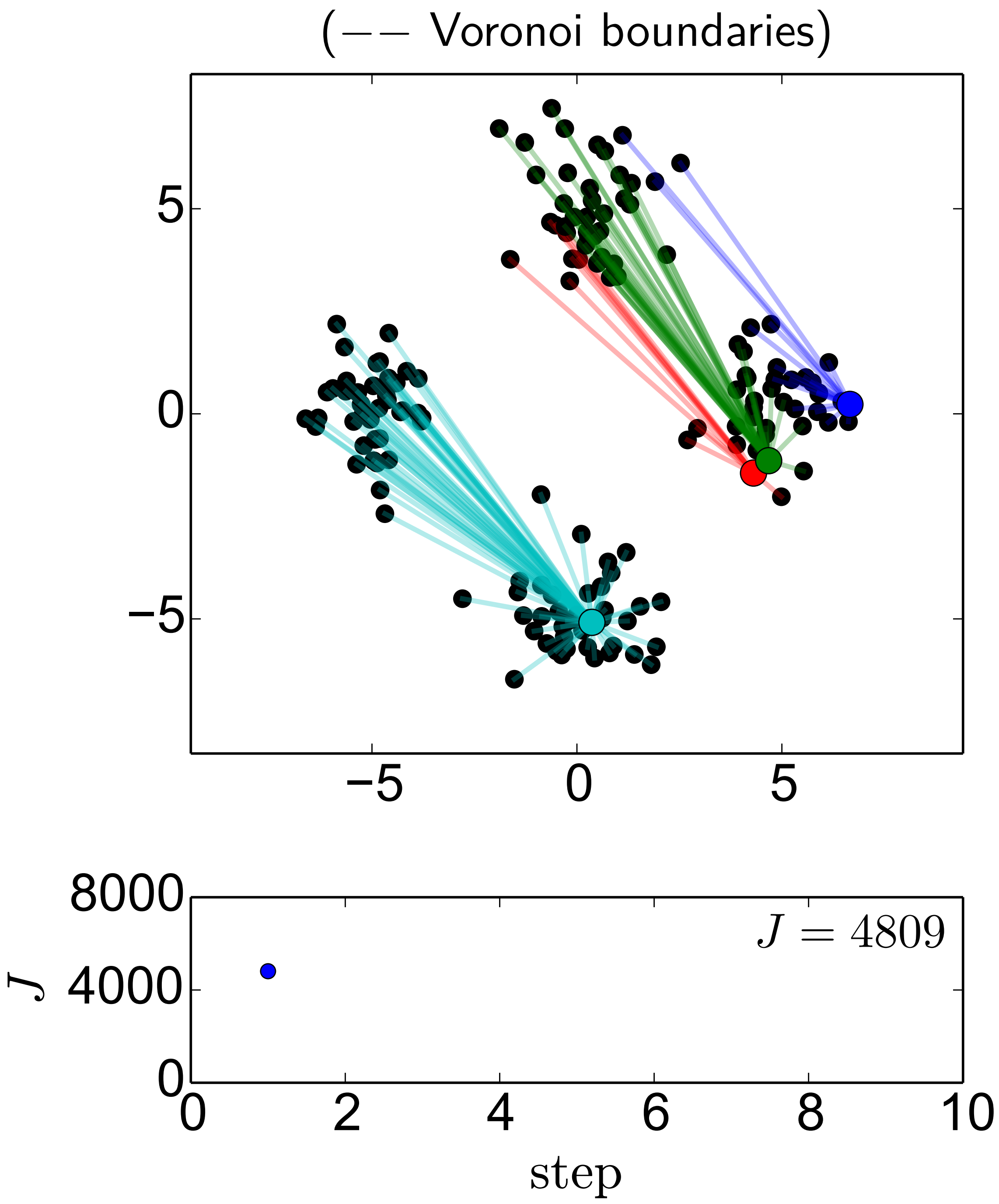
K-Means: Example with Reinitialization



Cluster the data points to $K = 4$ clusters

Initial cluster centers (here selected randomly from data points)

K-Means: Example with Reinitialization

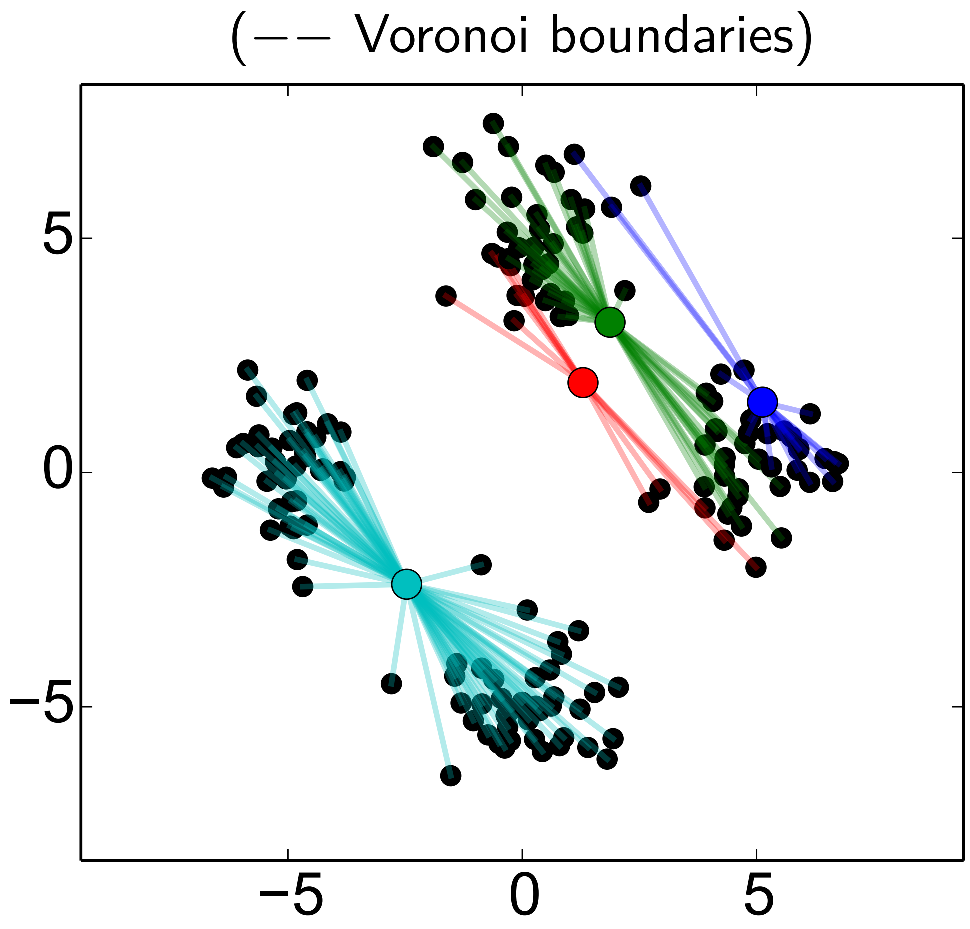


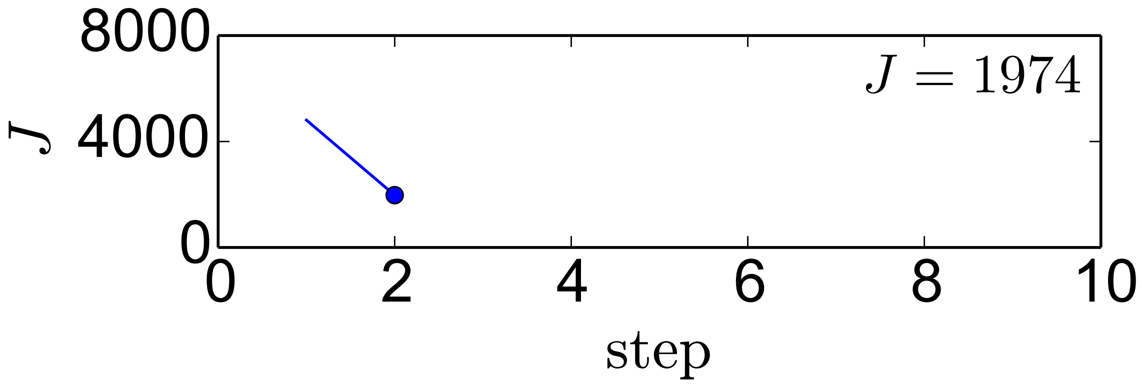
(−− Voronoi boundaries)

Cluster the data points to $K = 4$ clusters

Step 1, compute assignments

K-Means: Example with Reinitialization

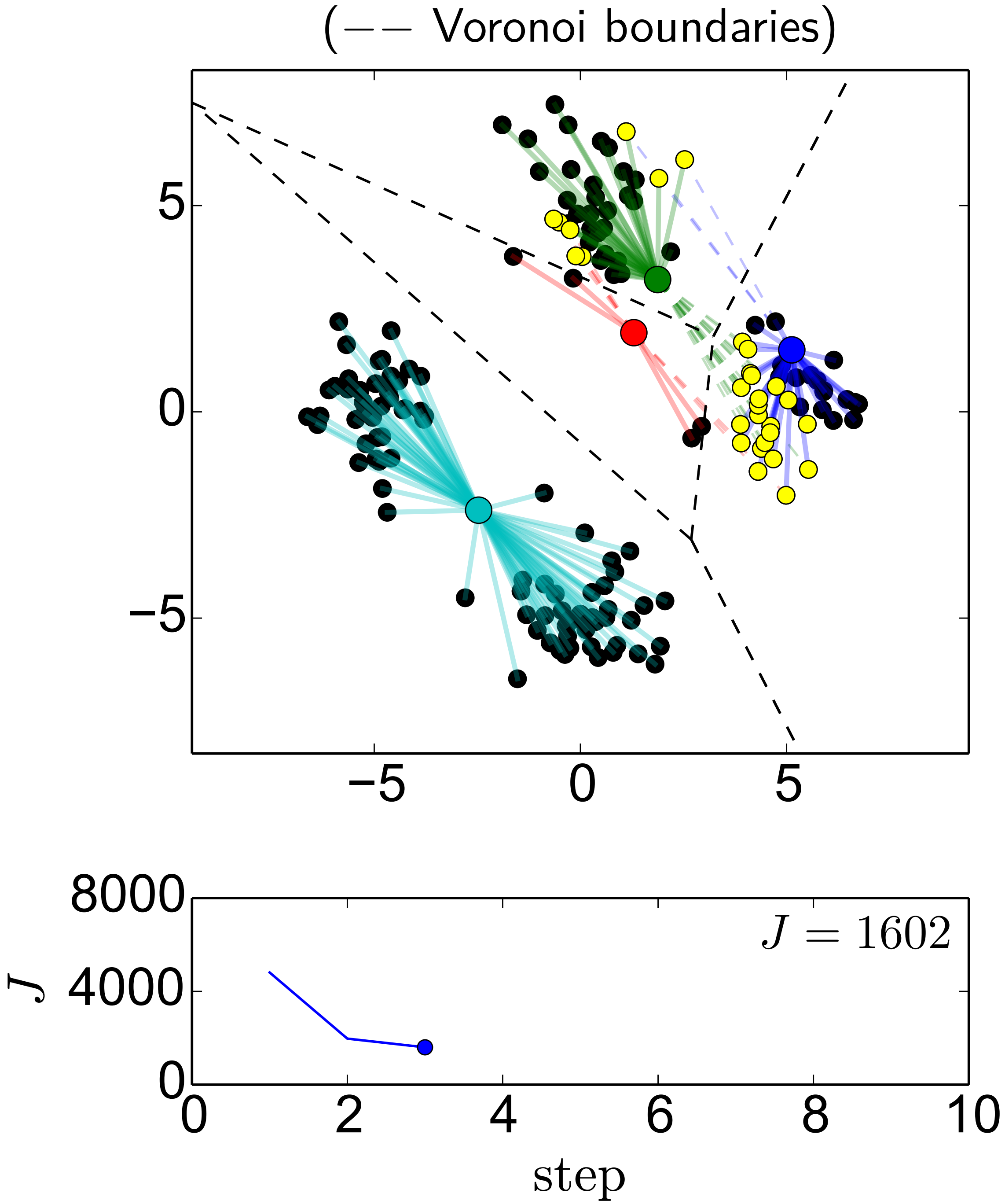




Cluster the data points to $K = 4$ clusters

Step 2, recompute centres

K-Means: Example with Reinitialization

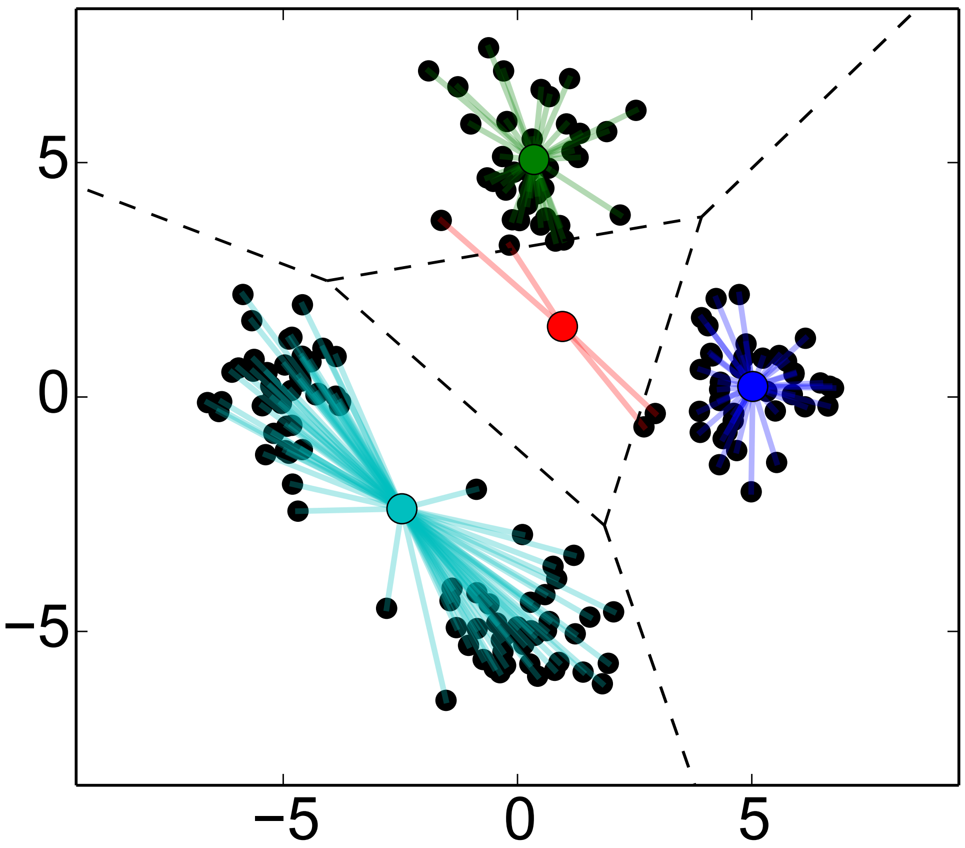


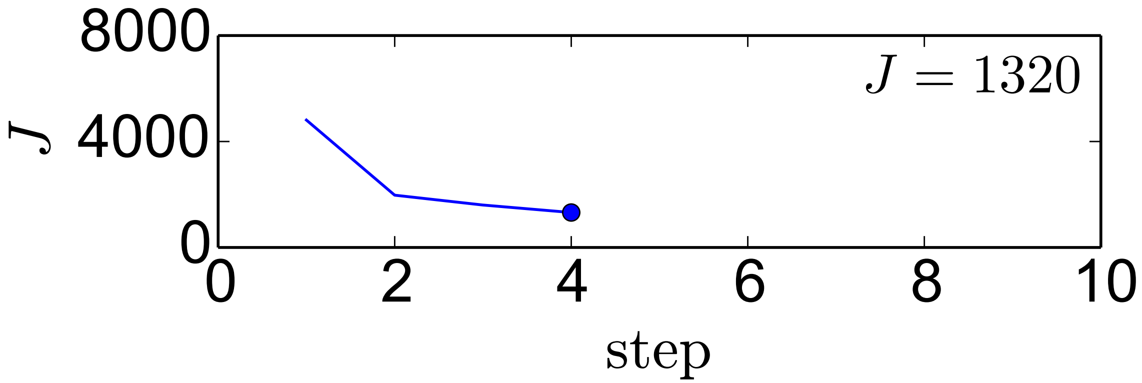
Cluster the data points to $K = 4$ clusters

Step 3, recompute assignments

K-Means: Example with Reinitialization

(−− Voronoi boundaries)



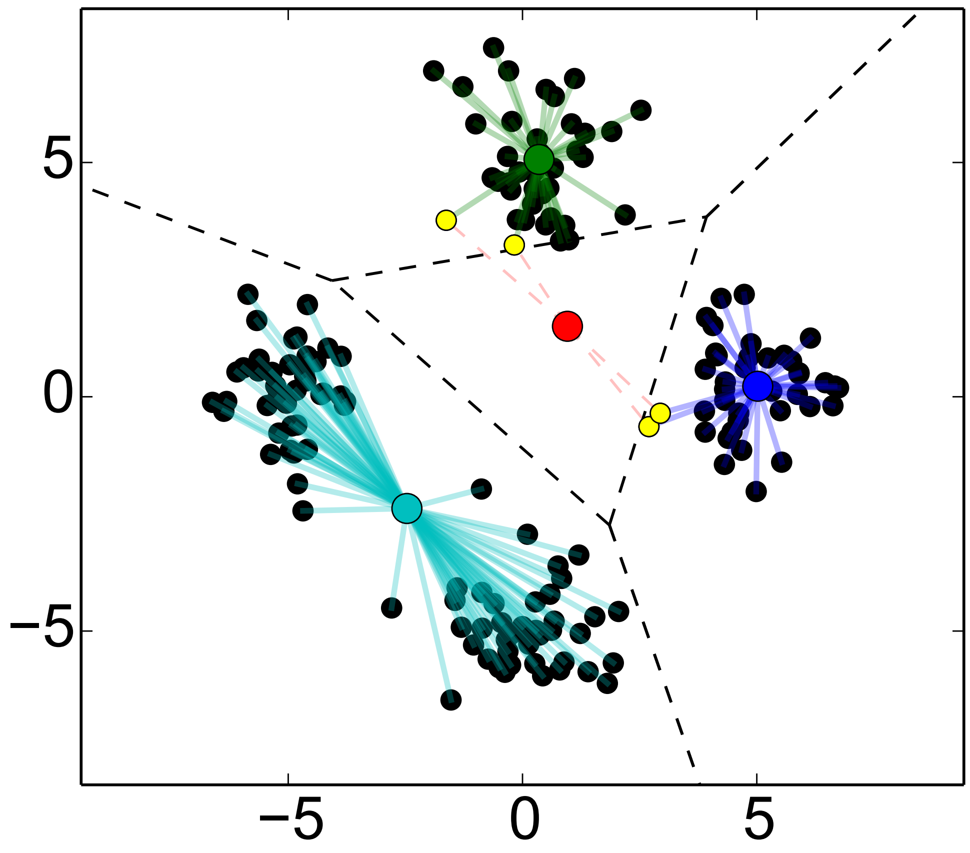


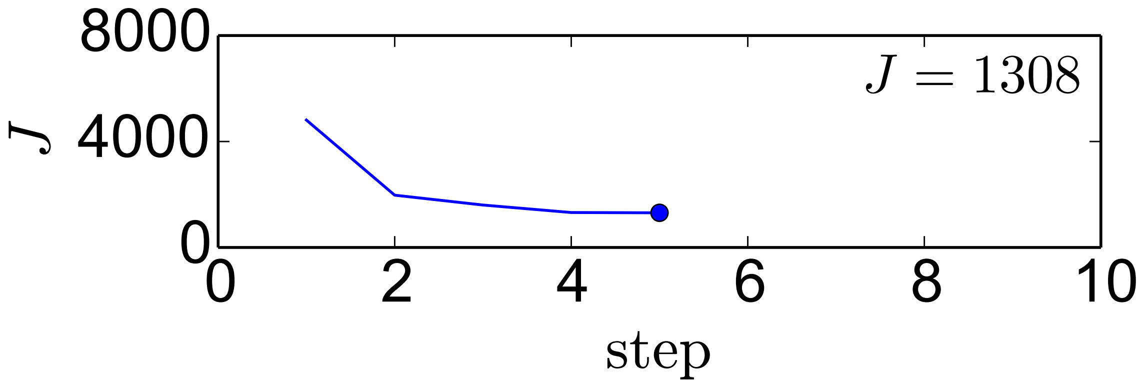
Cluster the data points to $K = 4$ clusters

