



Supervised Hierarchical Clustering with Exponential Linkage

Nishant Yadav



Ari Kobren



Nicholas Monath



Andrew McCallum



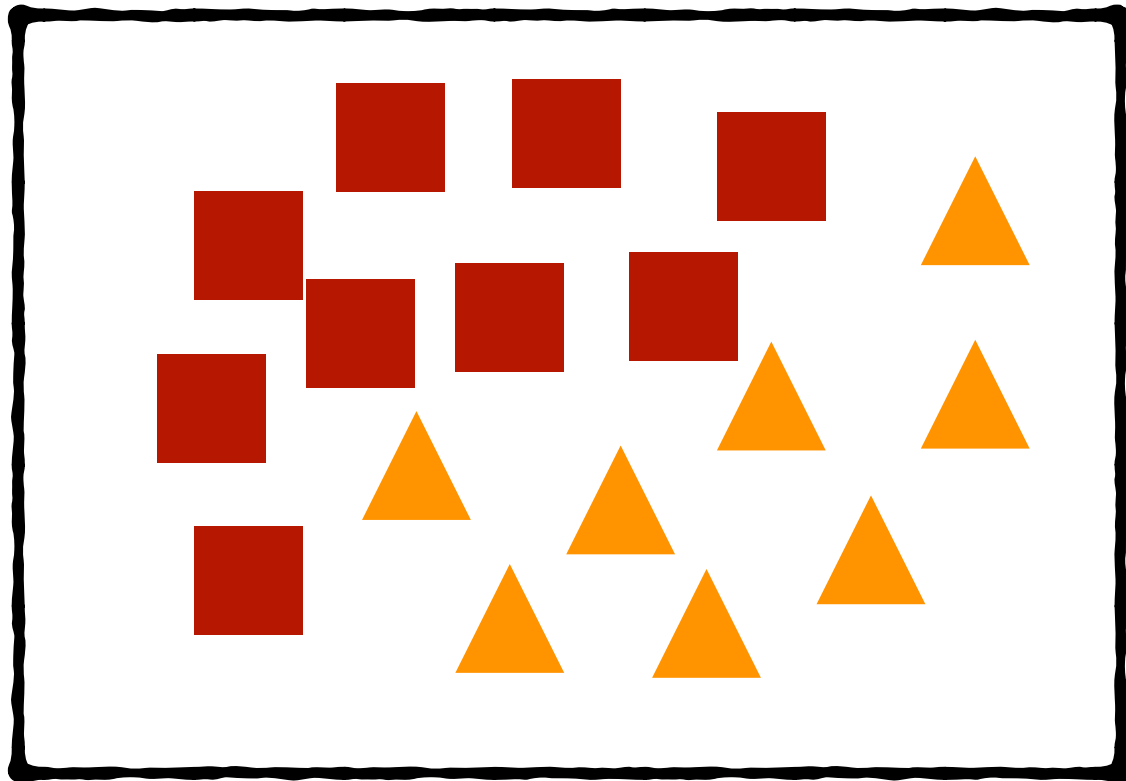
UMassAmherst



College of Information
and Computer Sciences

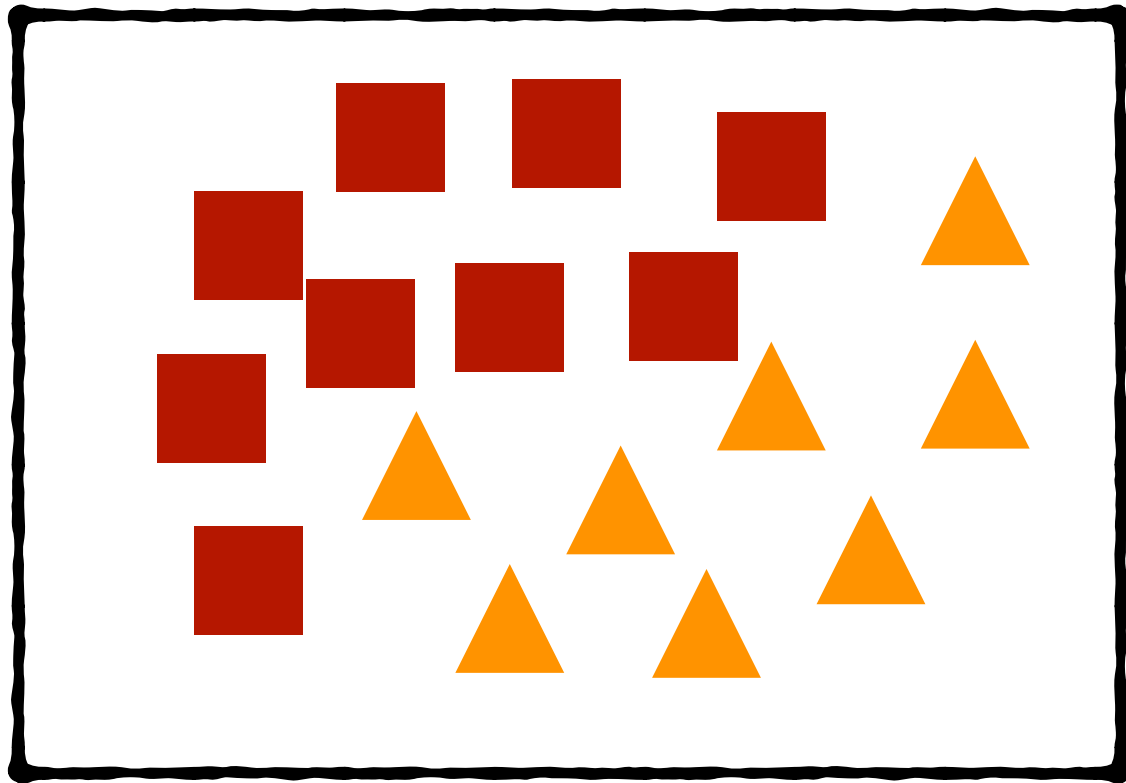
Supervised Clustering

At train time, learn $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{Y}$

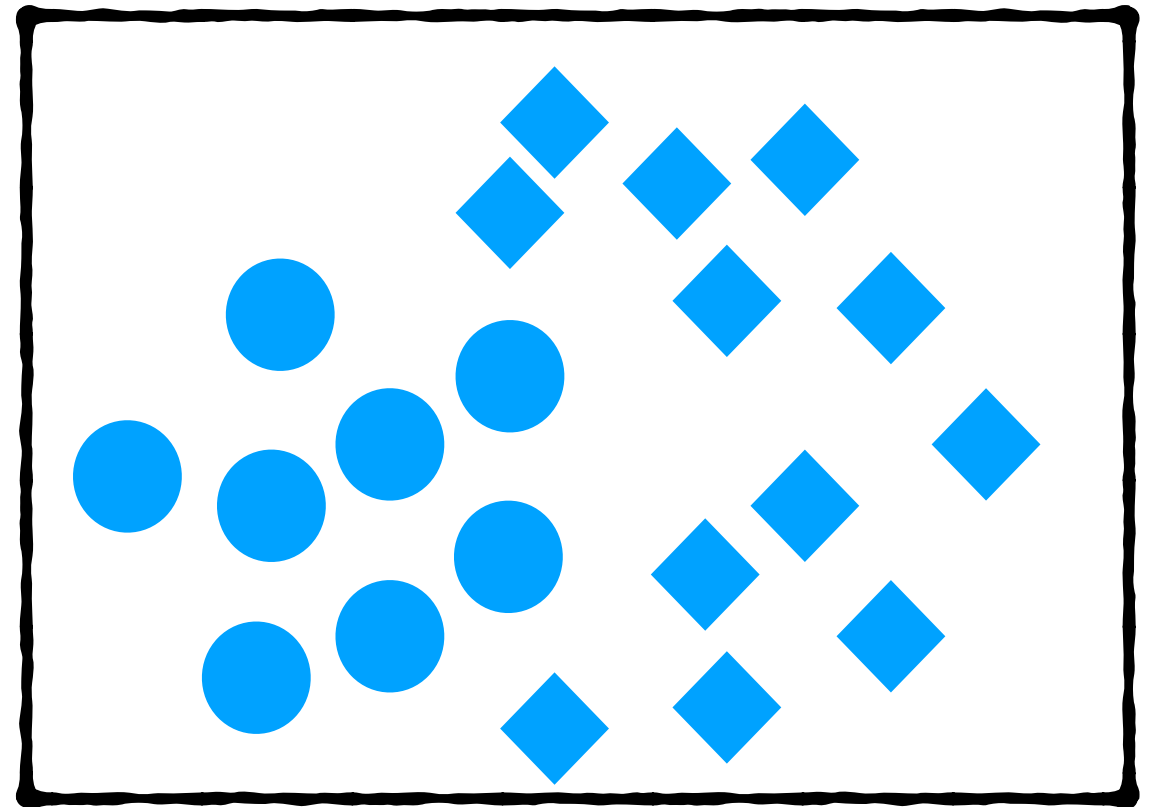


Supervised Clustering

At train time, learn $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{Y}$

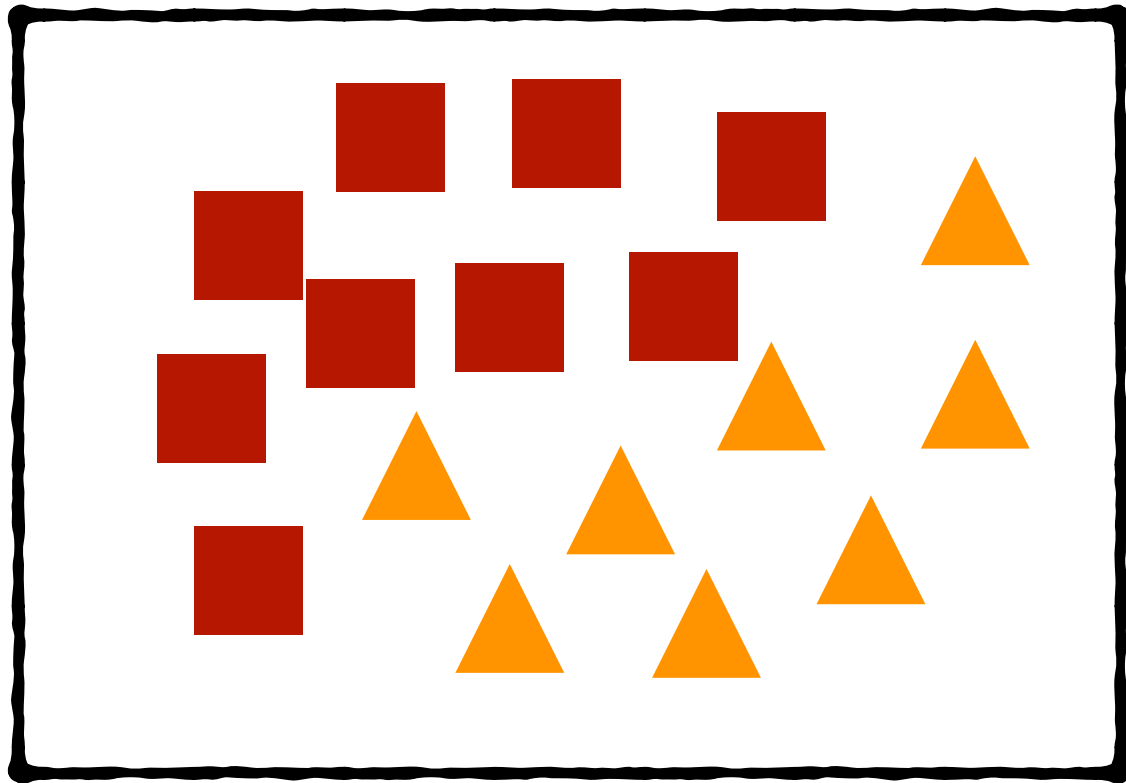


At test time, use \mathcal{A} on new set of points

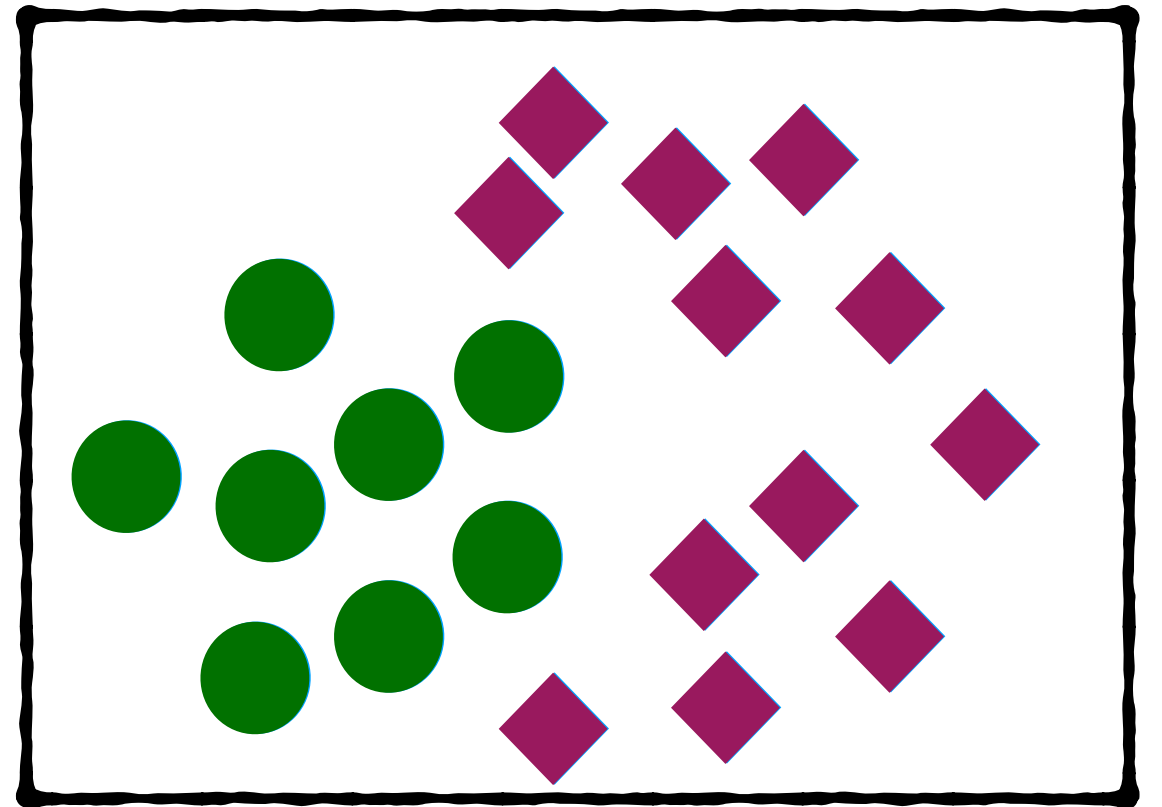


Supervised Clustering

At train time, learn $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{Y}$



At test time, use \mathcal{A} on new set of points



Supervised Clustering

At train time, learn $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{Y}$

At test time, use \mathcal{A} on new set of points



The diagram illustrates the supervised clustering process. It is divided into two main sections by a vertical line. The left section, representing training time, contains a collection of data points: red squares, orange triangles, and a single orange diamond. A green box