Machine Learning for Data Science (CS4786)
Lecture 11

Clustering + Linkage Clustering

Thorsten Joachims

Outline for Upcoming Lectures

- What is clustering?
- Hierarchical Clustering
  - Hierarchical Agglomerative Clustering (HAC)
- Non-Hierarchical Clustering
  - K-means
  - Mixtures of Gaussians and EM-Algorithm

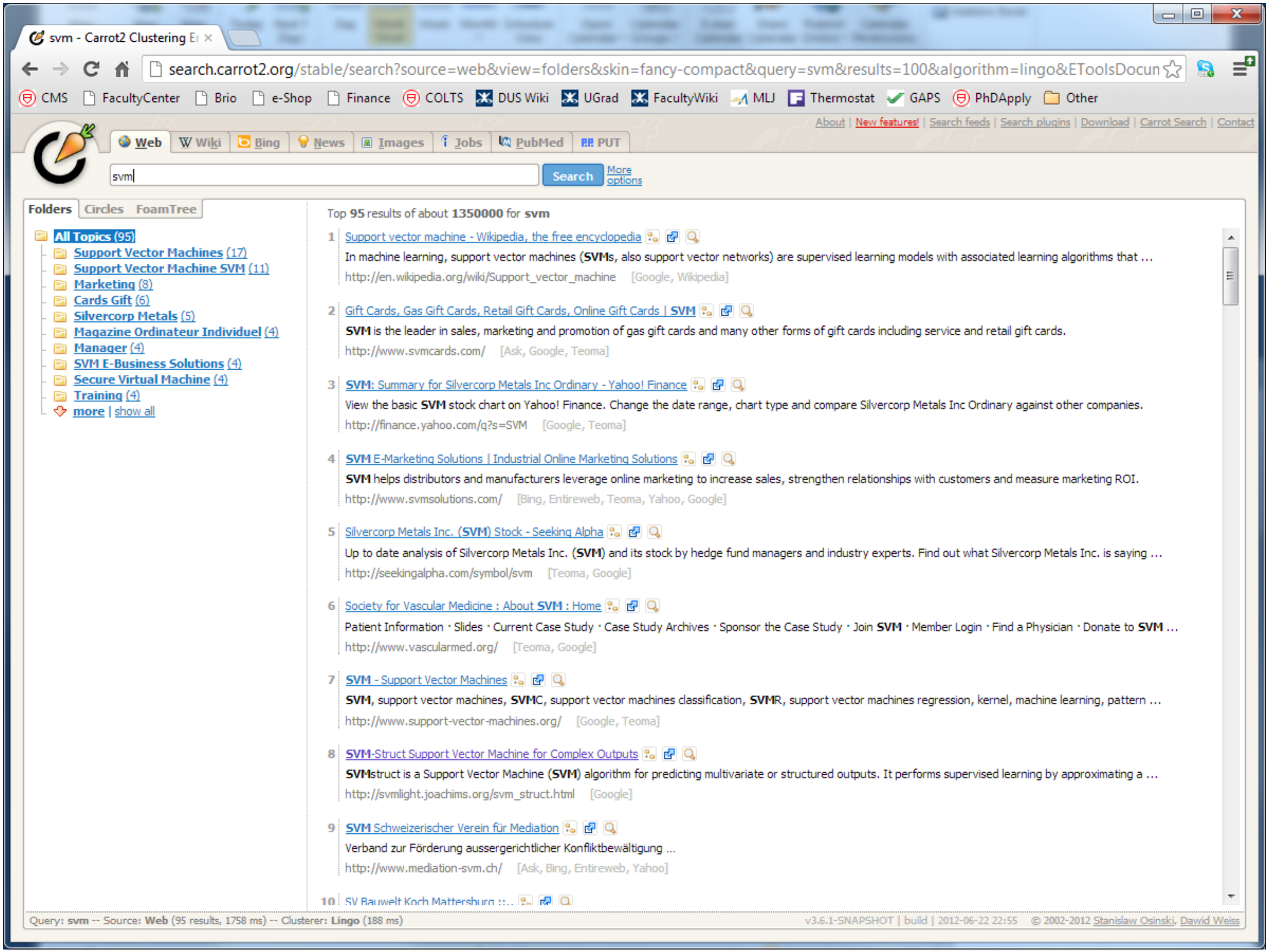
Intuitive Definition of Clustering

- Partition a set of data points into groups so that
  - points in the same group are similar, and
  - points in different groups are dissimilar.

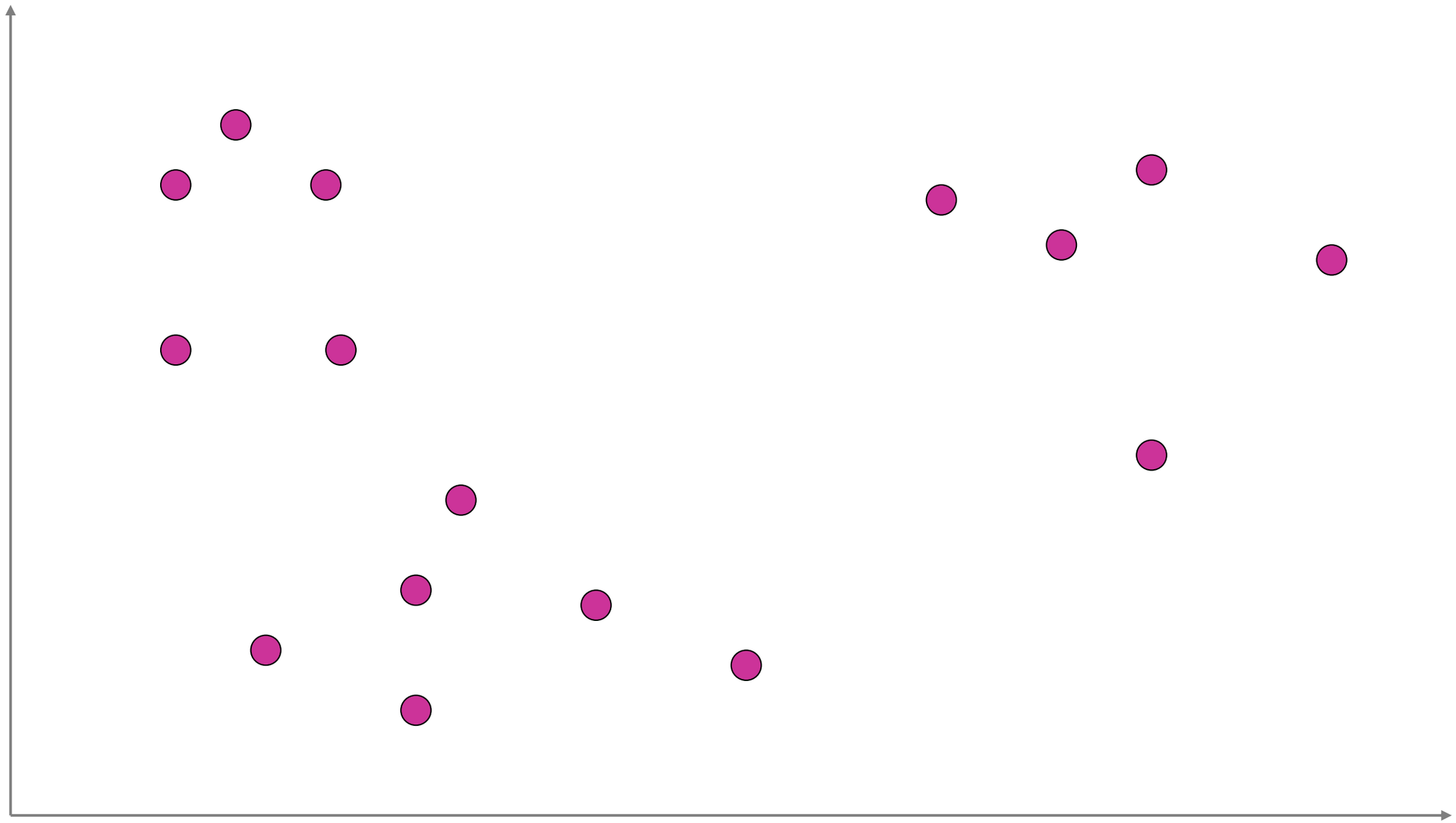
→ A form of “unsupervised classification” where there are no predefined class labels.

