Machine Learning for Data Science (CS4786)
Lecture 8

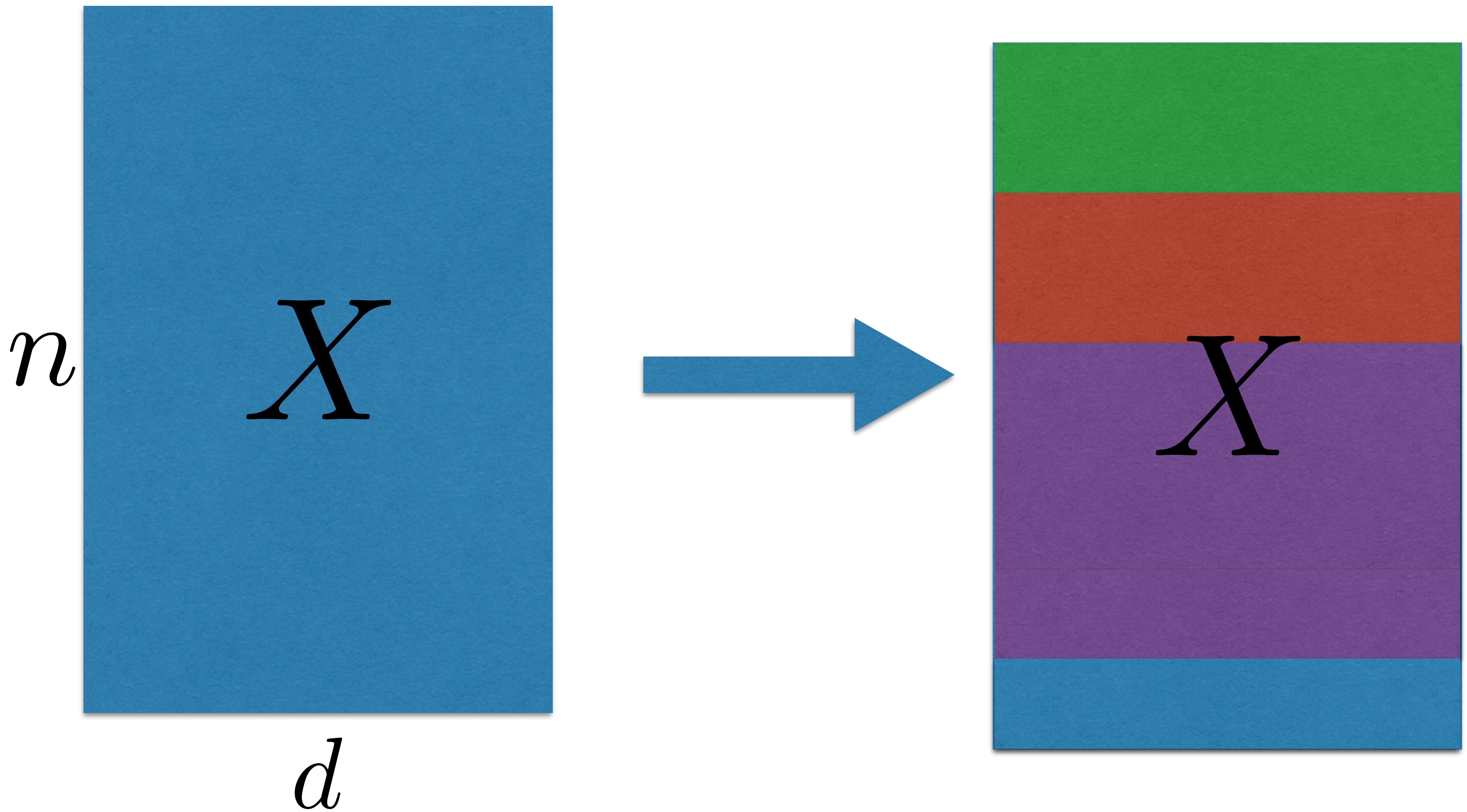
Clustering

Course Webpage :
http://www.cs.cornell.edu/Courses/cs4786/2016fa/

Announcement

- Those of you who submitted HW1 and are still on waitlist email me.

Clustering



CLUSTERING

- Grouping sets of data points s.t.
  - points in same group are similar
  - points in different groups are dissimilar

- A form of unsupervised classification where there are no predefined labels

Some Notations

- Kary clustering is a partition of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ into K groups
- For now assume the magical K is given to use
- Clustering given by $C_1, \ldots, C_K$, the partition of data points.
- Given a clustering, we shall use $c(\mathbf{x}_t)$ to denote the cluster identity of point $\mathbf{x}_t$ according to the clustering.
- Let n_j denote $|C_j|$, clearly $\sum_{j=1}^{K} n_j = n$.

How do we formalize a good clustering objective?

How do we formalize?

Say dissimilarity$(\mathbf{x}_t, \mathbf{x}_s)$ measures dissimilarity between $\mathbf{x}_t$ & $\mathbf{x}_s$

Given two clustering $\{C_1, \ldots, C_K\}$ (or c) and $\{C'_1, \ldots, C'_K\}$ (or c')

How do we decide which is better?

- points in same cluster are not dissimilar
- points in different clusters are dissimilar

Clustering Criterion

- Minimize total within-cluster dissimilarity

$$M_1 = \sum_{j=1}^{K} \sum_{s,t \in C_j} \text{dissimilarity}(x_t, x_s)$$

- Maximize between-cluster dissimilarity

$$M_2 = \sum_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \text{dissimilarity}(x_t, x_s)$$

- Maximize smallest between-cluster dissimilarity

$$M_3 = \min_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \text{dissimilarity}(x_t, x_s)$$

- Minimize largest within-cluster dissimilarity

$$M_4 = \max_{j \in [K]} \max_{s,t \in C_j} \text{dissimilarity}(x_t, x_s)$$

Clustering Criterion

- Minimize average dissimilarity within cluster

$$M_6 = \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{s \in C_j} \text{dissimilarity}\left(\mathbf{x}_s, C_j\right)$$

$$= \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{s \in C_j} \left(\sum_{t \in C_j, t \neq s} \text{dissimilarity}\left(\mathbf{x}_s, \mathbf{x}_t\right) \right)$$

$$= \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{s \in C_j} \left(\sum_{t \in C_j, t \neq s} \left\| \mathbf{x}_s - \mathbf{x}_t \right\|_2^2 \right)$$

- Minimize within-cluster variance: $\mathbf{r}_j = \frac{1}{n_j} \sum_{\mathbf{x} \in C_j} \mathbf{x}$

$$M_5 = \sum_{j=1}^{K} \sum_{t \in C_j} \left\| \mathbf{x}_t - \mathbf{r}_j \right\|_2^2$$

How different are these criteria?

Clustering Criterion

- minimizing $M_1 \equiv$ maximizing M_2
- minimizing $M_5 \equiv$ minimizing M_6

Clustering

- Multiple clustering criteria all equally valid
- Different criteria lead to different algorithms/solutions
- Which notion of distances or costs we use matter

Lets build algorithm for two criteria

1 $$M_5 = \sum_{j=1}^{K} \sum_{t \in C_j} \left\| \mathbf{x}_t - \mathbf{r}_j \right\|_2^2$$

2 $$M_3 = \min_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \mathrm{dissimilarity}(x_t, x_s)$$

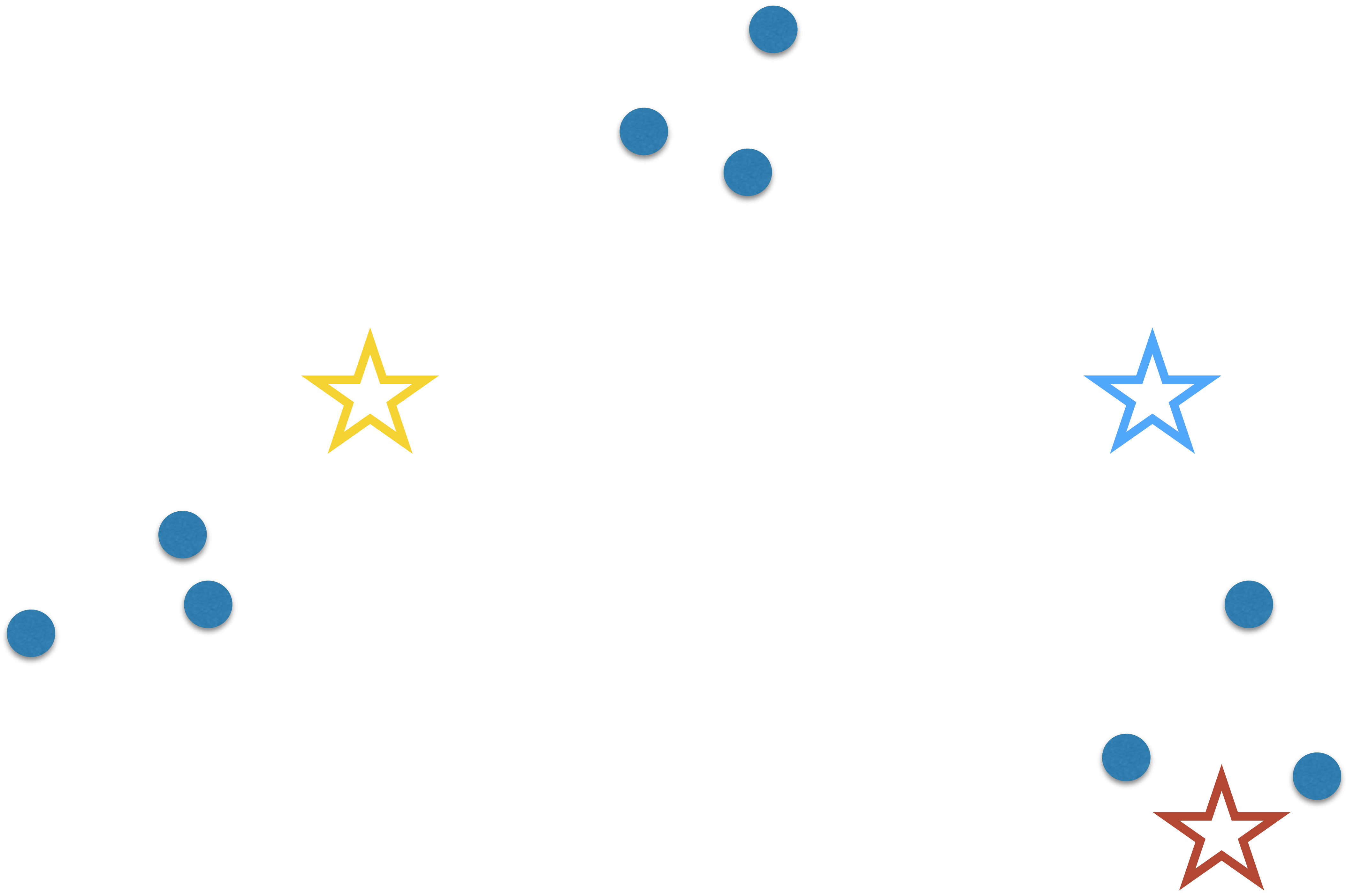
Lets build an Algorithm

$$M_5 = \sum_{j=1}^{K} \sum_{t \in C_j} \left\| \mathbf{x}_t - \mathbf{r}_j \right\|_2^2$$