with a dark green border is overlaid on this section, containing the text 'Select a clustering algorithm'. The right section, representing test time, contains a new set of data points: green circles and purple diamonds. A second green box with a dark green border is overlaid on this section, containing the text 'Learn a dissimilarity function'. The two boxes are connected by a horizontal line, suggesting a shared process or function.

Select a clustering algorithm

Learn a dissimilarity function

Supervised Clustering

At train time, learn $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{Y}$

At test time, use \mathcal{A} on new set of points

Select a clustering algorithm

Learn a dissimilarity function

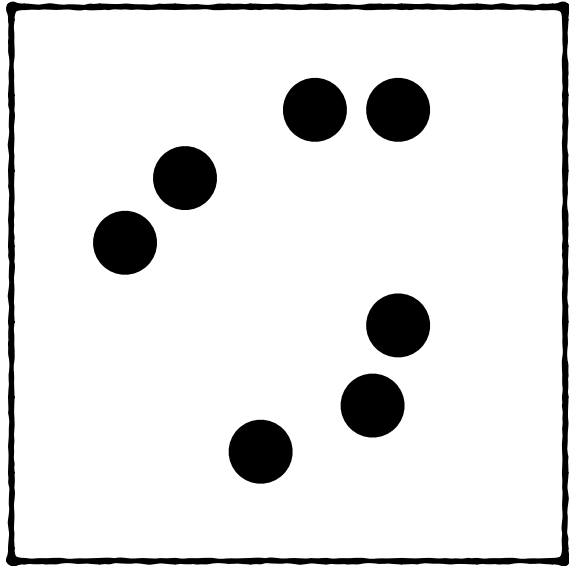
Clustering
Algorithm

Training
Procedure

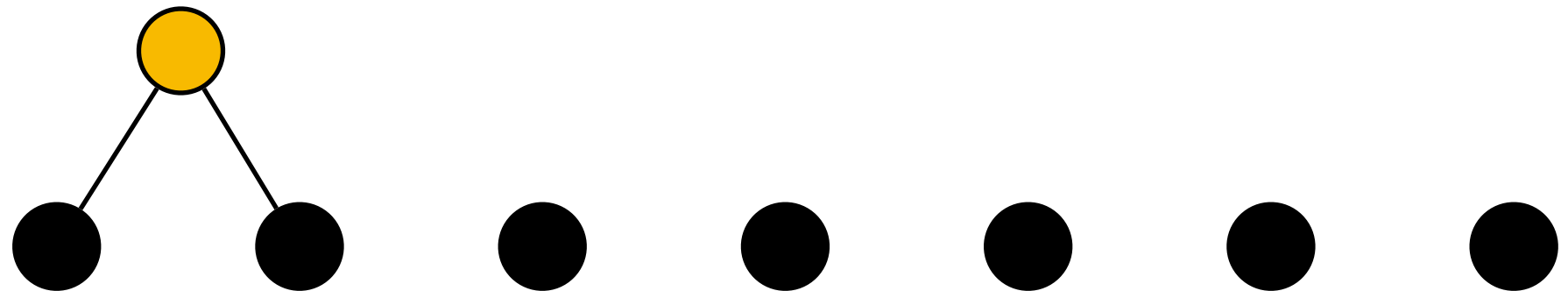
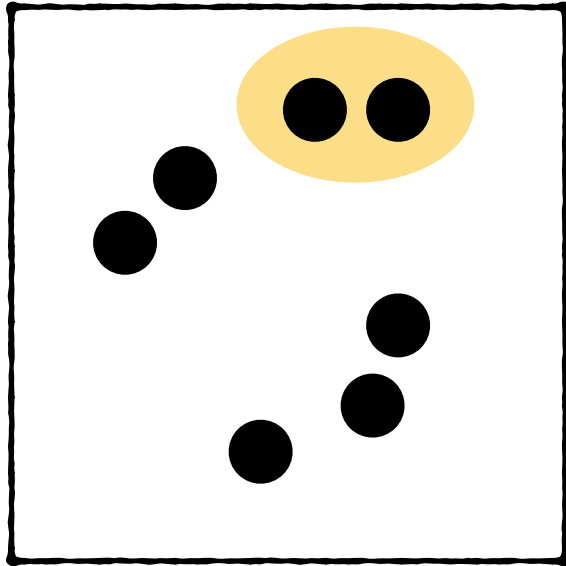
Mismatch leads to poor
generalization

Hierarchical Agglomerative Clustering

Hierarchical Agglomerative Clustering

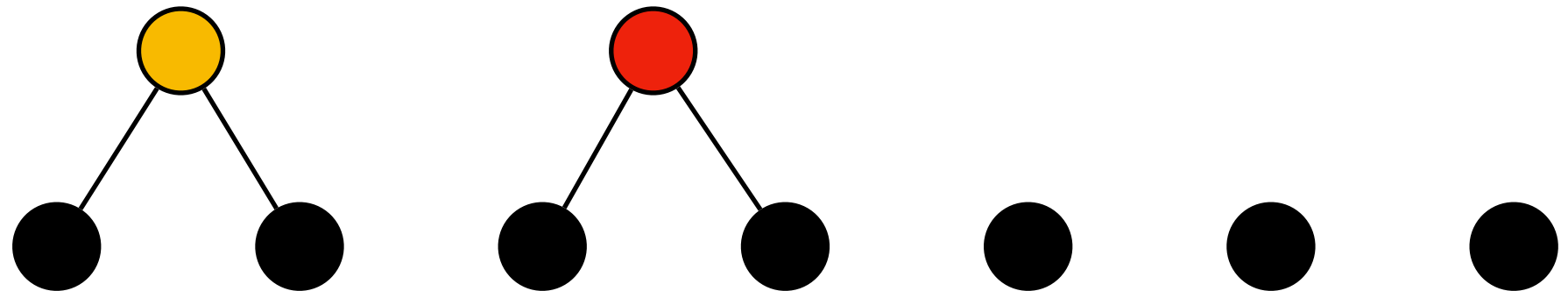
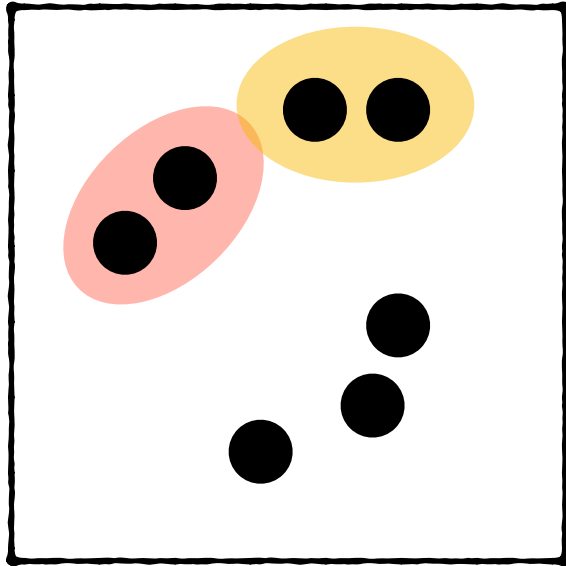


Hierarchical Agglomerative Clustering



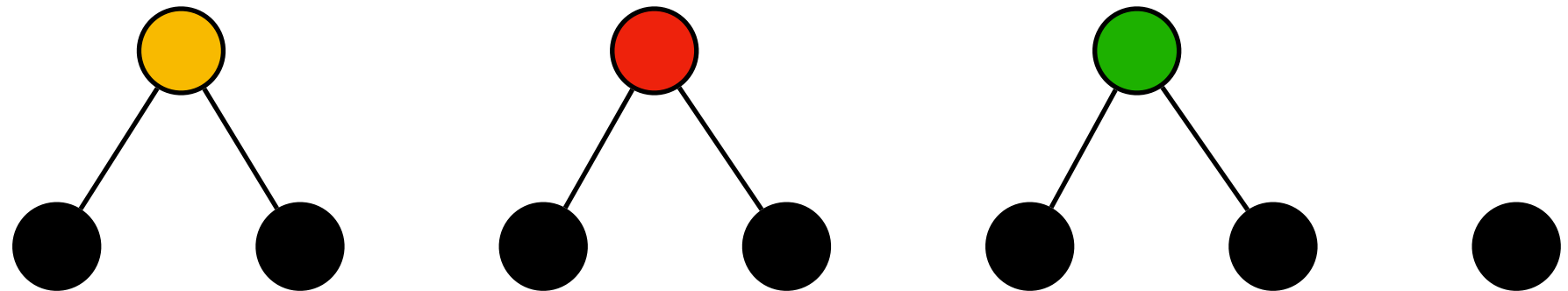
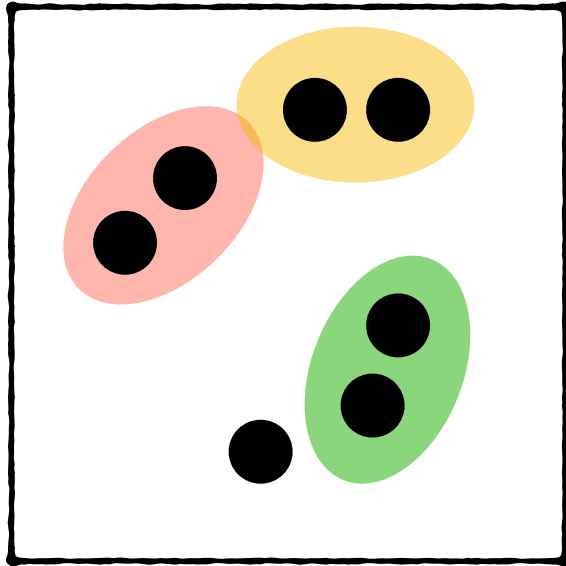
Iteratively merge two *“closest”* clusters