Applications of Clustering

- Exploratory data analysis
- Cluster retrieved documents
  - to present more organized and understandable results to user → “diversified retrieval”
- Detecting near duplicates
  - Entity resolution
    - E.g. “Thorsten Joachims” == “Thorsten B Joachims”
  - Cheating detection
- Automated (or semi-automated) creation of taxonomies
  - e.g. Yahoo, DMOZ, Wikipedia
- Compression

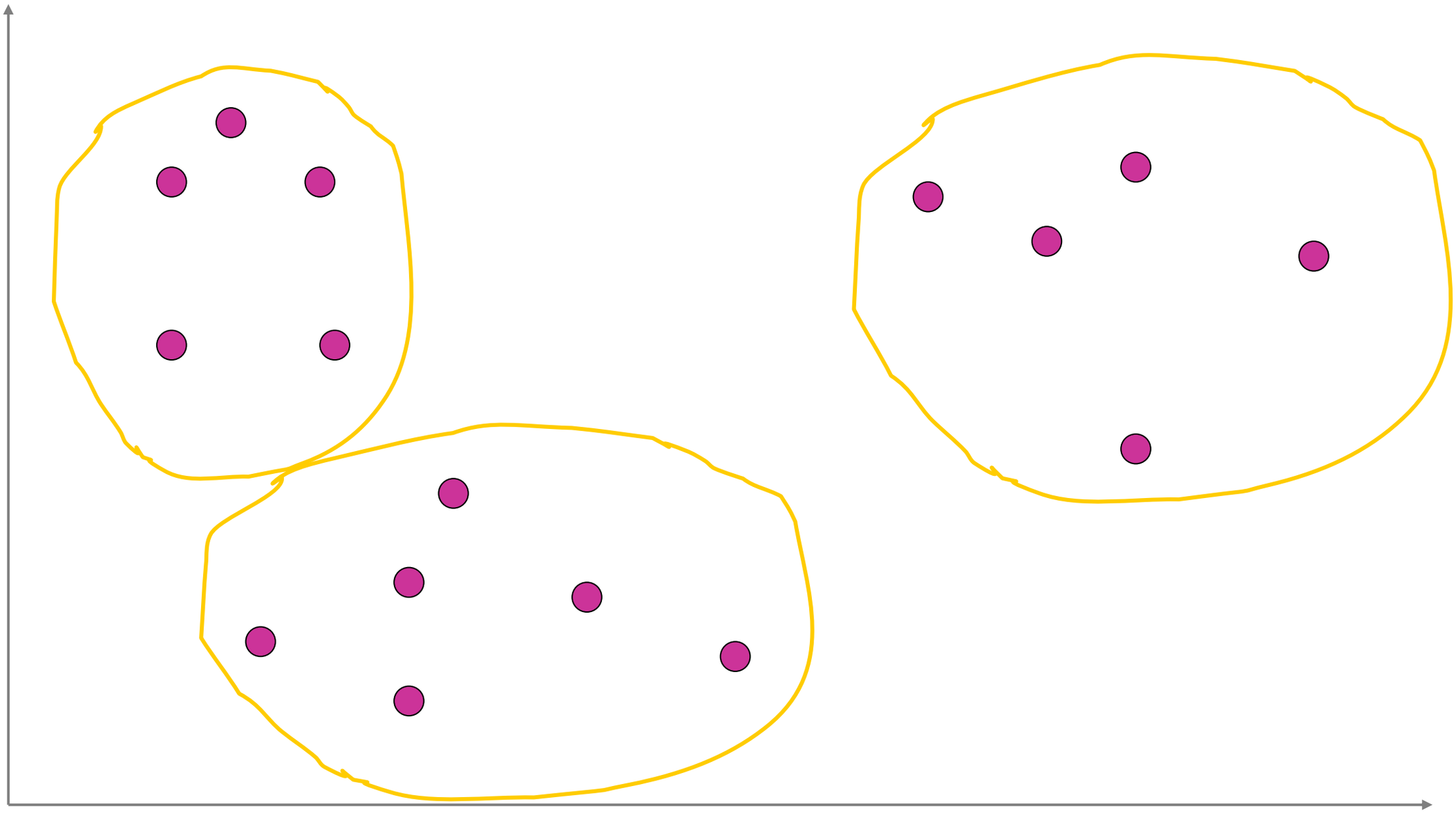

svm - Carrot2 Clustering E
search.carrot2.org/stable/search?source=web&view=folders&skin=fancy-compact&query=svm&results=100&algorithm=lingo&ETooIsDocun
CMS
FacultyCenter
Brio
e-Shop
Finance
COLTS
DUS Wiki
UGrad
FacultyWiki
MLJ
Thermostat
GAPS
PhDApply
Other
About | New features! | Search feeds | Search plugins | Download | Carrot Search | Contact
Web
Wiki
Bing
News
Images
Jobs
PubMed
PUT
svm
Search
More options
Folders
Circles
FoamTree
All Topics (95)
Support Vector Machines (17)
Support Vector Machine SVM (11)
Marketing (8)
Cards Gift (6)
Silvercorp Metals (5)
Magazine Ordinateur Individuel (4)
Manager (4)
SVM E-Business Solutions (4)
Secure Virtual Machine (4)
Training (4)
more | show all
Top 95 results of about 1350000 for svm
1 Support vector machine - Wikipedia, the free encyclopedia
In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that ...
http://en.wikipedia.org/wiki/Support_vector_machine [Google, Wikipedia]
2 Gift Cards, Gas Gift Cards, Retail Gift Cards, Online Gift Cards | SVM
SVM is the leader in sales, marketing and promotion of gas gift cards and many other forms of gift cards including service and retail gift cards.
http://www.svmcards.com/ [Ask, Google, Teoma]
3 SVM: Summary for Silvercorp Metals Inc Ordinary - Yahoo! Finance
View the basic SVM stock chart on Yahoo! Finance. Change the date range, chart type and compare Silvercorp Metals Inc Ordinary against other companies.
http://finance.yahoo.com/q?s=SVM [Google, Teoma]
4 SVM E-Marketing Solutions | Industrial Online Marketing Solutions
SVM helps distributors and manufacturers leverage online marketing to increase sales, strengthen relationships with customers and measure marketing ROI.
http://www.svmsolutions.com/ [Bing, Entireweb, Teoma, Yahoo, Google]
5 Silvercorp Metals Inc. (SVM) Stock - Seeking Alpha
Up to date analysis of Silvercorp Metals Inc. (SVM) and its stock by hedge fund managers and industry experts. Find out what Silvercorp Metals Inc. is saying ...
http://seekingalpha.com/symbol/svm [Teoma, Google]
6 Society for Vascular Medicine : About SVM : Home
Patient Information · Slides · Current Case Study · Case Study Archives · Sponsor the Case Study · Join SVM · Member Login · Find a Physician · Donate to SVM ...
http://www.vascularmed.org/ [Teoma, Google]
7 SVM - Support Vector Machines
SVM, support vector machines, SVMC, support vector machines classification, SVMR, support vector machines regression, kernel, machine learning, pattern ...
http://www.support-vector-machines.org/ [Google, Teoma]
8 SVM-Struct Support Vector Machine for Complex Outputs
SVMstruct is a Support Vector Machine (SVM) algorithm for predicting multivariate or structured outputs. It performs supervised learning by approximating a ...
http://svmlight.joachims.org/svm_struct.html [Google]
9 SVM Schweizerischer Verein für Mediation
Verband zur Förderung aussergerichtlicher Konfliktbewältigung ...
http://www.mediation-svm.ch/ [Ask, Bing, Entireweb, Yahoo]
10 SV Bauwelt Koch Mattersburg ::..
Query: svm -- Source: Web (95 results, 1758 ms) -- Clusterer: Lingo (188 ms)
v3.6.1-SNAPSHOT | build | 2012-06-22 22:55 © 2002-2012 Stanislaw Osinski, Dawid Weiss