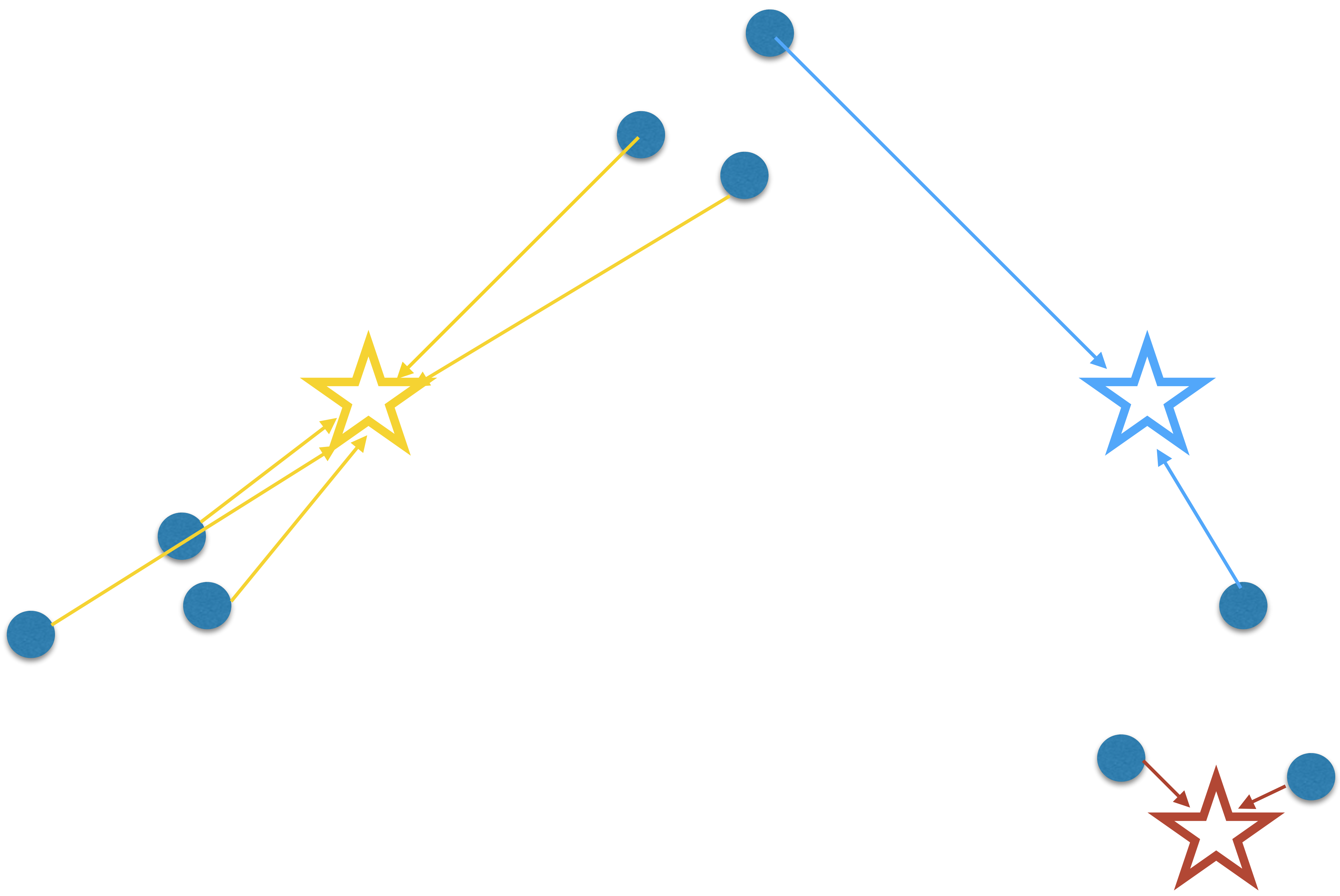
$$\text{where } \mathbf{r}_j = \frac{1}{|C_j|} \sum_{t \in C_j} \mathbf{x}_t$$