Hierarchical Agglomerative Clustering



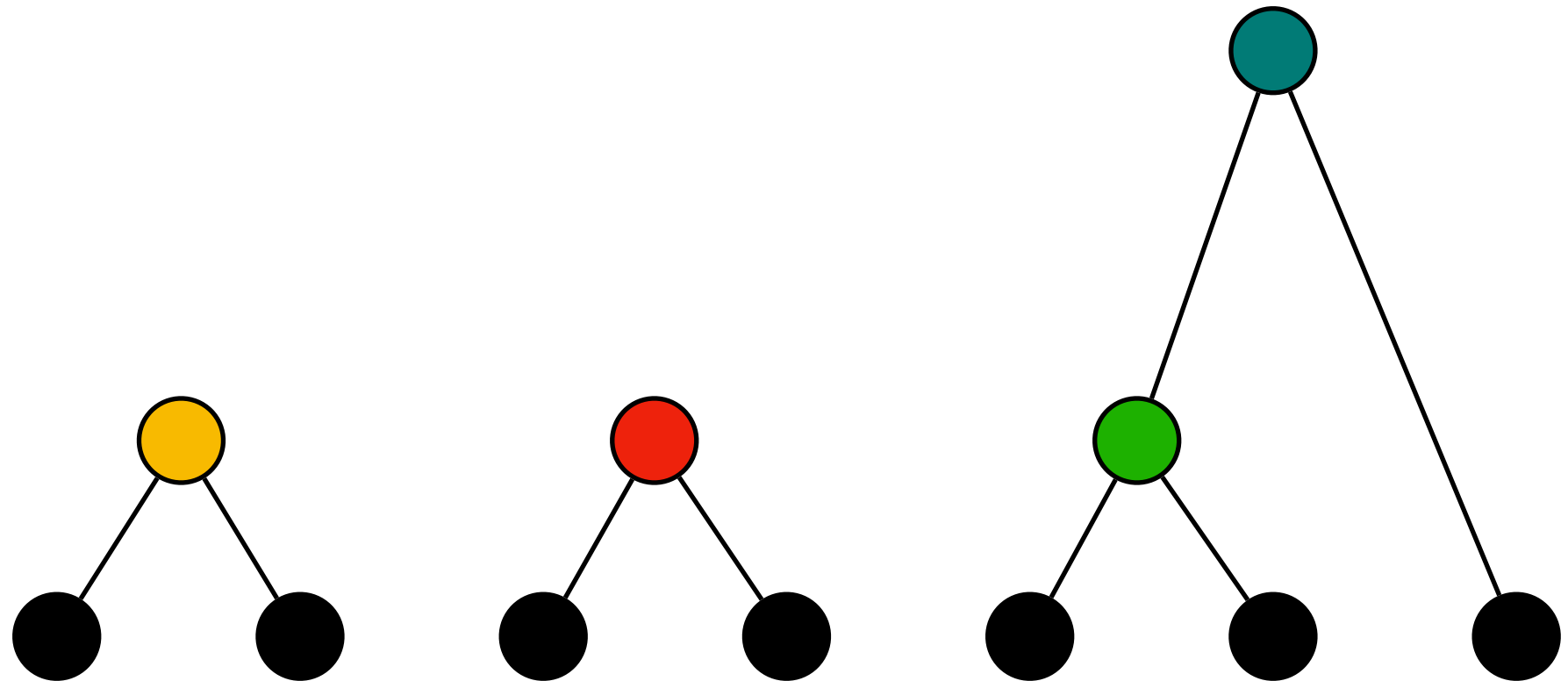
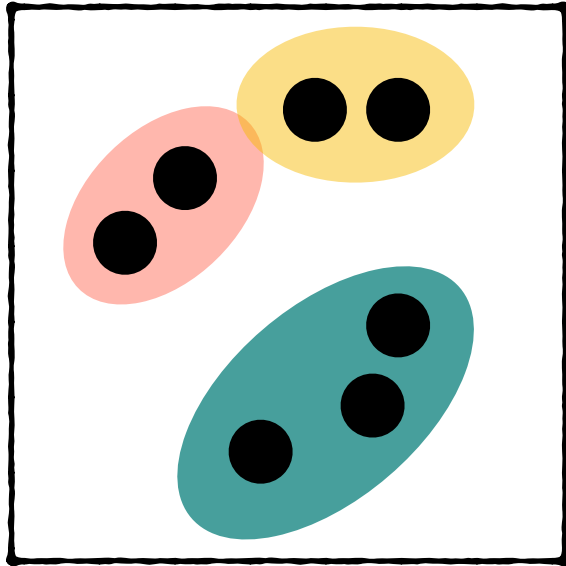
Iteratively merge two *“closest”* clusters

Hierarchical Agglomerative Clustering



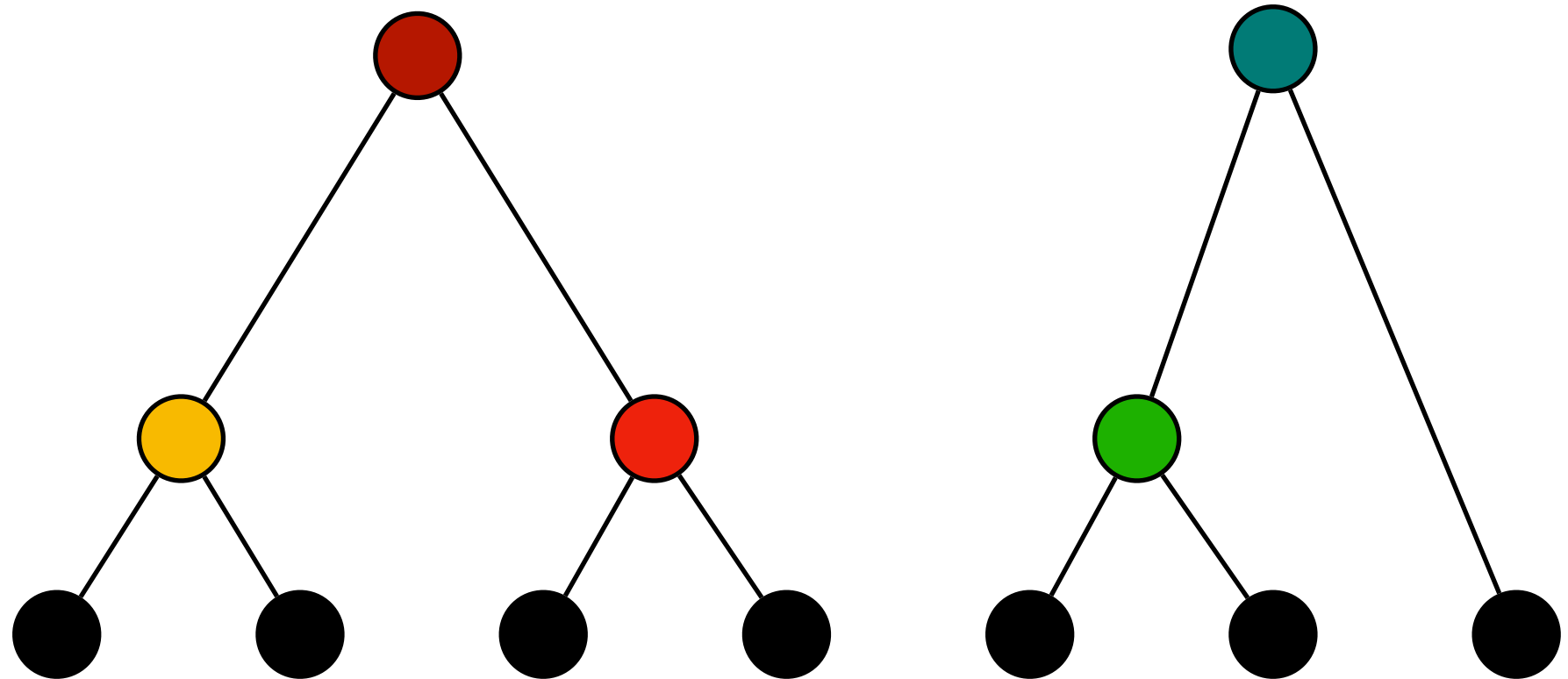
Iteratively merge two *“closest”* clusters

Hierarchical Agglomerative Clustering



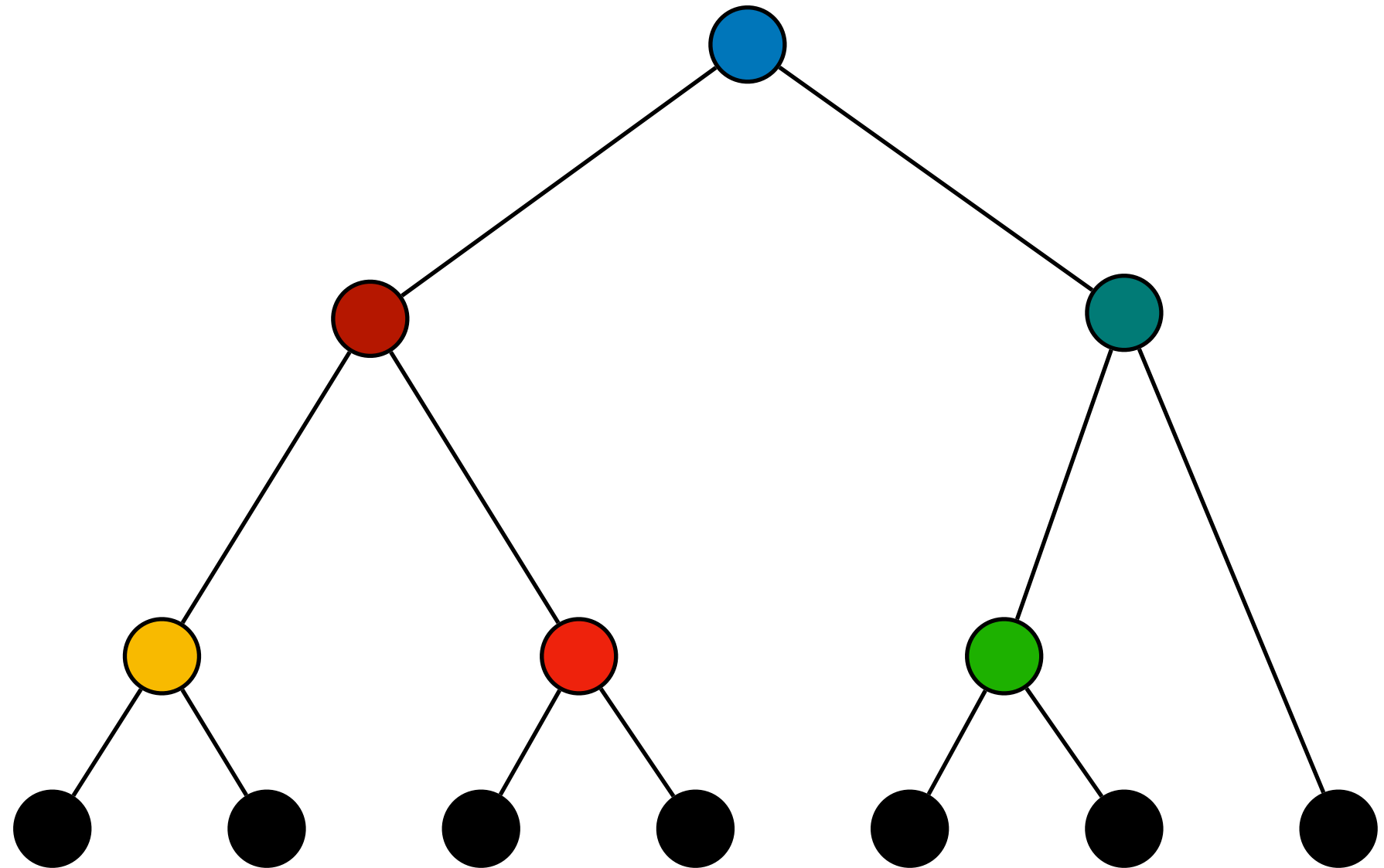
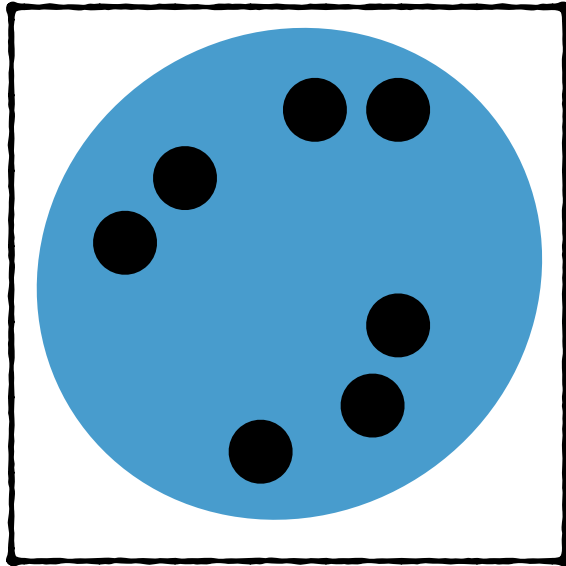
Iteratively merge two *“closest”* clusters