Clustering Example

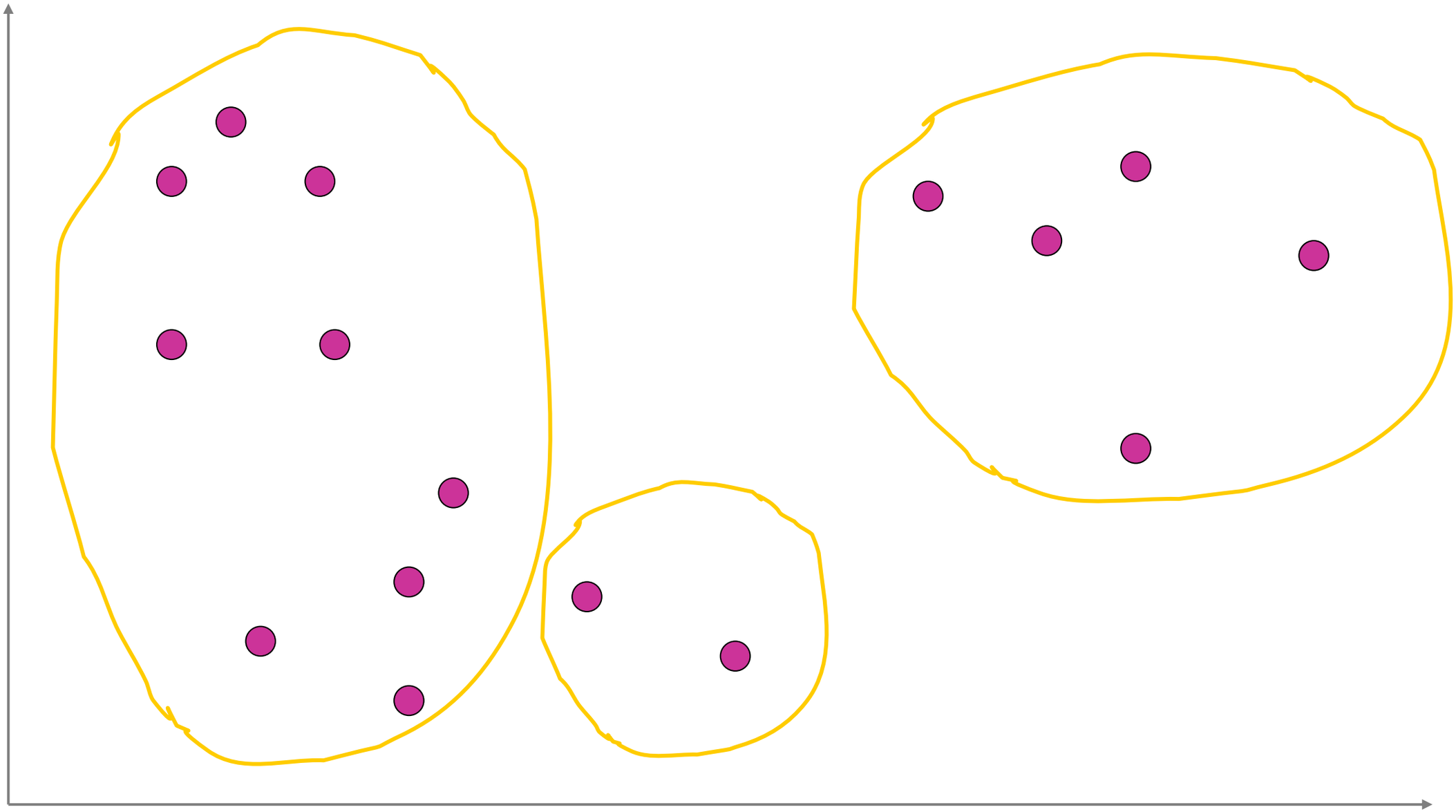


Clustering Example

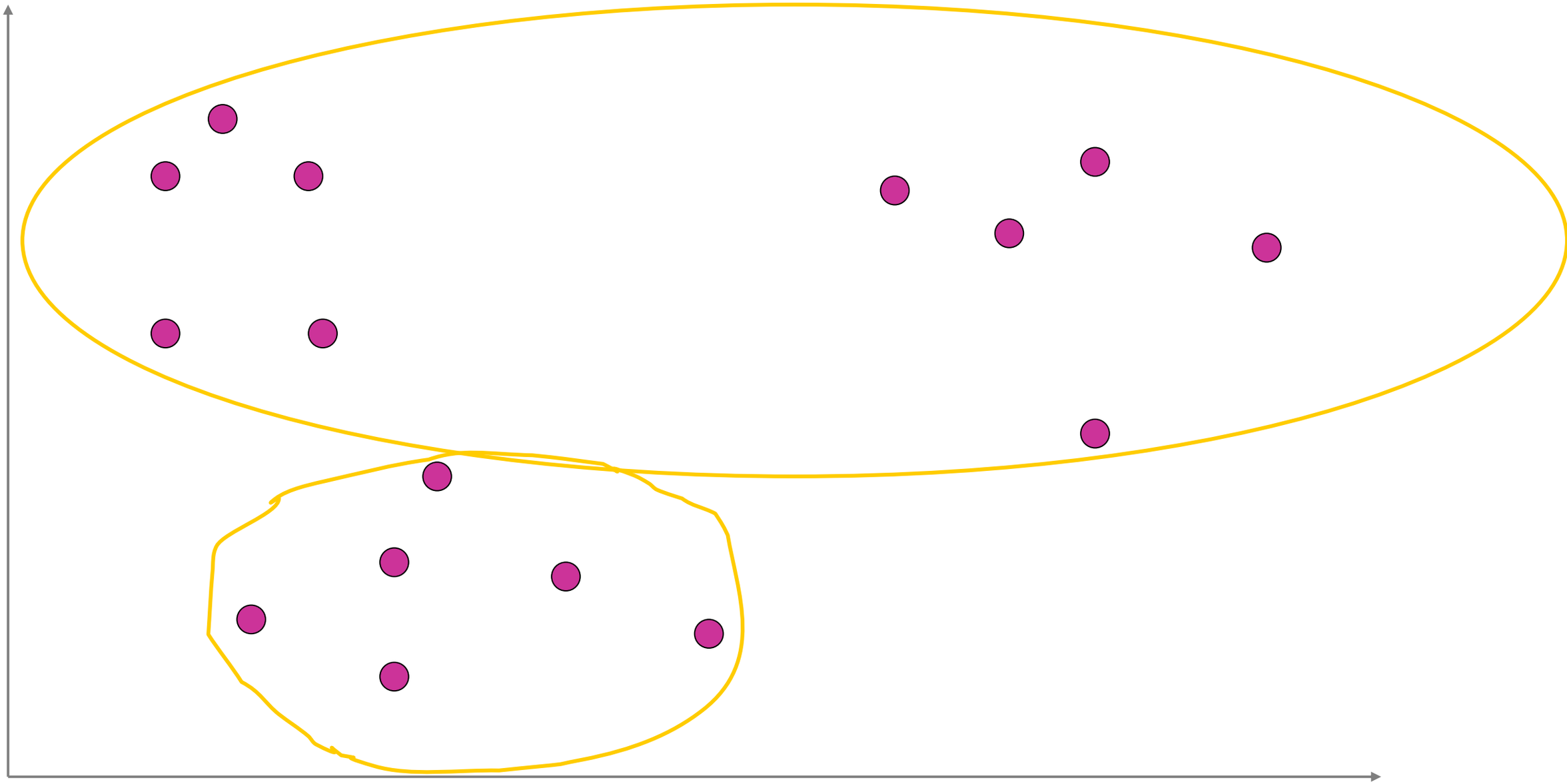
Clustering Example



Clustering Example



Clustering Example



Some Notation

- Set of n items to be clusters. Described by feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
- K-ary clustering partitions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ into K groups $C_1, \ldots, C_K$.
- Cluster C_i is the set of items in group i.
- $c(\boldsymbol{x}_i)$ denotes the cluster among $C_1, \ldots, C_K$ that contains $\boldsymbol{x}_i$.

Similarity/Distance Measures

- Euclidian distance (L_2 norm):

$$L_2(\boldsymbol{x}, \boldsymbol{x}') = \sqrt{\sum_{i=1}^{N} \left(x_i - x_i'\right)^2}$$

- L_1 norm:

$$L_1(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{N} |x_i - x_i'|$$

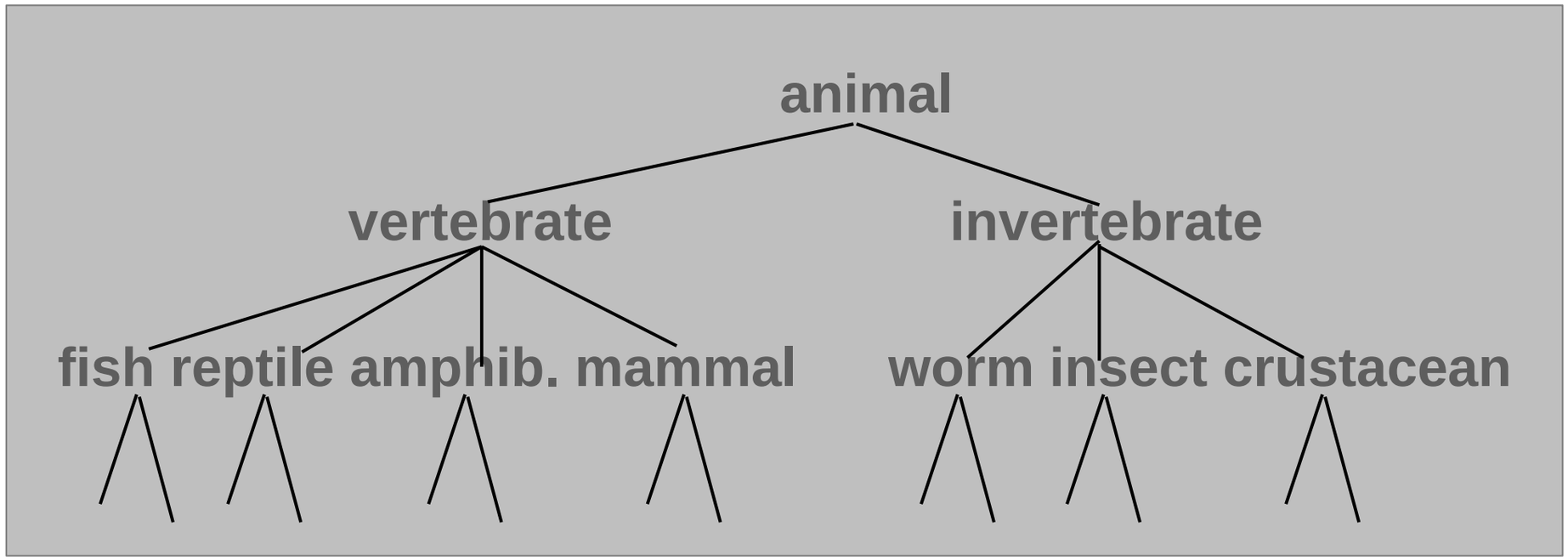
- Cosine similarity:

$$\cos(\boldsymbol{x}, \boldsymbol{x}') = \frac{\vec{x} * \vec{x}'}{\|\vec{x}\| \, \|\vec{x}'\|}$$

- Kernel-based similarity

Hierarchical Clustering

- Build a tree-based hierarchical taxonomy from a set of unlabeled examples.

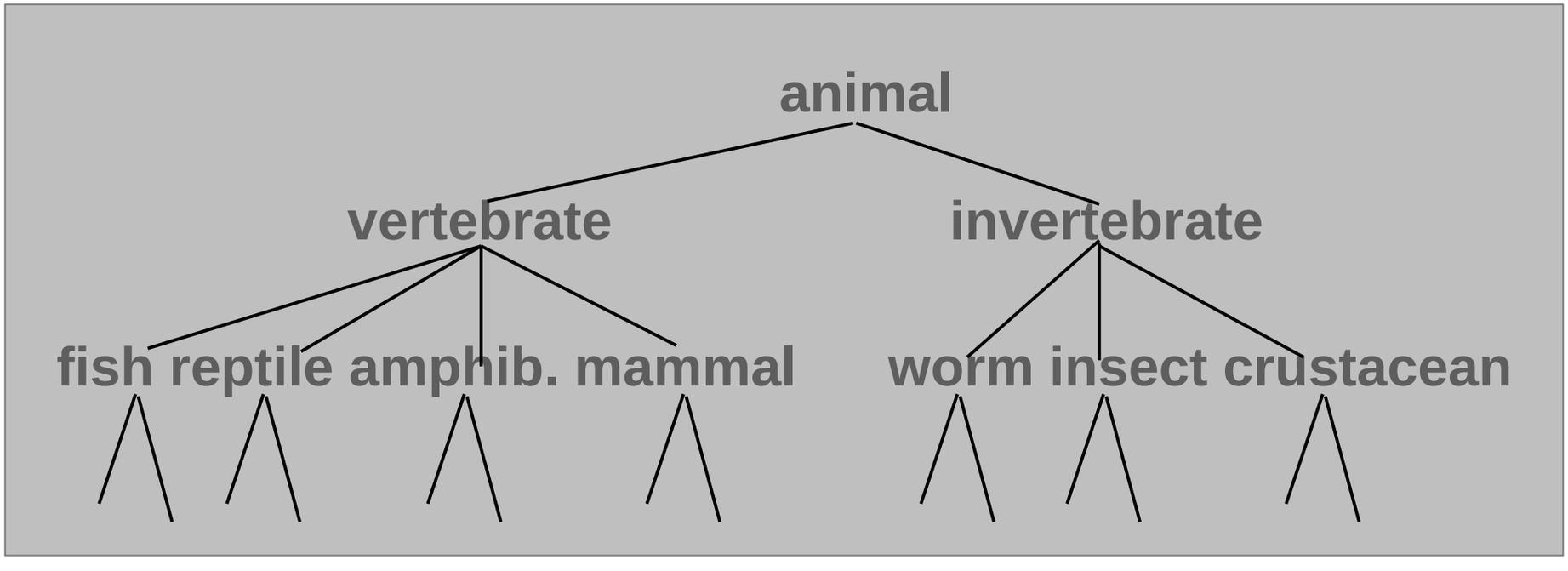


Question: Come up with an idea for an algorithm that produces such a hierarchical clustering!

Agglomerative vs. Divisive Clustering

Agglomerative (*bottom-up*) methods start with each example in its own cluster and iteratively combine them to form larger and larger clusters.

Divisive (*top-down*) separate all examples immediately into clusters.



Hierarchical Agglomerative Clustering (HAC)

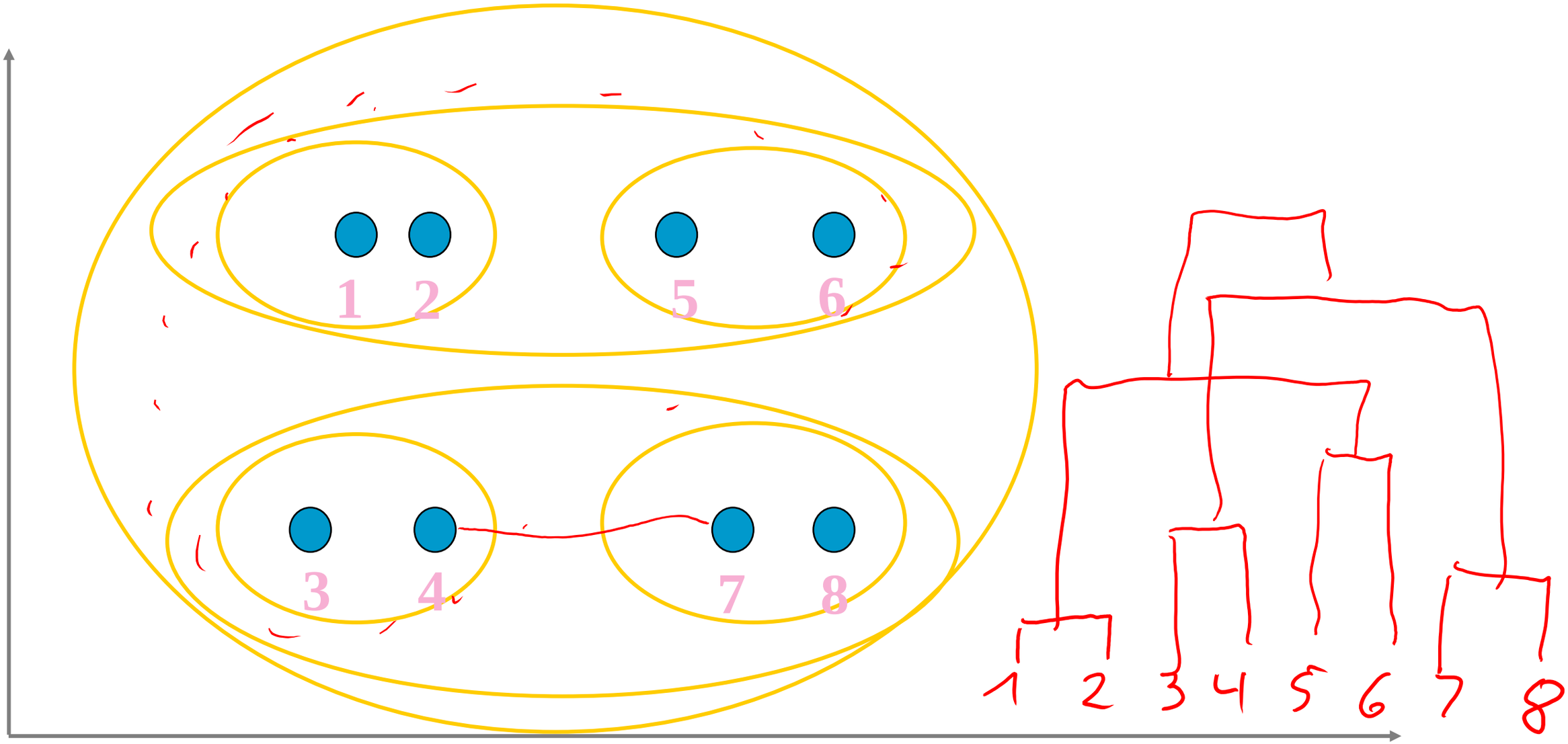
- Assumes a ***distance/similarity function*** for determining the similarity of two clusters.
- Starts with all instances in a separate cluster and then repeatedly joins the two clusters that are closest until there is only one cluster.
- The history of merging forms a binary dendogram or hierarchy.
- Basic algorithm:

  - Start with all instances in their own cluster.
  - Until there is only one cluster:
    - Among the current clusters, determine the two clusters, C_i and C_j, that are closest.
    - Replace C_i and C_j with a single cluster $C_i \cup C_j$

Cluster Distance

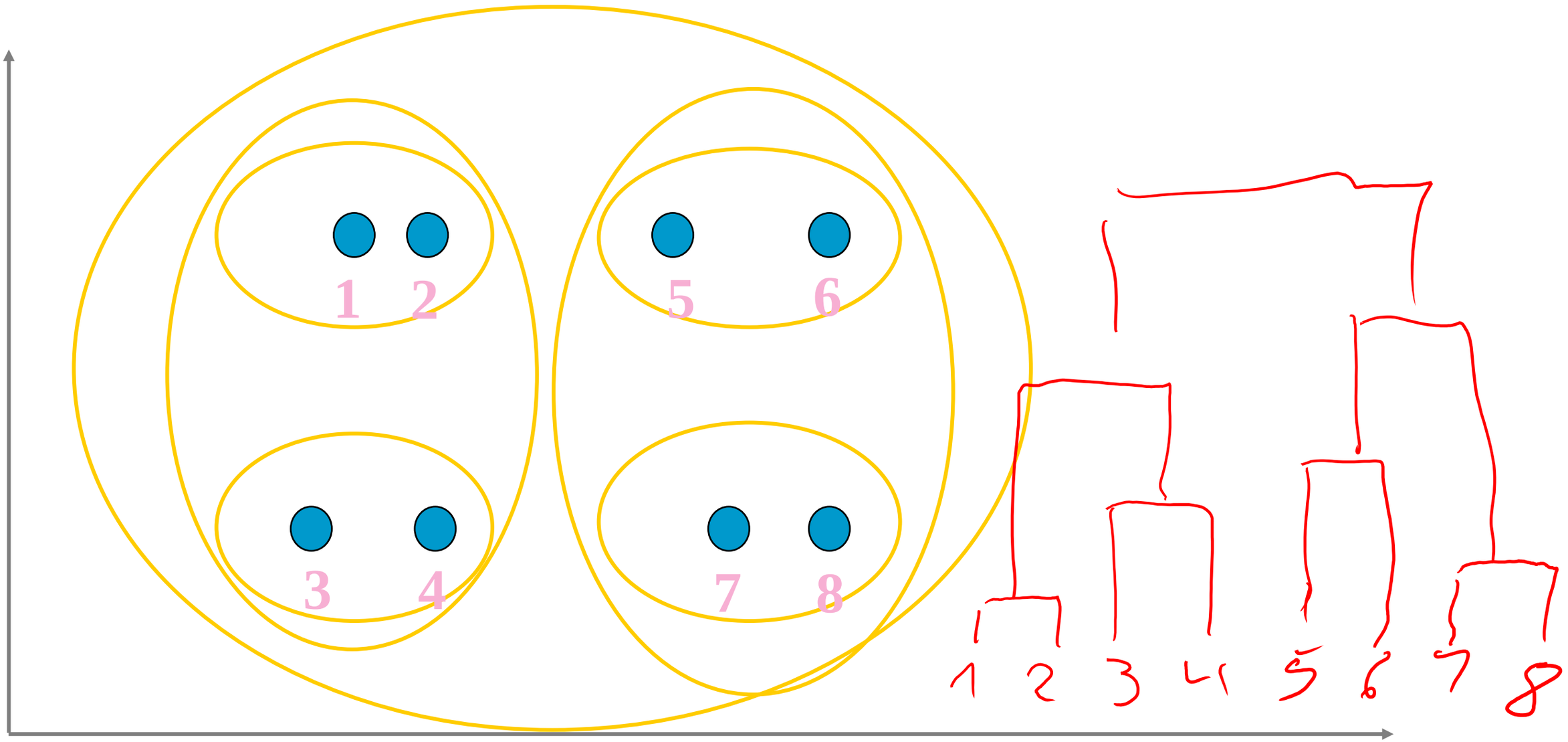
- How to compute distance between two clusters that each possibly contain multiple instances?
  - ***Single link***: Distance of two closest members.
  - ***Complete link***: Distance of two farthest members.
  - ***Average link***: Pairwise average distance between members.

Single-Link HAC



$$dist(C_i, C_j) = \min_{\boldsymbol{x} \in C_i, \boldsymbol{x}' \in C_j} dist(\boldsymbol{x}, \boldsymbol{x}')$$

Complete-Link HAC



$$dist(C_i, C_j) = \max_{\boldsymbol{x} \in C_i, \boldsymbol{x}' \in C_j} dist(\boldsymbol{x}, \boldsymbol{x}')$$

Question

- Can you design a dataset for which single-link is ideal?

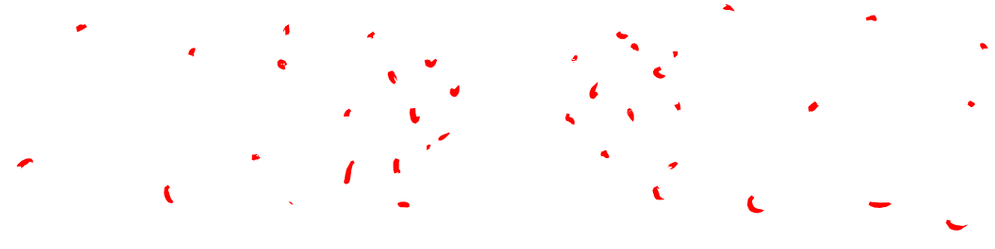
$$dist(C_i, C_j) = \min_{\boldsymbol{x} \in C_i, \boldsymbol{x}' \in C_j} dist(\boldsymbol{x}, \boldsymbol{x}')$$

→ **Forms "straggly" (long and thin) clusters due to chaining effect.**

- Can you design a dataset for which complete-link is ideal?

$$dist(C_i, C_j) = \max_{\boldsymbol{x} \in C_i, \boldsymbol{x}' \in C_j} dist(\boldsymbol{x}, \boldsymbol{x}')$$

→ **Results in more compact and round clusters.**



What is the time complexity of HAC?

Computational Complexity of HAC

- In the first iteration, all HAC methods need to compute similarity of all pairs of *n* individual instances which is $O(n^2)$.

- In each of the subsequent O(*n*) merging iterations, must find smallest distance pair of clusters → Maintain heap $O(n^2 \log n)$

- In each of the subsequent O(*n*) merging iterations, it must compute the distance between the most recently created cluster and all other existing clusters. Can this be done in linear time such that $O(n^2 \log n)$ overall?

Computing Cluster Distances

- After merging c_i and c_j, the distance of the resulting cluster to any other cluster, c_k, can be computed by:

- Single Link:

$$dist(C_i \cup C_j, C_k) = min\left(dist(C_i, C_k), dist(C_j, C_k)\right)$$

- Complete Link:

$$dist(C_i \cup C_j, C_k) = max\left(dist(C_i, C_k), dist(C_j, C_k)\right)$$

Single-Link Example

	x1	x2	x3	x4	x5
x1	0	0.2	0.8	0.3	0.7
x2	0.2	0	0.9	0.5	0.8
x3	0.8	0.9	0	0.1	0.5
x4	0.3	0.5	0.1	0	0.6
x5	0.7	0.8	0.5	0.6	0

Merge x3,x4
replace with min

	x1	x2	C1	x5
x1	0	0.2	0.3	0.7
x2	0.2	0	0.5	0.8
C1	0.3	0.5	0	0.5
x5	0.7	0.8	0.5	0

Merge x1,x2
replace with min

	C2	C1	x5
C2	0	0.3	0.7
C1	0.3	0	0.5
x5	0.7	0.5	0

Merge c1,c2
replace with min

	C3	x5
c3	0	0.5
x5	0.5	0

$$\boldsymbol{Total\ Time\ Complexity}\colon\ \boldsymbol{O}\left(n^2 \log n\right)$$

How do we formalize the clustering objective?

- Given two clusterings $C_1, \dots, C_K$ and $C'_1, \dots, C'_K$, which one is better?

- Informally so far:
  - Items in the same cluster should be similar
  - Items in different clusters should be dissimilar/distant

→ Numerical objective to evaluate clusterings!