Demo

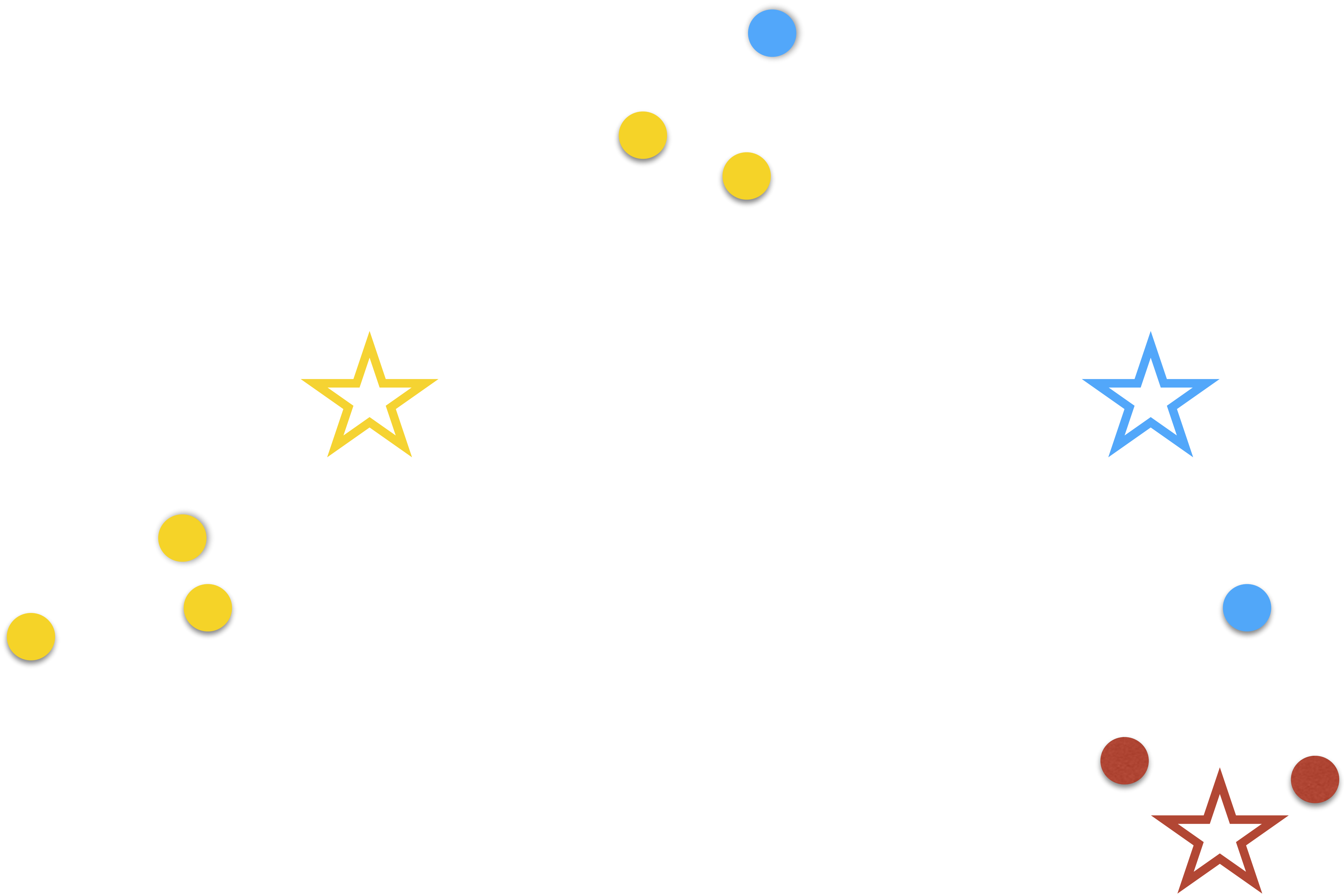
Demo



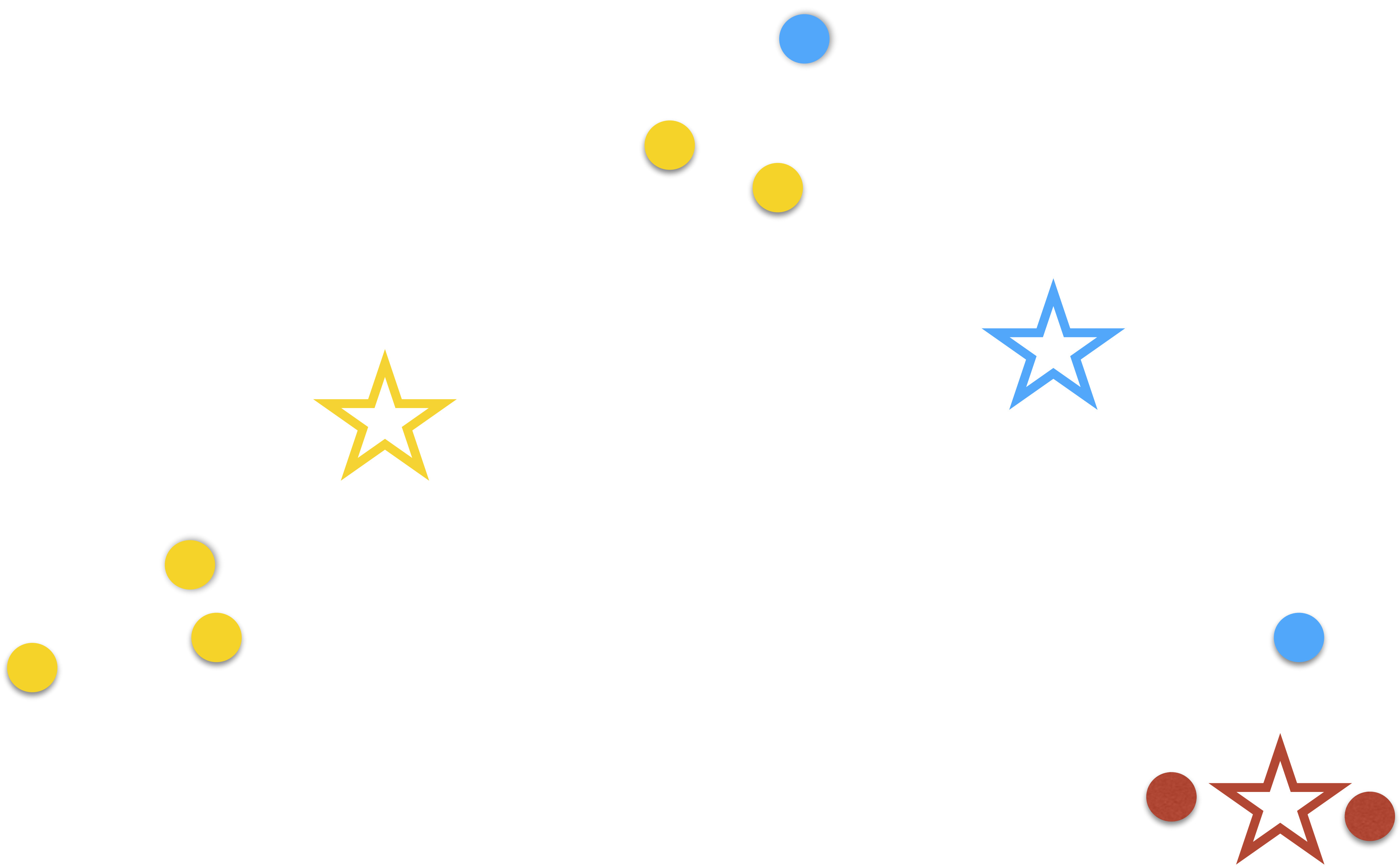
Demo



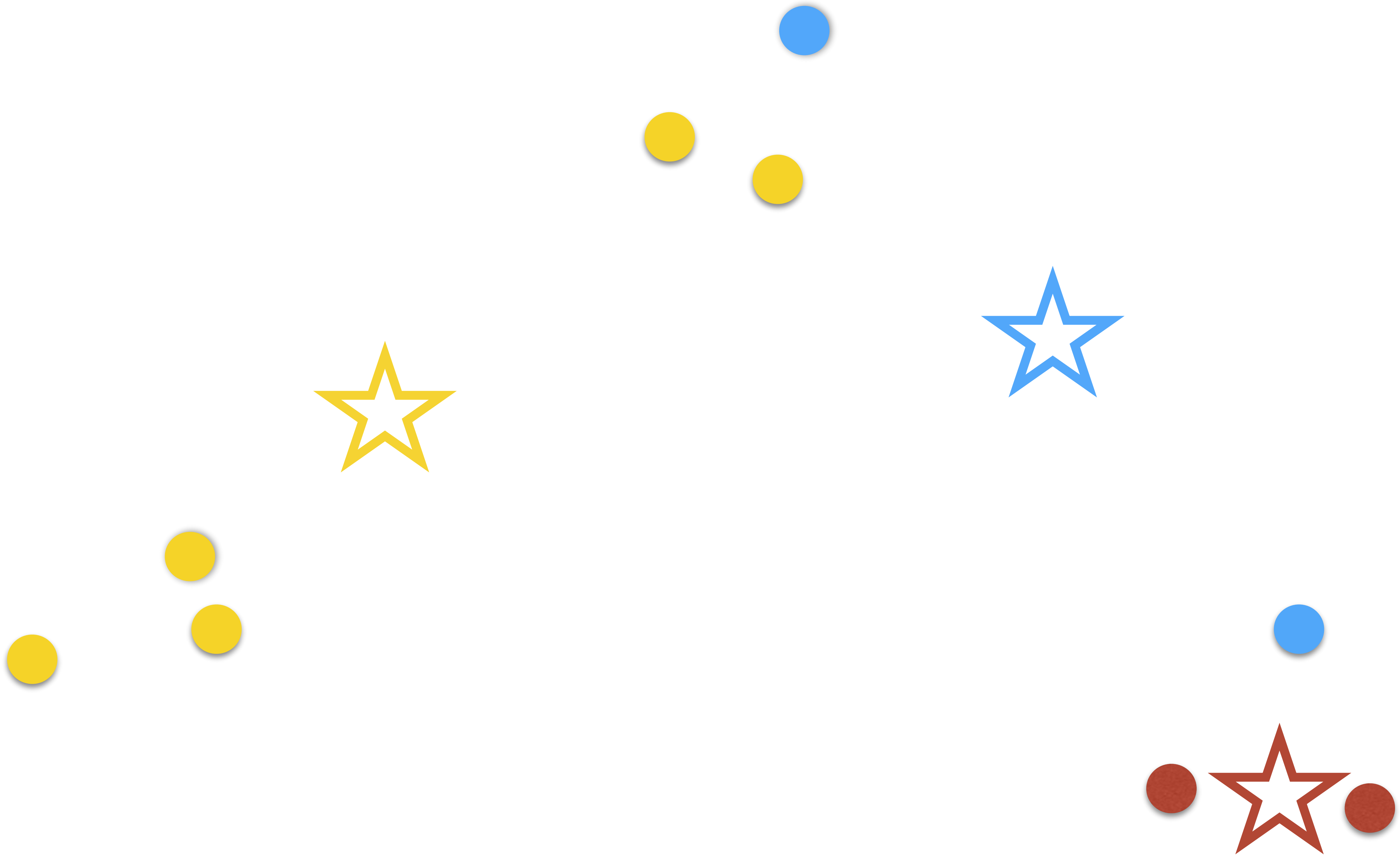
Demo



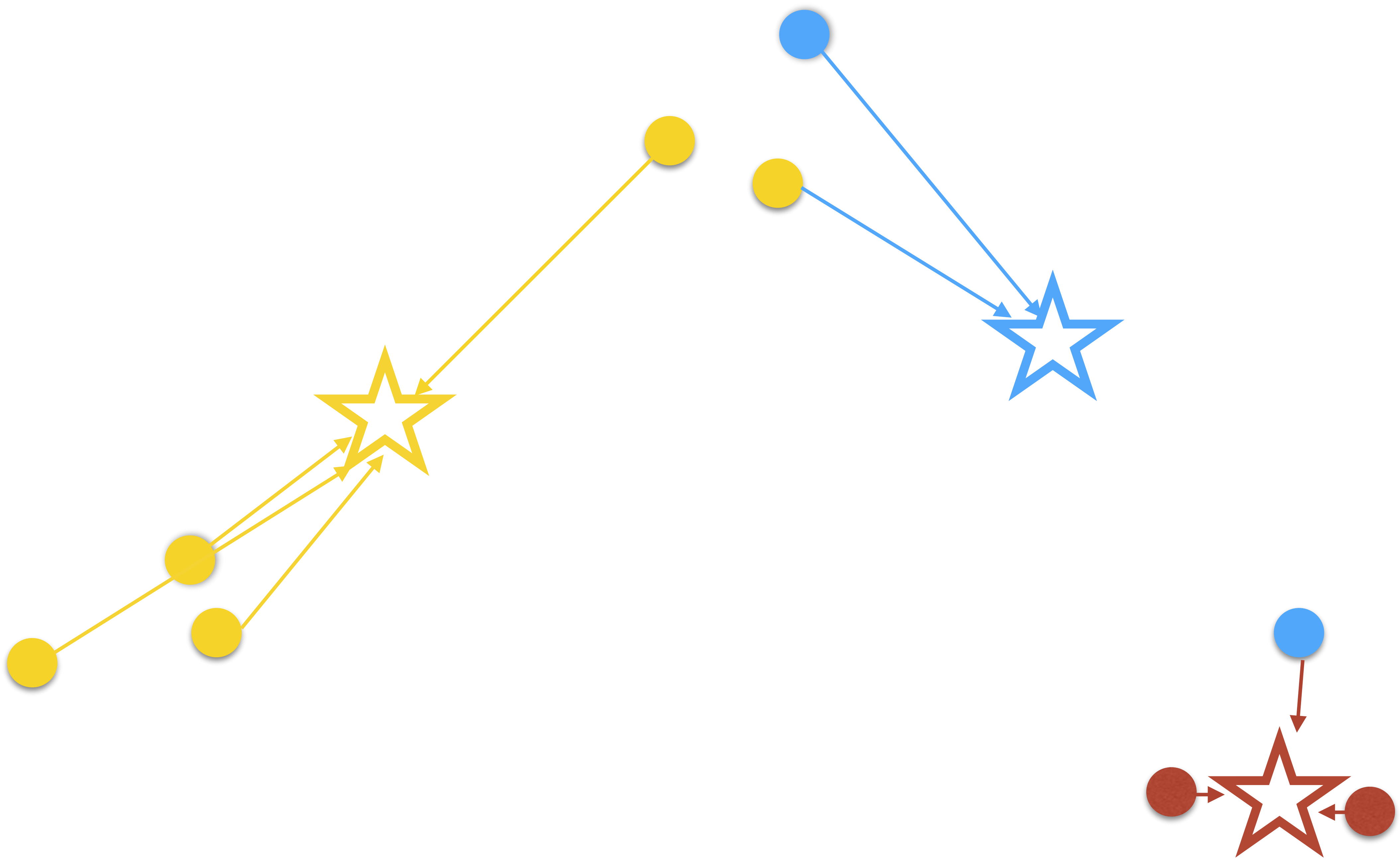
Demo



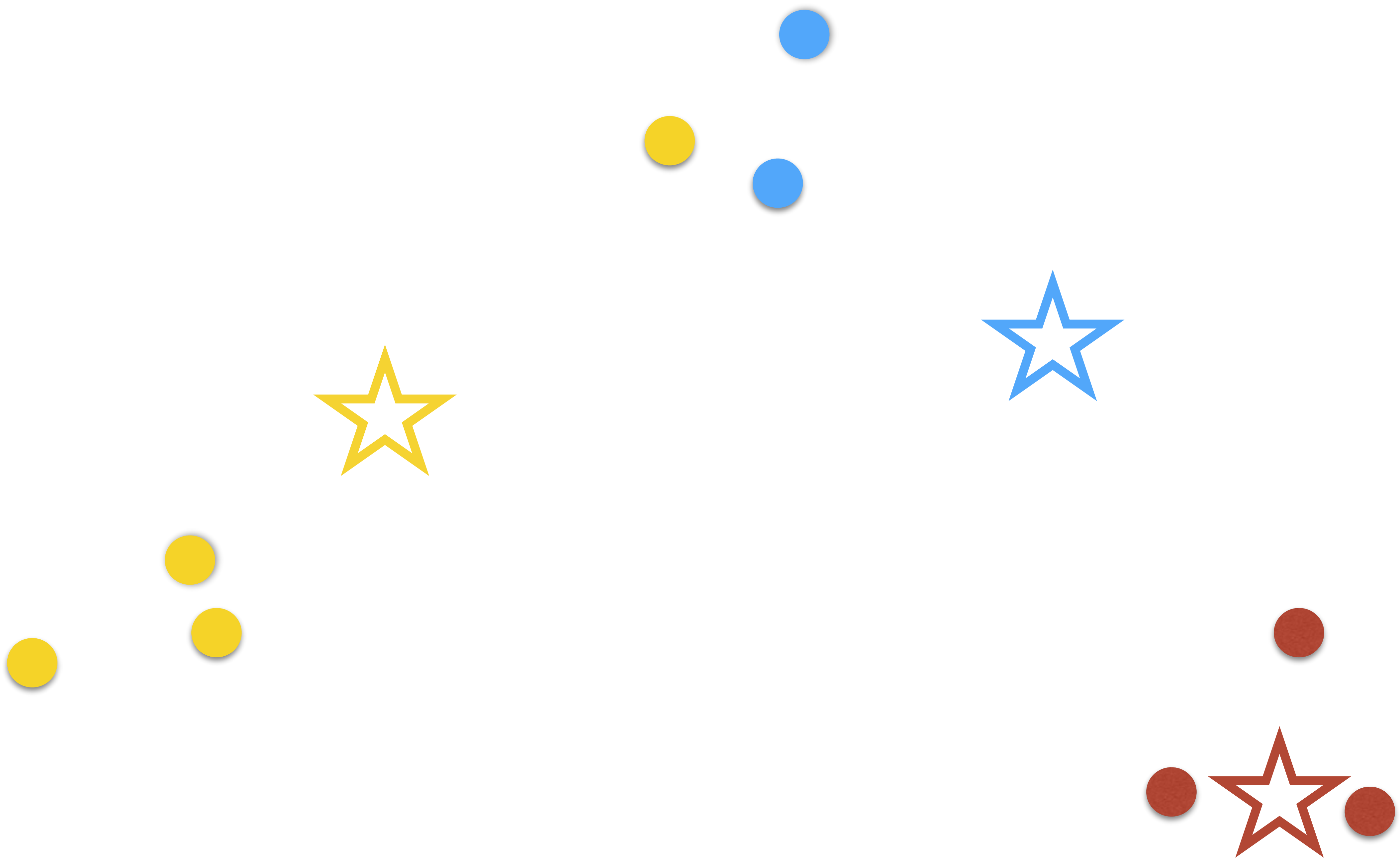
Demo



Demo

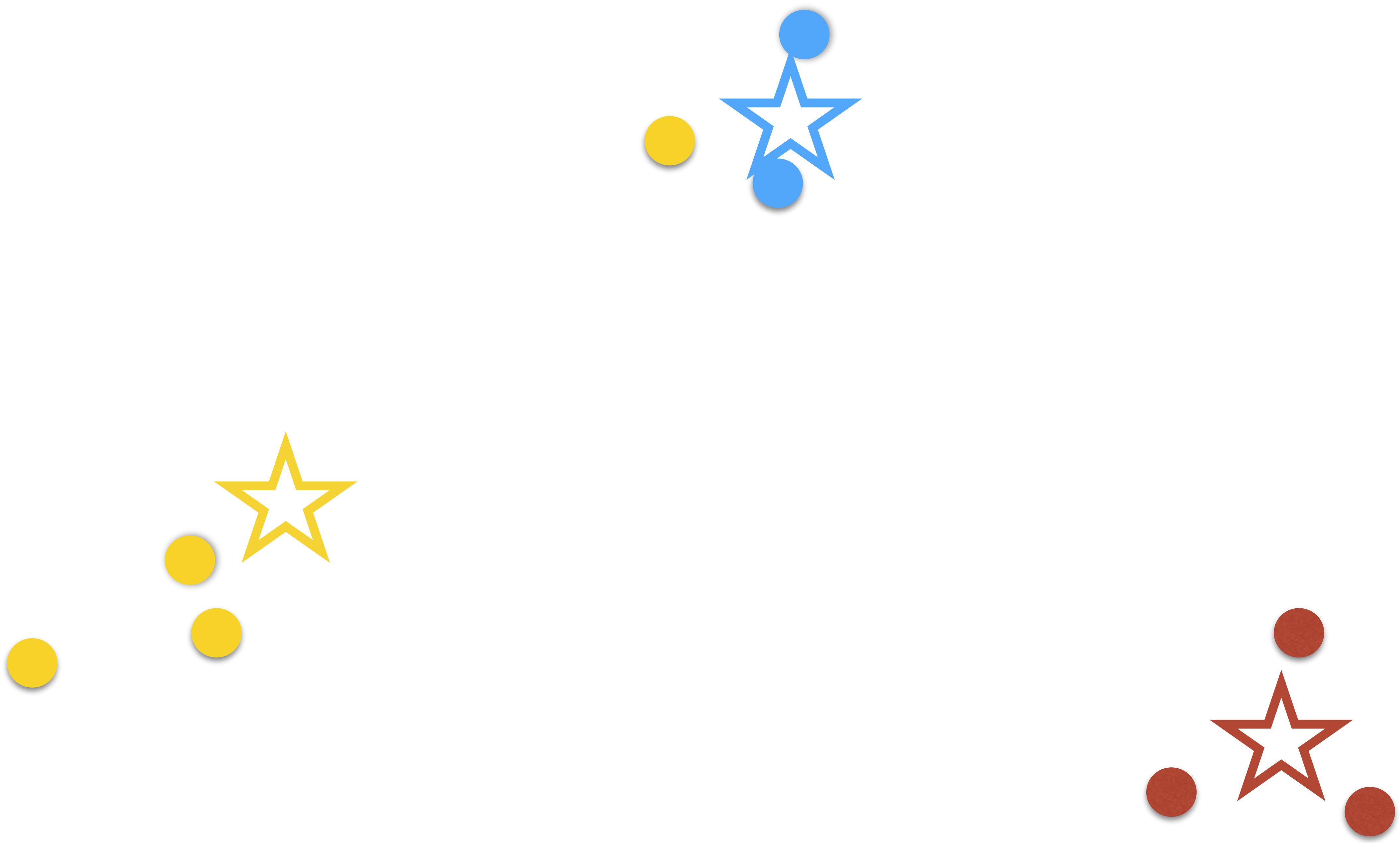


Demo



Demo

Demo



Demo

Demo

Demo

Demo

K-MEANS CLUSTERING

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^1$ randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

$$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\operatorname{argmin}} \|\mathbf{x}_t - \hat{\mathbf{r}}_j^m\|$$

  2. For each $j \in [K]$, set new representative as

$$\hat{\mathbf{r}}_j^{m+1} = \frac{1}{|\hat{C}_j^m|} \sum_{t \in \hat{C}_j^m} \mathbf{x}_t$$

  3. $m \leftarrow m + 1$

K-means Convergence

- K-means algorithm converges to local minima of objective

$$O(c; \mathbf{r}_1, \ldots, \mathbf{r}_K) = \sum_{j=1}^{K} \sum_{c(\mathbf{x}_t)=j} \|\mathbf{x}_t - \mathbf{r}_j\|_2^2$$

- Proof:
  Clustering assignment improves objective:

$$O\left(\hat{c}^{m-1}; \mathbf{r}_1^m, \ldots, \mathbf{r}_K^m\right) \geq O\left(\hat{c}^m; \mathbf{r}_1^m, \ldots, \mathbf{r}_K^m\right)$$

  (By definition of $\hat{c}^m(\mathbf{x}_t)$)
  Computing centroids improves objective:

$$O\left(\hat{c}^m; \mathbf{r}_1^m, \ldots, \mathbf{r}_K^m\right) \geq O\left(\hat{c}^m; \mathbf{r}_1^{m+1}, \ldots, \mathbf{r}_K^{m+1}\right)$$

  (By the fact about centroid)

Lets build an Algorithm

$$M_3 = \min_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \text{dissimilarity}(x_t, x_s)$$

Demo

Demo

$$\text{dissimilarity}(C_i, C_j) = \min_{t \in C_i, s \in C_j} \text{dissimilarity}(\mathrm{x}_t, \mathrm{x}_s)$$

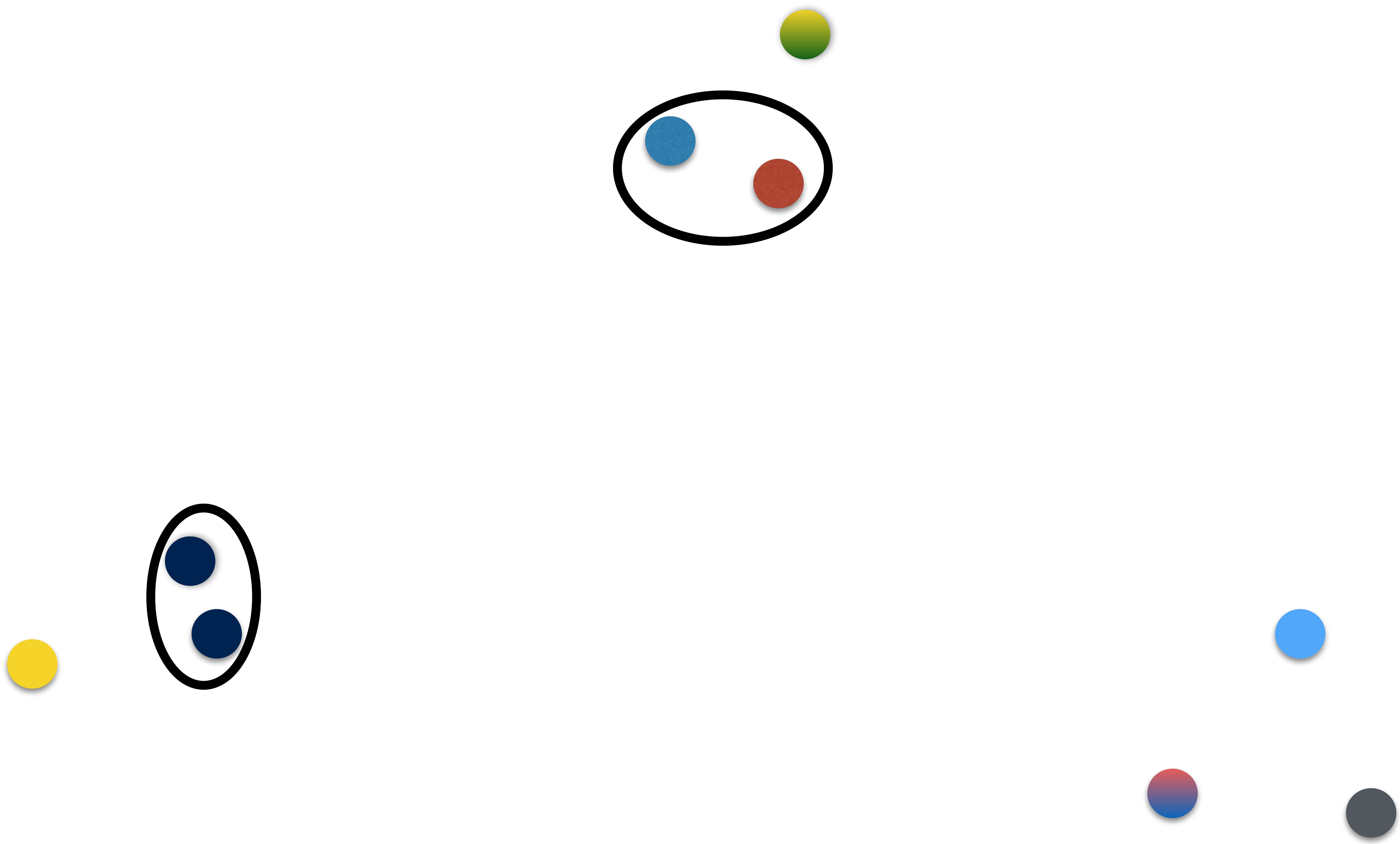
Demo

$$\text{dissimilarity}(C_i, C_j) = \min_{t \in C_i, s \in C_j} \text{dissimilarity}(\mathrm{x}_t, \mathrm{x}_s)$$

Demo

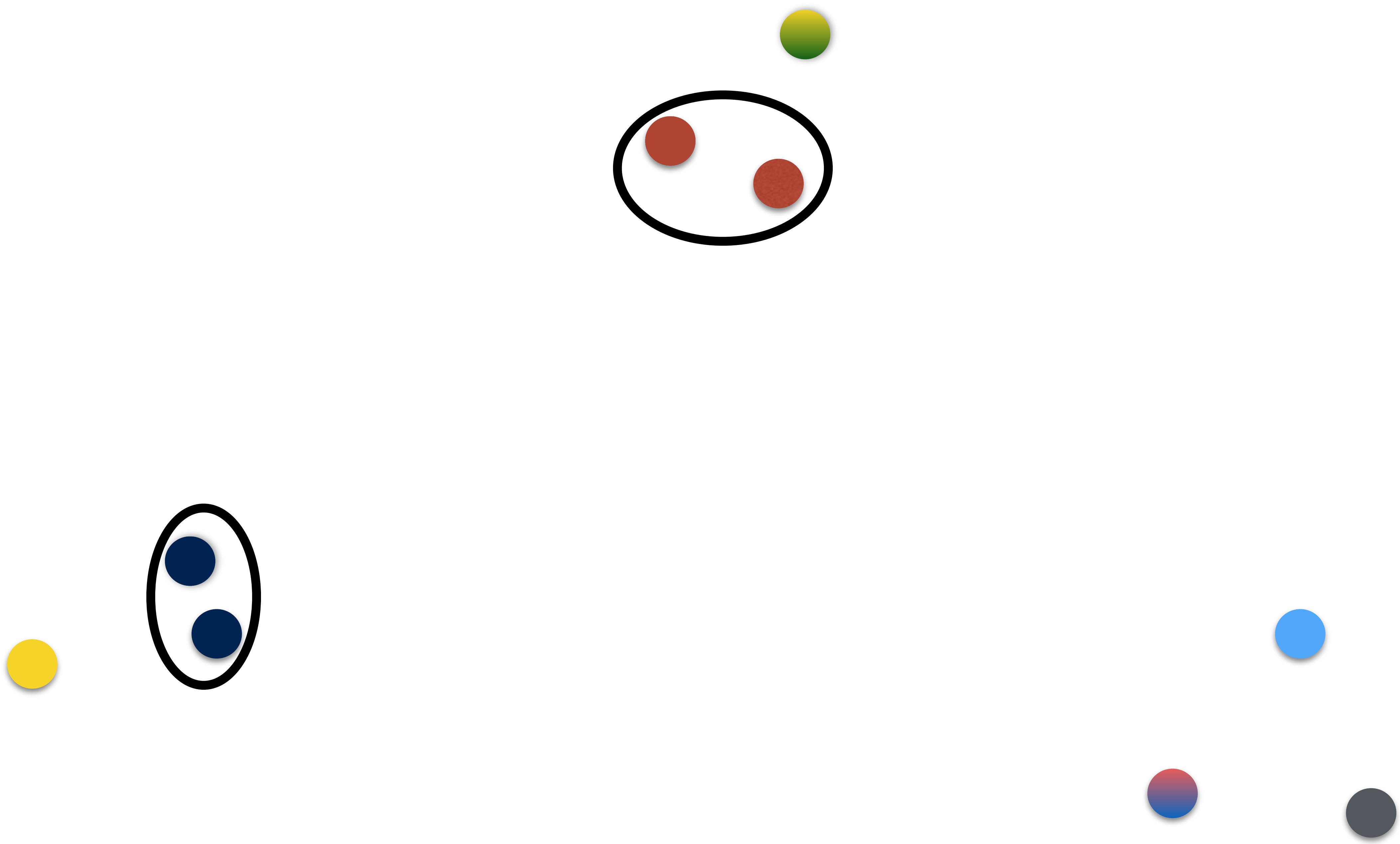
$$\text{dissimilarity}(C_i, C_j) = \min_{t \in C_i, s \in C_j} \text{dissimilarity}(\mathrm{x}_t, \mathrm{x}_s)$$

Demo



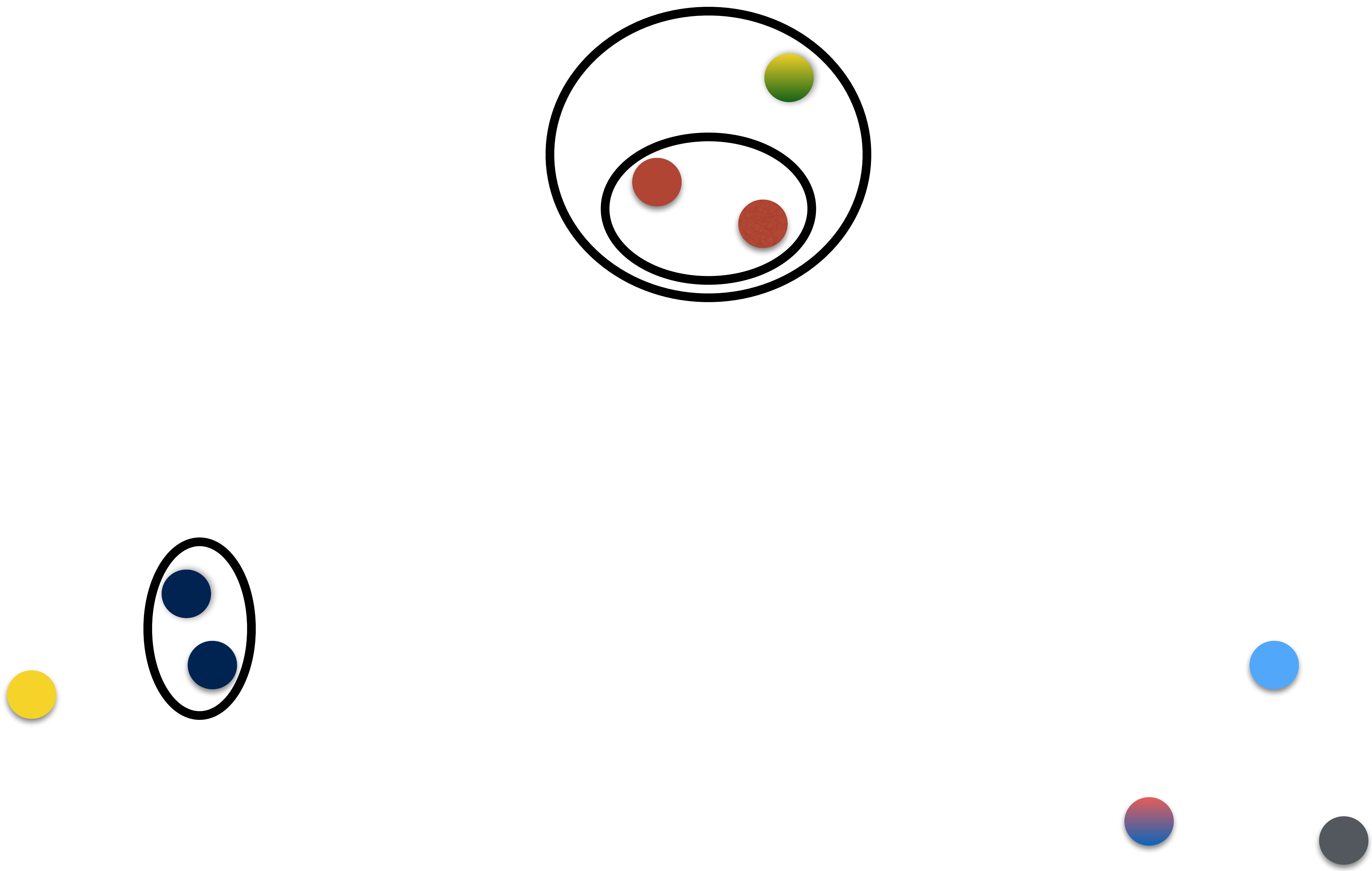
$$\text{dissimilarity}(C_i, C_j) = \min_{t \in C_i, s \in C_j} \text{dissimilarity}(\mathrm{x}_t, \mathrm{x}_s)$$

Demo



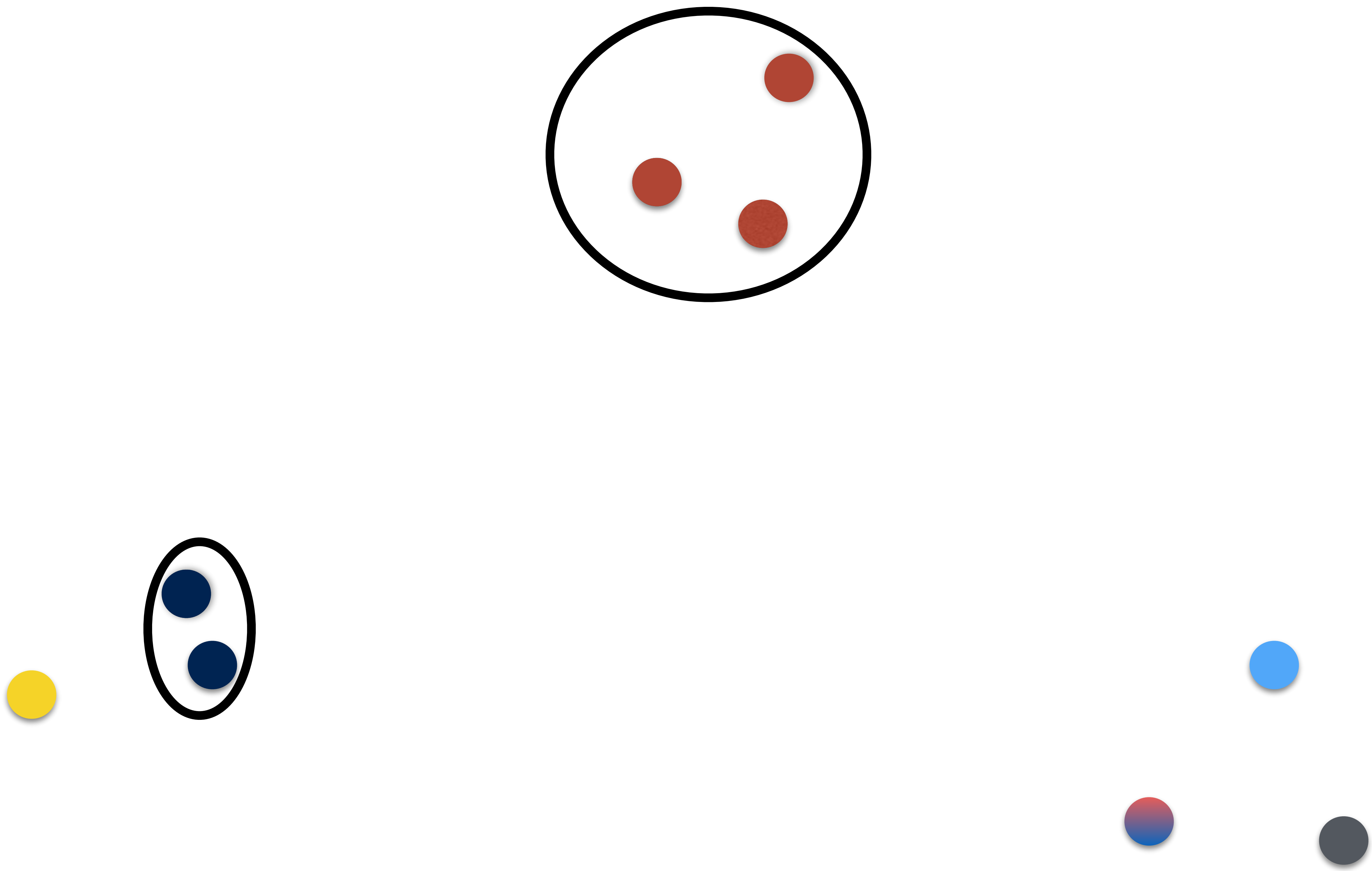
$$\text{dissimilarity}(C_i, C_j) = \min_{t \in C_i, s \in C_j} \text{dissimilarity}(\mathrm{x}_t, \mathrm{x}_s)$$

Demo



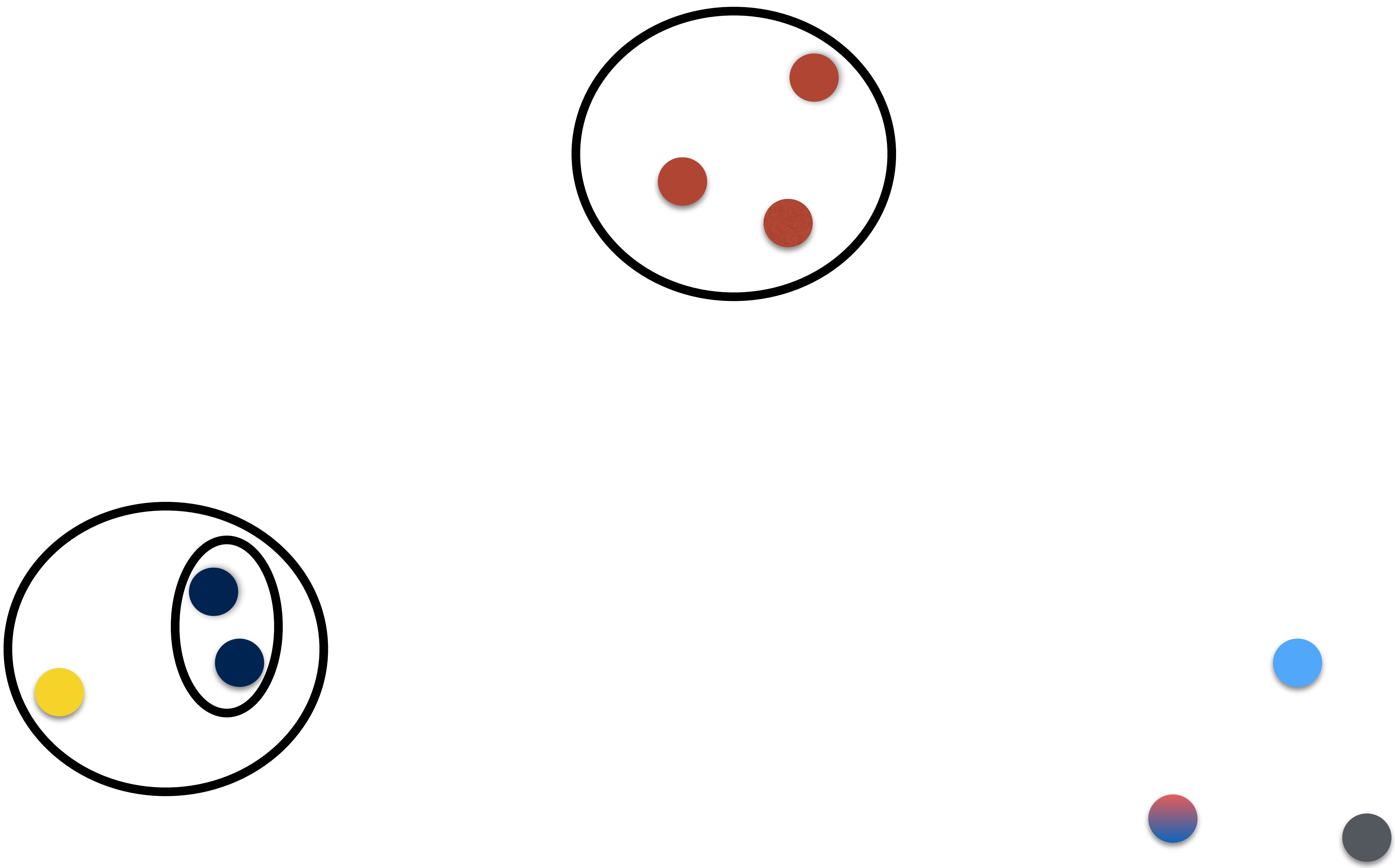
$$\text{dissimilarity}(C_i, C_j) = \min_{t \in C_i, s \in C_j} \text{dissimilarity}(\mathrm{x}_t, \mathrm{x}_s)$$