Hierarchical Agglomerative Clustering



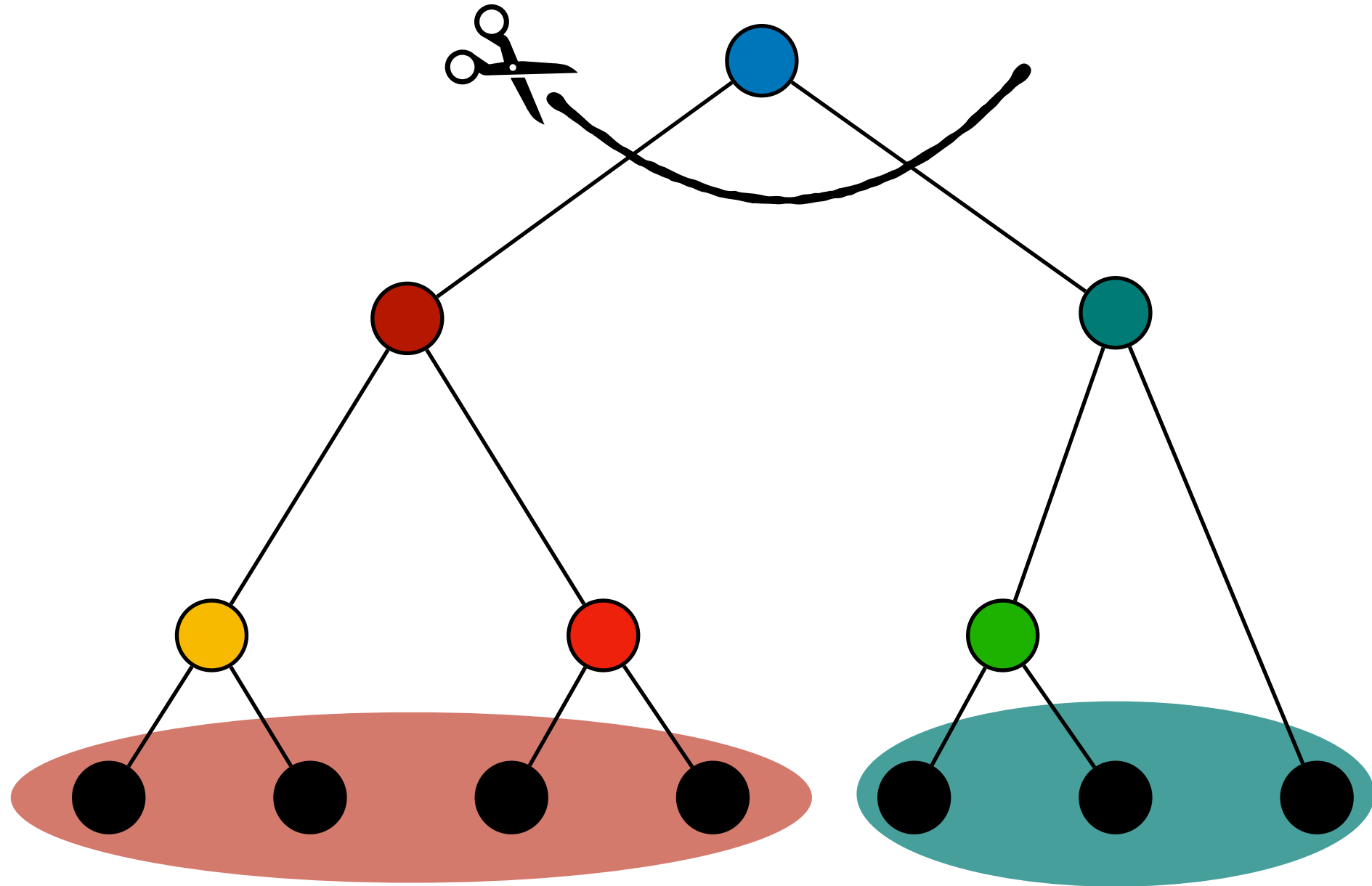
Iteratively merge two *“closest”* clusters

Hierarchical Agglomerative Clustering



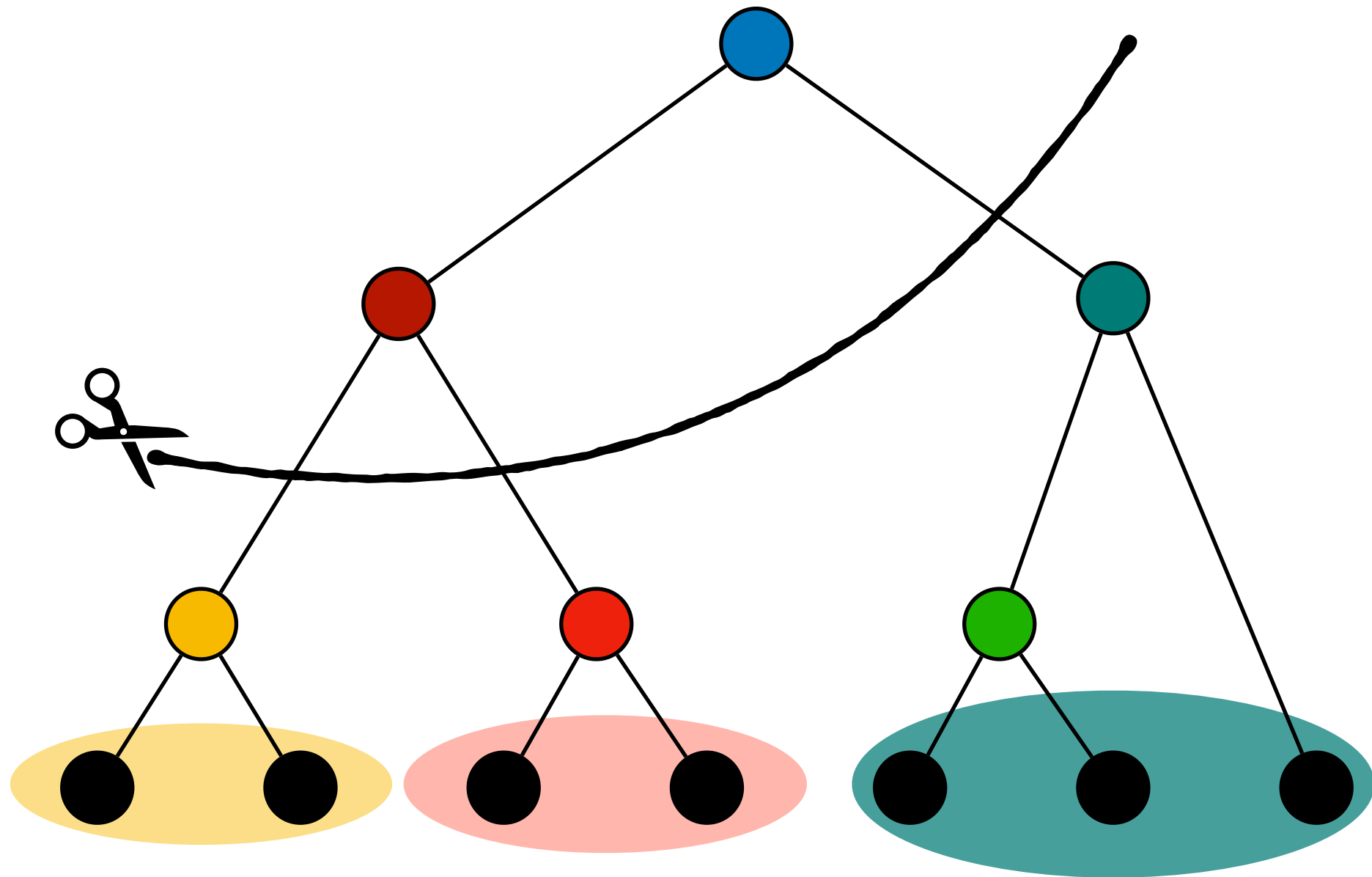
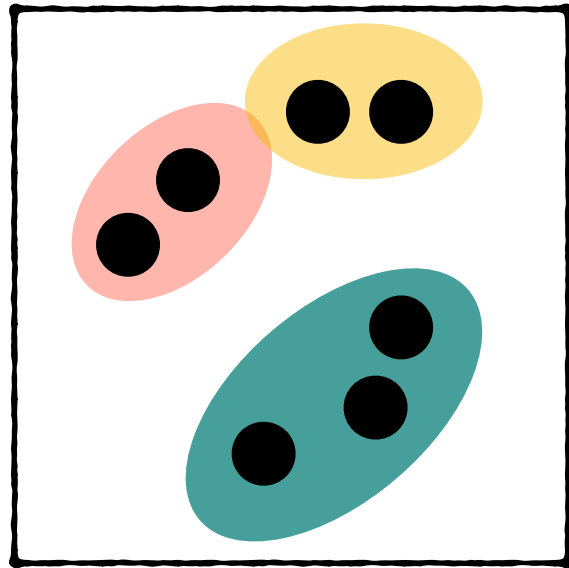
Iteratively merge two *“closest”* clusters

Hierarchical Agglomerative Clustering



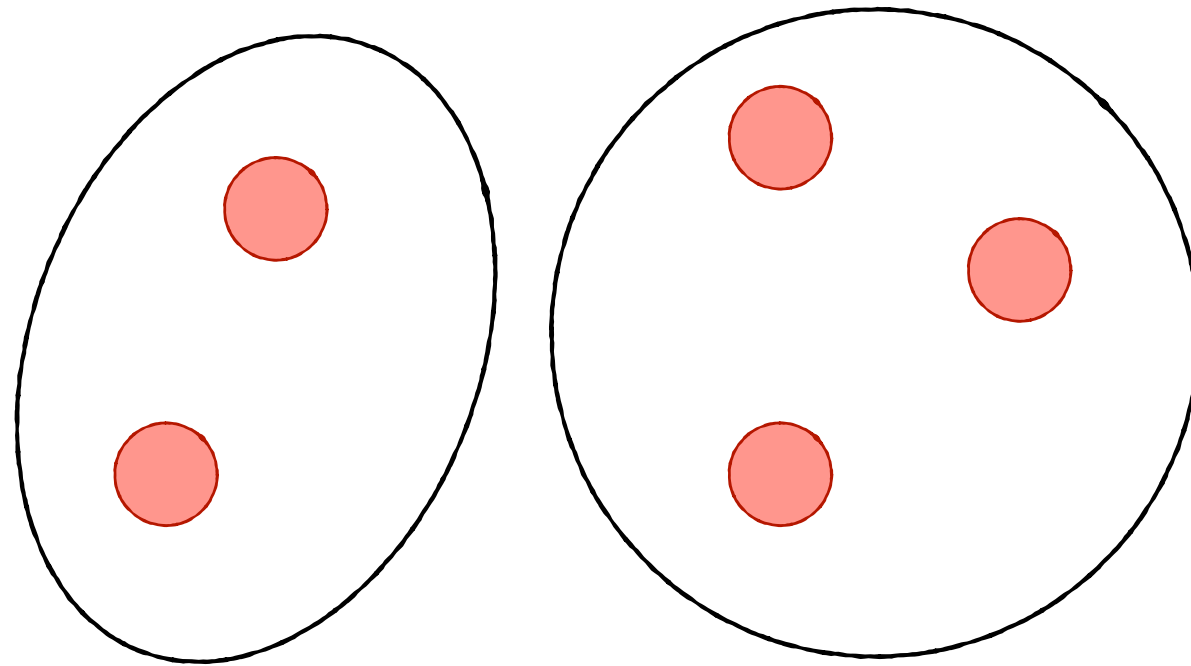
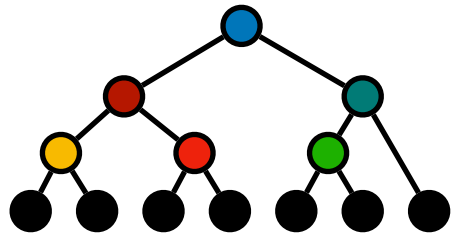
Iteratively merge two *“closest”* clusters