Sum of Distances

- Items in the same cluster should be similar
  - Minimize total within-cluster dissimilarity/distance

$$M_1 = \sum_{j=1}^{K} \sum_{s,t \in C_j} dissimilarity(\boldsymbol{x}_s, \boldsymbol{x}_t)$$

- Items in different clusters should be dissimilar
  - Maximize total between-cluster dissimilarity/distance

$$M_2 = \sum_{\boldsymbol{x}_s, \boldsymbol{x}_t : c(\boldsymbol{x}_s) \neq c(\boldsymbol{x}_t)} dissimilarity(\boldsymbol{x}_s, \boldsymbol{x}_t)$$

Question: Is optimizing M_1 equivalent to optimizing M_2?

$M_1 + M_2 = \sum_{s,t : s \neq t} dissimilarity(x_s, x_t) - M_2$

constant

→ Weighted MaxCut Problem for K=2

What does Single-Link HAC optimize?

- Single-link HAC finds the clustering $C_1, \ldots, C_K$ that maximizes the following objective:

$$M_3 = \min_{\boldsymbol{x}_s, \boldsymbol{x}_t : c(\boldsymbol{x}_s) \neq c(\boldsymbol{x}_t)} dissimilarity(\boldsymbol{x}_s, \boldsymbol{x}_t)$$

[Maximize smallest between-cluster distance]

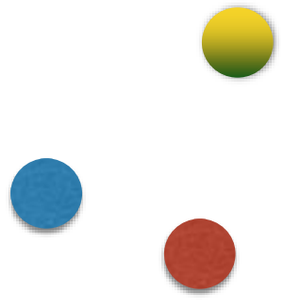
Observation

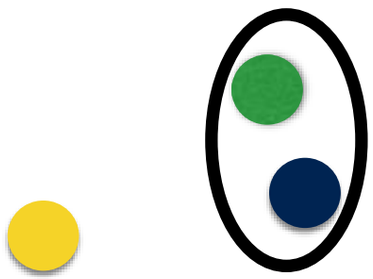
Say c is the clustering produced by single-link HAC.

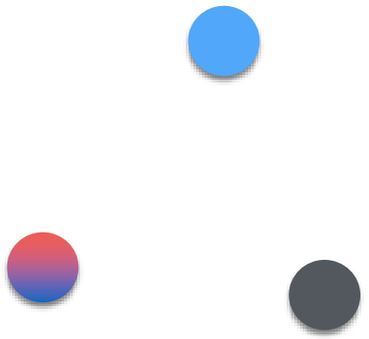
Then it always holds

$$\min_{\boldsymbol{x}_s, \boldsymbol{x}_t : c(\boldsymbol{x}_s) \neq c(\boldsymbol{x}_t)} dissimilarity(\boldsymbol{x}_s, \boldsymbol{x}_t) > \text{merged distances in tree}$$

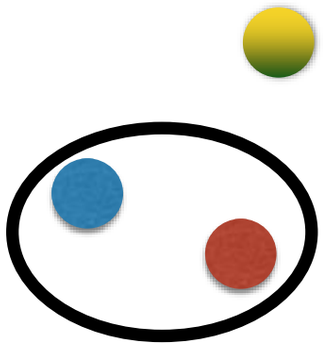
Demo

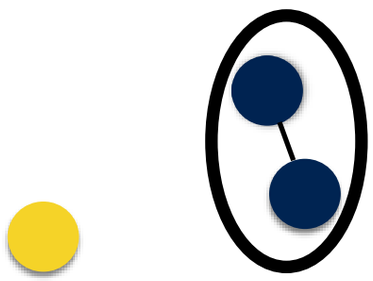


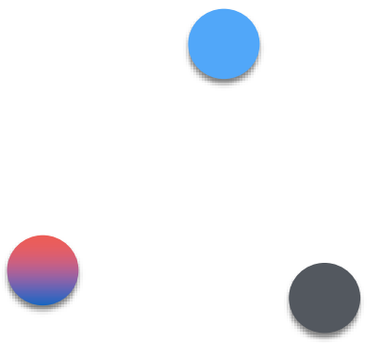




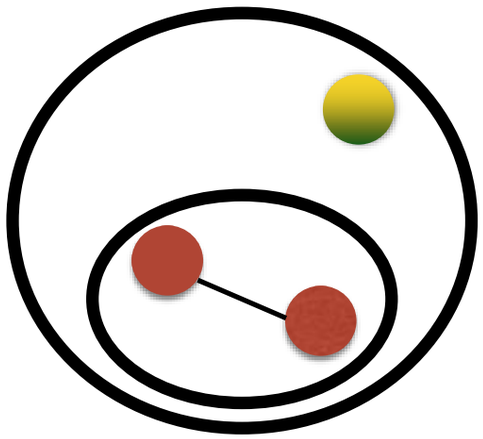
Demo

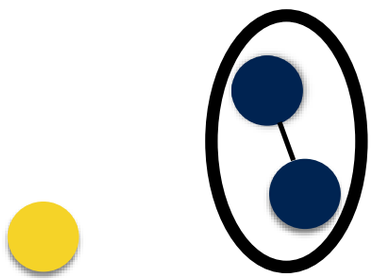


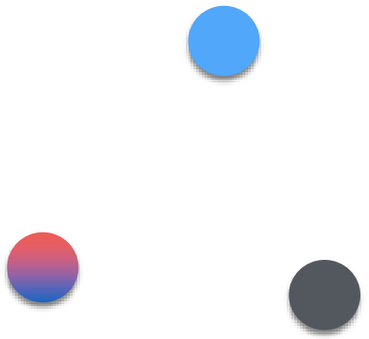




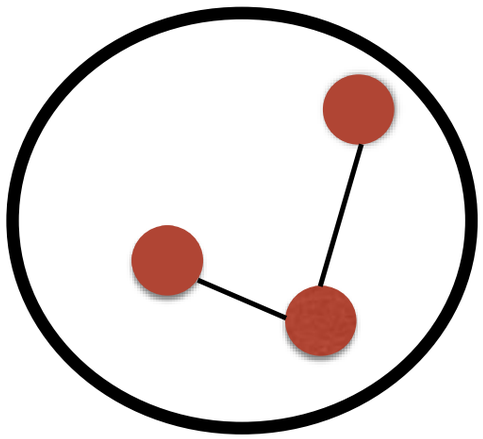
Demo

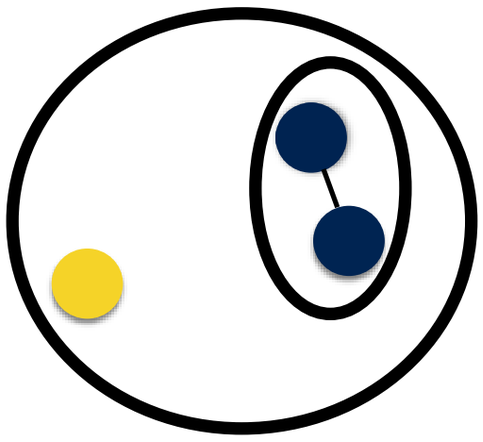


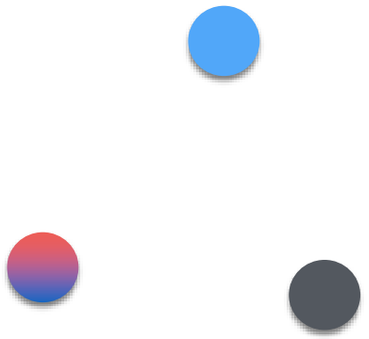




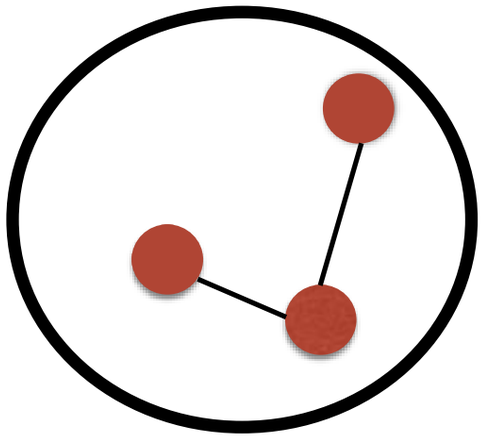
Demo

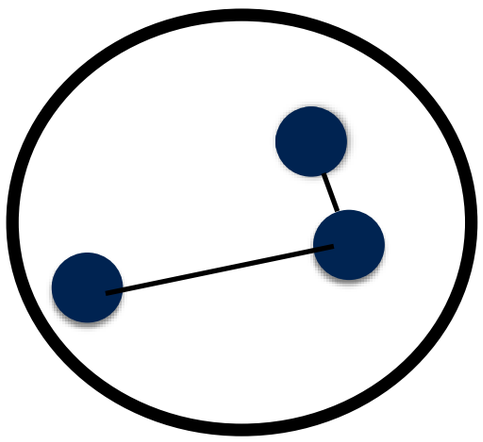


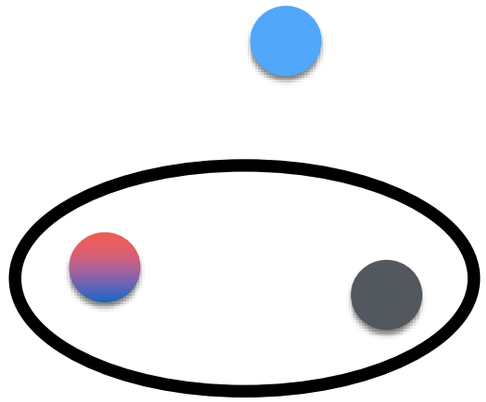




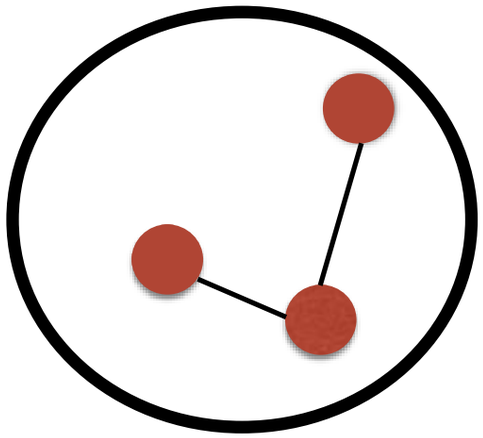
Demo

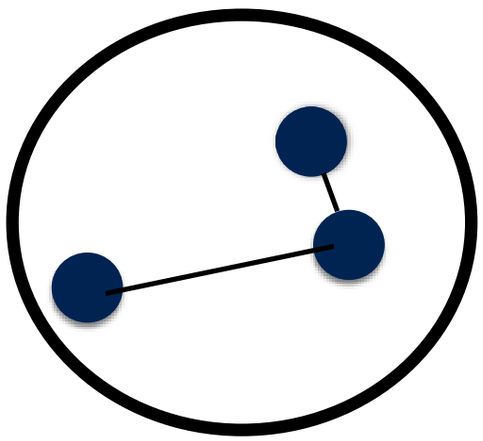


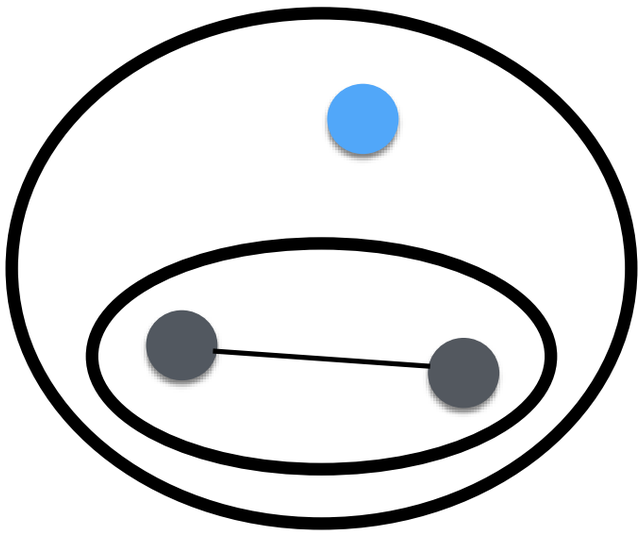




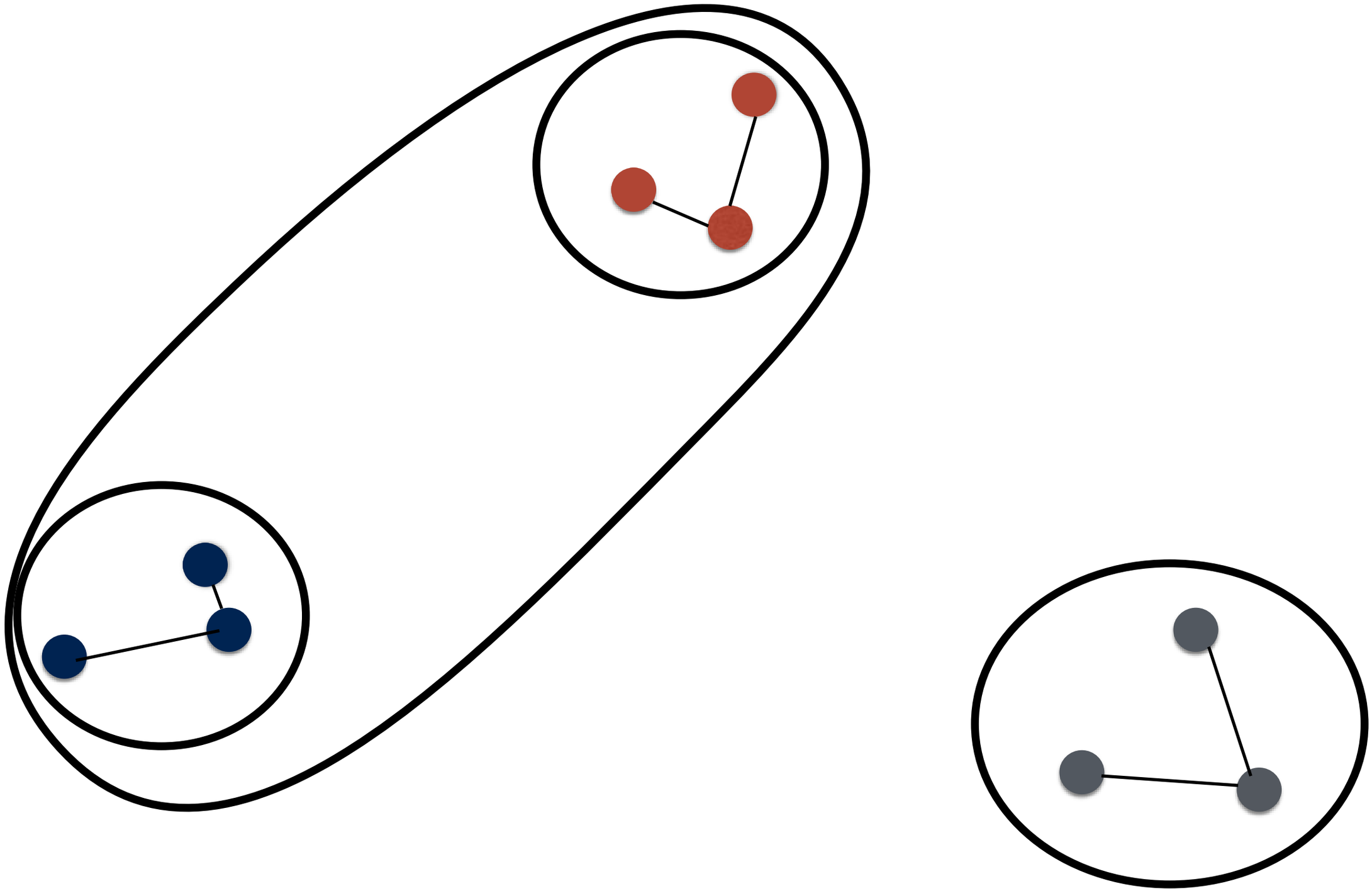