Demo



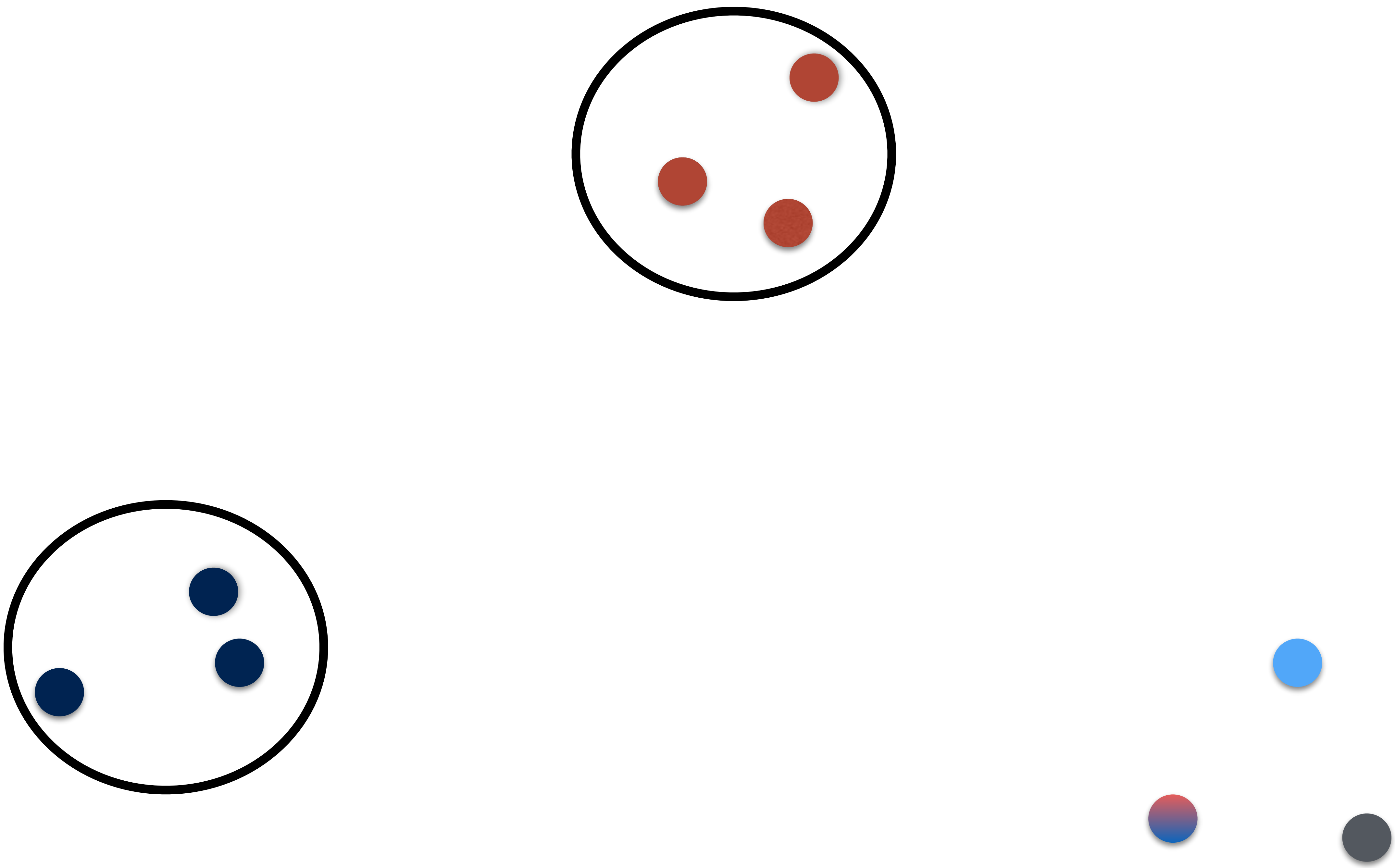
$$\text{dissimilarity}(C_i, C_j) = \min_{t \in C_i, s \in C_j} \text{dissimilarity}(\mathrm{x}_t, \mathrm{x}_s)$$

Demo



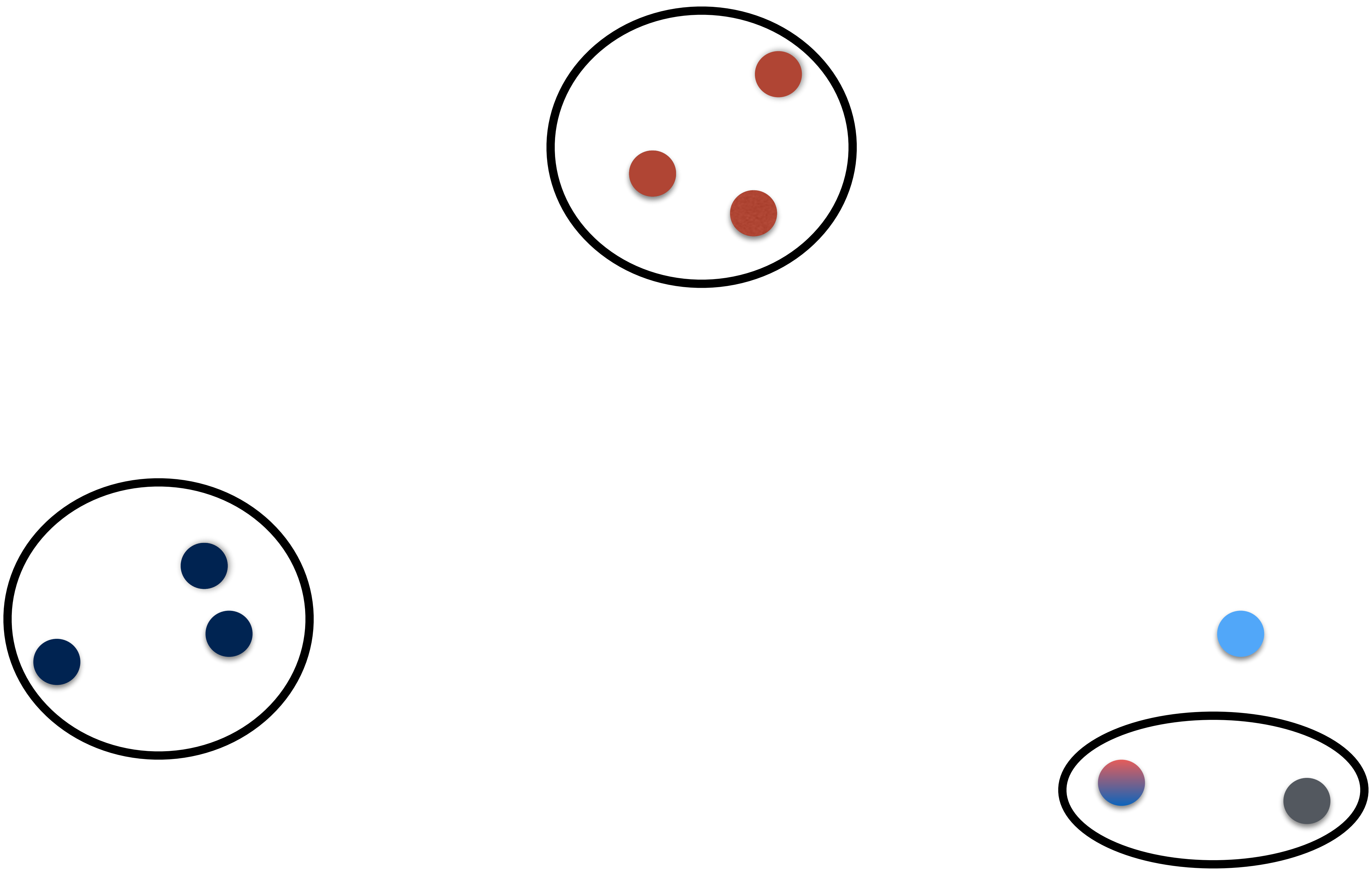
$$\text{dissimilarity}(C_i, C_j) = \min_{t \in C_i, s \in C_j} \text{dissimilarity}(\mathrm{x}_t, \mathrm{x}_s)$$

Demo



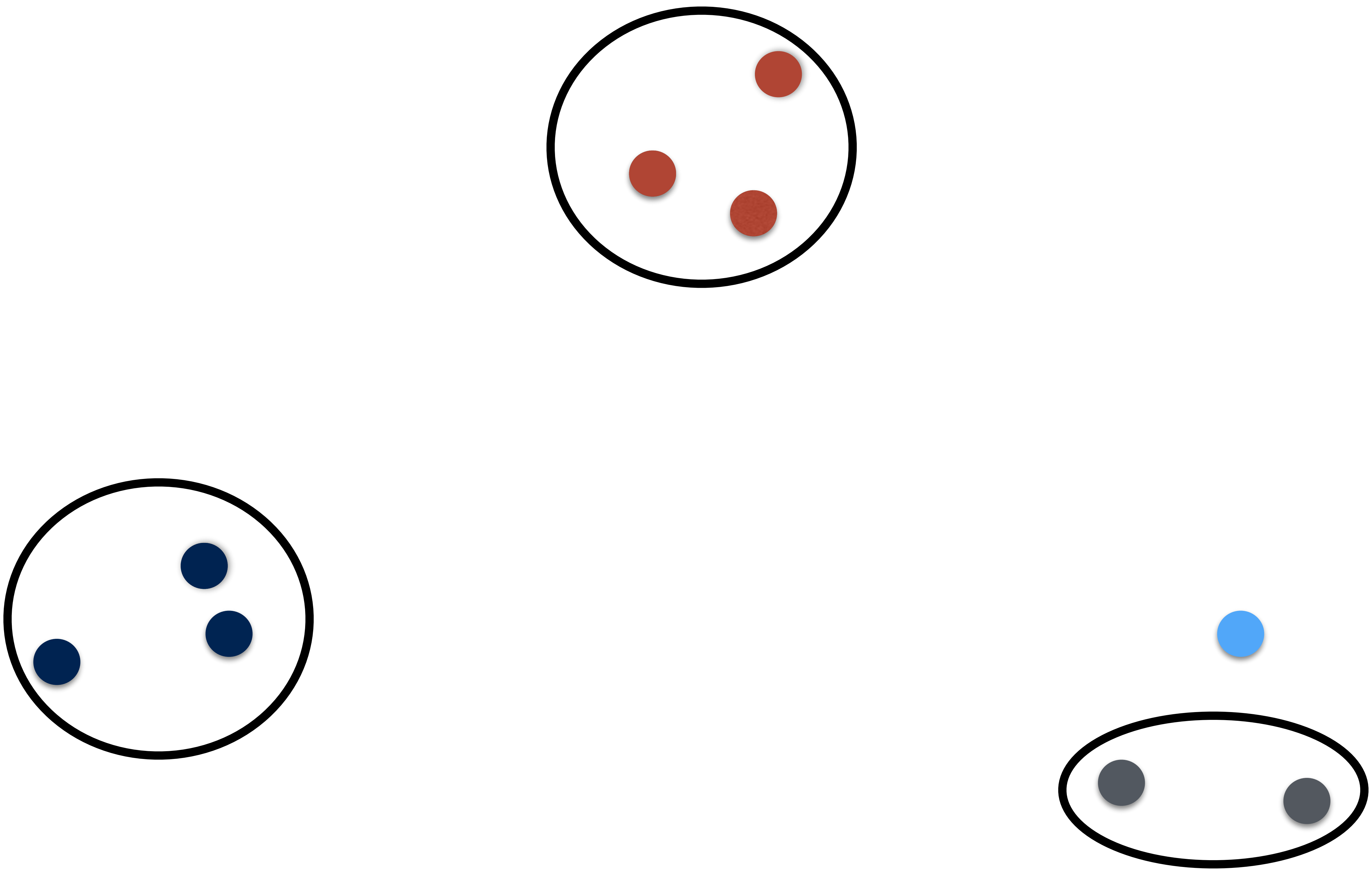
$$\text{dissimilarity}(C_i, C_j) = \min_{t \in C_i, s \in C_j} \text{dissimilarity}(\mathrm{x}_t, \mathrm{x}_s)$$

Demo

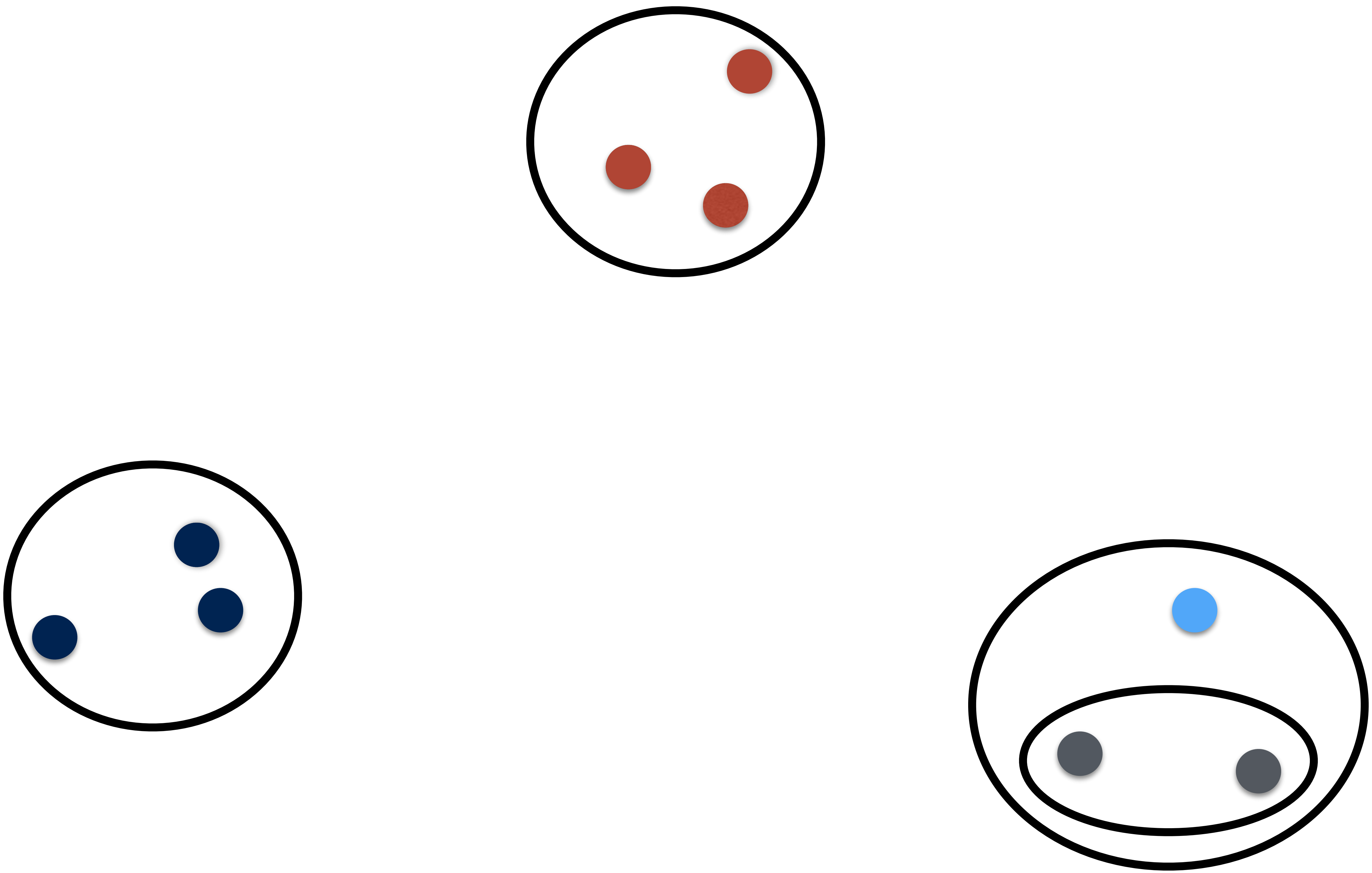


$$\text{dissimilarity}(C_i, C_j) = \min_{t \in C_i, s \in C_j} \text{dissimilarity}(\mathrm{x}_t, \mathrm{x}_s)$$