Hierarchical Agglomerative Clustering



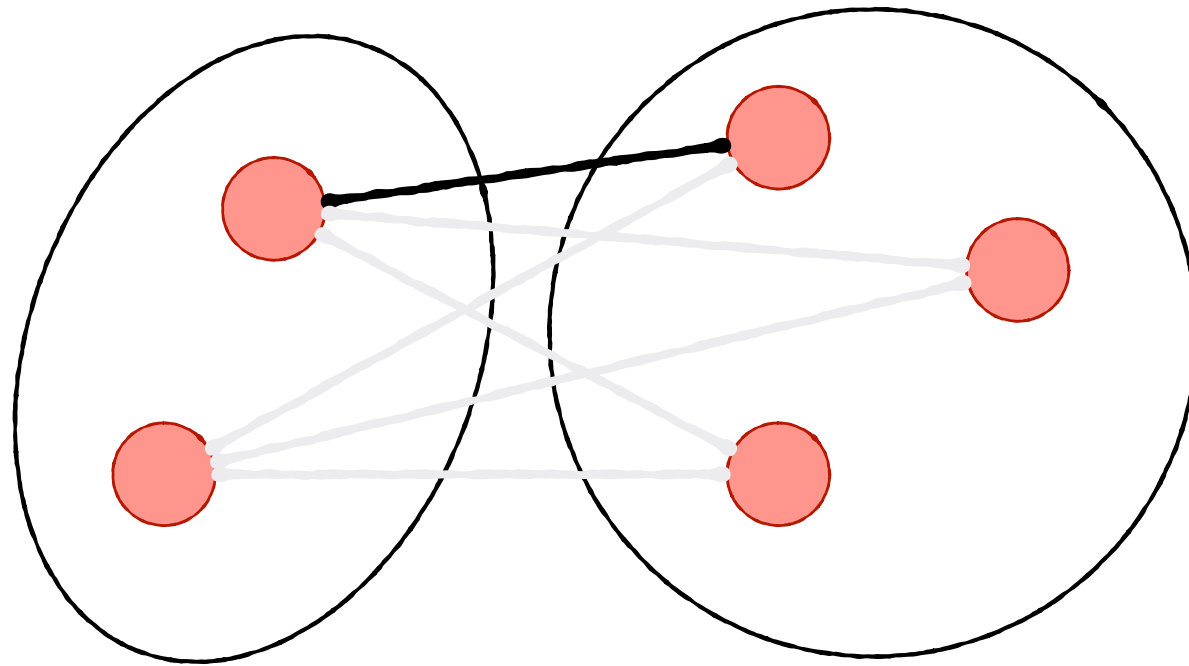
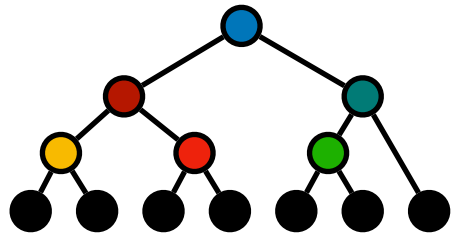
Iteratively merge two *“closest”* clusters

Linkage Function



Inter-Cluster distance given by **Linkage Function**

Linkage Function

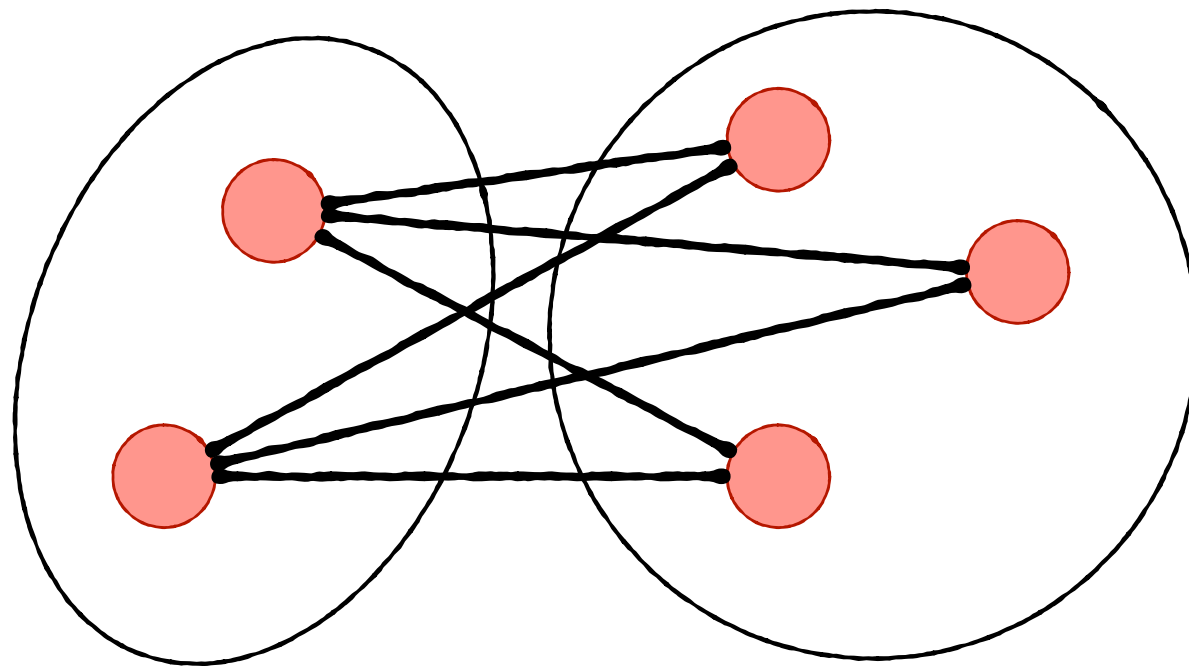
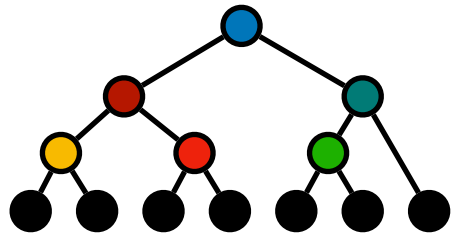


Single Linkage

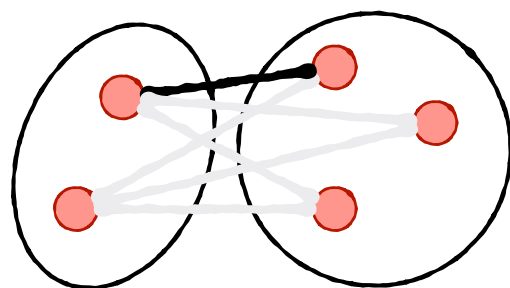
(Minimum Pairwise Dissimilarity)

Inter-Cluster distance given by **Linkage Function**

Linkage Function

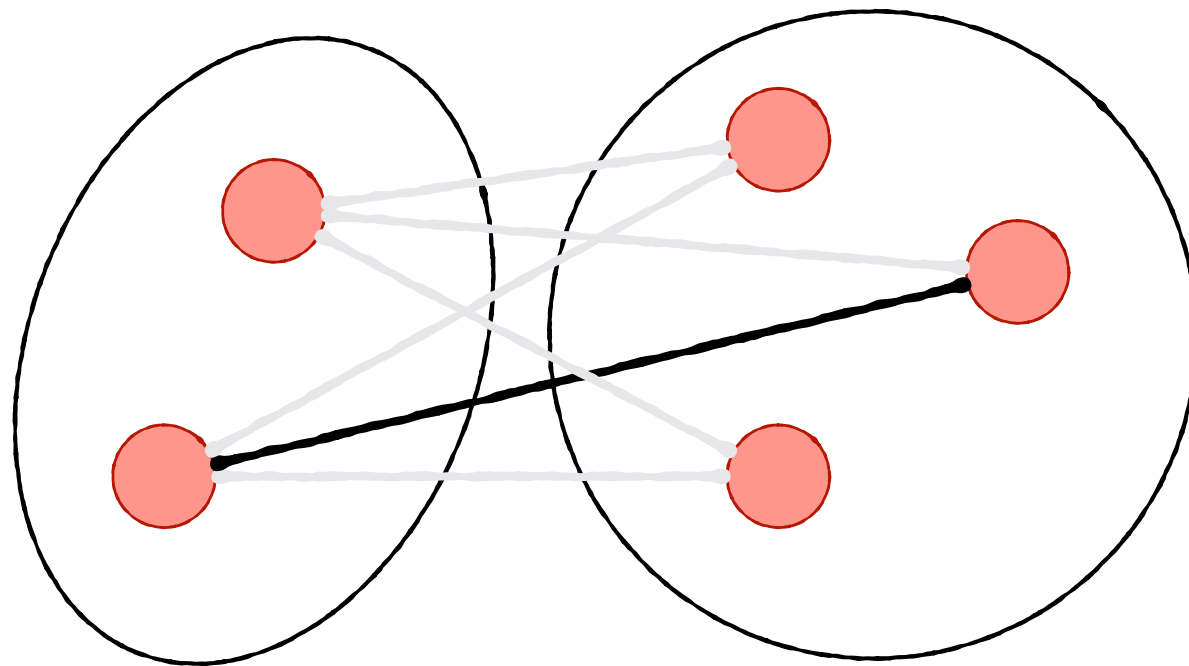
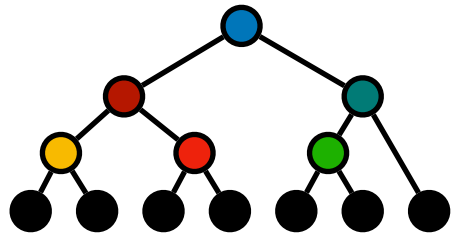


Average Linkage
(Average Pairwise Dissimilarity)

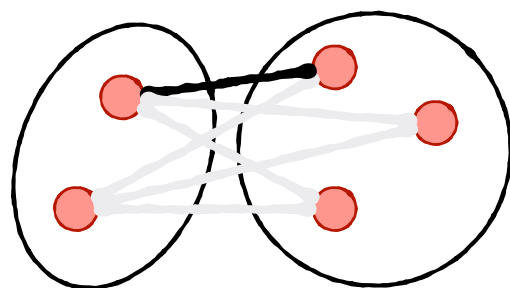


Single Linkage

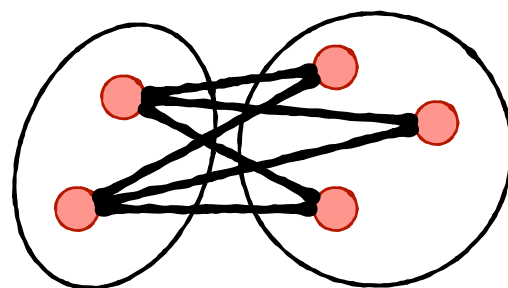
Linkage Function



Complete Linkage
(Maximum Pairwise Dissimilarity)

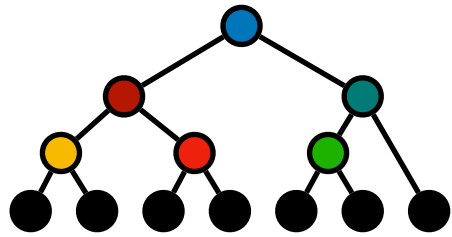


Single Linkage

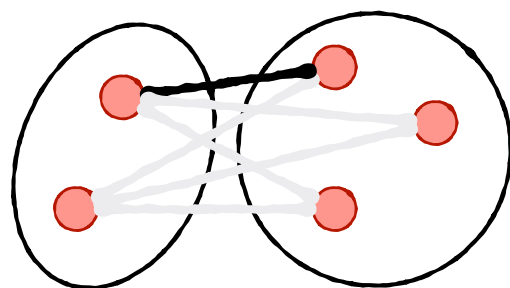


Average Linkage
6

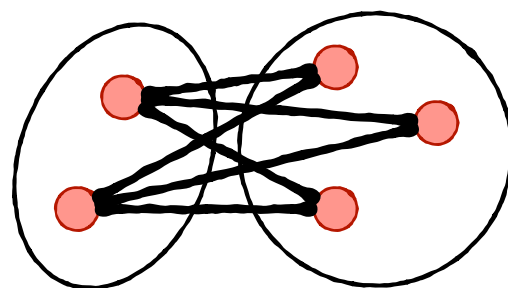
Linkage Function



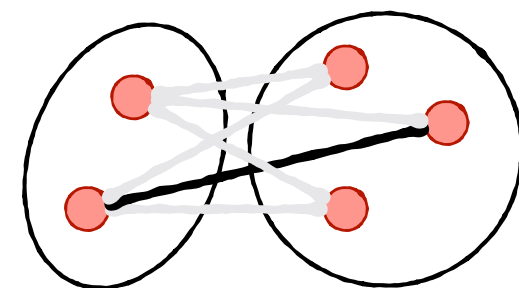
Best linkage function for a dataset is, *a priori*, unknown



Single Linkage



Average Linkage

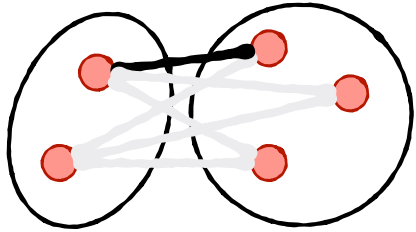


Complete Linkage

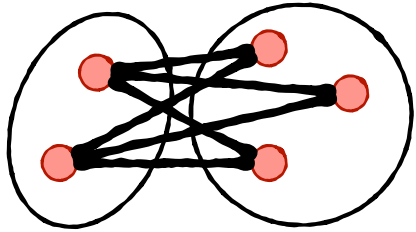
In this work:

1. **Exponential Linkage:** Learnable family of linkage functions
2. **Training objective** to jointly optimize linkage & dissimilarity function

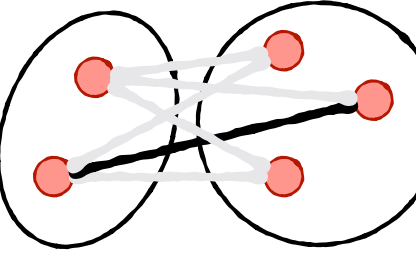
Exponential Linkage



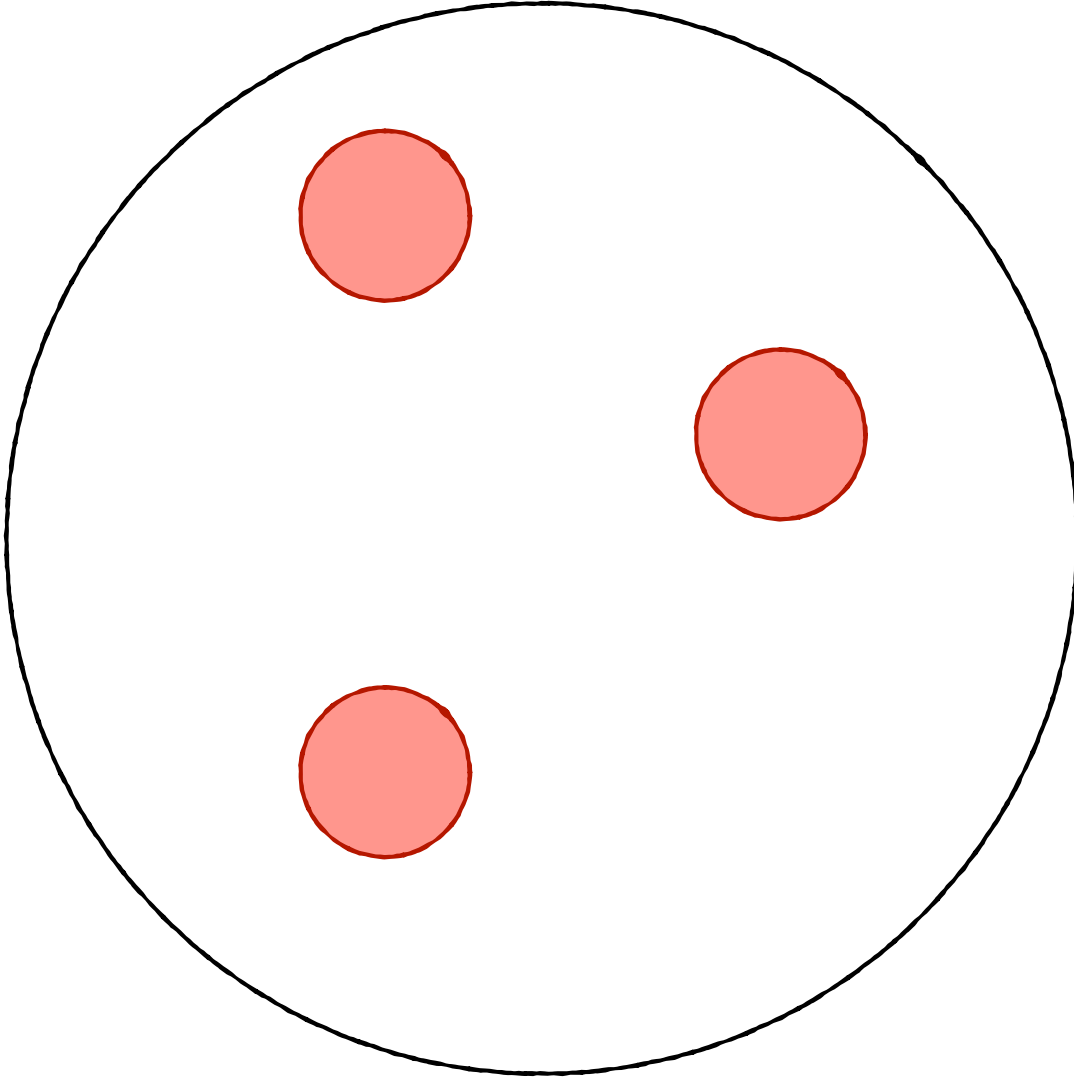
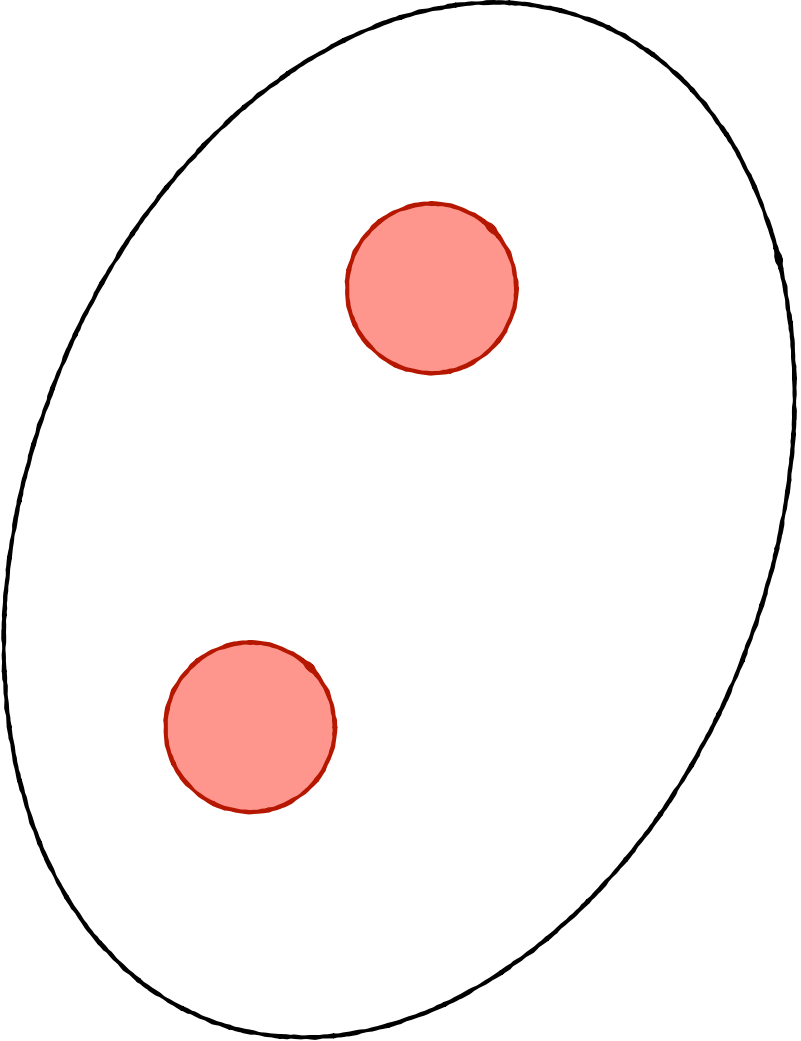
Single Linkage



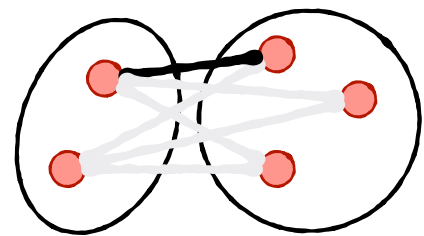
Average Linkage



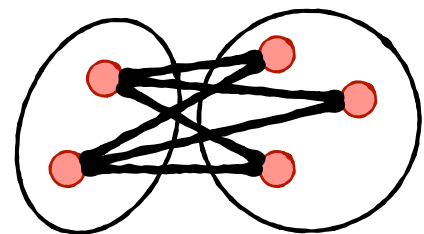
Complete Linkage



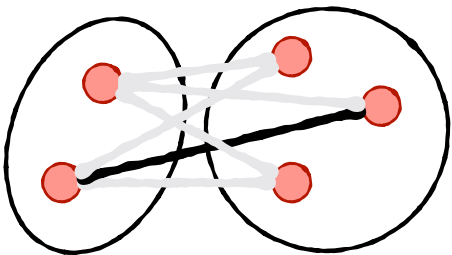
Exponential Linkage



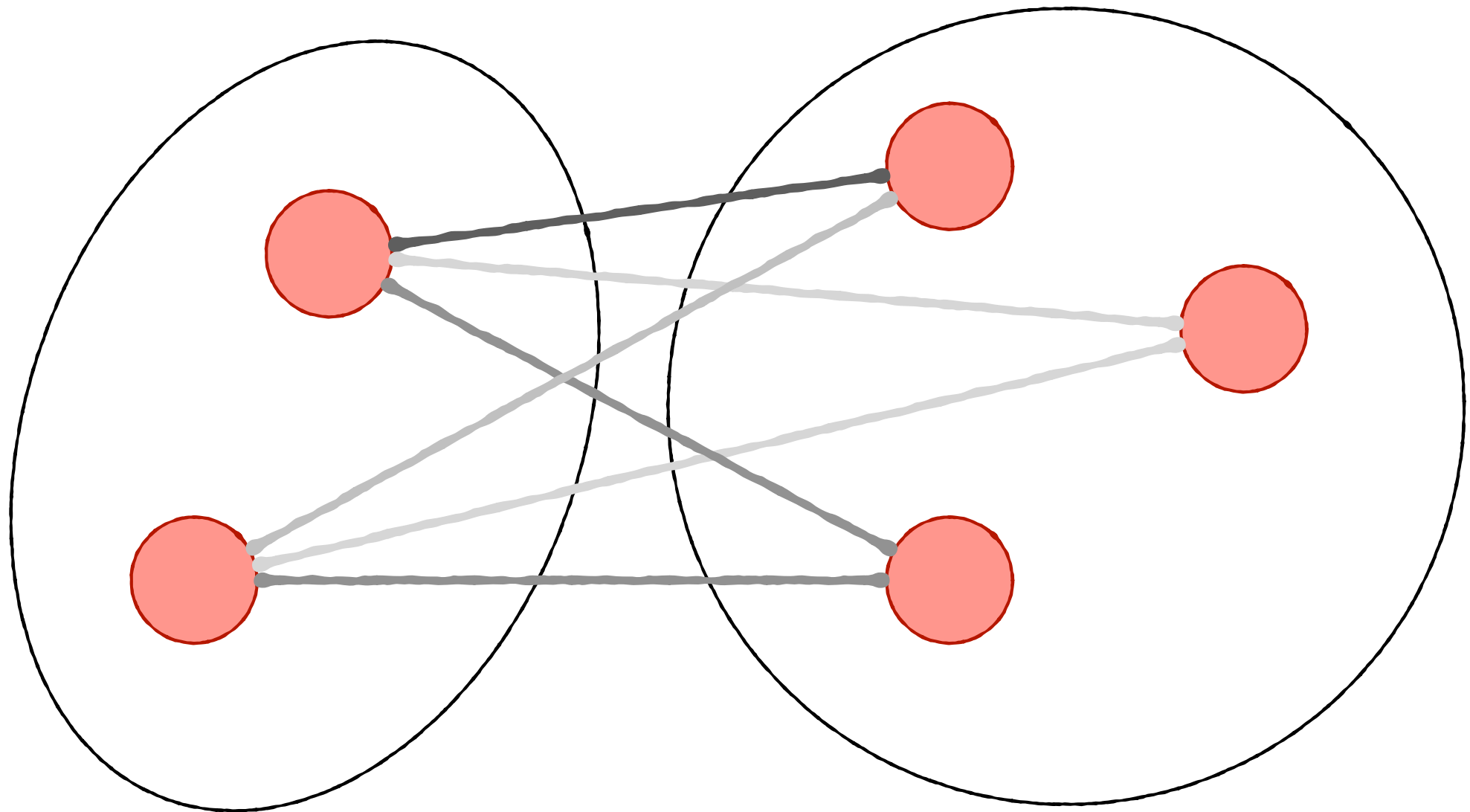
Single Linkage



Average Linkage



Complete Linkage



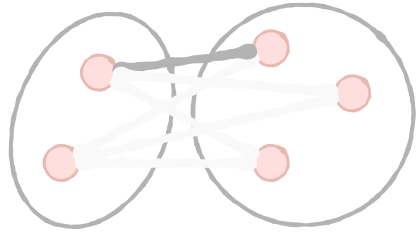
0

Weight

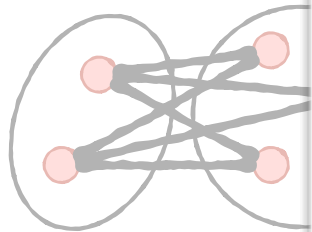
1

Weighted Average with **Learnable Parameter**

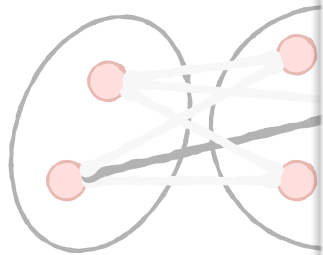
Exponential Linkage



Single Linkage



Average Link



Complete Li

$$J(\theta, \alpha) = \sum_{i=1}^{n'} \sum_{\mathbf{C}_{u,v} \in \mathcal{P}_-^{(i)}} \max \left\{ 0, \Psi^\alpha(\mathbf{C}_{u',v'}) - \Psi^\alpha(\mathbf{C}_{u,v}) \right\}$$