Step 4, recompute centres

K-Means: Example with Reinitialization

(−− Voronoi boundaries)





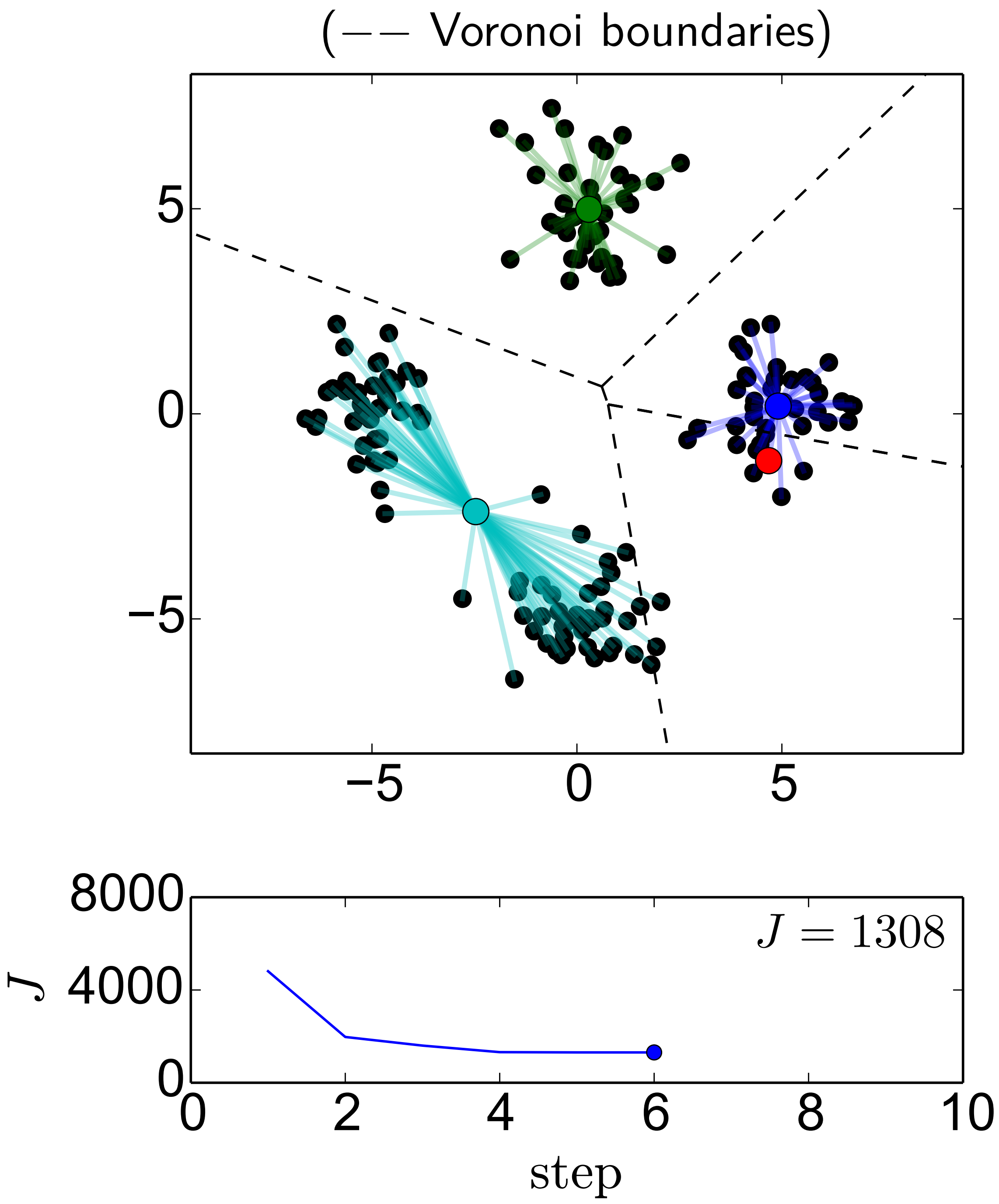
Cluster the data points to $K = 4$ clusters

Step 5, recompute assignments

No points assigned to cluster 🔴

$\Rightarrow$ it will be reinitialized.

K-Means: Example with Reinitialization



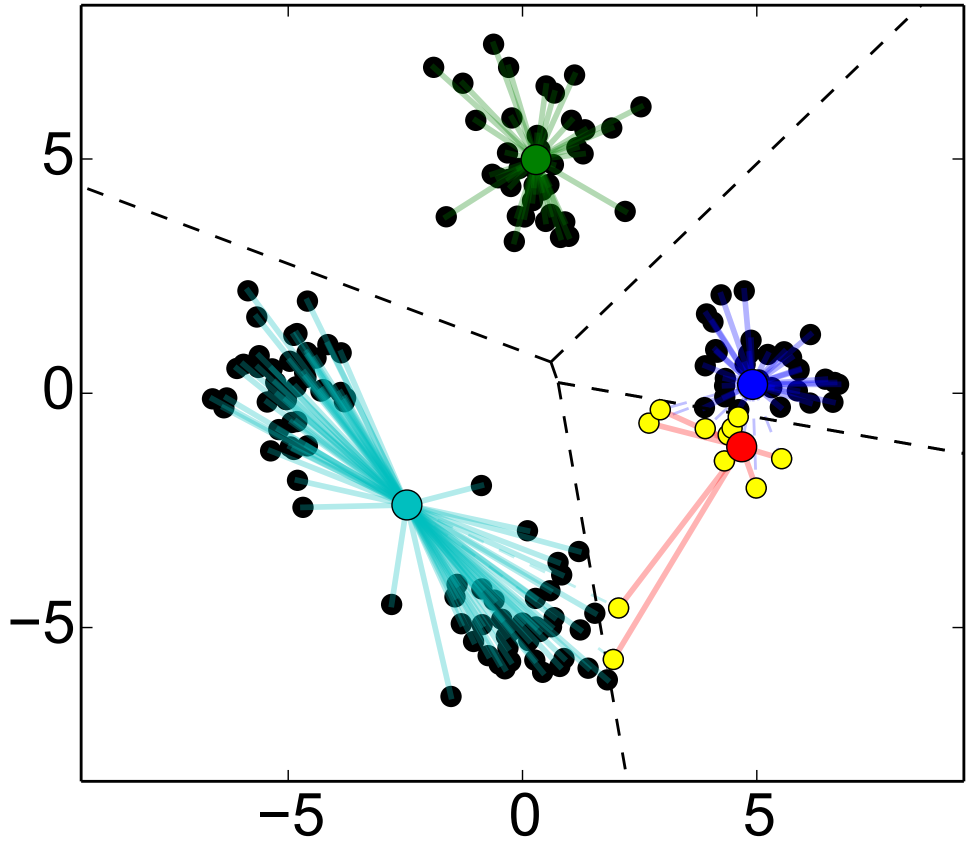
Cluster the data points to $K = 4$ clusters

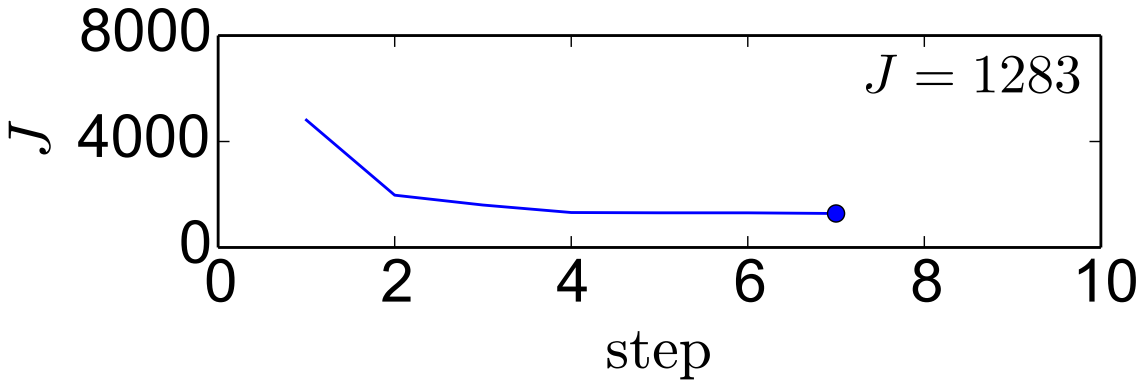
Step 6

– recompute ● ● ●

– reinitialize ●

K-Means: Example with Reinitialization

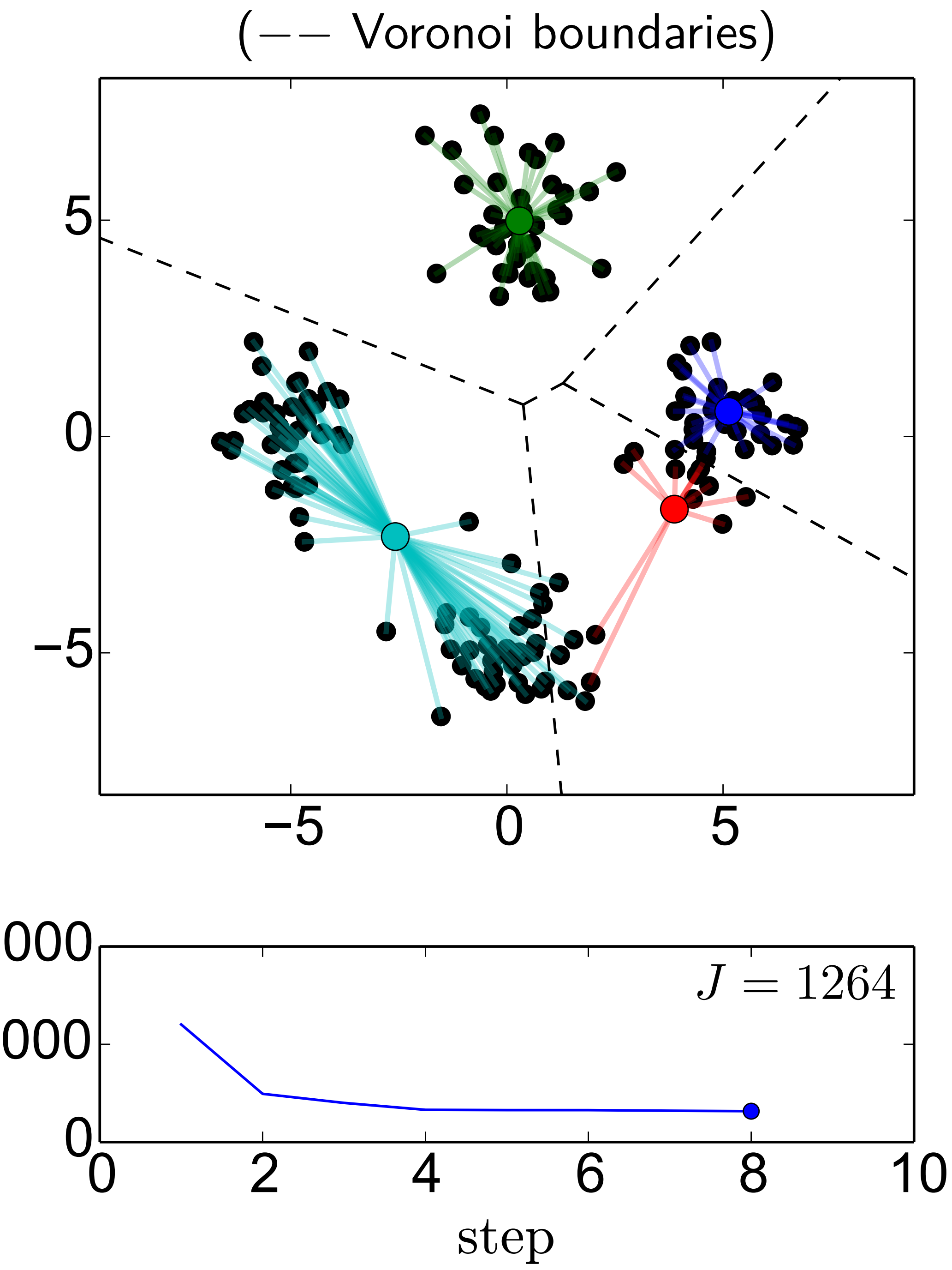

(−− Voronoi boundaries)



Cluster the data points to $K = 4$ clusters

Step 7, recompute assignments

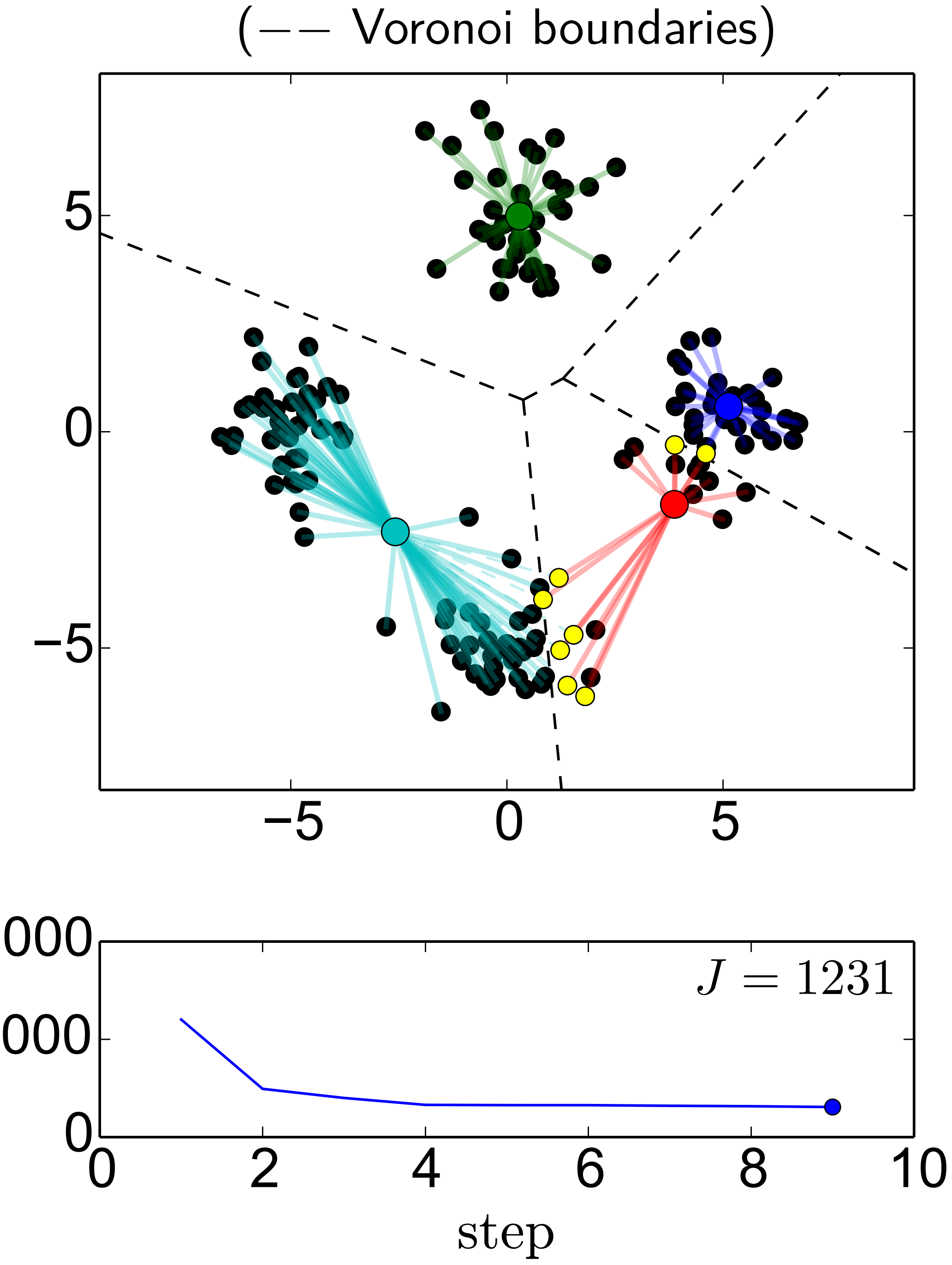
K-Means: Example with Reinitialization



Cluster the data points to $K = 4$ clusters

Step 8, recompute centres

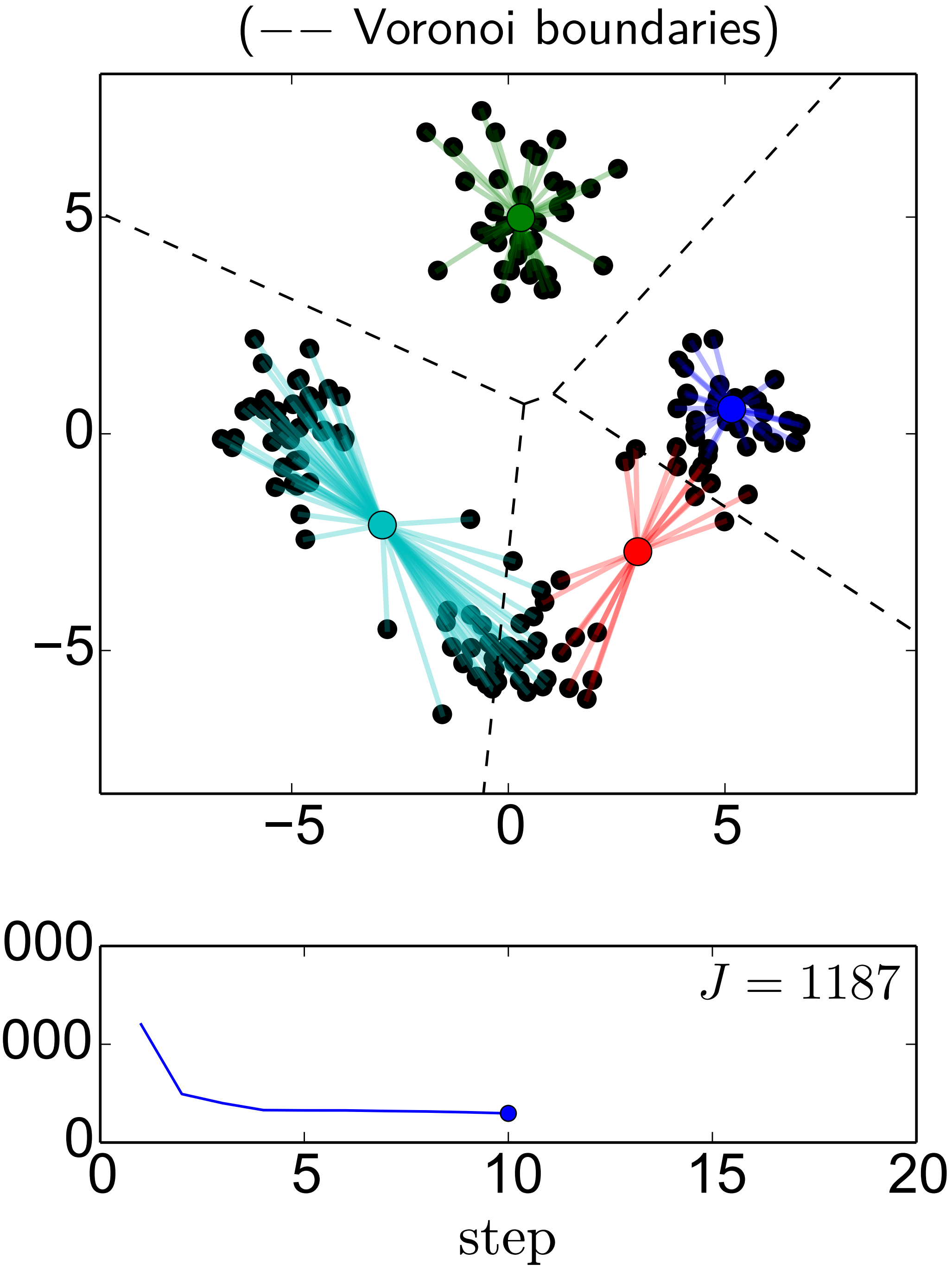
K-Means: Example with Reinitialization



Cluster the data points to $K = 4$ clusters

Step 9, recompute assignments

K-Means: Example with Reinitialization

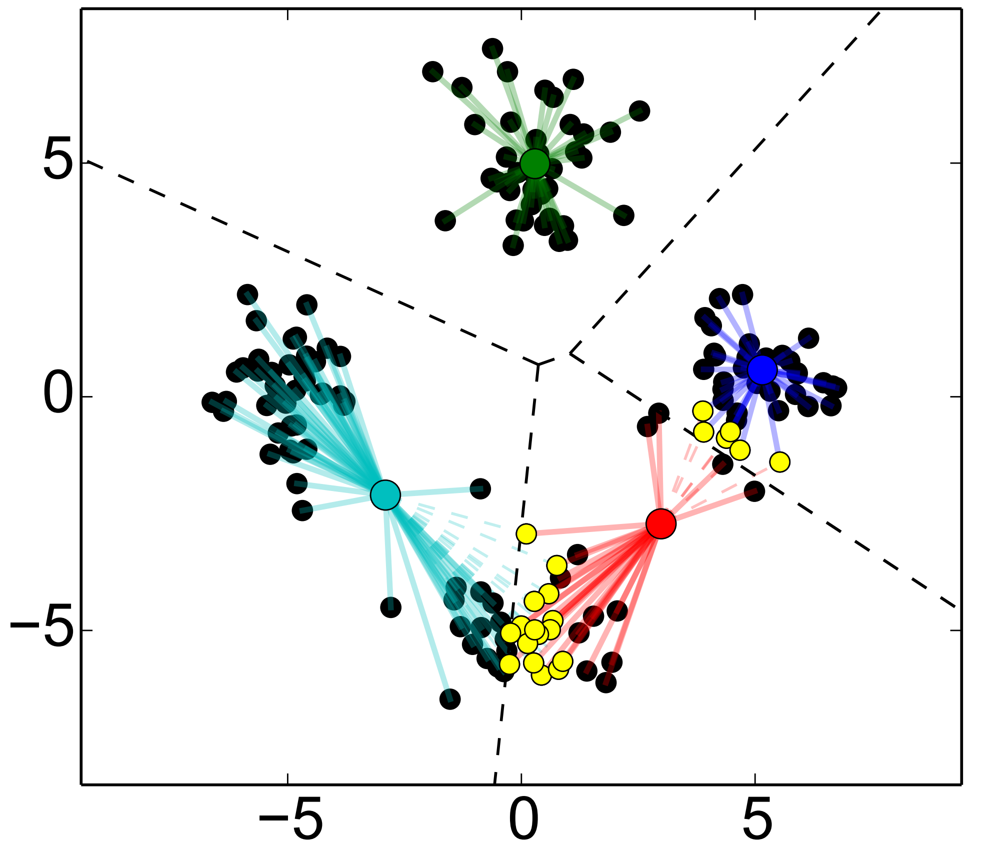


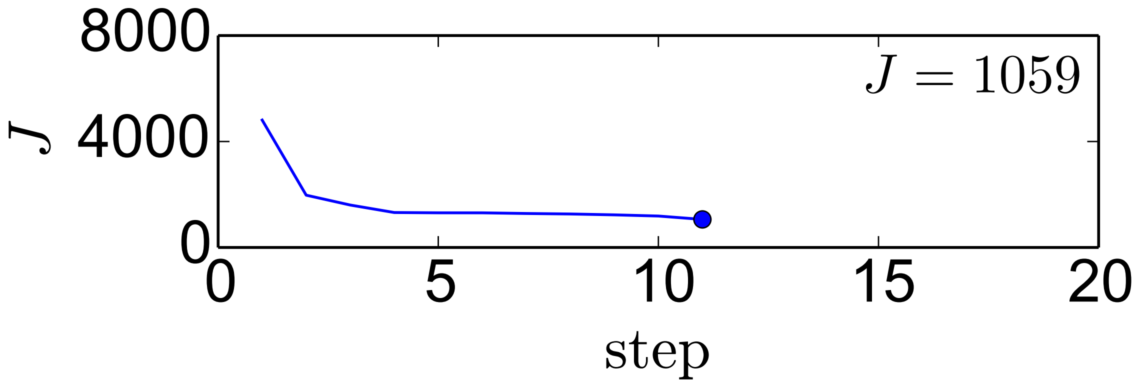
Cluster the data points to $K = 4$ clusters

Step 10, recompute centres

K-Means: Example with Reinitialization

(−− Voronoi boundaries)



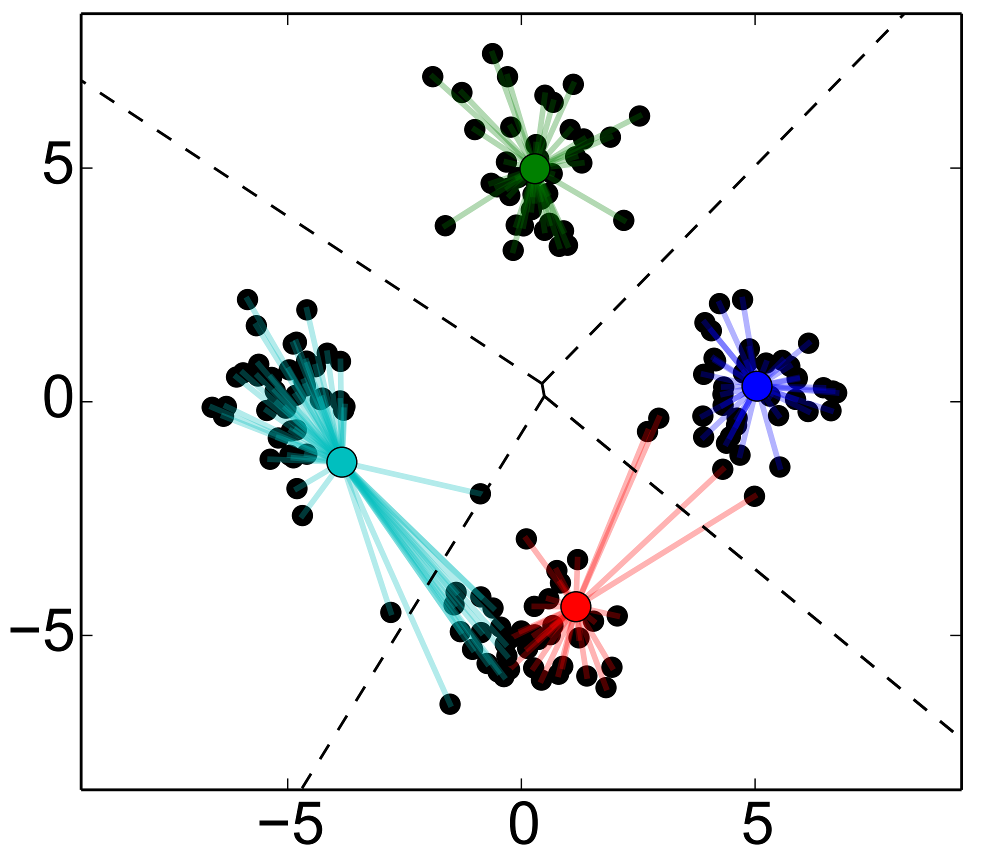


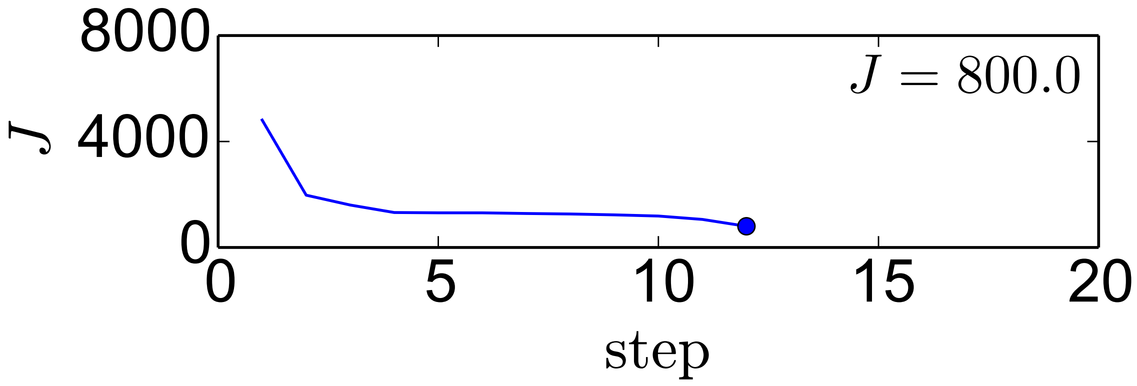
Cluster the data points to $K = 4$ clusters

Step 11, recompute assignments

K-Means: Example with Reinitialization

(−− Voronoi boundaries)

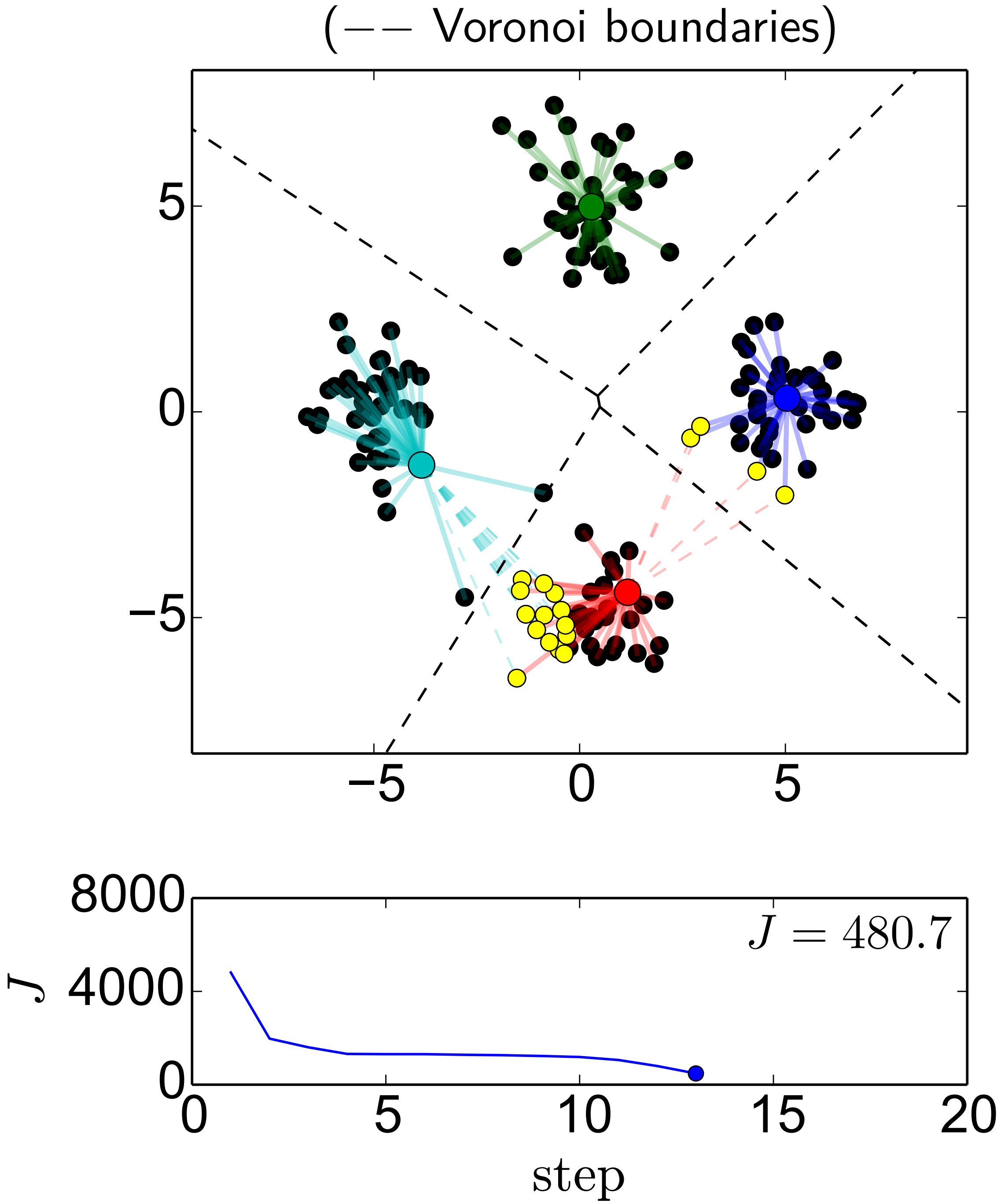




Cluster the data points to $K = 4$ clusters

Step 12, recompute centres

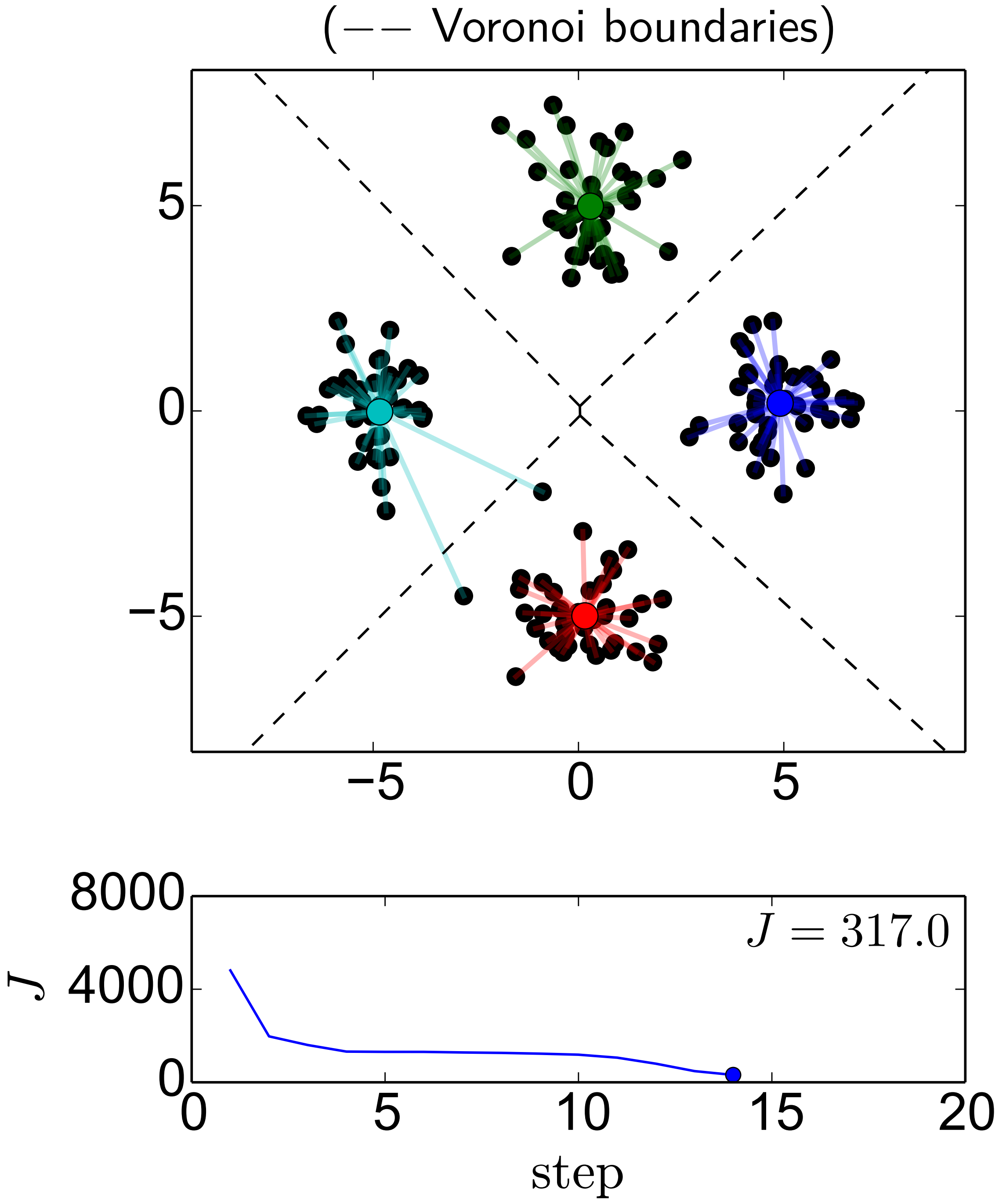
K-Means: Example with Reinitialization



Cluster the data points to $K = 4$ clusters

Step 13, recompute assignments

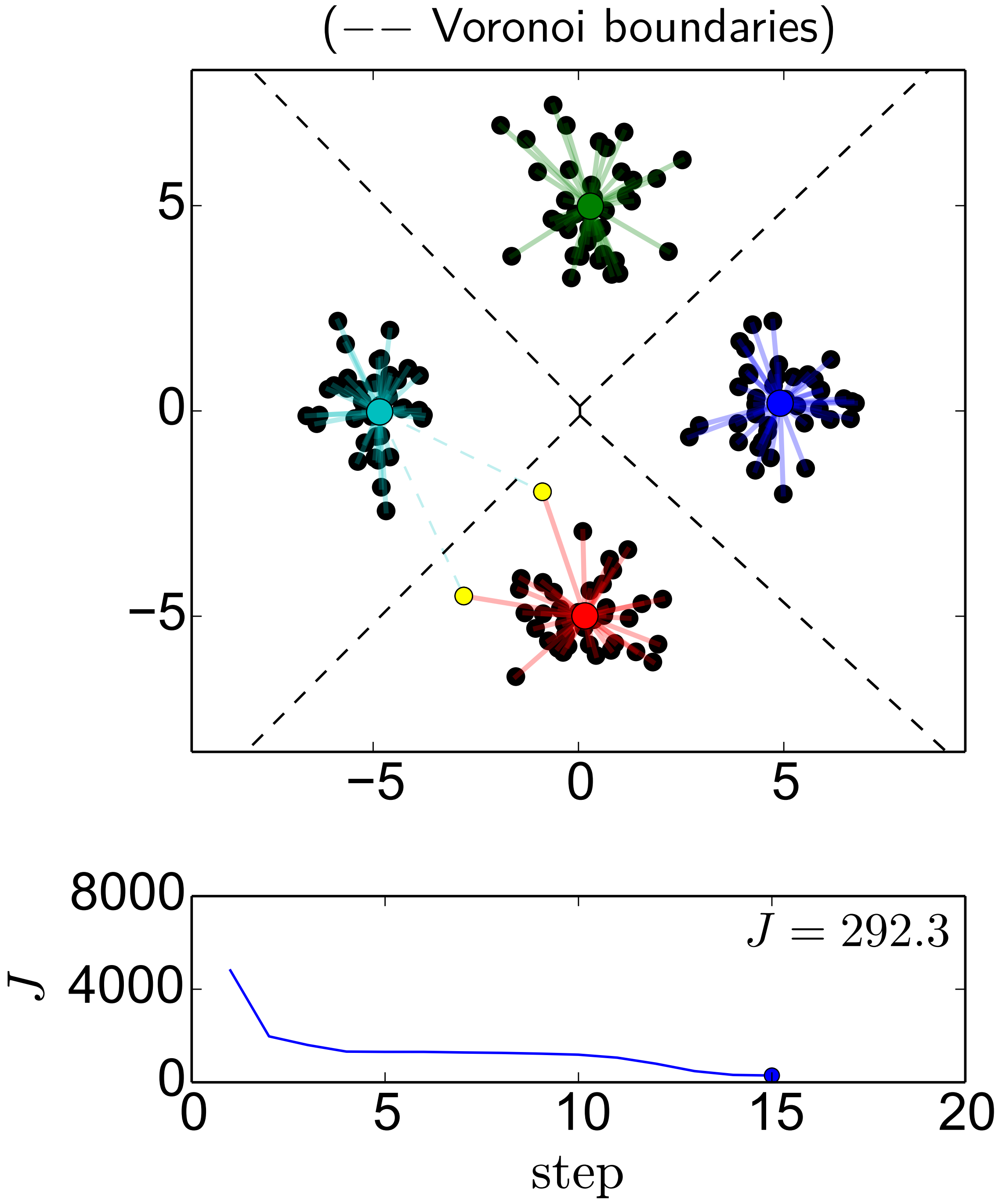
K-Means: Example with Reinitialization



Cluster the data points to $K = 4$ clusters

Step 14, recompute centres

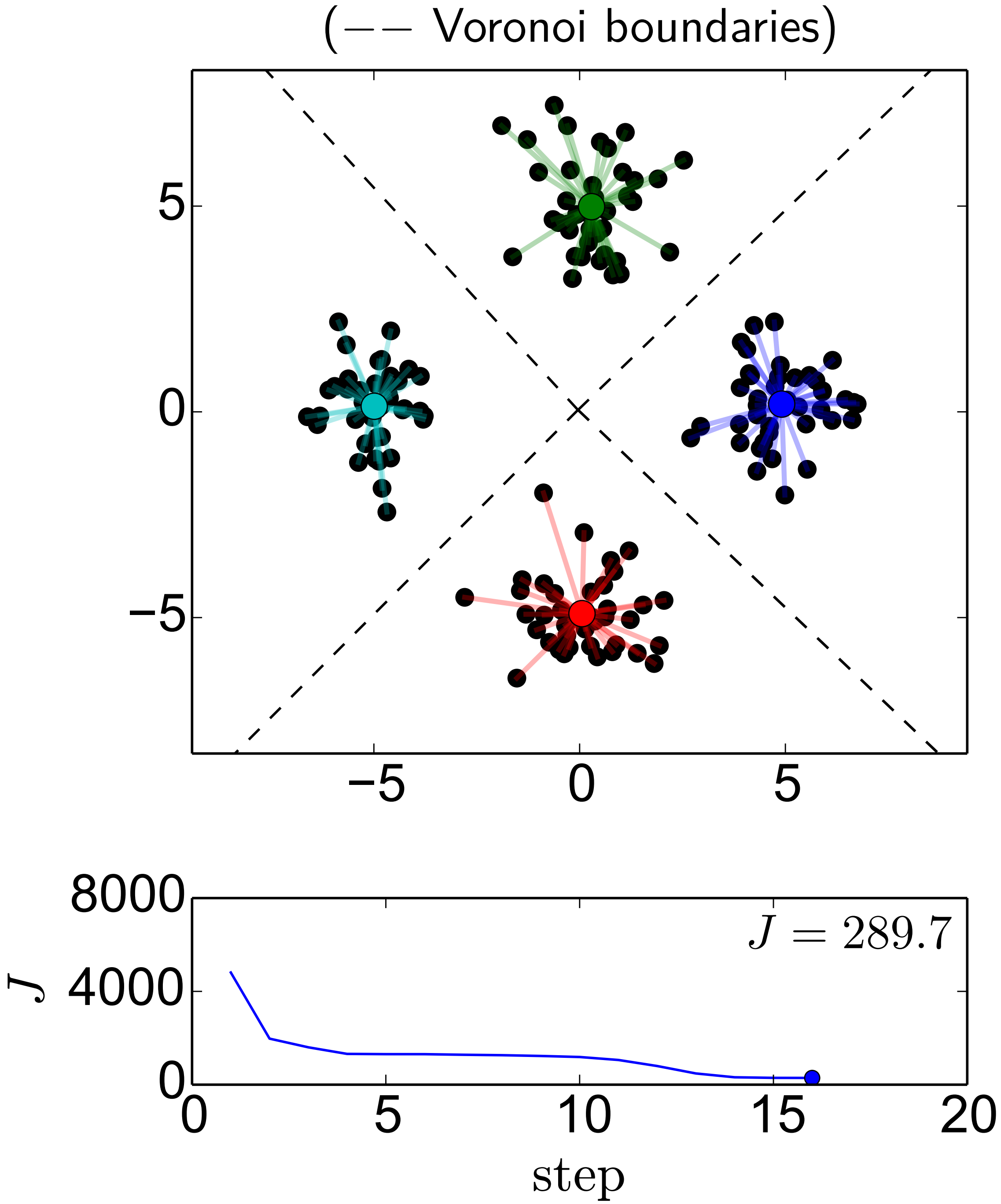
K-Means: Example with Reinitialization



Cluster the data points to $K = 4$ clusters

Step 15, recompute assignments

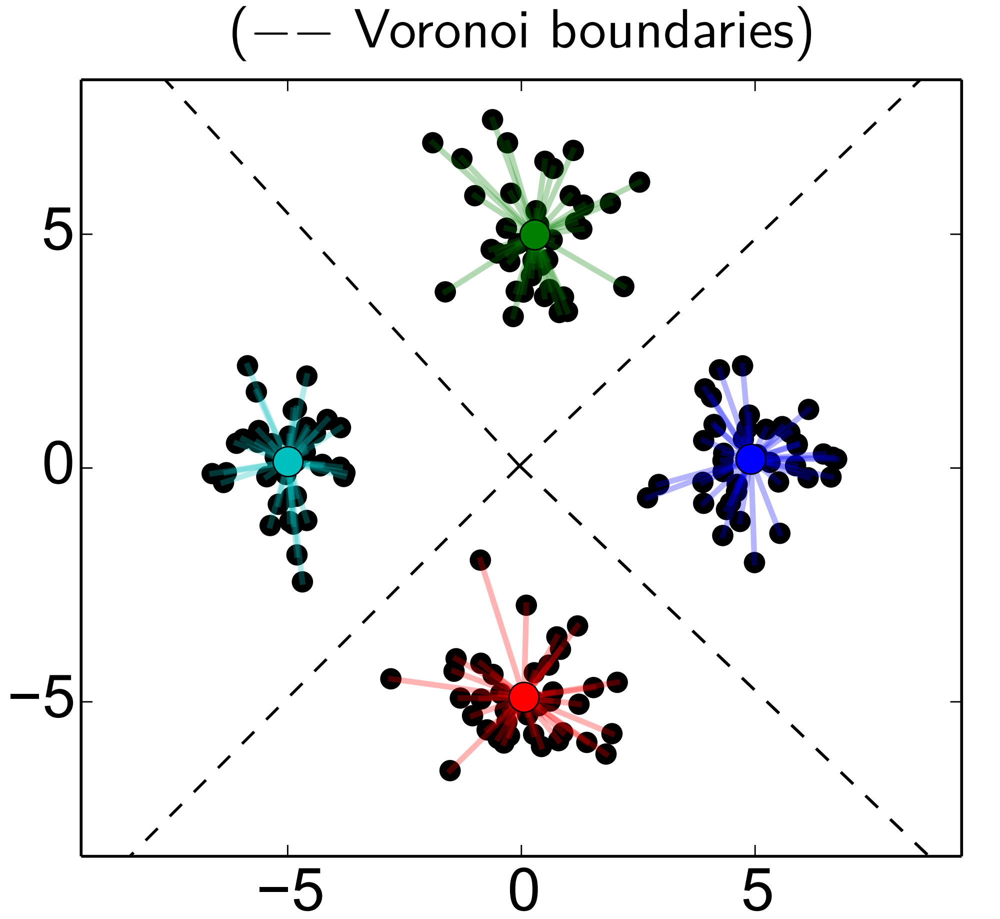
K-Means: Example with Reinitialization

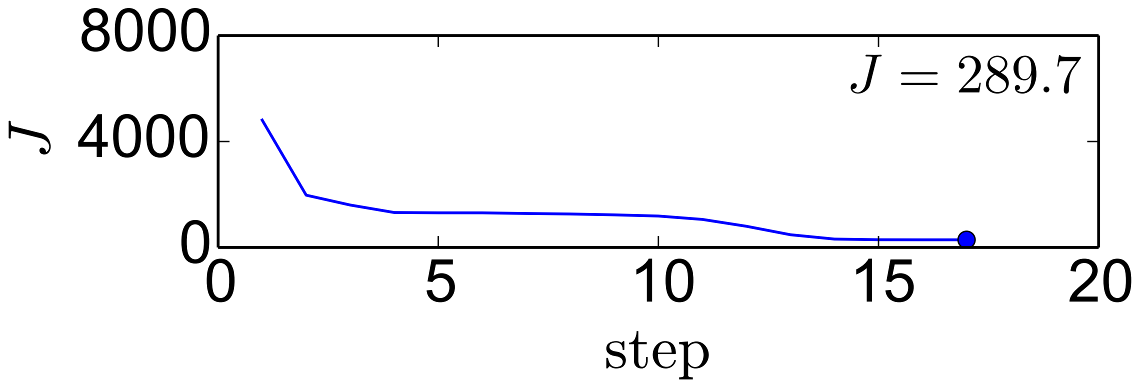


Cluster the data points to $K = 4$ clusters

Step 16, recompute centres

K-Means: Example with Reinitialization





Cluster the data points to $K = 4$ clusters

Step 17, recompute assignments

Assigments haven't changed. **Done**.

K-Means: Properties

Clustering criterion: $J(\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_K; \mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_K) = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{T}_k} \|\mathbf{x} - \mathbf{c}_k\|^2 .$

K-means algorithm skeleton:

1. Initialization
2. Assignment optimization (assign to closest etalon)
3. Cluster centres optimization ($\mathbf{c}_k$ set to average of data in $\mathcal{T}_k$)
4. Goto 2 if the assignments have changed

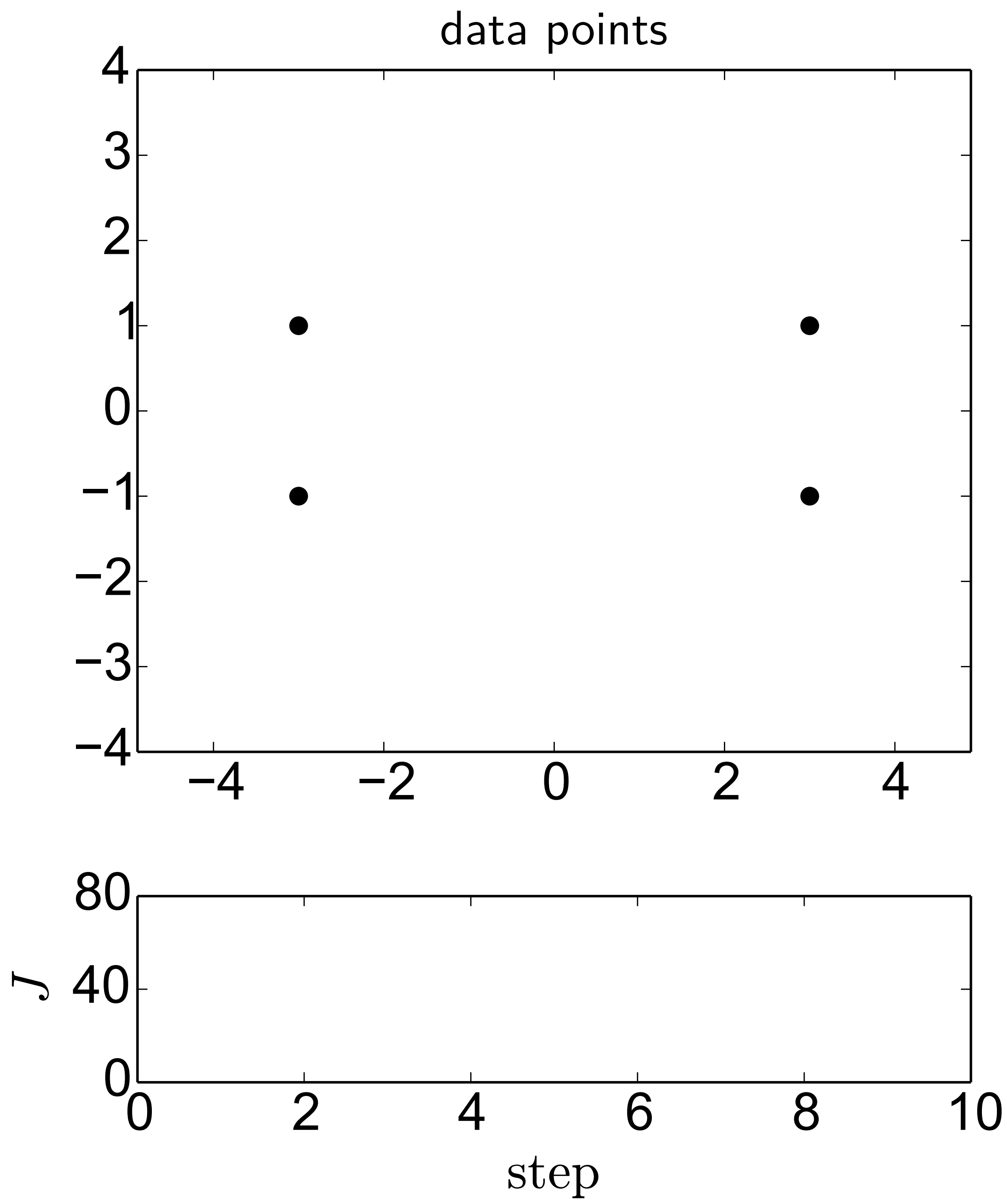
Convergence:

- During the run of the algorithm, J monotonically decreases (though not necessarily strictly) because:
  - Step 2: The contribution of each $\mathbf{x}_l$ to J either stays the same, or gets lower,
  - Step 3: For a fixed assignment $\mathcal{T}_k$, the mean of the data points in $\mathcal{T}_k$ is the optimal solution under the least squares criterion J. If $\mathcal{T}_k$ is empty and re-initizalization is done for $\mathbf{c}_k$ then this has no effect on J at this point, but it can only cause additional decrease in J in the subsequent Step 2.
- Since there is a **finite number of assignmens** (how many?) and no assignment is visited twice (why?), the K-means algorithm **reaches** a **local** minimum after a **finite** number of steps.

K-Means: Notes

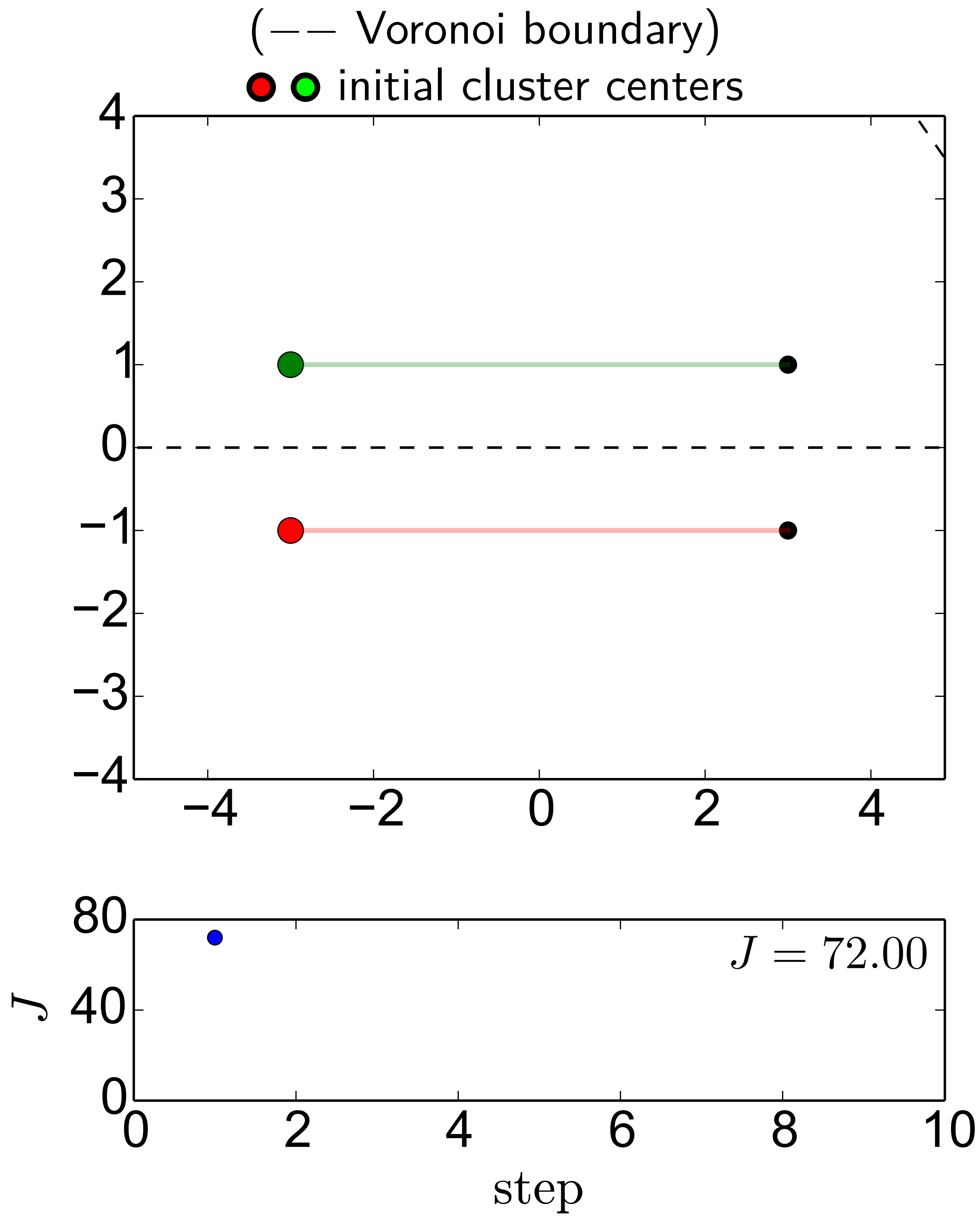
- K-means is clearly not a guaranteed global minimum minimizer.
- In theory, there may be a problem of infinite looping through a set of assignments with equal J, but this is not the case when breaking the ties in Step 2 is done consistently (e.g. assigning to cluster with the lowest index if a point is equidistant to multiple cluster centres.)
- As for the computational time, the complexity of assignment computation dominates, as for every observation the nearest prototype is sought. Trivially implemented, this requires $O(LK)$ distance computations per iteration. Any idea for a speed-up?
- The algorithm is sometimes modified in a way that initialization is done by setting $\mathcal{T}_k$'s and swapping steps 2 and 3 in the iteration loop.

Example of Local Minima



Cluster the data points to $K = 2$ clusters

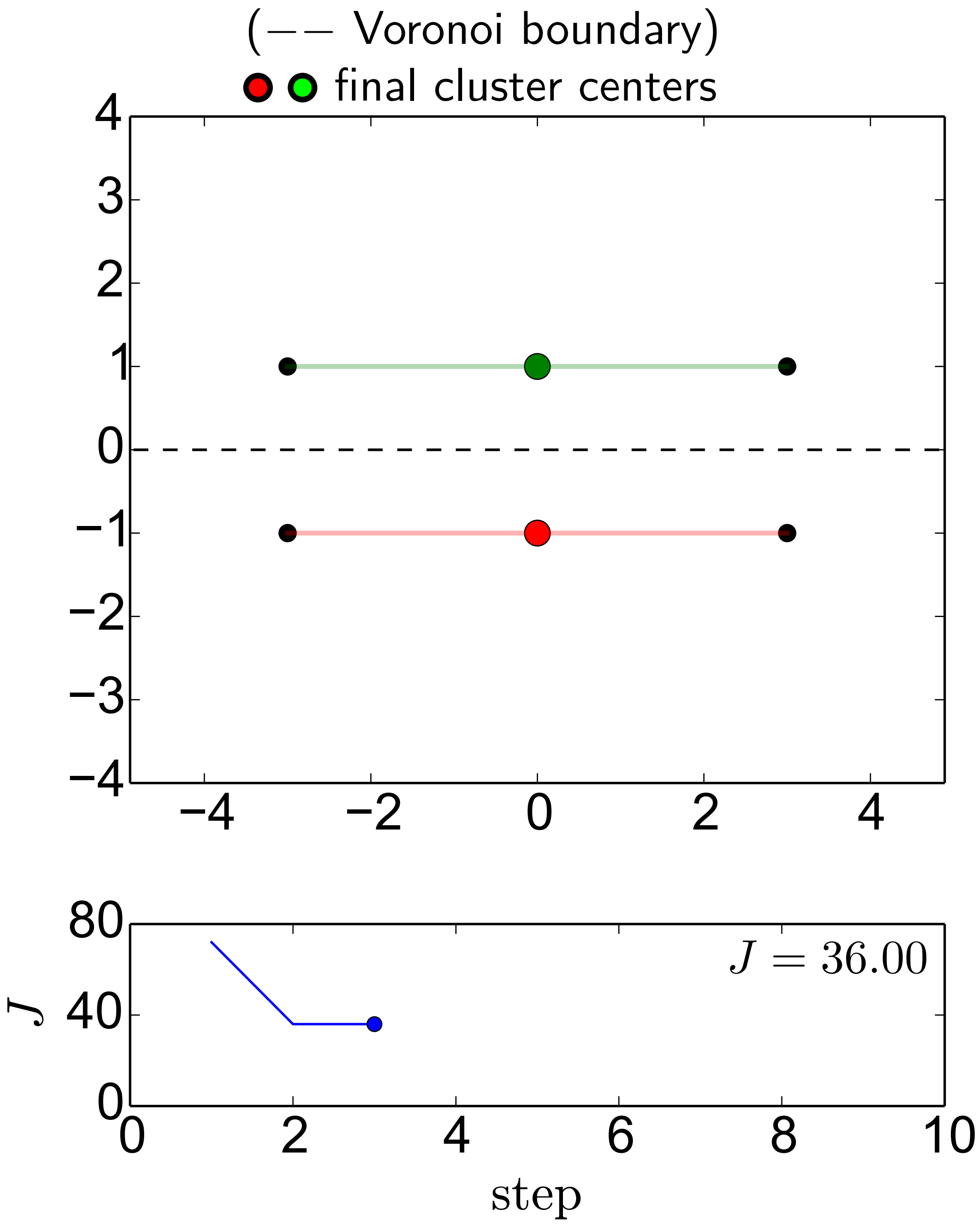
Example of Local Minima



Cluster the data points to $K = 2$ clusters

Initial cluster centers (selected randomly from data points)

Local Minimum 1, $J = 36$



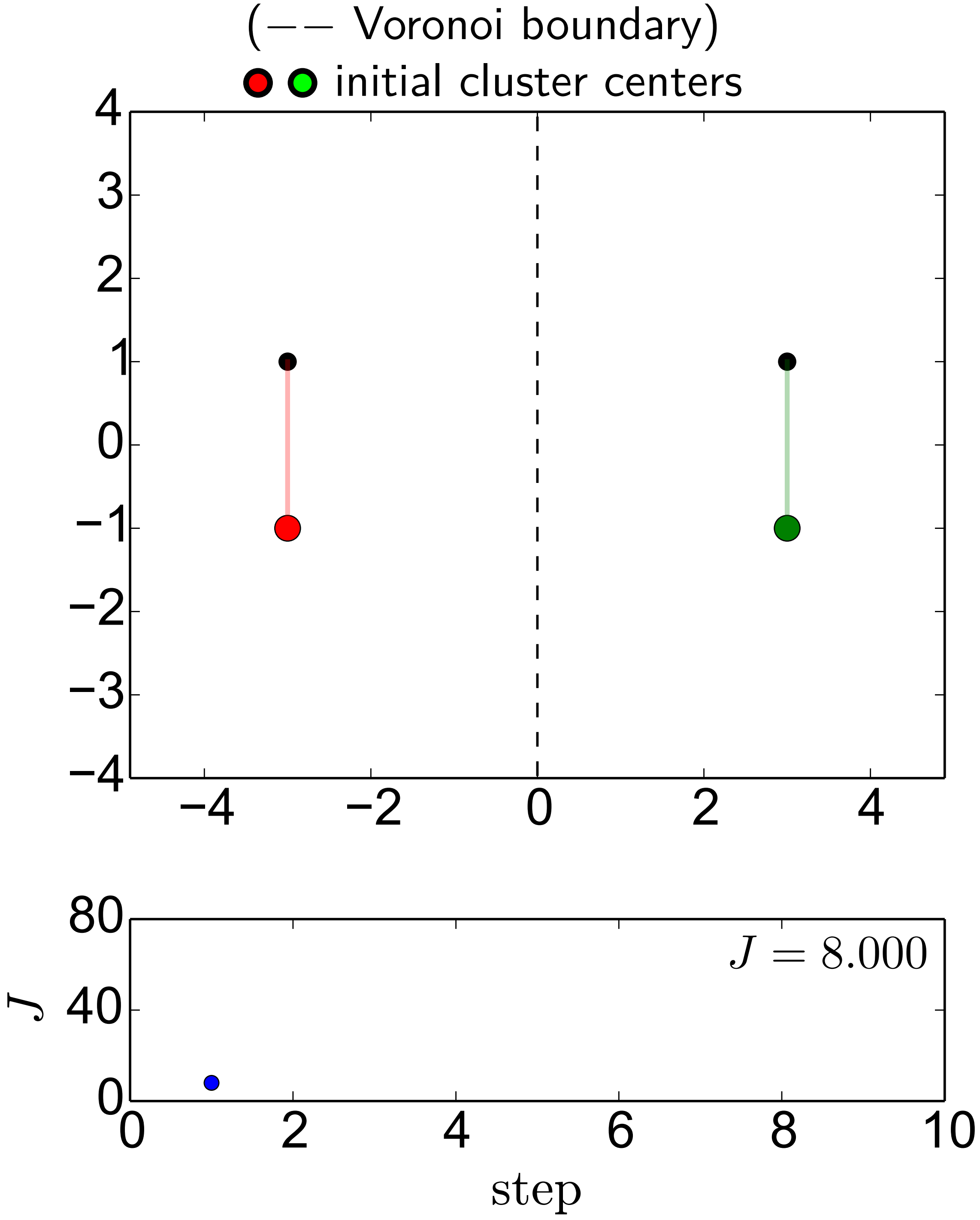
Cluster the data points to $K = 2$ clusters

Initial cluster centers (selected randomly from data points)

step 1, compute assignments
step 2, recompute centers
step 3, recompute assignments

Done.

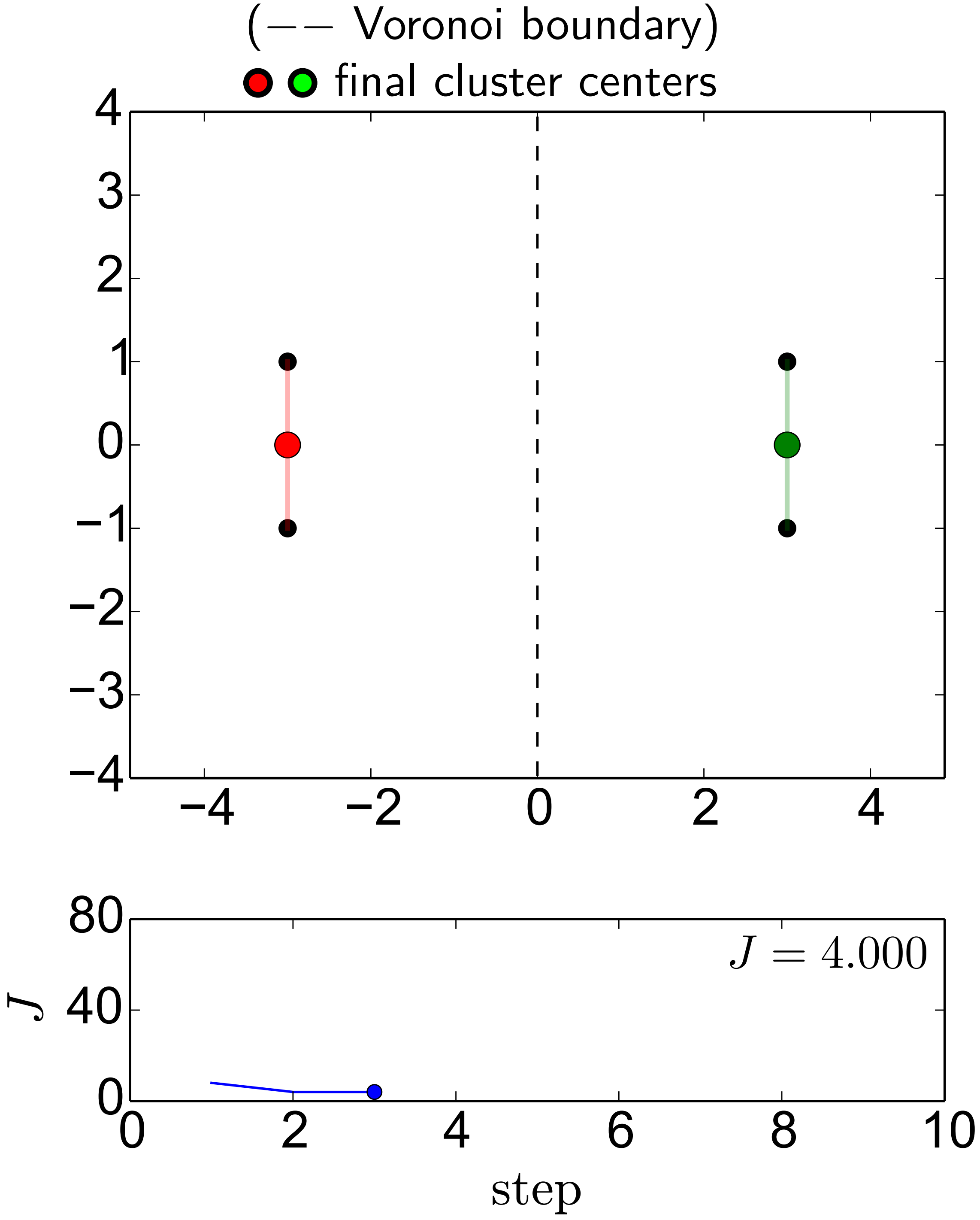
Different Initialization



Cluster the data points to $K = 2$ clusters

Initial cluster centers (selected randomly from data points)

Local Minimum 2, $J = 4$



Cluster the data points to $K = 2$ clusters

Initial cluster centers (selected randomly from data points)

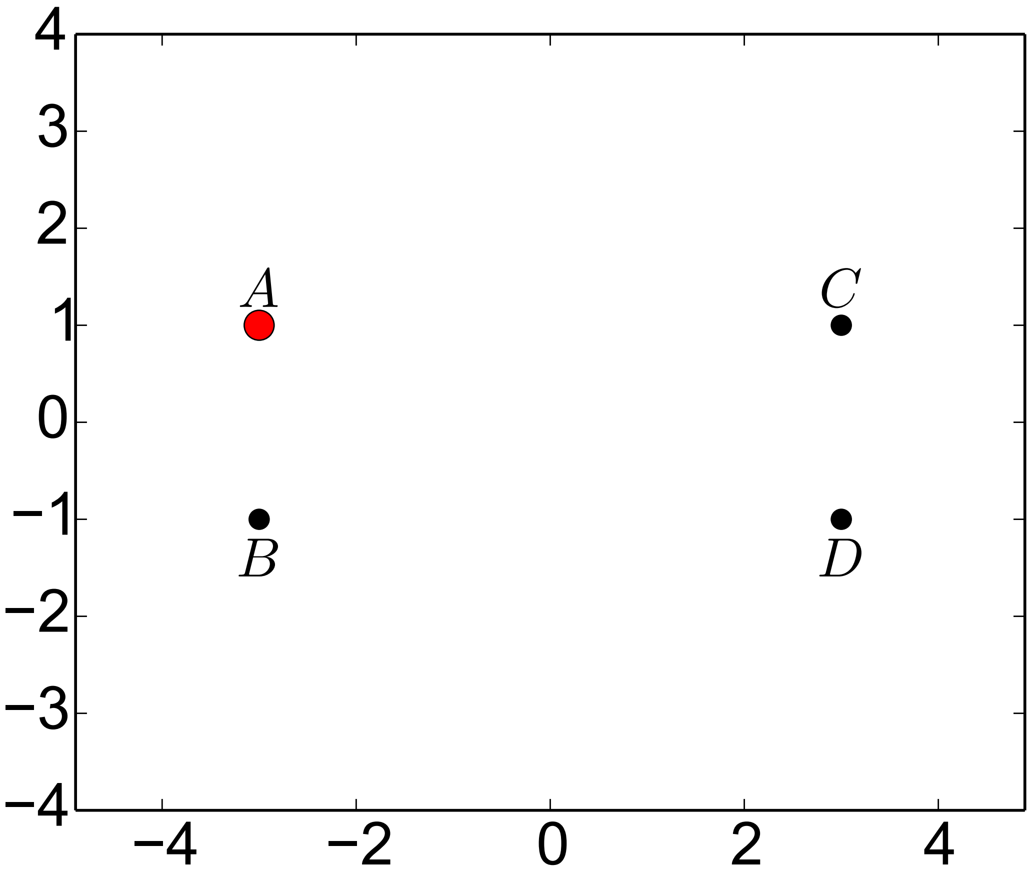
step 1, compute assignments
step 2, recompute centers
step 3, recompute assignments

Done.

This minimum is the global one, reaching the optimum $J_{\text{opt}} = 4$.

There is no other local minimum besides these two.

Which Local Minimum Will Be Reached?



This depends on initialization. Let us assume that A has been randomly selected from data as the first cluster centre (●).

If B is selected as the second cluster centre, the output will be Minimum 1 ($J = 36$).

If C or D is selected as the second cluster centre, the output will be Minimum 2 ($J = J_{\text{opt}} = 4$).

All of the points B, C, D have equal chance of being randomly selected as the second cluster centre. Thus, the algorithm outcome can be summarized as follows:

K-means output	value of J	odds
Minimum 1	36	$1/3$
Minimum 2	4	$2/3$

K-Means++

K-Means++ is the K-means with clever **initialization** of cluster centers. The motivation is to make initializations which make K-means more likely to end up in better local minima (minima with lower J).

K-Means++ uses the following randomized sampling strategy for constructing the initial cluster centers set $\mathcal{C}$:

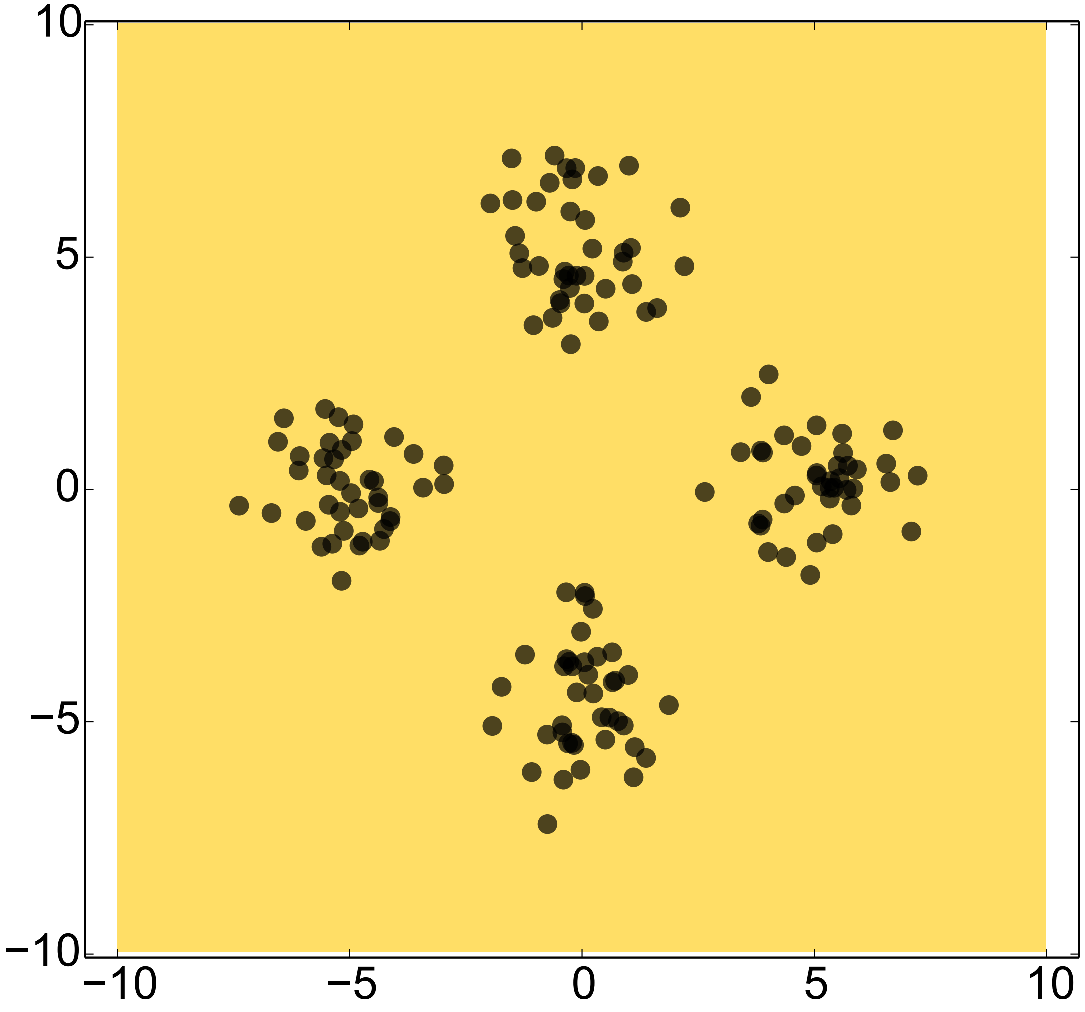
1. Choose the first cluster centre $\mathbf{c}_1$ uniformly at random from $\mathcal{T}$. Set $\mathcal{C} = \{\mathbf{c}_1\}$.
2. For each data point $\mathbf{x}_l$, compute the distance d_l to its nearest cluster in $\mathcal{C}$:

$$d_l = \min_{\mathbf{c} \in \mathcal{C}} \|\mathbf{x}_l - \mathbf{c}\| \qquad (\forall l = 1, 2, ..., L) \tag{8}$$

3. Select a point $\mathbf{x}_l$ from $\mathcal{T}$ with probability proportional to d_l^2. This involves constructing a distribution $p(l)$ from d_l as $p(l) = \frac{d_l^2}{\sum_{l=1}^{L} d_l^2}$ and sampling from it to get the index l.
4. $\mathcal{C} \leftarrow \mathcal{C} \cup \mathbf{x}_l$.
5. Stop if $|\mathcal{C}| = K$, otherwise goto 2.

After this initialization, standard K-means algorithm is employed.

K-means++, Example



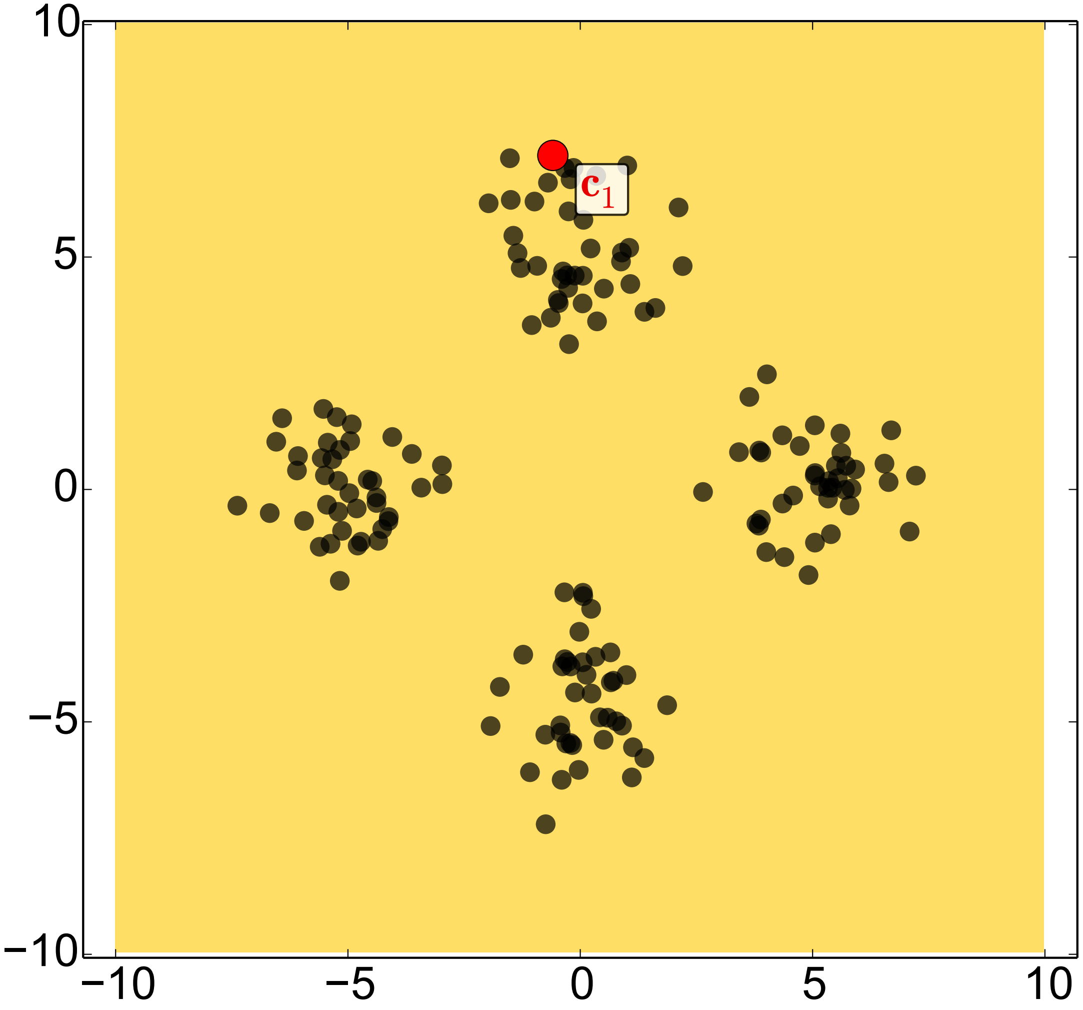
Sampling distribution $p(l)$ is shown as the scatter plot. Area of circle shown at $\mathbf{x}_l$ is proportional to $p(l)$.

Data: Points sampled from normal distribution with unit variance, at each of the following four positions (40 samples each):

$[-5, 0]$, $[5, 0]$, $[0, -5]$, $[0, 5]$.

Problem: Initialize $K = 4$ cluster centres using K-means++.

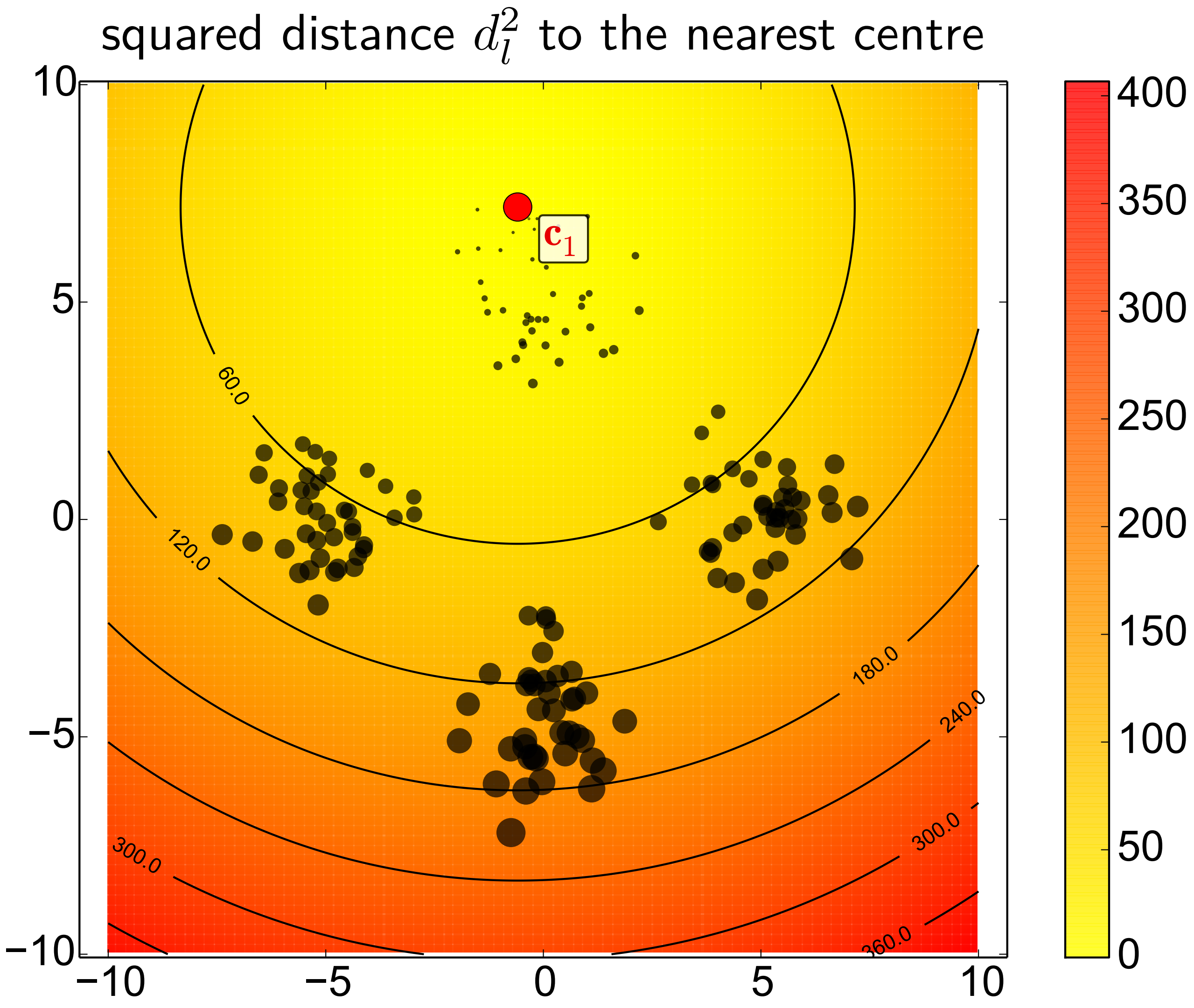
K-means++, Example



Sampling distribution $p(l)$ is shown as the scatter plot. Area of circle shown at $\mathbf{x}_l$ is proportional to $p(l)$.

– Select $\mathbf{c}_1$

K-means++, Example

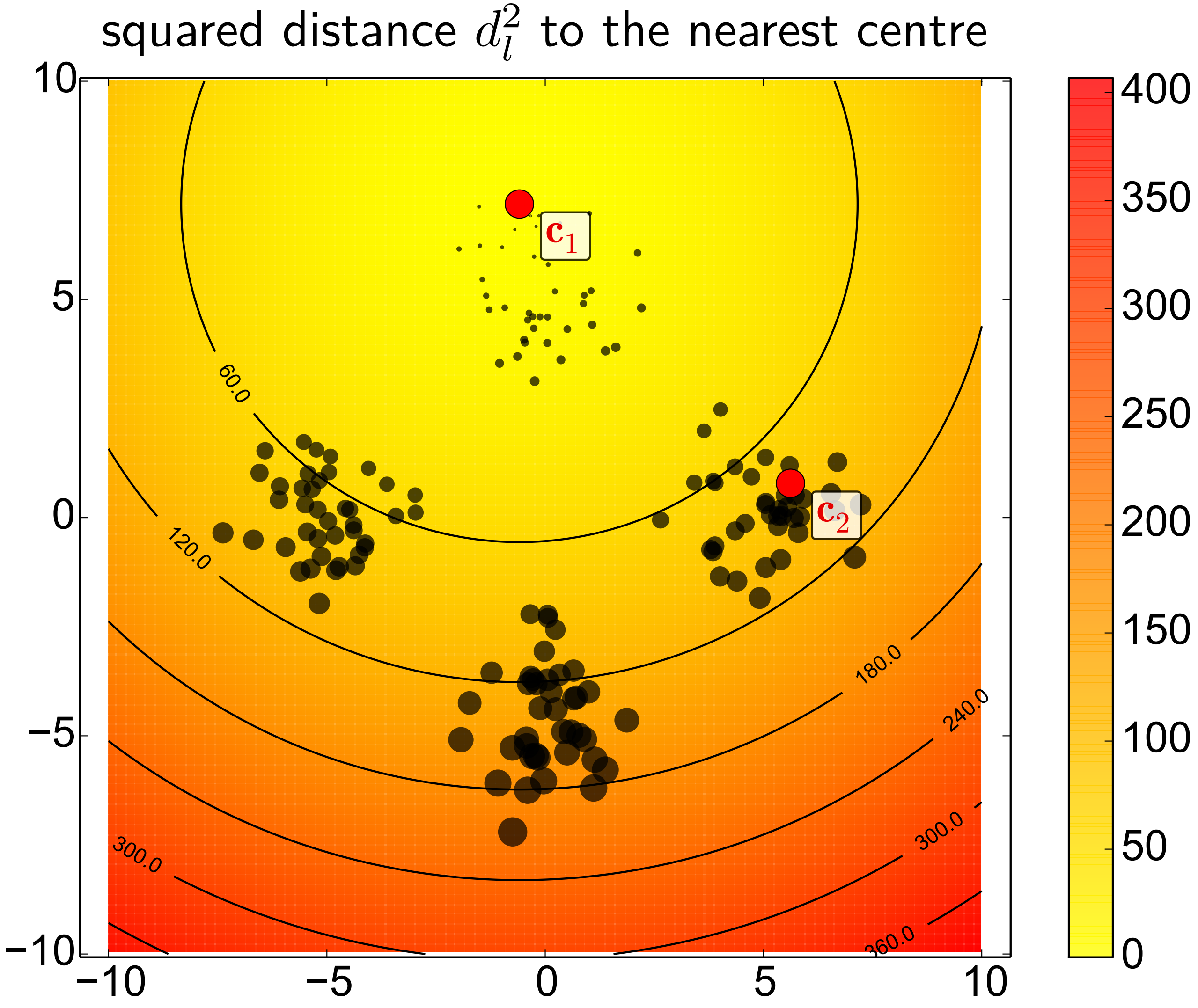


– Select $\mathbf{c}_1$

– update $p(l)$

Sampling distribution $p(l)$ is shown as the scatter plot. Area of circle shown at $\mathbf{x}_l$ is proportional to $p(l)$.

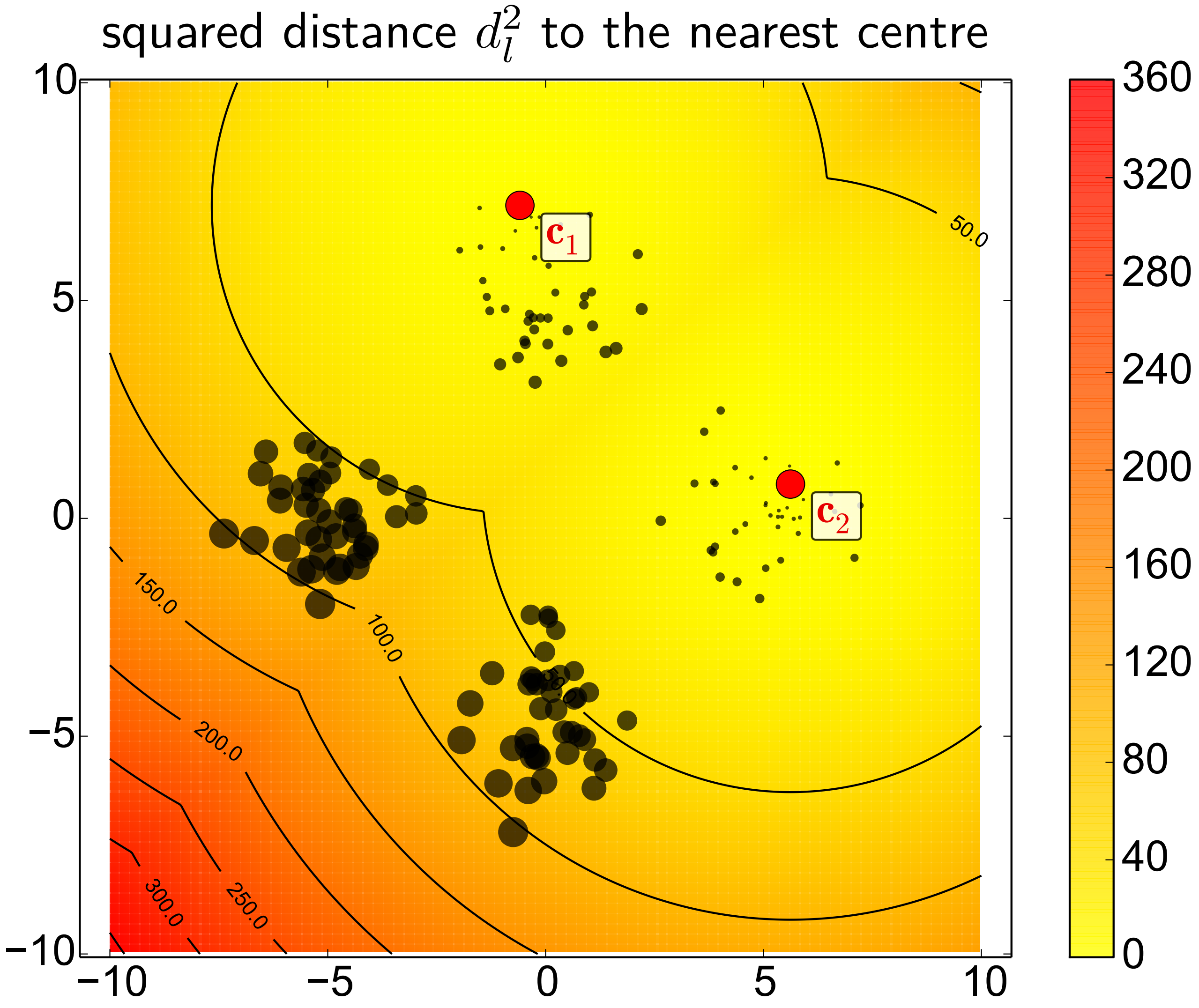
K-means++, Example



- Select $\mathbf{c}_1$
- update $p(l)$
- Select $\mathbf{c}_2$

Sampling distribution $p(l)$ is shown as the scatter plot. Area of circle shown at $\mathbf{x}_l$ is proportional to $p(l)$.

K-means++, Example

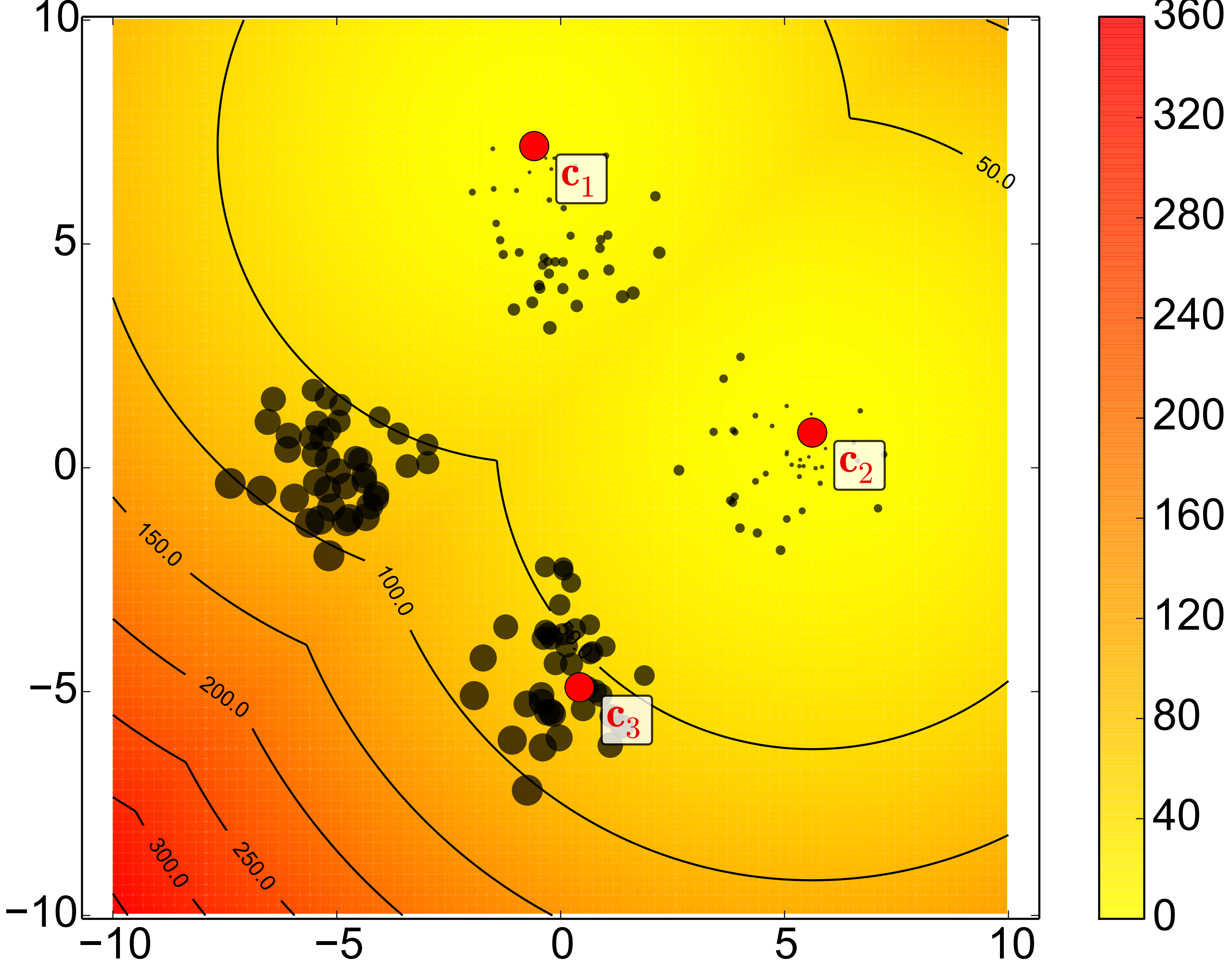


- – Select $\mathbf{c}_1$
- – update $p(l)$
- – Select $\mathbf{c}_2$
- – update $p(l)$

Sampling distribution $p(l)$ is shown as the scatter plot. Area of circle shown at $\mathbf{x}_l$ is proportional to $p(l)$.

K-means++, Example

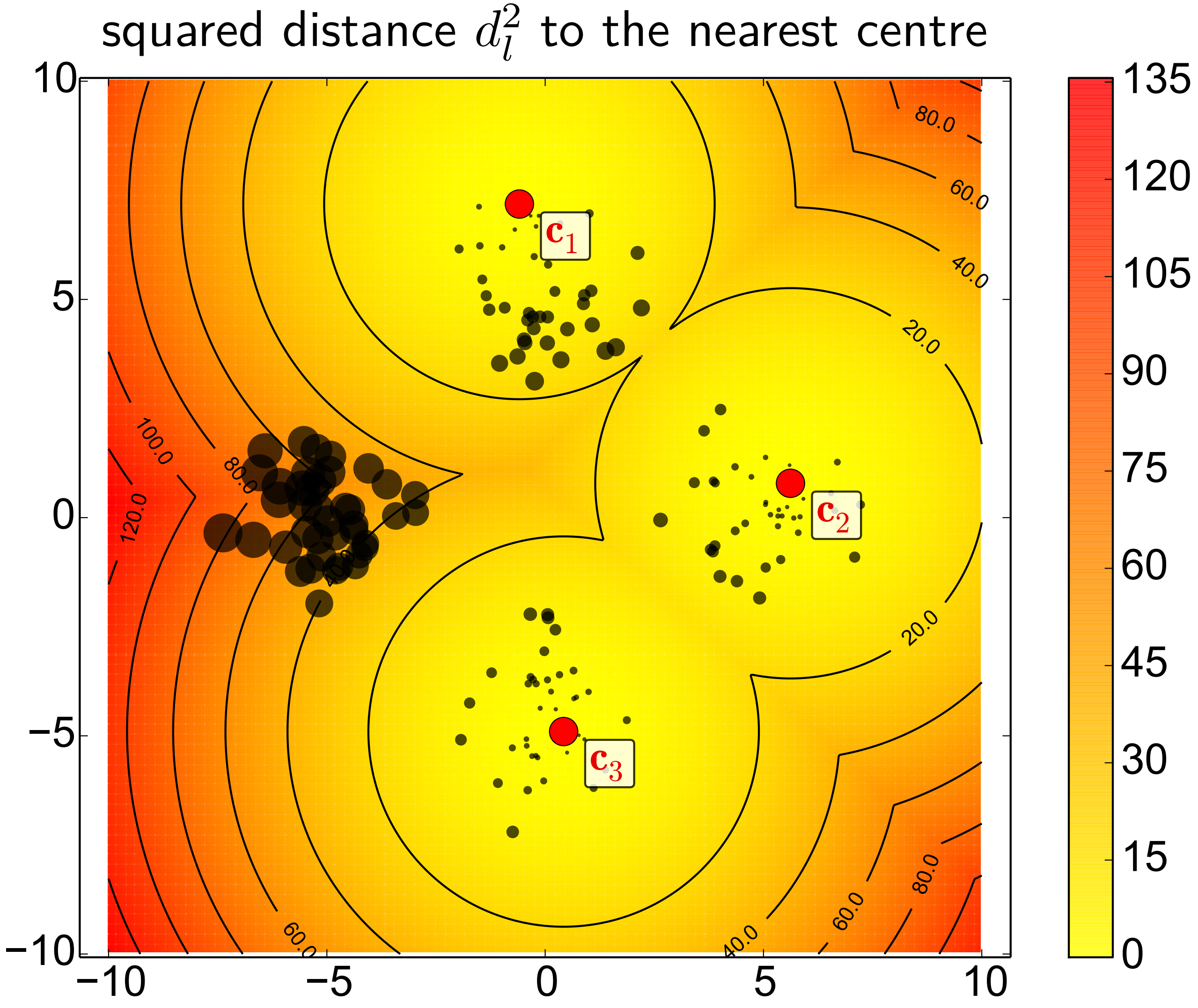
squared distance d_l^2 to the nearest centre



- Select $\mathbf{c}_1$
- update $p(l)$
- Select $\mathbf{c}_2$
- update $p(l)$
- Select $\mathbf{c}_3$

Sampling distribution $p(l)$ is shown as the scatter plot. Area of circle shown at $\mathbf{x}_l$ is proportional to $p(l)$.

K-means++, Example

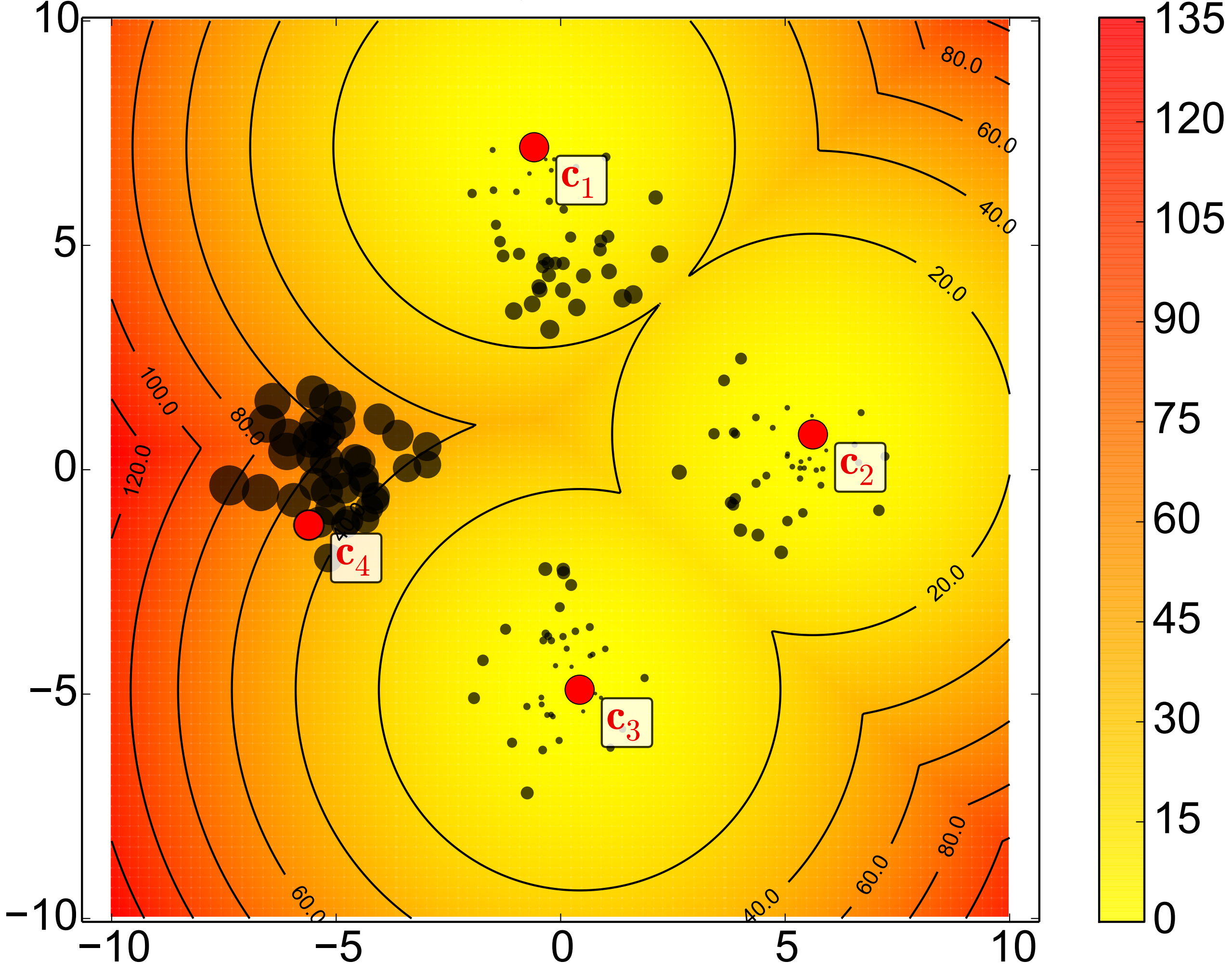


- Select $\mathbf{c}_1$
- update $p(l)$
- Select $\mathbf{c}_2$
- update $p(l)$
- Select $\mathbf{c}_3$
- update $p(l)$

Sampling distribution $p(l)$ is shown as the scatter plot. Area of circle shown at $\mathbf{x}_l$ is proportional to $p(l)$.

K-means++, Example

squared distance d_l^2 to the nearest centre



- Select $\mathbf{c}_1$
- update $p(l)$
- Select $\mathbf{c}_2$
- update $p(l)$
- Select $\mathbf{c}_3$
- update $p(l)$
- Select $\mathbf{c}_4$

Done.

Sampling distribution $p(l)$ is shown as the scatter plot. Area of circle shown at $\mathbf{x}_l$ is proportional to $p(l)$.

K-means++, Bound on $E(J)$

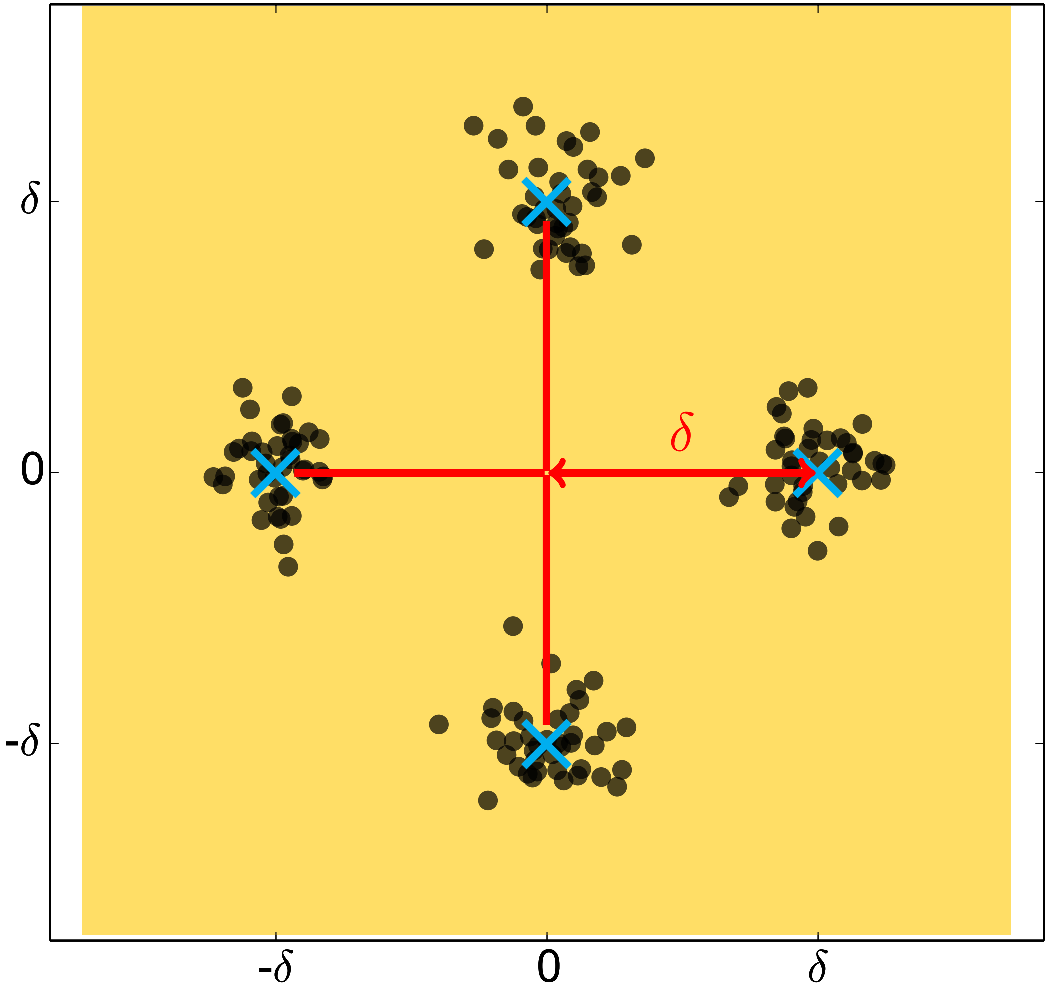
- The following bound on expectation $E(J)$ of the criterion value J exists when K-means++ is used for initialization:

  $$E(J) \leq 8(\ln K + 2) J_{\text{opt}} \tag{9}$$

  In the classical initialization (selecting all centres from data uniformly at random), no such bound exists.

- Arthur, D. and Vassilvitskii, S. (2007). *"k-means++: the advantages of careful seeding"*. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035.

K-means++, Effect on K-means outcome, Example 1



(shown for $\delta = 7$)

$J_{\text{opt}} = 289.7$

Data: points sampled from normal distribution with unit variance, at four different positions (40 samples each):

$$\mu_1 = [-\delta, 0]$$
$$\mu_2 = [\delta, 0]$$
$$\mu_3 = [0, \delta]$$
$$\mu_4 = [0, -\delta]$$

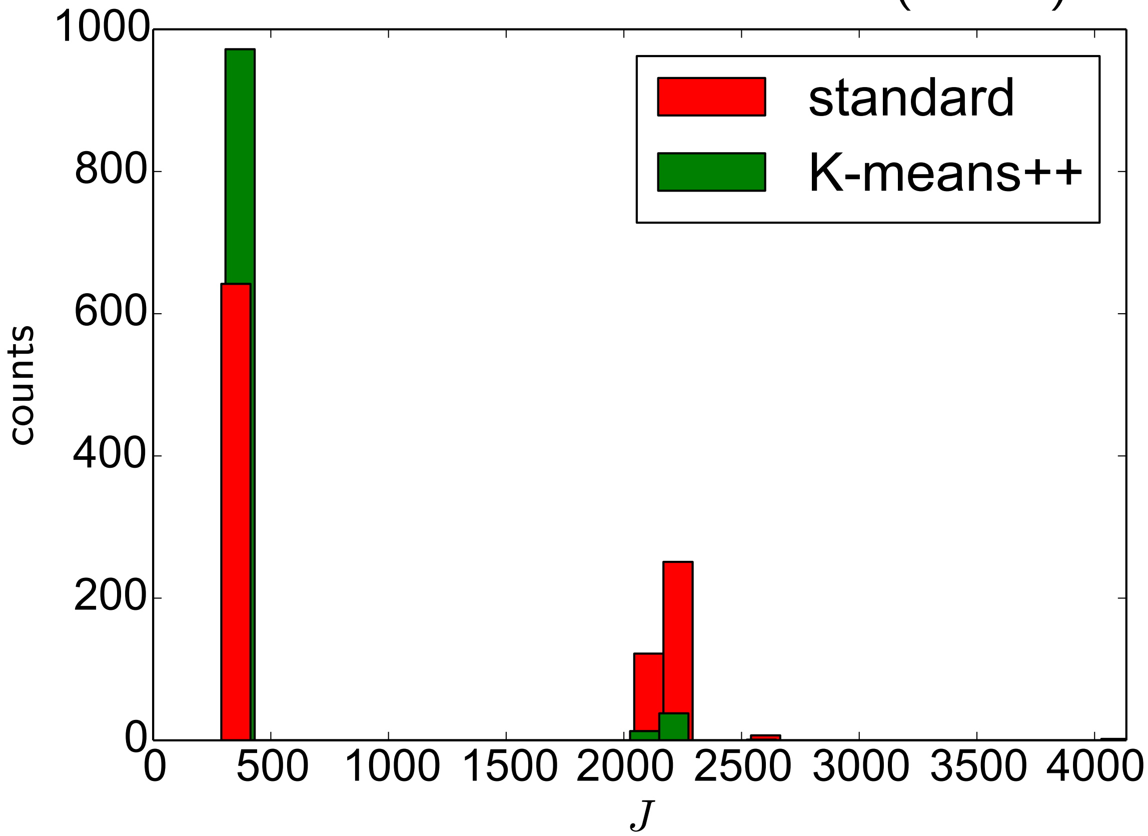
Experiment: Cluster the data repeatedly to $K = 4$ clusters, using (i) standard and (ii) K-means++ initializations. Store the values of J obtained in individual runs of K-means. Compare distributions of J for the two initializations.

K-means++, Effect on K-means outcome, Example 1

Results (for $\delta = 7$):

	J_{mean}	J_{min}	J_{max}
standard init.	1002	289.7	4135
K-means++	386.5	289.7	2637

histogram of values of J obtained across 1024 runs of K-means ($\delta = 7$)



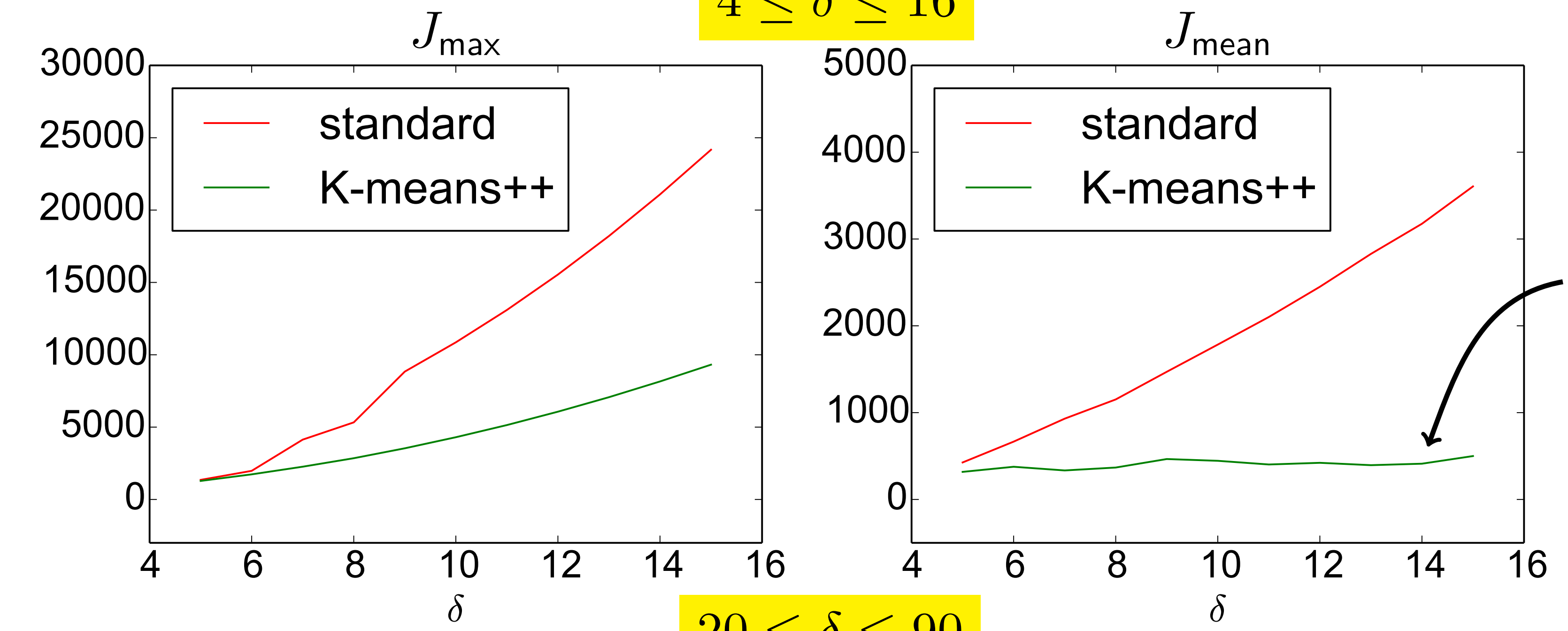
Things to note:

- both initialization methods found the optimal clustering and reach $J_{\text{opt}} = 289.7$
- K-means++ achieved better clustering on average (lower J_{mean})
- K-means++ also achieved better worst case (lower J_{max})

K-means++, Effect on K-means outcome, Example 1

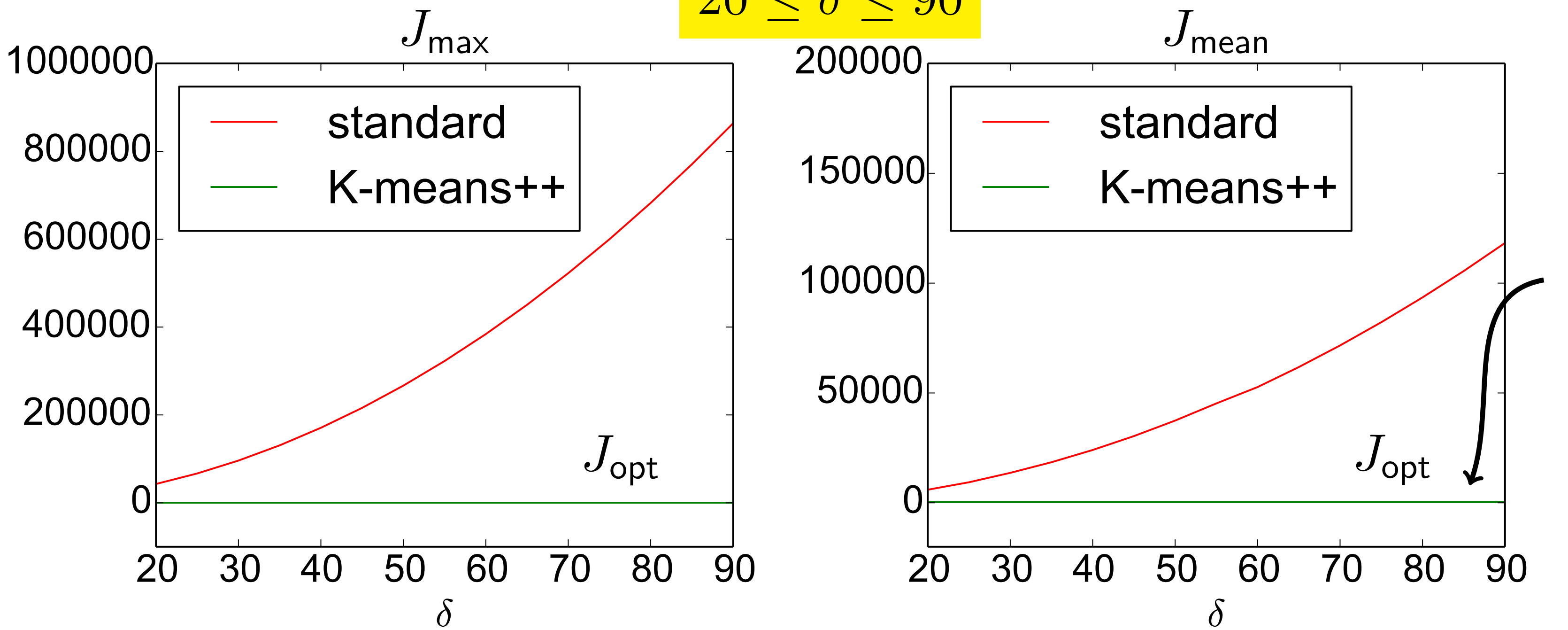
Dependence on δ. Results obtained by running K-means 128× for each δ (Note: $J_{\text{min}} = J_{\text{opt}}$ for all δ's and both methods.)

$4 \leq \delta \leq 16$



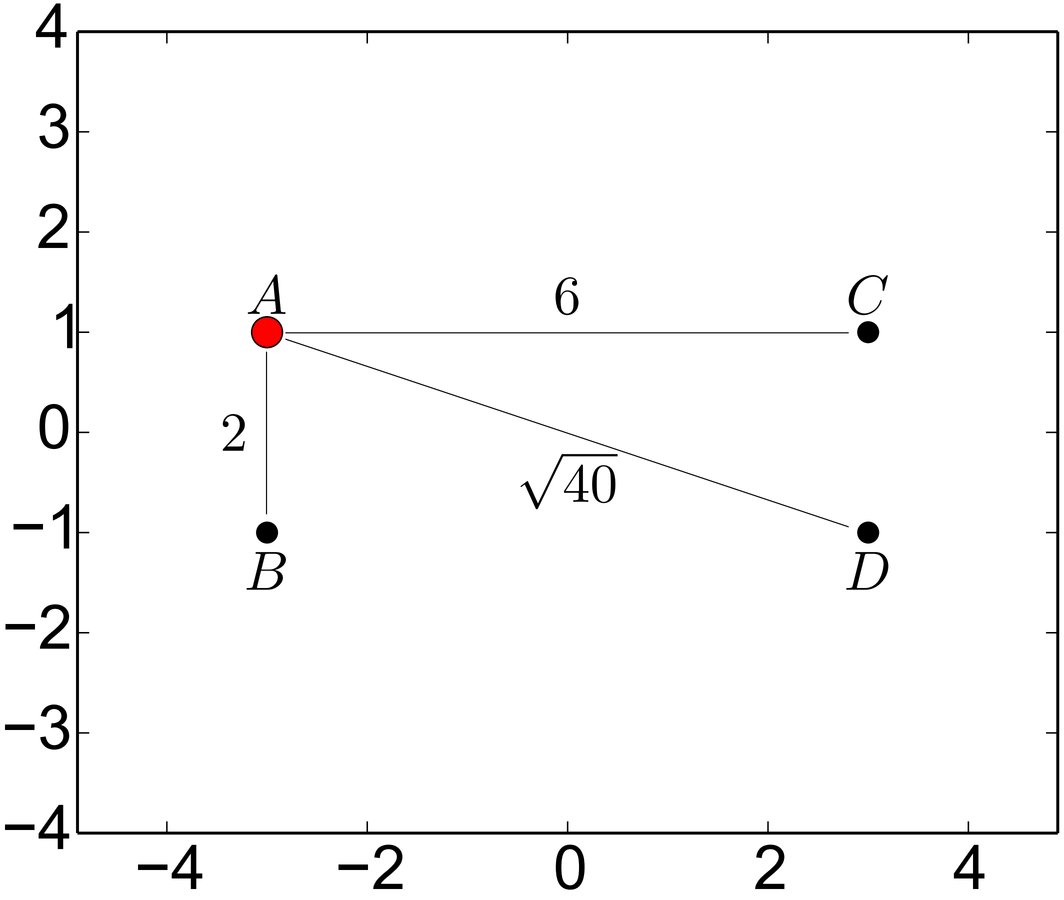
Note J_{mean} stays low for K-means++

$20 \leq \delta \leq 90$



Probability of generating initialization resulting in non-optimal outcome is so low that **no** non-optimal outcome is encountered across the 128 runs.

K-means++, Effect on K-means outcome, Example 2



Suppose A has been selected as the first cluster centre (●).

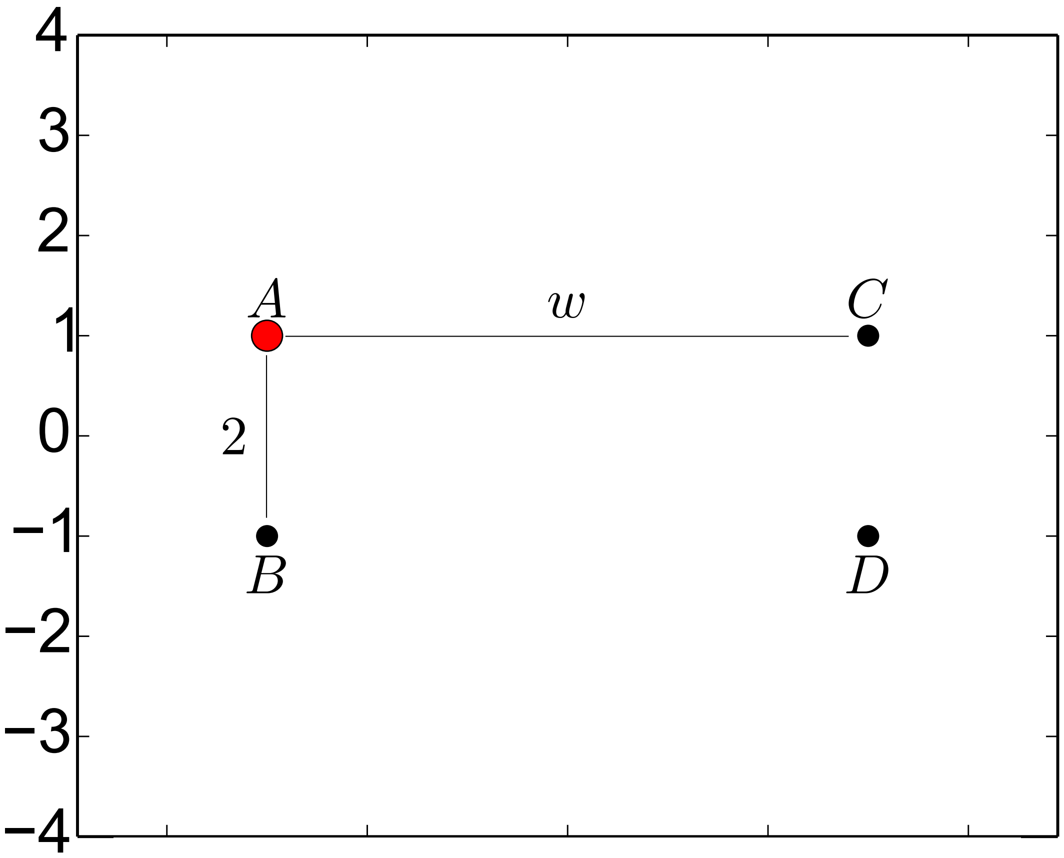
The points B, C, D will be selected to be the second cluster centre with odds $B : C : D = 4 : 36 : 40$. Hence the probabilities of being selected are:

$$p(B) = 1/20,$$
$$p(C) = 9/20,$$
$$p(D) = 1/2.$$

The algorithm outcome can be summarized as follows:

K-means output	value of J	odds (K-means++)	odds (standard init.)
Minimum 1	36	**1/20**	1/3
Minimum 2	4	**19/20**	2/3

K-means++, Effect on K-means outcome, Example 2



Suppose we let the points C and D go further away from A and B, with $|AC| = |BD| = w \quad (w \geq 2)$.

Using the same arguments as before,

$$p(B) = 4/Z,$$
$$p(C) = w^2/Z,$$
$$p(D) = (w^2 + 4)/Z,$$

(Z is the normalization constant), and we arrive at the result summarized in this table:

K-means output	value of J	odds (K-means++)	odds (standard init.)
Minimum 1	w^2	$\frac{2}{w^2+4}$	$1/3$
Minimum 2	4	$1 - \frac{2}{w^2+4}$	$2/3$

K-means++, Effect on K-means outcome, Example 2

The expectation $E(J)$ of J is $(w \geq 2)$:

$$E(J) = w^2 \frac{2}{w^2+4} + 4\left(1 - \frac{2}{w^2+4}\right) = 6 - \frac{16}{w^2+4} \qquad \text{(K-means++)} \qquad (10)$$

$$E(J) = \frac{w^2+8}{3} \qquad \text{(standard initialization)} \qquad (11)$$

There is a striking difference between the two as w increases:

	K-means++	standard init.
$E(J)$ for $w = 2$	4	4
$E(J)$ for $w = 6$	5.6	14.7
$E(J)$ for $w \to \infty$	6	∞

K-means output	value of J	odds (K-means++)	odds (standard init.)
Minimum 1	w^2	$\frac{2}{w^2+4}$	$1/3$
Minimum 2	4	$1 - \frac{2}{w^2+4}$	$2/3$

K-means Generalizations (K-medians, K-medoids, . . .)

K-means can be generalized for minimizing criterion other than squared Euclidean.

Given:

$\mathcal{T} = \{\mathbf{x}_l\}_{l=1}^{L}$	the set of observations
K	the desired number of cluster prototypes
$d(\cdot,\cdot)$	'distance function' (not necessarily a metric)

Output:

$\{\mathbf{c}_k\}_{k=1}^{K}$	the set of cluster prototypes (etalons)
$\{\mathcal{T}_k\}_{k=1}^{K}$	the clustering (partitioning) of the data
	$\cup_{k=1}^{K}\mathcal{T}_k = \mathcal{T}, \mathcal{T}_i \cap \mathcal{T}_j =$ for $i \neq j$

1. Initialize the cluster centres $\{\mathbf{c}_k\}_{k=1}^{K}$ (e.g. by random selection from the data points $\mathcal{T}$, without replacement)
2. Assignment optimization (assign to closest etalon):

$$\mathcal{T}_k = \{\mathbf{x} \in \mathcal{T} : \forall j, \quad d(\mathbf{x}, \mathbf{c}_k) \leq d(\mathbf{x}, \mathbf{c}_j)\} \qquad (\forall k = 1, 2, ..., K) \tag{12}$$

3. Prototype optimization:

$$\mathbf{c}_k = \begin{cases} \operatorname*{argmin}_{\mathbf{c}} \sum_{\mathbf{x} \in \mathcal{T}_k} d(\mathbf{x}, \mathbf{c}) & \text{if } |\mathcal{T}_k| > 0 \\ \text{re-initialize} & \text{if } \mathcal{T}_k = \emptyset \end{cases} \qquad (\forall k = 1, 2, ..., K) \tag{13}$$

4. Terminate if $\forall k : \mathcal{T}_k^{t+1} = \mathcal{T}_k^{t}$, otherwise goto 2

K-means Generalization: K-medians

Given:

$\mathcal{T} = \{\mathbf{x}_l\}_{l=1}^{L}$ the set of observations, $\mathbf{x} \in \mathbb{R}^D$

K the desired number of cluster prototypes

$d(\cdot,\cdot)$ $\|\mathbf{c} - \mathbf{x}\|_1 = \sum_{i=1}^{D} |c_i - x_i|$ (L_1 metric)

Output:

$\{\mathbf{c}_k\}_{k=1}^{K}$ the set of cluster prototypes (etalons)

$\{\mathcal{T}_k\}_{k=1}^{K}$ the clustering (partitioning) of the data

$\cup_{k=1}^{K} \mathcal{T}_k = \mathcal{T}, \mathcal{T}_i \cap \mathcal{T}_j =$ for $i \neq j$

1. Initialize the cluster centres $\{\mathbf{c}_k\}_{k=1}^{K}$ (e.g. by random selection from the data points $\mathcal{T}$, without replacement)
2. Assignment optimization (assign to closest etalon):

$$\mathcal{T}_k = \{\mathbf{x} \in \mathcal{T} : \forall j, \quad d(\mathbf{x}, \mathbf{c}_k) \leq d(\mathbf{x}, \mathbf{c}_j)\} \qquad (\forall k = 1, 2, ..., K) \tag{14}$$

3. Prototype optimization:

$$\mathbf{c}_k = \begin{cases} \text{median}\{\mathcal{T}_k\} & \\ \text{re-initialize} & \text{if } \mathcal{T}_k = \emptyset \end{cases} \qquad (\forall k = 1, 2, ..., K) \tag{15}$$

4. Terminate if $\forall k : \mathcal{T}_k^{t+1} = \mathcal{T}_k^{t}$, otherwise goto 2

K-means Generalization: Clustering Strings

Given:

$\mathcal{T} = \{\mathbf{x}_l\}_{l=1}^{L}$ observations are strings
K the desired number of cluster prototypes
$d(\mathbf{s}_1, \mathbf{s}_2)$ Levenshtein distance, number of edit operations to transform $\mathbf{s}_1$ to $\mathbf{s_2}$

Output:

$\{\mathbf{c}_k\}_{k=1}^{K}$ the set of cluster prototypes (etalons)
$\{\mathcal{T}_k\}_{k=1}^{K}$ the clustering (partitioning) of the data
$\cup_{k=1}^{K} \mathcal{T}_k = \mathcal{T}, \mathcal{T}_i \cap \mathcal{T}_j =$ for $i \neq j$

1. Initialize the cluster centres $\{\mathbf{c}_k\}_{k=1}^{K}$ (e.g. by random selection from the data points $\mathcal{T}$, without replacement)
2. Assignment optimization (assign to closest etalon):

$$\mathcal{T}_k = \{\mathbf{x} \in \mathcal{T} : \forall j, \quad d(\mathbf{x}, \mathbf{c}_k) \leq d(\mathbf{x}, \mathbf{c}_j)\} \qquad (\forall k = 1, 2, ..., K) \tag{16}$$

3. Prototype optimization:

$$\mathbf{c}_k = \begin{cases} \operatorname*{argmin}_{\mathbf{c}} \sum_{\mathbf{x} \in \mathcal{T}_k} d(\mathbf{x}, \mathbf{c}) & \text{if } |\mathcal{T}_k| > 0 \\ \text{re-initialize} & \text{if } \mathcal{T}_k = \emptyset \end{cases} \qquad (\forall k = 1, 2, ..., K) \tag{17}$$

4. Terminate if $\forall k : \mathcal{T}_k^{t+1} = \mathcal{T}_k^{t}$, otherwise goto 2

K-means Generalization: Clustering Strings, Notes

- the calculation of $d(\cdot, \cdot)$ may be non trivial
- it may be hard to minimize $\sum_{\mathbf{x} \in \mathcal{T}_k} d(\mathbf{x}, \mathbf{c})$ over the space of all strings. The minimization may be restricted to $\mathbf{c} \in \mathcal{T}$.
- is the algorithm guaranteed to terminate if step 2 (step 3) is only improving J, not finding the minimum (given $\mathcal{T}_k$ or $\mathbf{c}_k$), respectively?

K-means Generalization: Euclidean Clustering

Given:

$\mathcal{T} = \{\mathbf{x}_l\}_{l=1}^{L}$	the set of observations, $\mathbf{x} \in \mathbb{R}^D$
K	the desired number of cluster prototypes
$d(\cdot,\cdot)$	$\|\mathbf{c} - \mathbf{x}\|$ (L_2 metric)

Output:

$\{\mathbf{c}_k\}_{k=1}^{K}$	the set of cluster prototypes (etalons)
$\{\mathcal{T}_k\}_{k=1}^{K}$	the clustering (partitioning) of the data
	$\cup_{k=1}^{K} \mathcal{T}_k = \mathcal{T}, \mathcal{T}_i \cap \mathcal{T}_j =$ for $i \neq j$

1. Initialize the cluster centres $\{\mathbf{c}_k\}_{k=1}^{K}$ (e.g. by random selection from the data points $\mathcal{T}$, without replacement)
2. Assignment optimization (assign to closest etalon):

$$\mathcal{T}_k = \{\mathbf{x} \in \mathcal{T} : \forall j, \quad d(\mathbf{x}, \mathbf{c}_k) \leq d(\mathbf{x}, \mathbf{c}_j)\} \qquad (\forall k = 1, 2, ..., K) \tag{18}$$

3. Prototype optimization: no closed-form solution for *geometric median*. Use e.g. iterative Weiszfeld's algorithm.

$$\mathbf{c}_k = \begin{cases} \operatorname*{argmin}_{\mathbf{c}} \sum_{\mathbf{x} \in \mathcal{T}_k} \|\mathbf{x} - \mathbf{c}\| & \text{if } |\mathcal{T}_k| > 0 \\ \text{re-initialize} & \text{if } \mathcal{T}_k = \emptyset \end{cases} \qquad (\forall k = 1, 2, ..., K) \tag{19}$$

4. Terminate if $\forall k : \mathcal{T}_k^{t+1} = \mathcal{T}_k^{t}$, otherwise goto 2

Weiszfeld Algorithm for Computing Geometric Median

- Uses iteratively re-weighted least squares
- Given $\mathbf{x}_i \in \mathbb{R}^D$ $(i = 1, 2, .., I)$, the geometric median $\mathbf{m} \in \mathbb{R}^D$:

$$\mathbf{m} = \operatorname*{argmin}_{\mathbf{m}'} \sum_{i=1}^{I} \|\mathbf{x}_i - \mathbf{m}'\| \,. \tag{20}$$

- Algorithm:

1. $t = 0$. Initialize $\mathbf{m}^{(0)}$ (e. g. take the mean of $\mathbf{x}_i$'s)
2. Compute weights w_i:

$$w_i = \frac{1}{\|\mathbf{x}_i - \mathbf{m}^{(t)}\|} \qquad (i = 1, 2, ..., I) \tag{21}$$

3. Obtain new estimate for $\mathbf{m}$ as a weighted average of $\mathbf{x}_i$'s:

$$\mathbf{m}^{(t+1)} = \frac{\sum_{i=1}^{I} w_i \mathbf{x}_i}{\sum_{i=1}^{I} w_i} \tag{22}$$

4. Finish if the termination condition is met. Otherwise, $t \leftarrow t + 1$ and goto 2.