Demo



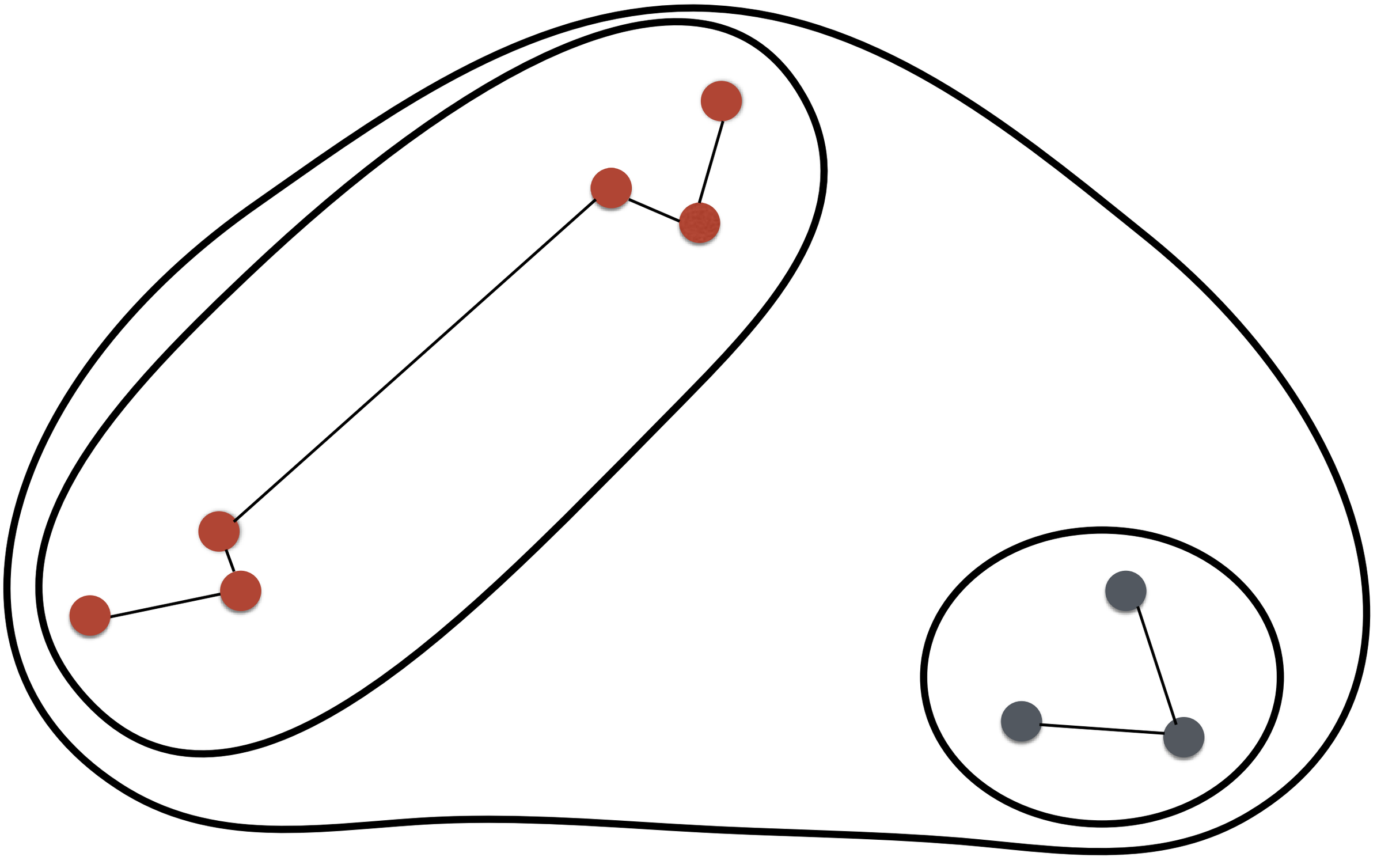




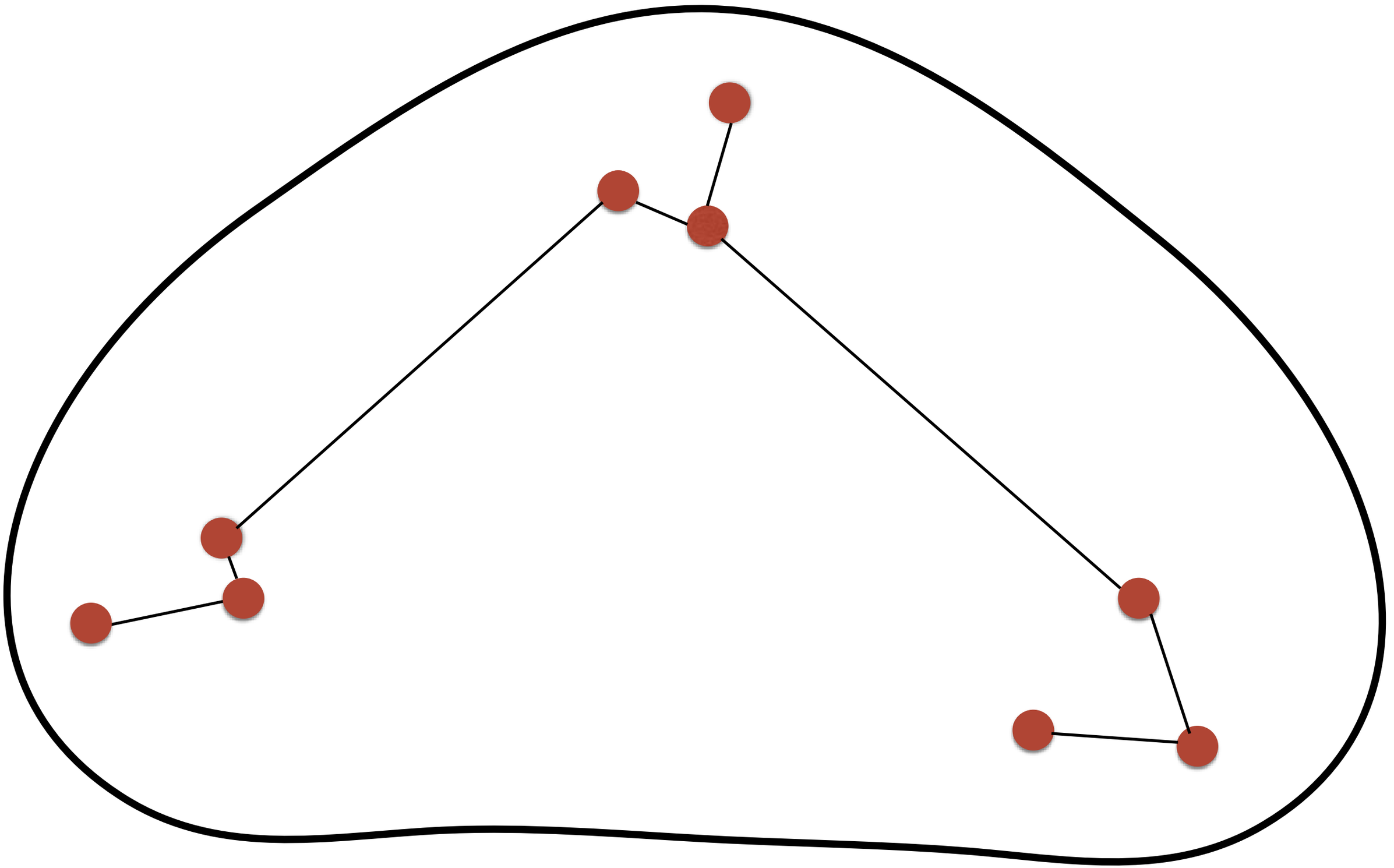
Demo



Demo



Demo



Single Link Objective

Proof:

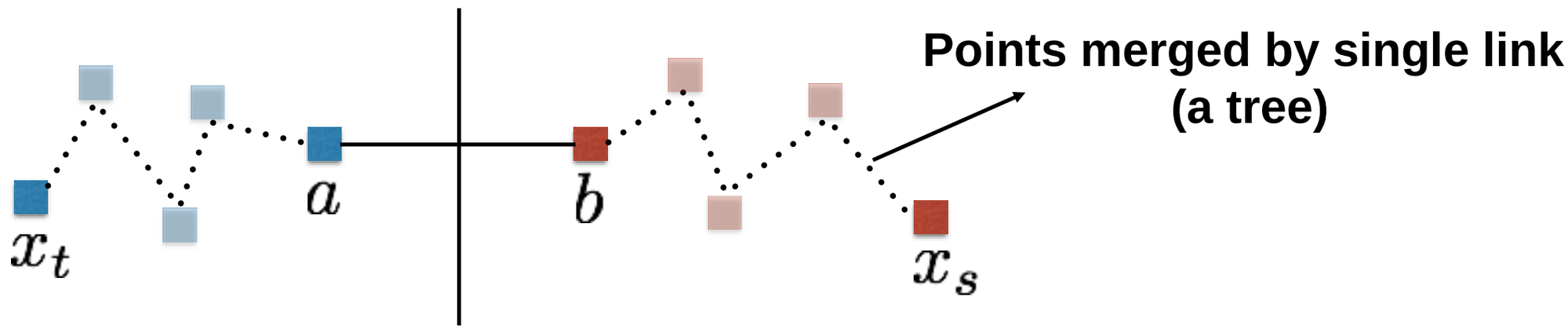
Say c is solution produced by single-link clustering

Key observation:

$$\min_{t,s:c(x_i)\neq c(x_j)} \text{dissimilarity}(x_i, x_j) >$$ Merged distances in tree

Say $c' \neq c$ then,

$$\exists\ t, s \text{ s.t. } c'(x_t) \neq c'(x_s) \text{ but } c(x_t) = c(x_s)$$



c' boundary

Summary

- Clustering: Find partitioning of the data points.
- HAC: Repeatedly merge clusters bottom up.
  - Design: Distance measure and merge criterion.
  - Efficiency: $O(n^2 \log n)$
- Single-link optimizes M_3 criterion.