Algorithm 1 train_ExpLink($\mathcal{X}, \mathcal{C}^*, T, \gamma_1, \gamma_2$)

Init: θ, α

for $t = 1, \dots, T$ **do**

$J \leftarrow 0$

$\mathcal{T}_j^{(0)} \leftarrow \{x_j\} \quad \forall x_j \in \mathcal{X}$

for round $i = 1, \dots, n'$ **do**

$\{\mathcal{T}_k^{(i)}\}_k^{l_i} \leftarrow \text{HAC-Round}(\{\mathcal{T}_k^{(i-1)}\}_k^{l_{i-1}})$

$\{\mathcal{C}^{(i)}\}_k^{l_i} \leftarrow \{1 \text{ vs } \mathcal{T}_k^{(i)}\}_k^{l_i}$

$\mathcal{C}^{(i)} \leftarrow \{\mathcal{C}^{(i)}\}_k^{l_i}$

$\mathcal{P}^{(i)} \leftarrow \{\mathbf{C}_{u,v} \in \mathcal{C}^{(i)} \times \mathcal{C}^{(i)} : C_u \neq C_v\}$

$\mathcal{P}_+^{(i)} \leftarrow \{\mathbf{C}_{u,v} \in \mathcal{P}^{(i)} : \exists C_j^* \text{ s.t. } C_u, C_v \subset C_j^*\}$

$\mathcal{P}_-^{(i)} \leftarrow \mathcal{P}^{(i)} \setminus \mathcal{P}_+^{(i)}$

$\mathbf{C}_{u',v'} \leftarrow \arg \min_{\mathbf{C}_{u,v} \in \mathcal{P}_+^{(i)}} \Psi^\alpha(\mathbf{C}_{u,v})$

for $\mathbf{C}_{u,v} \in \mathcal{P}_-^{(i)}$ **do**

$J \leftarrow J + \max \left\{ 0, \Psi^\alpha(\mathbf{C}_{u',v'}) - \Psi^\alpha(\mathbf{C}_{u,v}) \right\}$

$\theta \leftarrow \theta - \gamma_1 \frac{\partial J}{\partial \theta}$

$\alpha \leftarrow \alpha - \gamma_2 \frac{\partial J}{\partial \alpha}$

Experimental Setup

Entity Resolution

REXA

AMINER

Type Classes in Haskell. Hall, C. V. and Hammond, K. and **Jones, S.** and Wadler, P. *Programming Languages and Systems.* 1996.

Imperative Function Programming. **Peyton Jones, S.** and Wadler, P. *Principles of Programming Languages.* 1993.

The Implementer's Dilemma: A Mathematical Model of Compile Time Garbage Collection. **Jones, S.** and Tyas, A. *Functional Programming.* 1993.



UMIST Faces



Noun Phrase
Coreference

Julie Foudy played in four FIFA Women's World Cup tournaments, winning two FIFA Women's World Cups—in 1991 and 1999. **She** played in three Summer Olympic Games, winning an Olympic Gold Medal in 1996, Silver in 2000, and Gold again in 2004.

Experimental Setup

Entity Resolution

REXA

AMINER

Type Classes in Haskell. Hall, C. V. and Hammond, K. and Jones, S. and Wadler, P. *Programming Languages and Systems*. 1996.

Imperative Function Programming. Peyton Jones, S. and Wadler, P. *Principles of Programming Languages*. 1993.

The Implementer's Dilemma: A Mathematical Model of Compile Time Garbage Collection. Jones, S. and Tyas, A. *Functional Programming*. 1993.



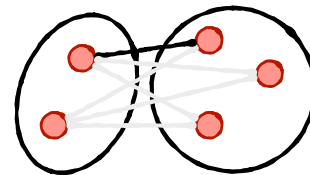
UMIST Faces



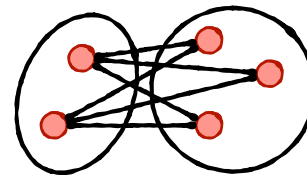
Noun Phrase Coreference

Julie Foudy played in four FIFA Women's World Cup tournaments, winning two FIFA Women's World Cups—in 1991 and 1999. She played in three Summer Olympic Games, winning an Olympic Gold Medal in 1996, Silver in 2000, and Gold again in 2004.

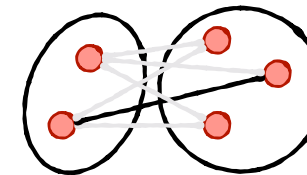
Four Linkage Functions



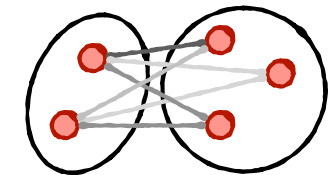
Single Linkage



Average Linkage



Complete Linkage



Exponential Linkage

Experimental Setup

Entity Resolution

REXA

AMINER

Type Classes in Haskell. Hall, C. V. and Hammond, K. and Jones, S. and Wadler, P. *Programming Languages and Systems*. 1996.

Imperative Function Programming. Peyton Jones, S. and Wadler, P. *Principles of Programming Languages*. 1993.

The Implementer's Dilemma: A Mathematical Model of Compile Time Garbage Collection. Jones, S. and Tyas, A. *Functional Programming*. 1993.



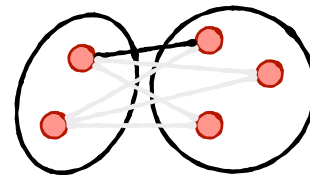
UMIST Faces



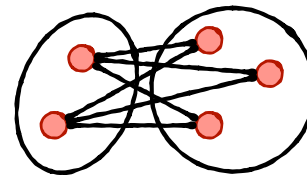
Noun Phrase Coreference

Julie Foudy played in four FIFA Women's World Cup tournaments, winning two FIFA Women's World Cups—in 1991 and 1999. She played in three Summer Olympic Games, winning an Olympic Gold Medal in 1996, Silver in 2000, and Gold again in 2004.