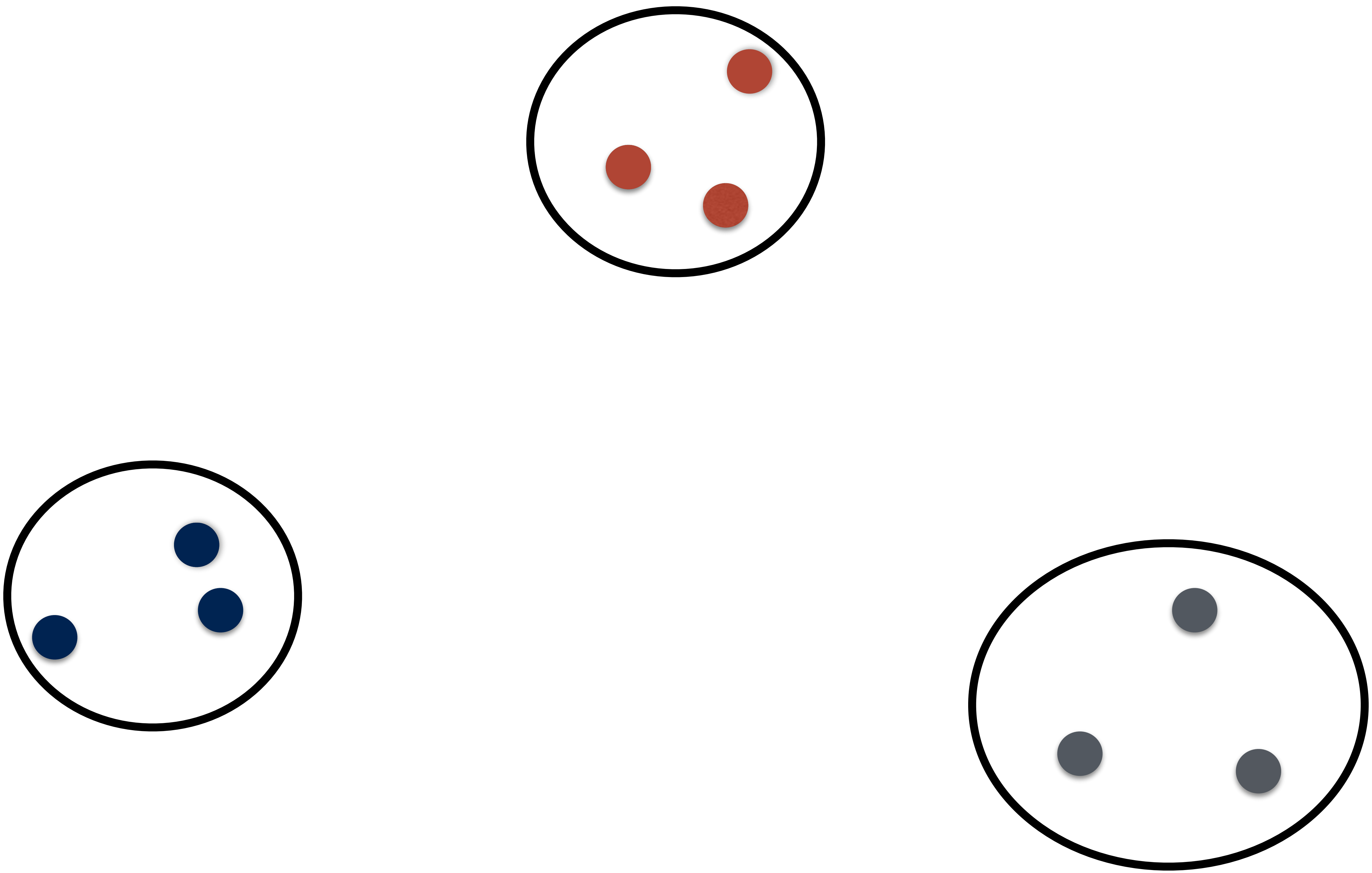
Demo



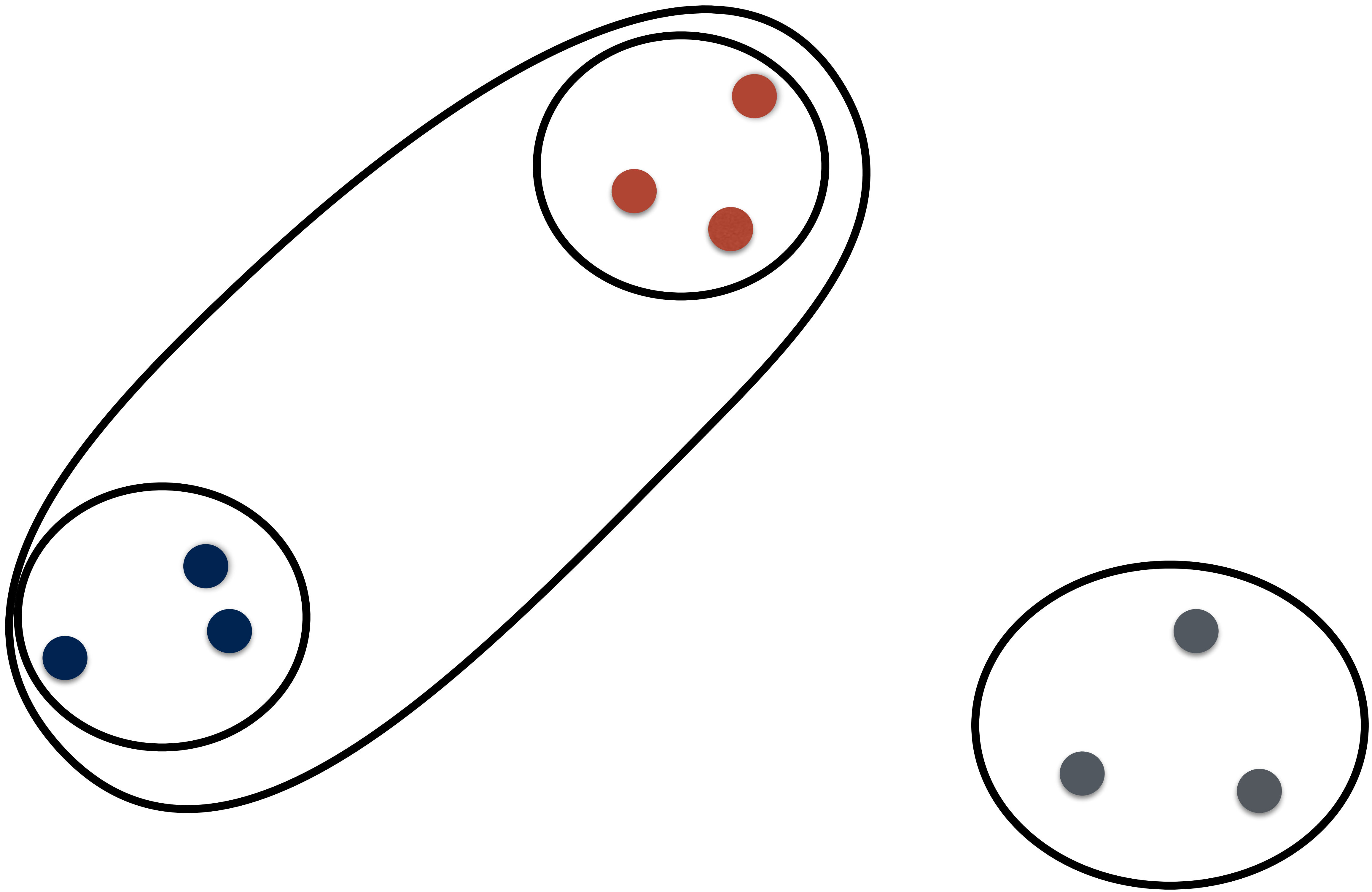
Demo



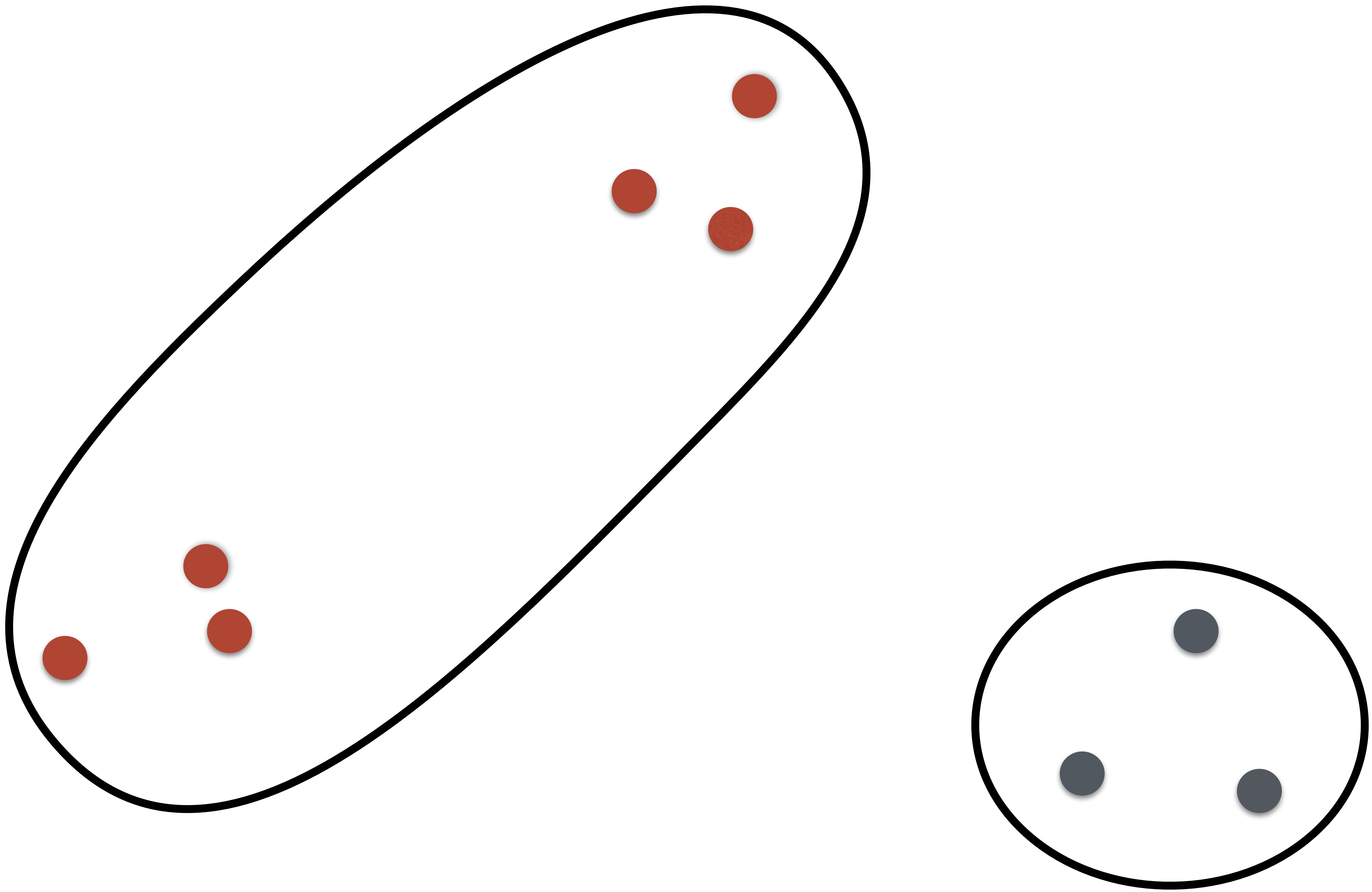
Demo



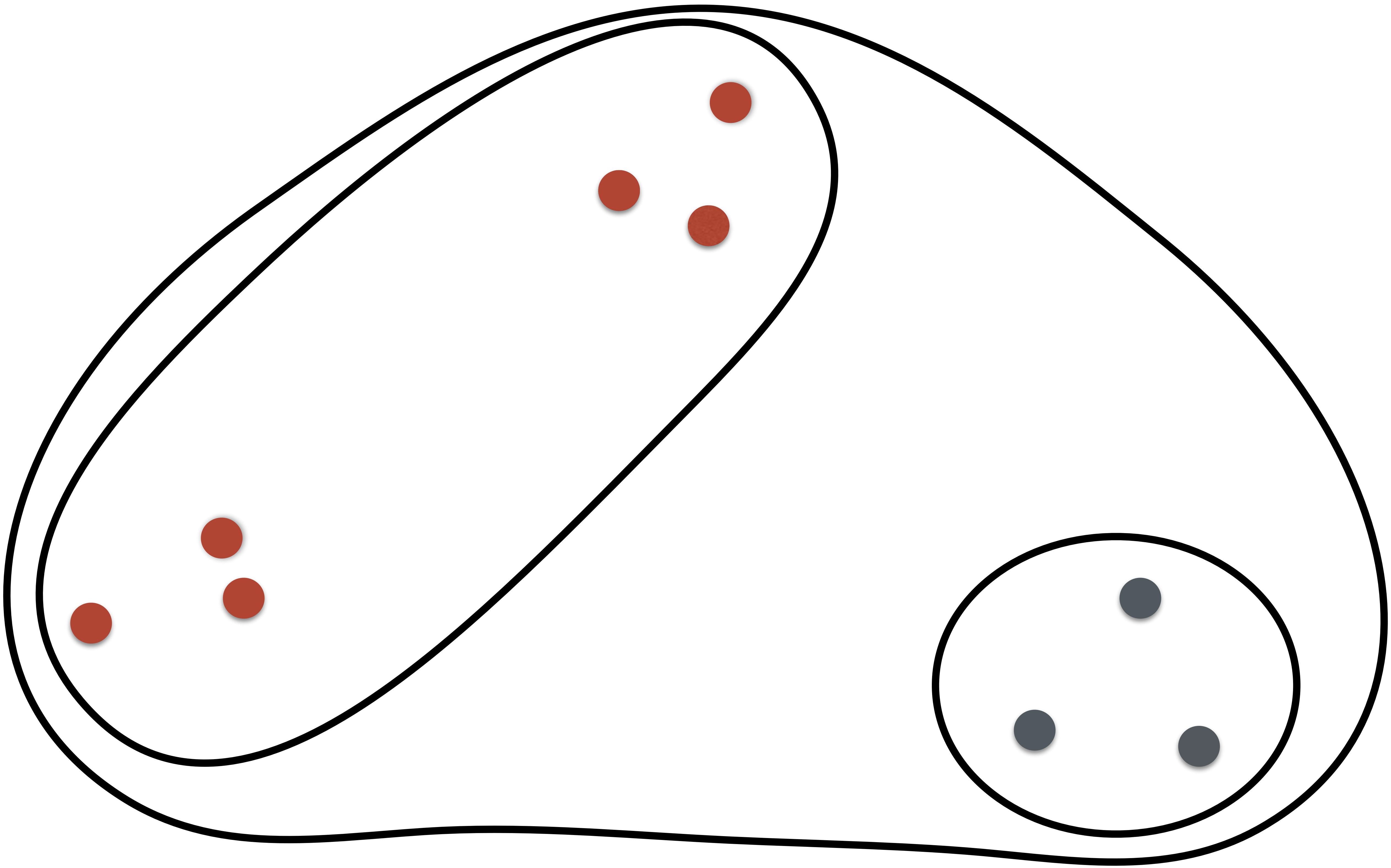
Demo



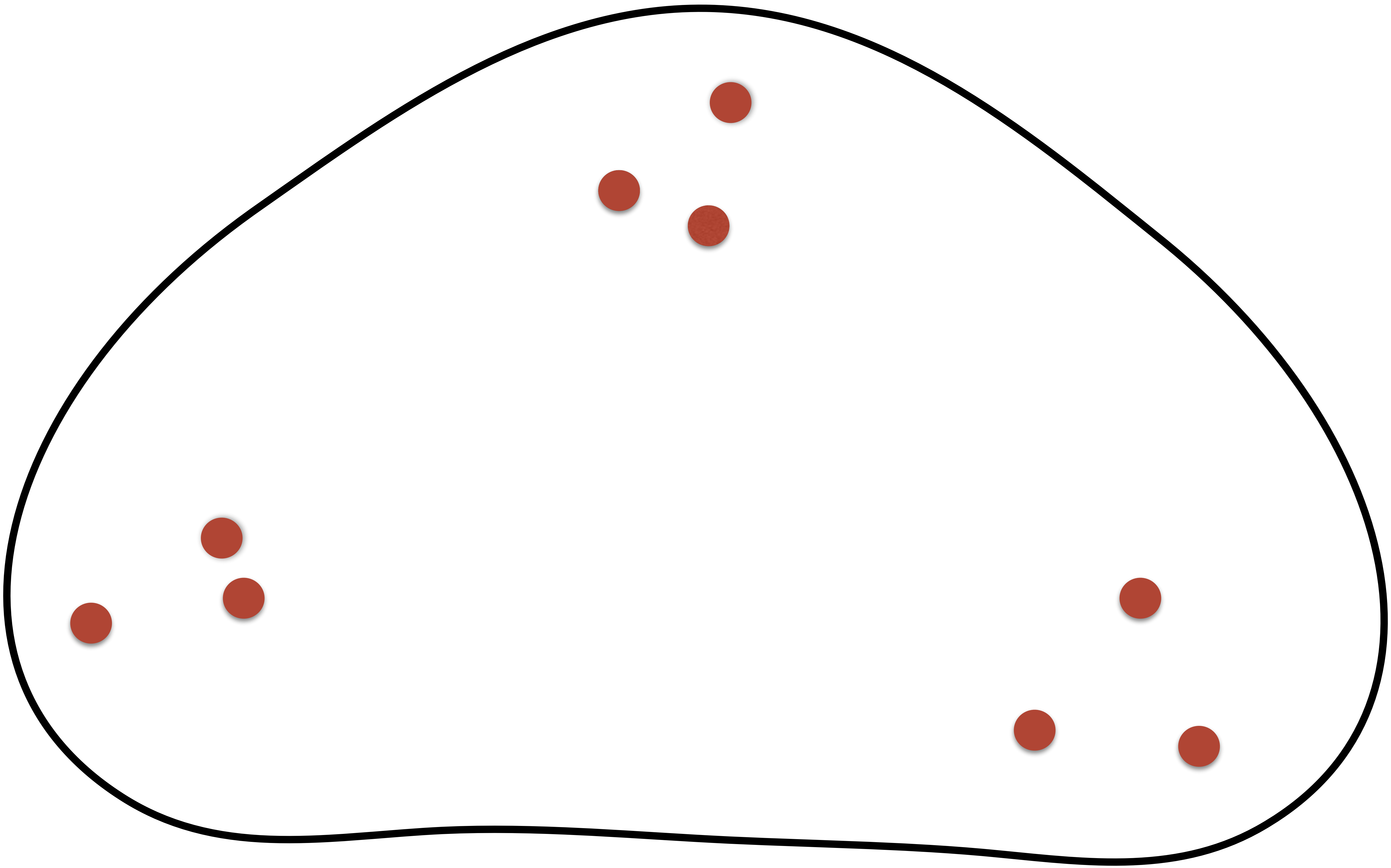
Demo



Demo



Demo



SINGLE LINK CLUSTERING

- Initialize n clusters with each point $\mathbf{x}_t$ to its own cluster
- Until there are only K clusters, do
  1. Find closest two clusters and merge them into one cluster
  2. Update between cluster distances (called proximity matrix)

Single Link Clustering

- Initialize n clusters with each point $\mathbf{x}_t$ to its own cluster
- Until there are only K clusters, do
  1. Find closest two clusters and merge them into one cluster
  2. Update between cluster distances (called proximity matrix)

$$\text{dissimilarity}(C_i, C_j) = \min_{t \in C_i, s \in C_j} \text{dissimilarity}(\mathrm{x}_t, \mathrm{x}_s)$$

Single Link Objective

Objective for single-link:

$$M_3 = \min_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \text{dissimilarity}(x_t, x_s)$$

Single link clustering is optimal for above objective!