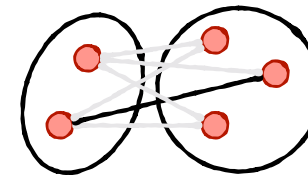
Four Linkage Functions



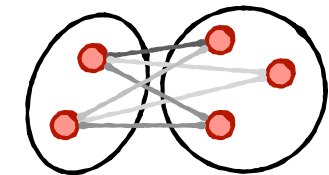
Single Linkage



Average Linkage



Complete Linkage



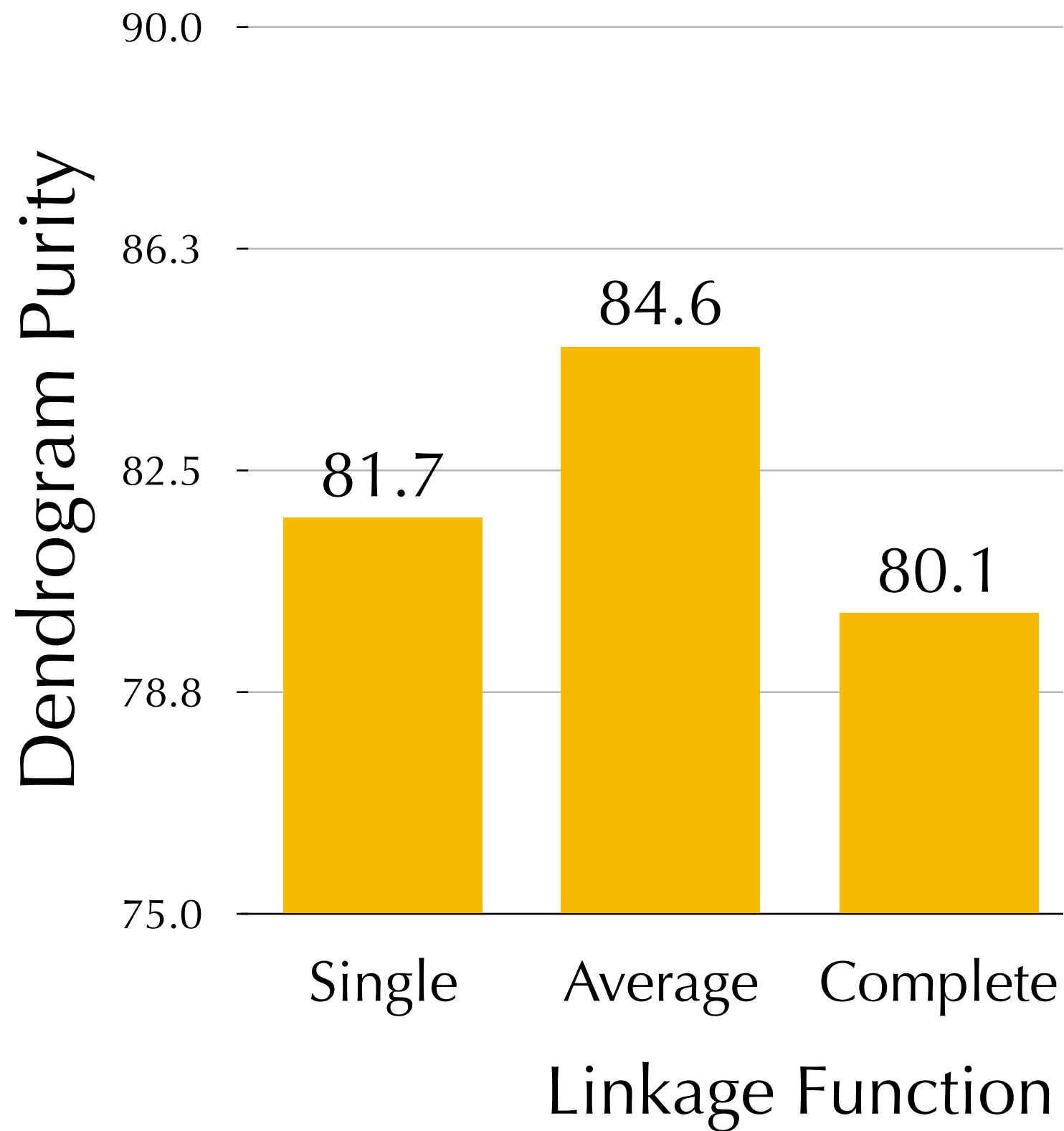
Exponential Linkage

Evaluated using **Dendrogram Purity**

Averaged across 50 different train/test/dev splits

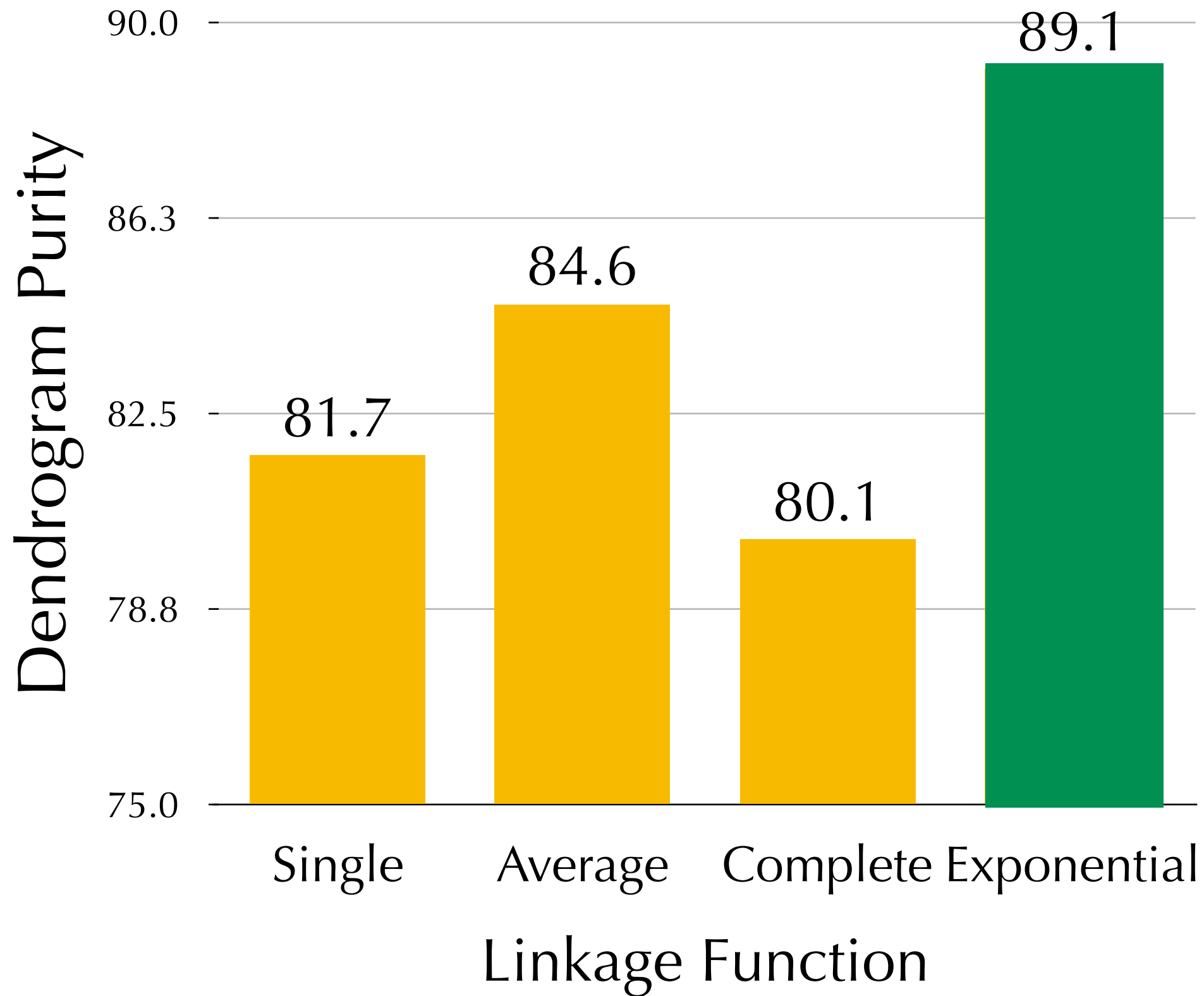
Results

Dataset: Rexa



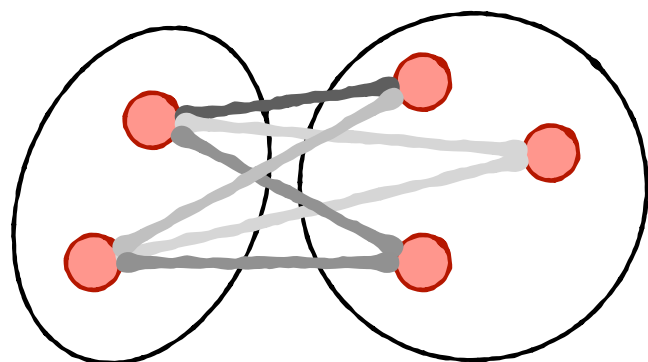
Results

Dataset: Rexa



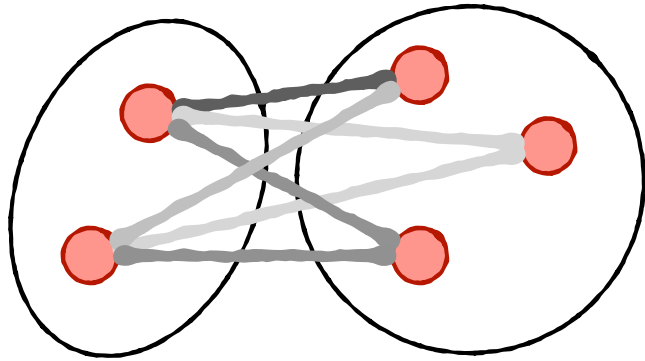
Summary

Summary



Exponential Linkage: Learnable family of linkage functions

Summary



$$J(\theta, \alpha) = \sum_{i=1}^{n'} \sum_{\mathbf{C}_{u,v} \in \mathcal{P}_-^{(i)}} \max \{0, \Psi^\alpha(\mathbf{C}_{u',v'}) - \Psi^\alpha(\mathbf{C}_{u,v})\}$$

Algorithm 1 train_ExpLink($\mathcal{X}, \mathcal{C}^*, T, \gamma_1, \gamma_2$)

```

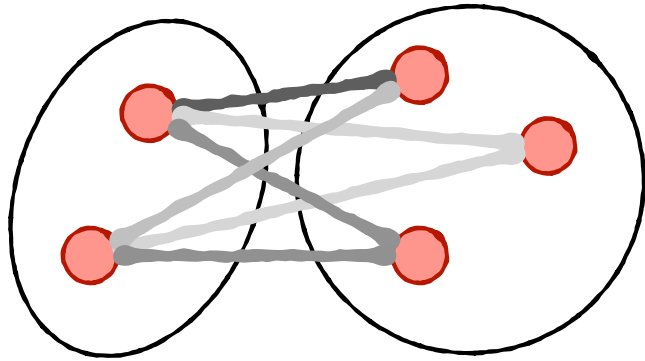
Init:  $\theta, \alpha$ 
for  $t = 1, \dots, T$  do
   $J \leftarrow 0$ 
   $\mathcal{T}_j^{(0)} \leftarrow \{x_j\} \quad \forall x_j \in \mathcal{X}$ 
  for round  $i = 1, \dots, n'$  do
     $\{\mathcal{T}_k^{(i)}\}_k^{l_i} \leftarrow \text{HAC-Round}(\{\mathcal{T}_k^{(i-1)}\}_k^{l_{i-1}})$ 
     $\{\mathcal{C}^{(i)}\}_k^{l_i} \leftarrow \{\text{1vs}\mathcal{T}_k^{(i)}\}_k^{l_i}$ 
     $\mathcal{C}^{(i)} \leftarrow \{\mathcal{C}^{(i)}\}_k^{l_i}$ 
     $\mathcal{P}^{(i)} \leftarrow \{\mathbf{C}_{u,v} \in \mathcal{C}^{(i)} \times \mathcal{C}^{(i)} : C_u \neq C_v\}$ 
     $\mathcal{P}_+^{(i)} \leftarrow \{\mathbf{C}_{u,v} \in \mathcal{P}^{(i)} : \exists \mathcal{C}_j^* \text{ s.t. } C_u, C_v \subset \mathcal{C}_j^*\}$ 
     $\mathcal{P}_-^{(i)} \leftarrow \mathcal{P}^{(i)} \setminus \mathcal{P}_+^{(i)}$ 
     $\mathbf{C}_{u',v'} \leftarrow \arg \min_{\mathbf{C}_{u,v} \in \mathcal{P}_+^{(i)}} \Psi^\alpha(\mathbf{C}_{u,v})$ 
    for  $\mathbf{C}_{u,v} \in \mathcal{P}_-^{(i)}$  do
       $J \leftarrow J + \max \{0, \Psi^\alpha(\mathbf{C}_{u',v'}) - \Psi^\alpha(\mathbf{C}_{u,v})\}$ 
   $\theta \leftarrow \theta - \gamma_1 \frac{\partial J}{\partial \theta}$ 
   $\alpha \leftarrow \alpha - \gamma_2 \frac{\partial J}{\partial \alpha}$ 

```

Exponential Linkage: Learnable family of linkage functions

Training Objective & Algorithm: Jointly Optimizing Dissimilarity & Linkage Function

Summary



$$J(\theta, \alpha) = \sum_{i=1}^{n'} \sum_{\mathbf{C}_{u,v} \in \mathcal{P}_-^{(i)}} \max \{0, \Psi^\alpha(\mathbf{C}_{u',v'}) - \Psi^\alpha(\mathbf{C}_{u,v})\}$$

Algorithm 1 train_ExpLink($\mathcal{X}, \mathcal{C}^*, T, \gamma_1, \gamma_2$)

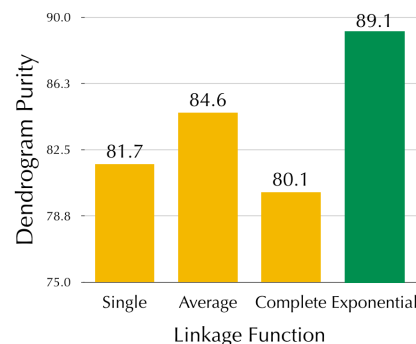
```

Init:  $\theta, \alpha$ 
for  $t = 1, \dots, T$  do
   $J \leftarrow 0$ 
   $\mathcal{T}_j^{(0)} \leftarrow \{x_j\} \quad \forall x_j \in \mathcal{X}$ 
  for round  $i = 1, \dots, n'$  do
     $\{\mathcal{T}_k^{(i)}\}_k \leftarrow \text{HAC-Round}(\{\mathcal{T}_k^{(i-1)}\}_k^{l_{i-1}})$ 
     $\{\mathcal{C}_k^{(i)}\}_k \leftarrow \{\text{1vs}\mathcal{T}_k^{(i)}\}_k^{l_i}$ 
     $\mathcal{C}^{(i)} \leftarrow \{\mathcal{C}^{(i)}\}_k^{l_i}$ 
     $\mathcal{P}^{(i)} \leftarrow \{\mathbf{C}_{u,v} \in \mathcal{C}^{(i)} \times \mathcal{C}^{(i)} : \mathbf{C}_u \neq \mathbf{C}_v\}$ 
     $\mathcal{P}_+^{(i)} \leftarrow \{\mathbf{C}_{u,v} \in \mathcal{P}^{(i)} : \exists \mathcal{C}_j^* \text{ s.t. } \mathbf{C}_u, \mathbf{C}_v \subset \mathcal{C}_j^*\}$ 
     $\mathcal{P}_-^{(i)} \leftarrow \mathcal{P}^{(i)} \setminus \mathcal{P}_+^{(i)}$ 
     $\mathbf{C}_{u',v'} \leftarrow \arg \min_{\mathbf{C}_{u,v} \in \mathcal{P}_-^{(i)}} \Psi^\alpha(\mathbf{C}_{u,v})$ 
    for  $\mathbf{C}_{u,v} \in \mathcal{P}_-^{(i)}$  do
       $J \leftarrow J + \max \{0, \Psi^\alpha(\mathbf{C}_{u',v'}) - \Psi^\alpha(\mathbf{C}_{u,v})\}$ 
   $\theta \leftarrow \theta - \gamma_1 \frac{\partial J}{\partial \theta}$ 
   $\alpha \leftarrow \alpha - \gamma_2 \frac{\partial J}{\partial \alpha}$ 

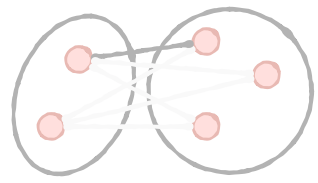
```

Exponential Linkage: Learnable family of linkage functions

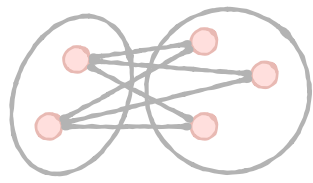
Training Objective & Algorithm: Jointly Optimizing Dissimilarity & Linkage Function



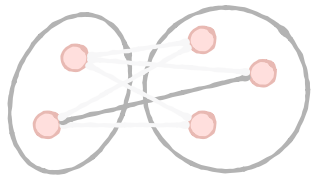
Effective Empirical Results



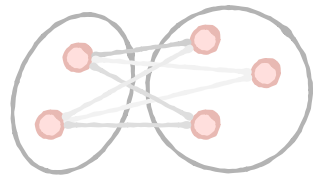
Single Linkage



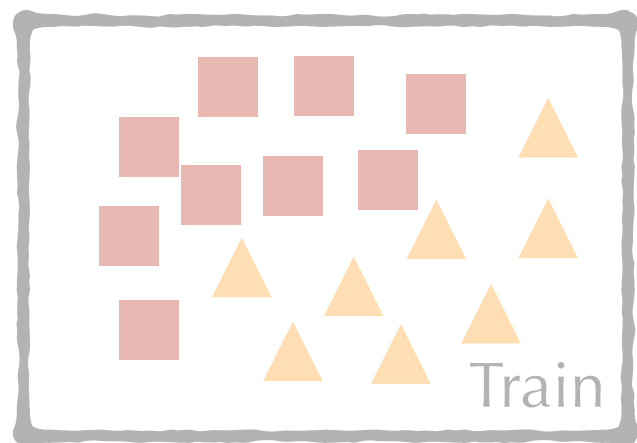
Average Linkage



Complete Linkage



Exponential Linkage



Thanks for listening!

Check out our poster #196 today at 6:30pm in Pacific Ballroom!

Paper:



Code:

