Clustering

Diplomatura en Ciencia de Datos,
Aprendizaje Automático y sus Aplicaciones
FaMAF-UNC
agosto 2018

Mapa de ruta

1. Embeddings
2. **Clustering**, y revisitar todos los conceptos que vimos hasta ahora
3. Reglas de Asociación
4. K-nn y recomendación
5. Grafos
6. Aprendizaje Semi-supervisado

Entregables:

- Clustering
- Recomendación

Mapa de ruta

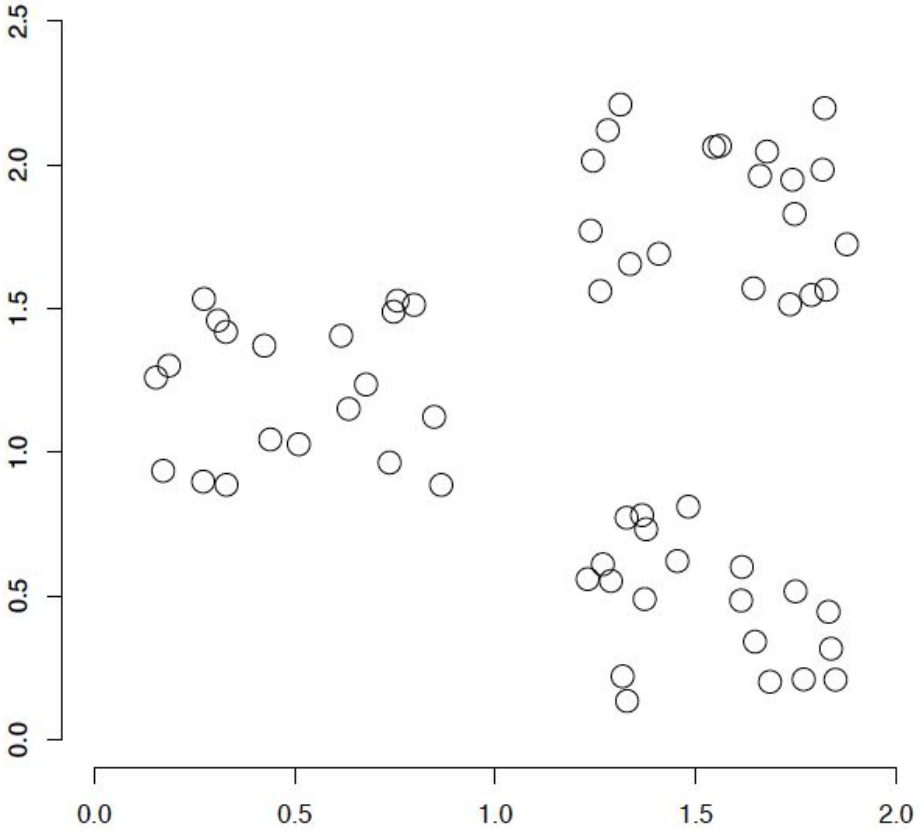
1. Cómo funciona
2. Evaluación
3. Qué puedo esperar
4. Metodología iterativa
5. Ejemplos con notebooks

Cómo funciona clustering

Agrupar objetos semejantes

- Entrada: objetos en un espacio n-dimensional
- Salida: una **solución** con grupos (**clusters**) de objetos semejantes → cercanos en el espacio
  - Se minimiza la distancia entre los objetos de un mismo grupo
  - Se maximiza la distancia entre los objetos de distintos clusters
- Los centros de cada cluster son los **centroides**

Dataset con clara estructura de clusters



¿Cómo sería un algoritmo para encontrar clusters en este espacio?

Cuestiones cruciales

- ¿Cómo es el espacio? ¿Cómo represento mis problemas?
- ¿Cómo se calcula la distancia (semejanza) en este espacio?
- ¿Cuántos clusters quiero distinguir?
- ¿Qué distribución tienen estos clusters? ¿Gaussiana? ¿En serie?
- ¿Busco una estructura jerárquica o plana?
- ¿Cómo veo qué hay en cada cluster?
- ¿Cómo evalúo la bondad de cada solución?

Semejanza (Distancia)

- La semejanza debería acercarse a las causas latentes
  - Entre documentos: semántica
  - Entre clientes: motivación para las compras
  - Entre imágenes: objetos físicos que representan
  - Entre propiedades inmobiliarias: elementos que otorgan valor
- Idealmente, debería calcularse de forma independiente para cada dimensión

Distancias (semejanzas)

- Euclídea
- Coseno ➜ normalizado por longitud, producto punto ➜ correlación!
- Distancia de Manhattan
- Distancia de Edición (Levenshtein)

- Divergencia de Kullback-Leibler

Hard clustering vs. Soft clustering

- Hard clustering: cada objeto pertenece a un cluster
- Soft clustering: cada objeto tiene una probabilidad de pertenecer a uno o más clusters ➜ LDA

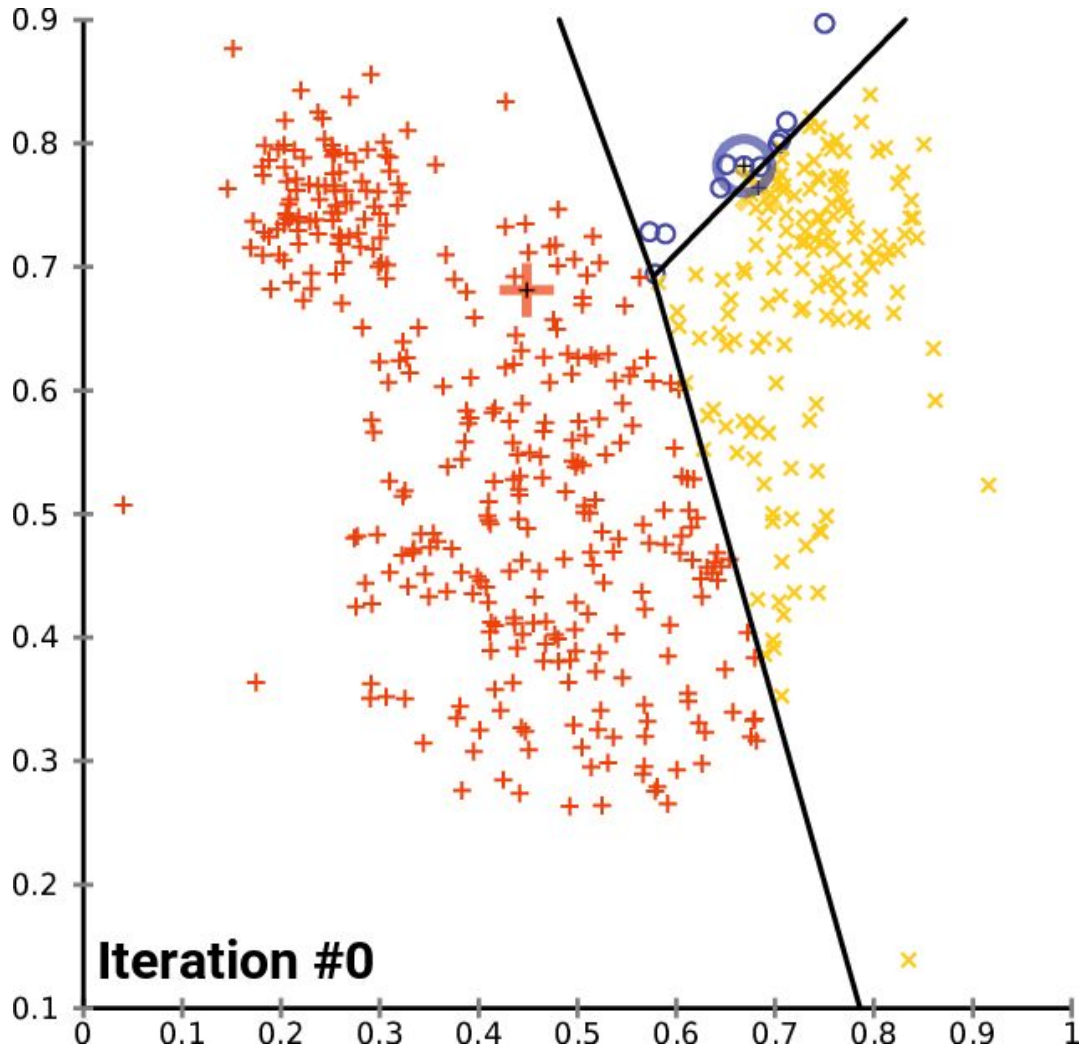
Plano vs. Jerárquico

- Algoritmos partitivos
  a. Empezar con una partición aleatoria
  b. Refinarla iterativamente
- Algoritmos jerárquicos
  a. Aglomerativos (bottom-up)
  b. Divisivos (top-down)

K-means

- El algoritmo partitivo más popular
- Toma objetos en un espacio y el número k de clusters que deseamos
- Cuando no se especifica el número k de clusters, es Expectation Maximization (muy costoso)

1. Toma unos centros de clase iniciales (en general, aleatorios)
2. Asigna cada objeto al centro que le queda más cercano, creando un cluster
3. Encuentra el centro de los objetos de un cluster
4. Vuelve a 2 hasta que se alcanza un criterio de terminación:
    a. Convergencia
    b. Número máximo de iteraciones


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Iteration #0

K-means

Problemas

- Inestabilidad
- Mínimos locales (mucha sensibilidad a las semillas)
- Soluciones globales ➔ sensibles a outliers
- El número de clusters k suele ser desconocido

Parámetros

- Inicialización
- número de veces que se vuelven a tirar las semillas
- cuántas iteraciones hasta que termina la búsqueda

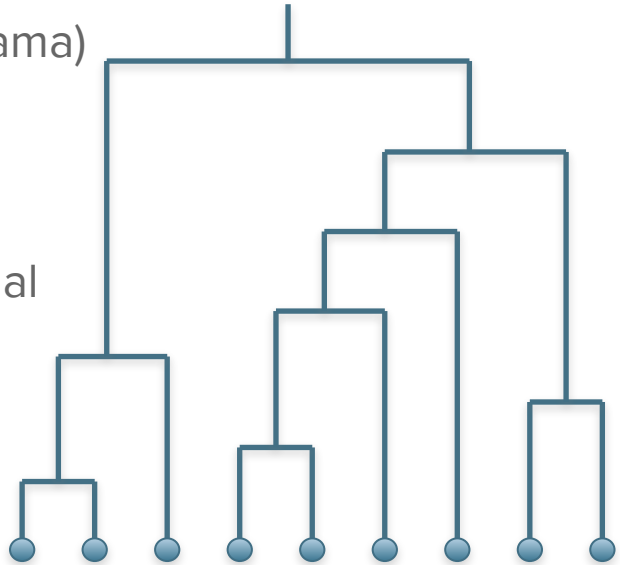
Clustering jerárquico

Si no queremos especificar k...

Algoritmos jerárquicos que generan una

taxonomía jerárquica de clusters (dendrograma)

- Interpretación más rica
- Más difícil de interpretar
- El corte del árbol tiene que ser ortogonal



Clustering jerárquico aglomerativo

Bottom-up

- Cada objeto es su propio cluster
- Se unen en un solo cluster el par de clusters más semejantes
- La historia de uniones forma un árbol binario (jerarquía)

Semejanza entre clusters

Single-link

1. Para cada par de clusters A y B, el par de objetos a, b más cercanos tal que a pertenece a A y b pertenece a B
2. Se unen los clusters con el par de objetos más semejante

Complete-link

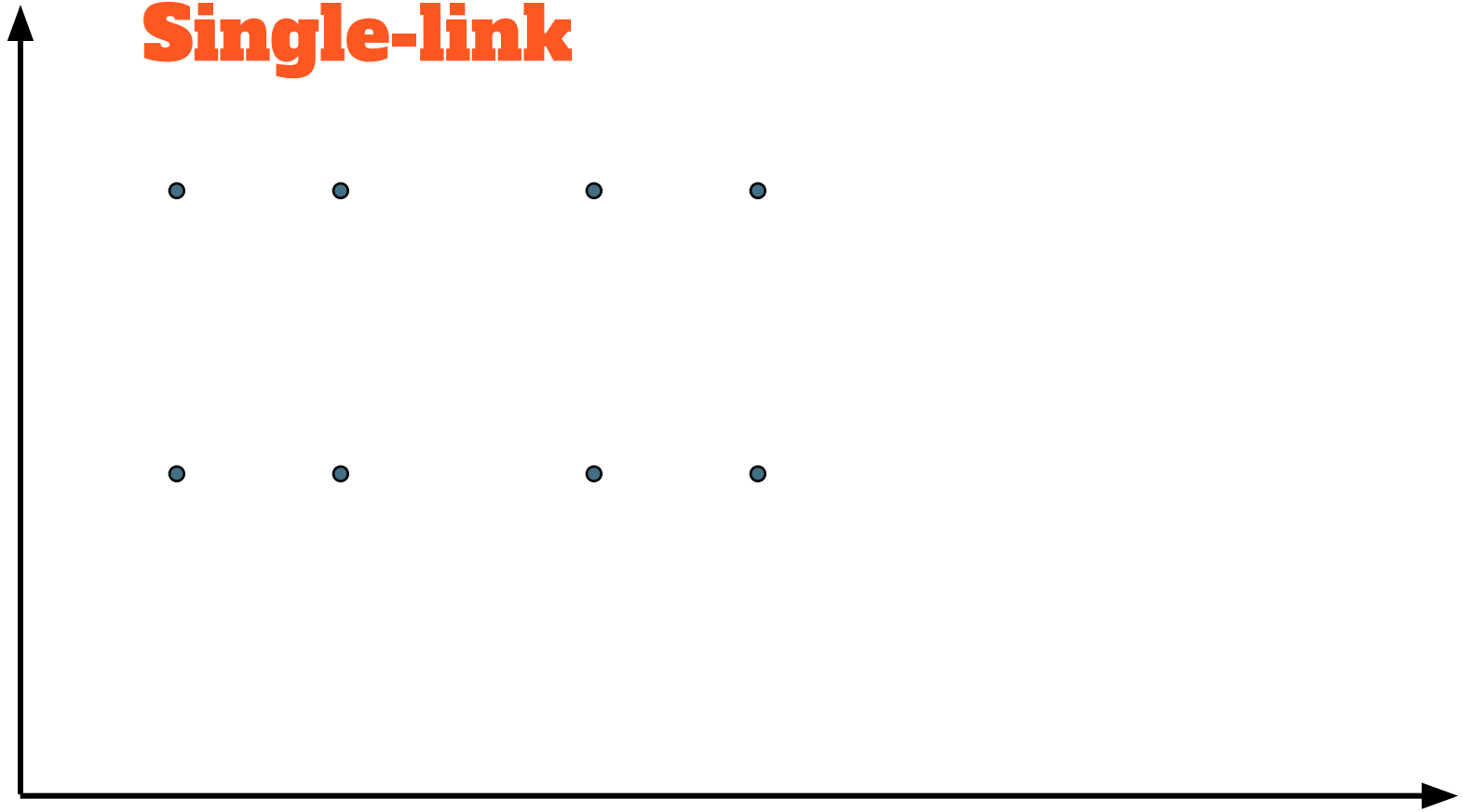
1. Para cada par de clusters A y B, el par de objetos a, b más distantes tal que a pertenece a A y b pertenece a B
2. Se unen los clusters con el par de objetos más semejante

Average-link

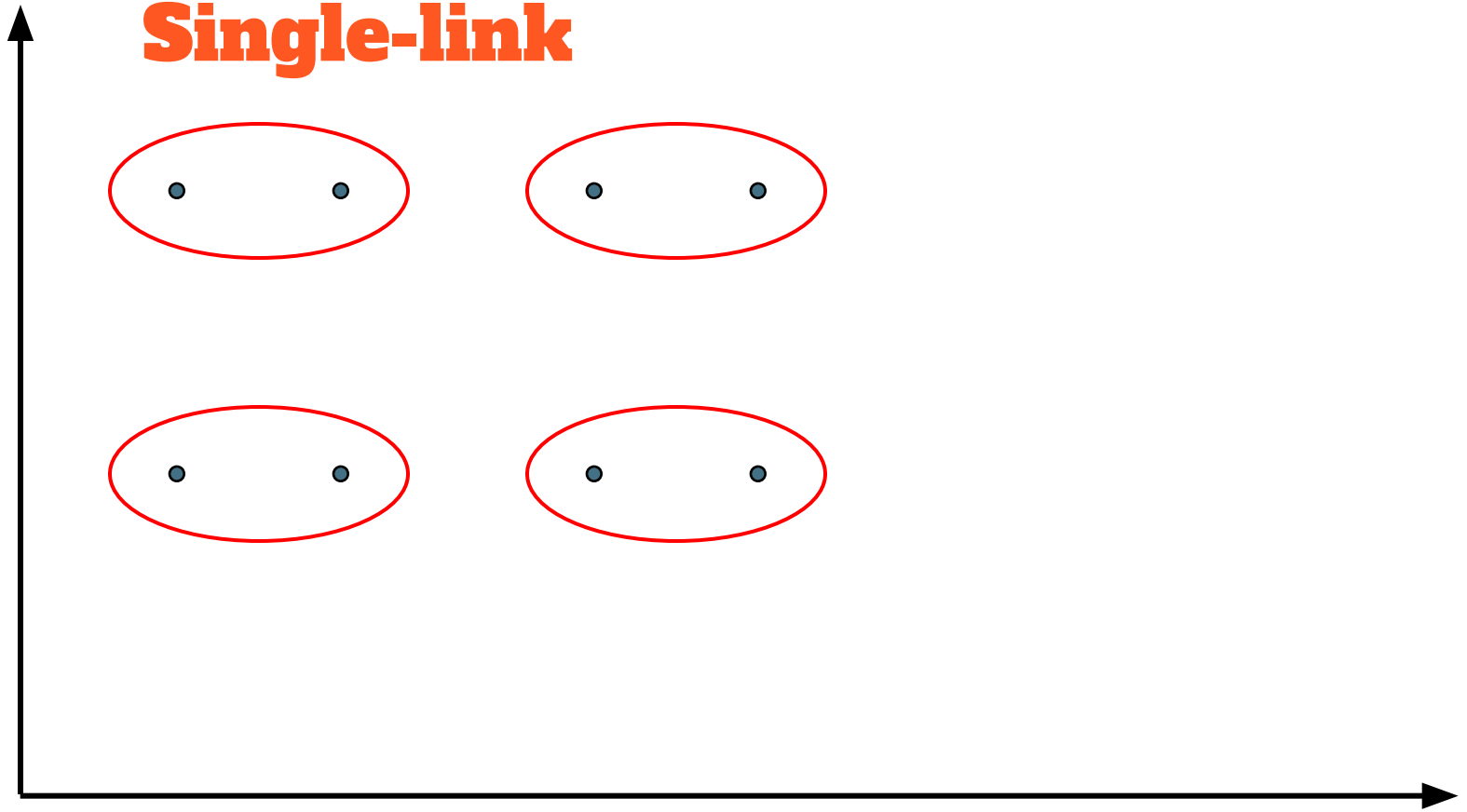
1. Para cada par de clusters A y B, se calcula la distancia entre todo par de objetos a, b tal que a pertenece a A y b pertenece a B
2. Se unen los clusters con el promedio de distancia más bajo

Centroid: Se unen los clusters con los centroides más cercanos

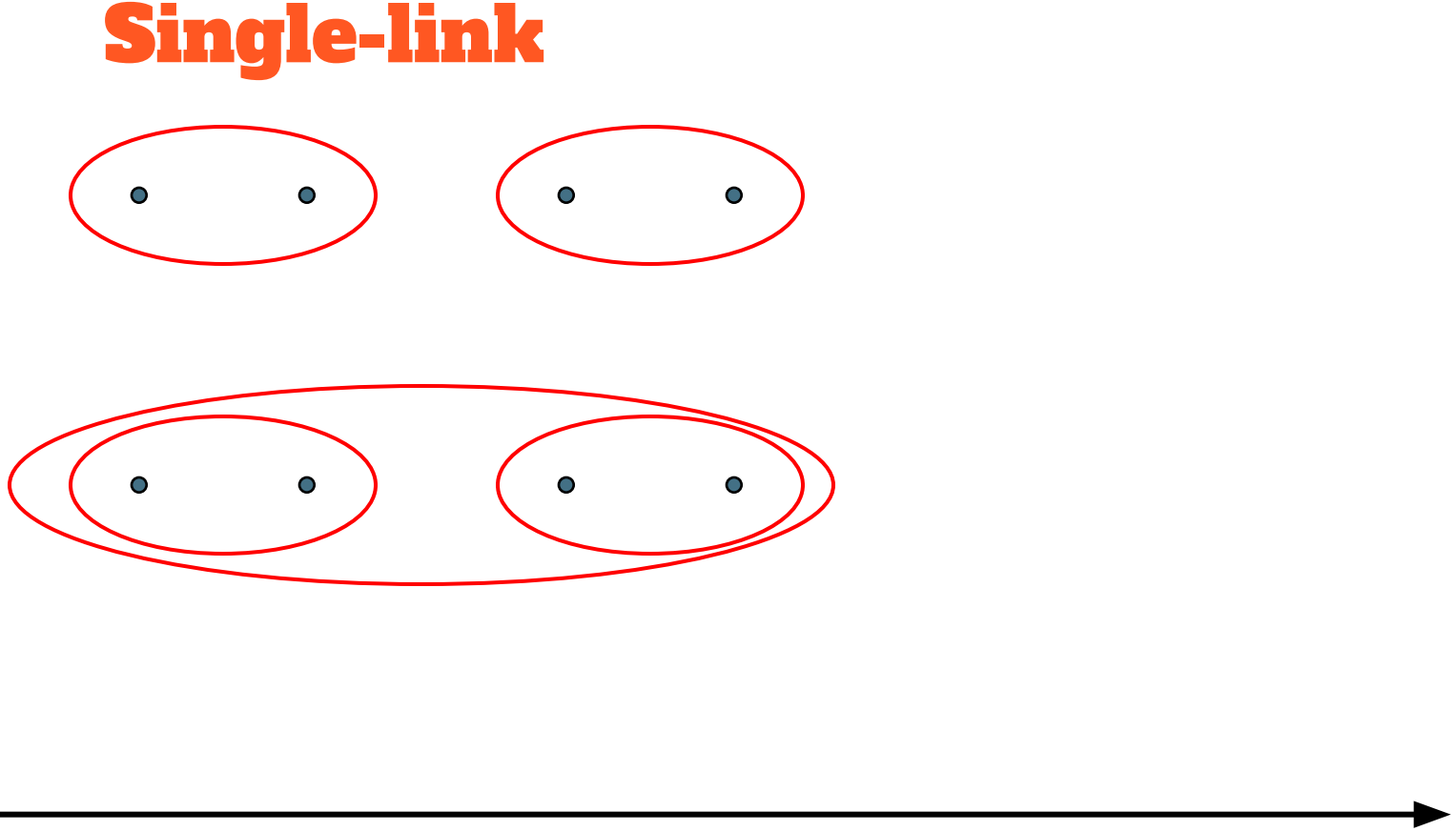
Single-link



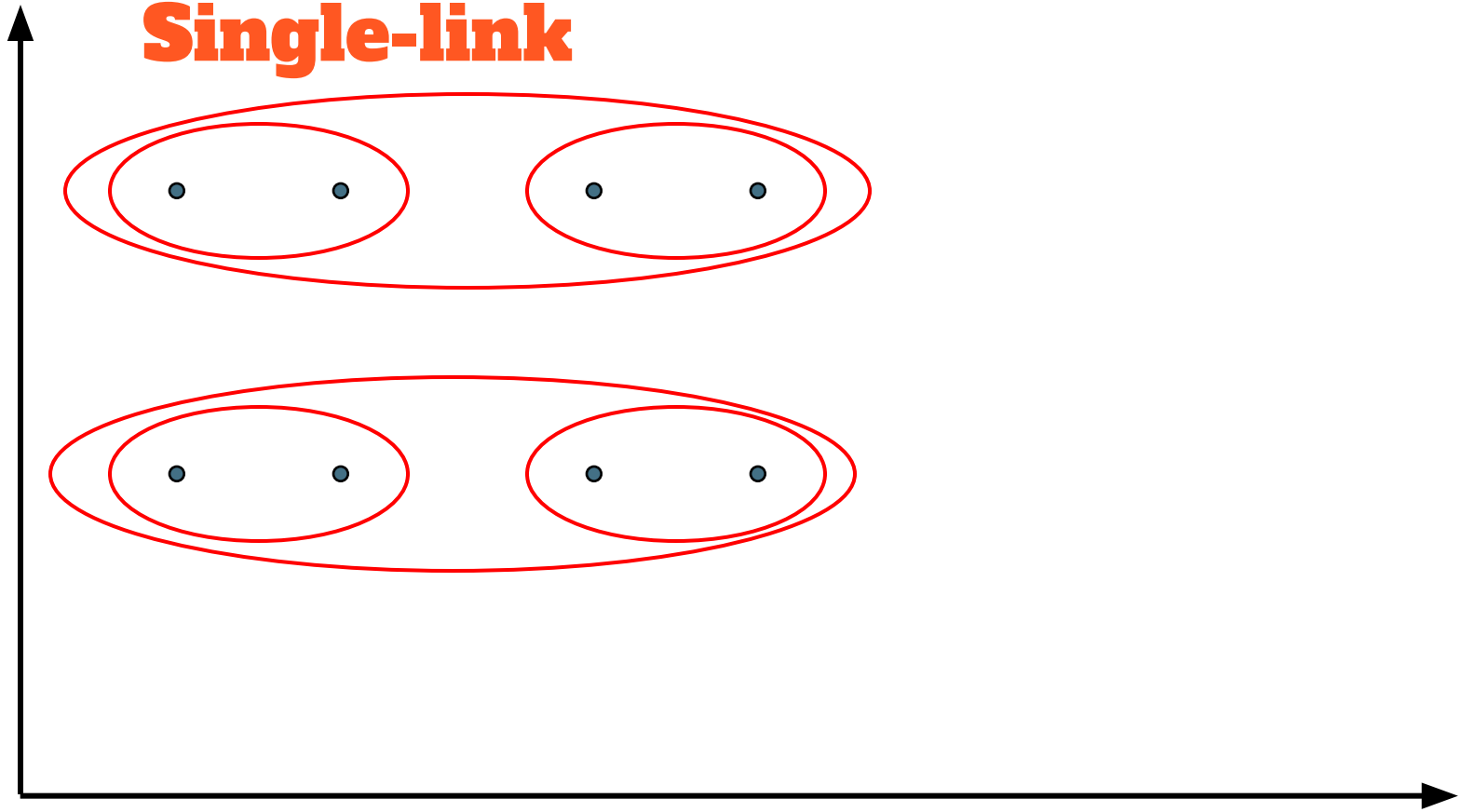
Single-link

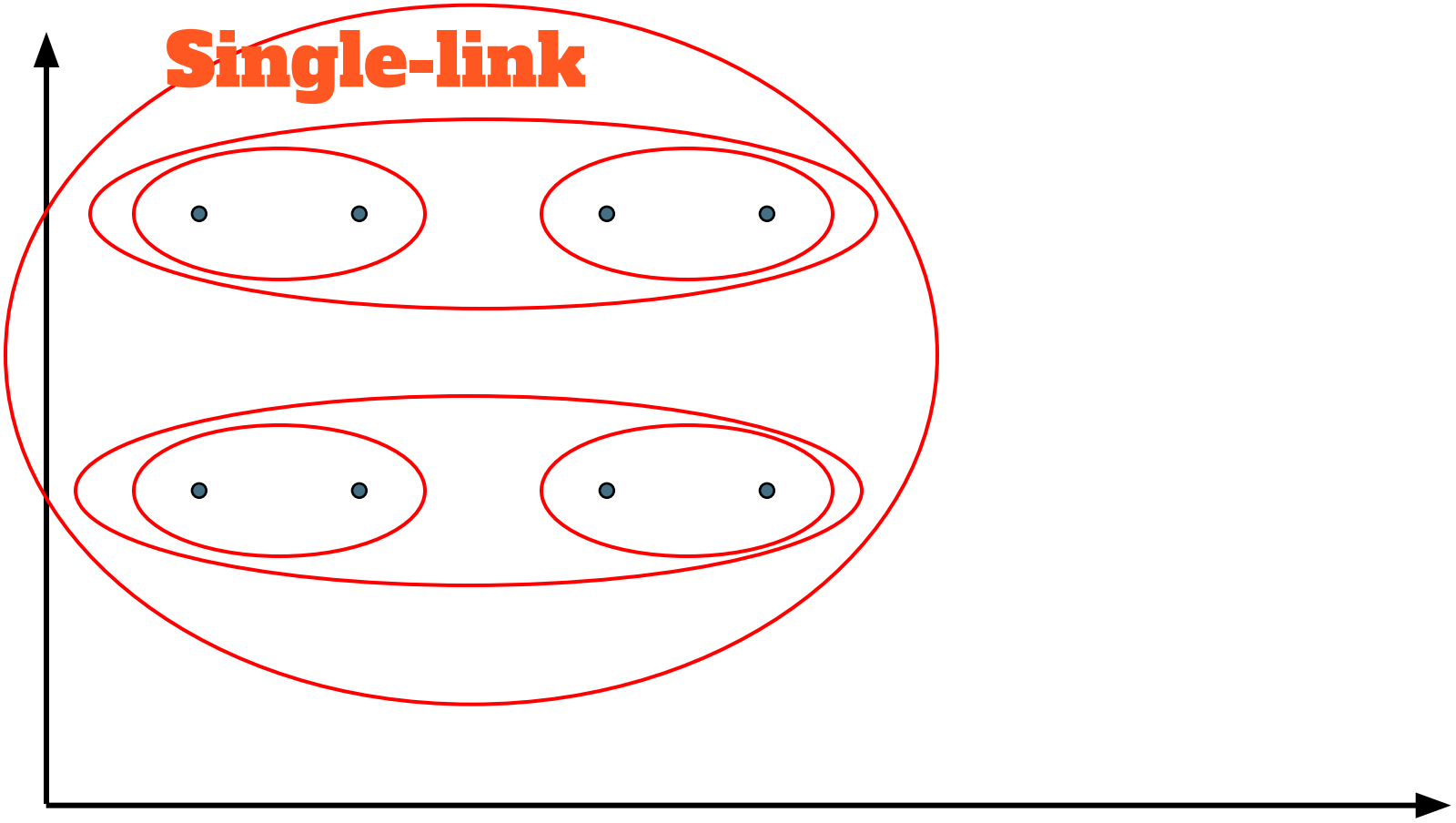


Single-link

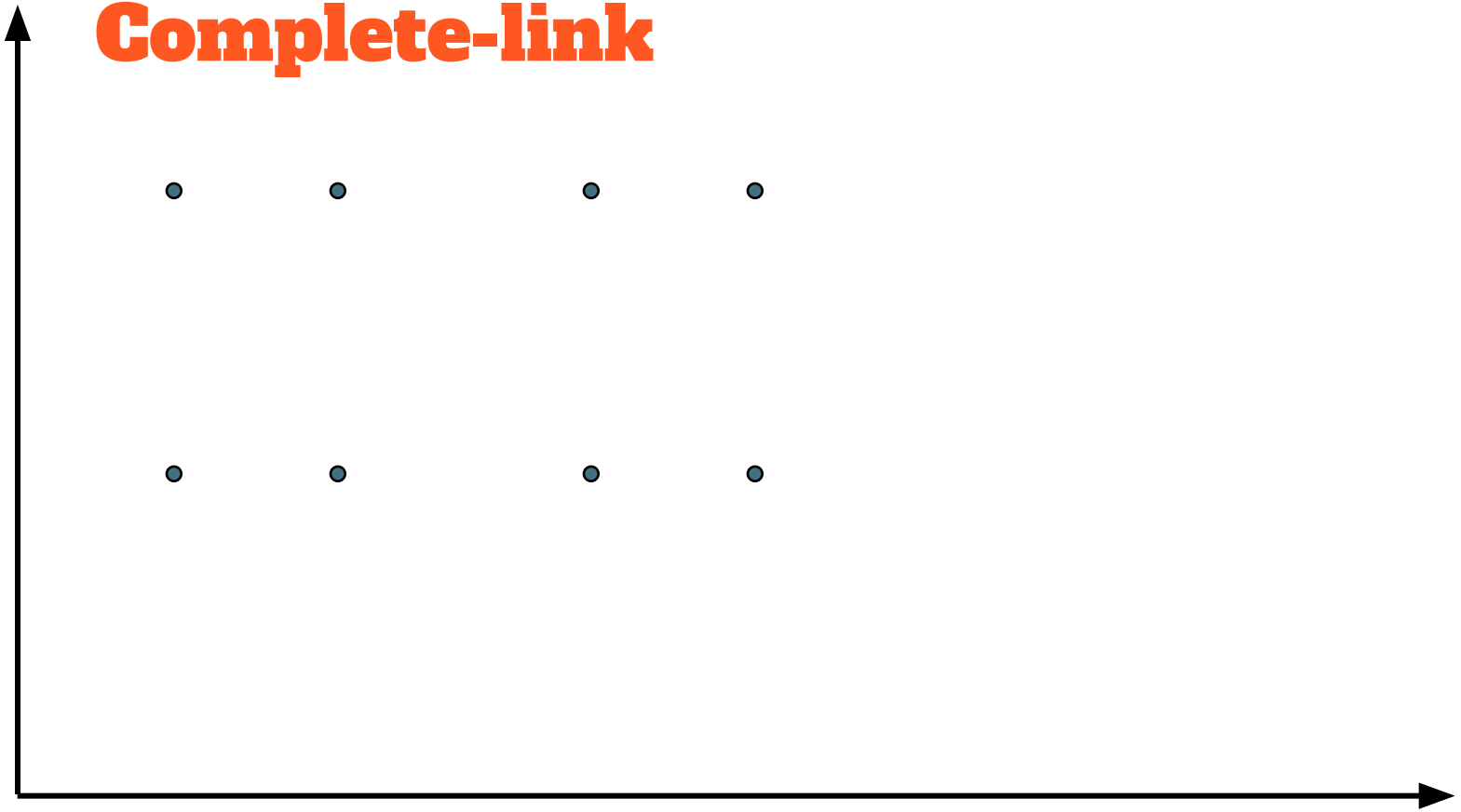


Single-link

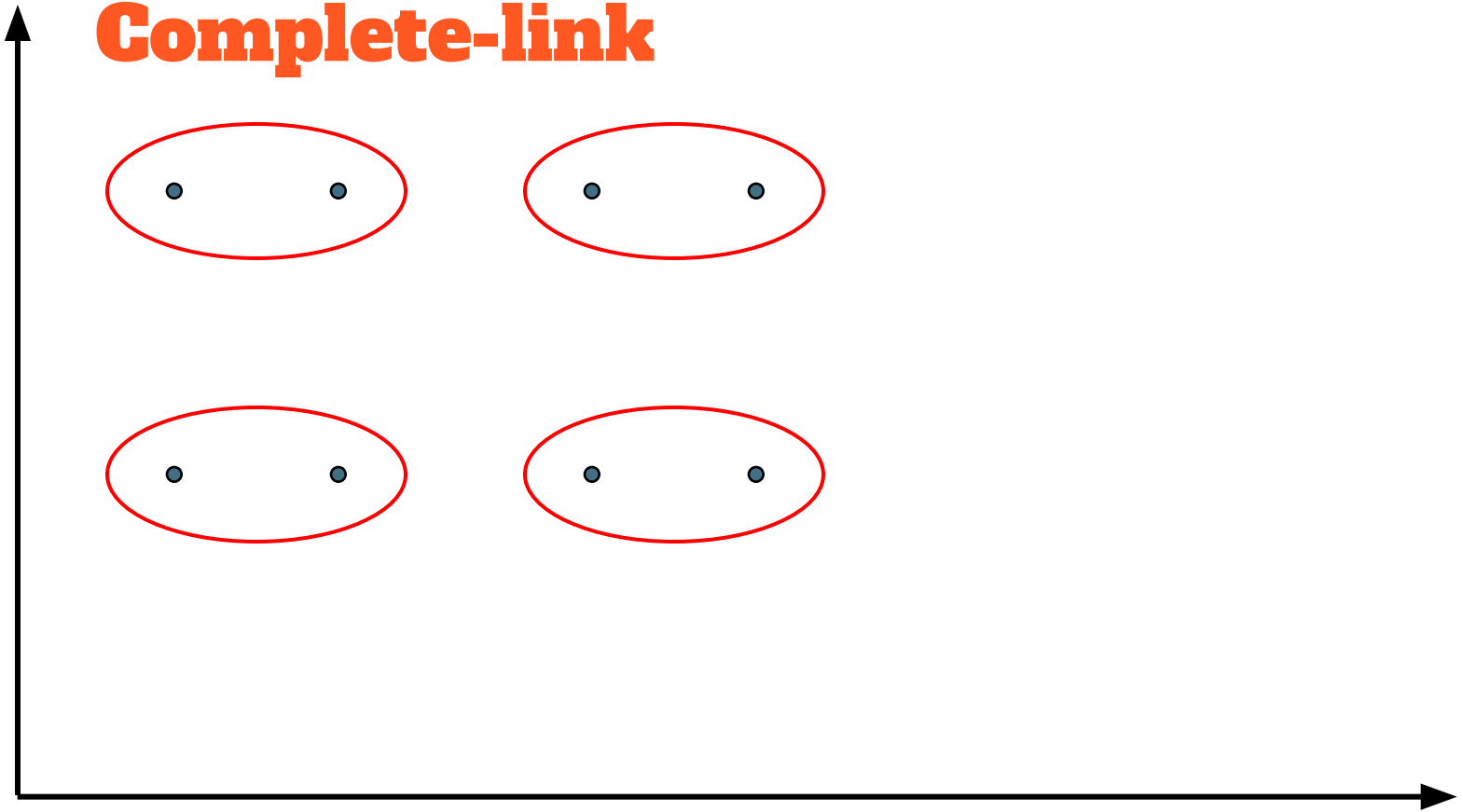



Single-link

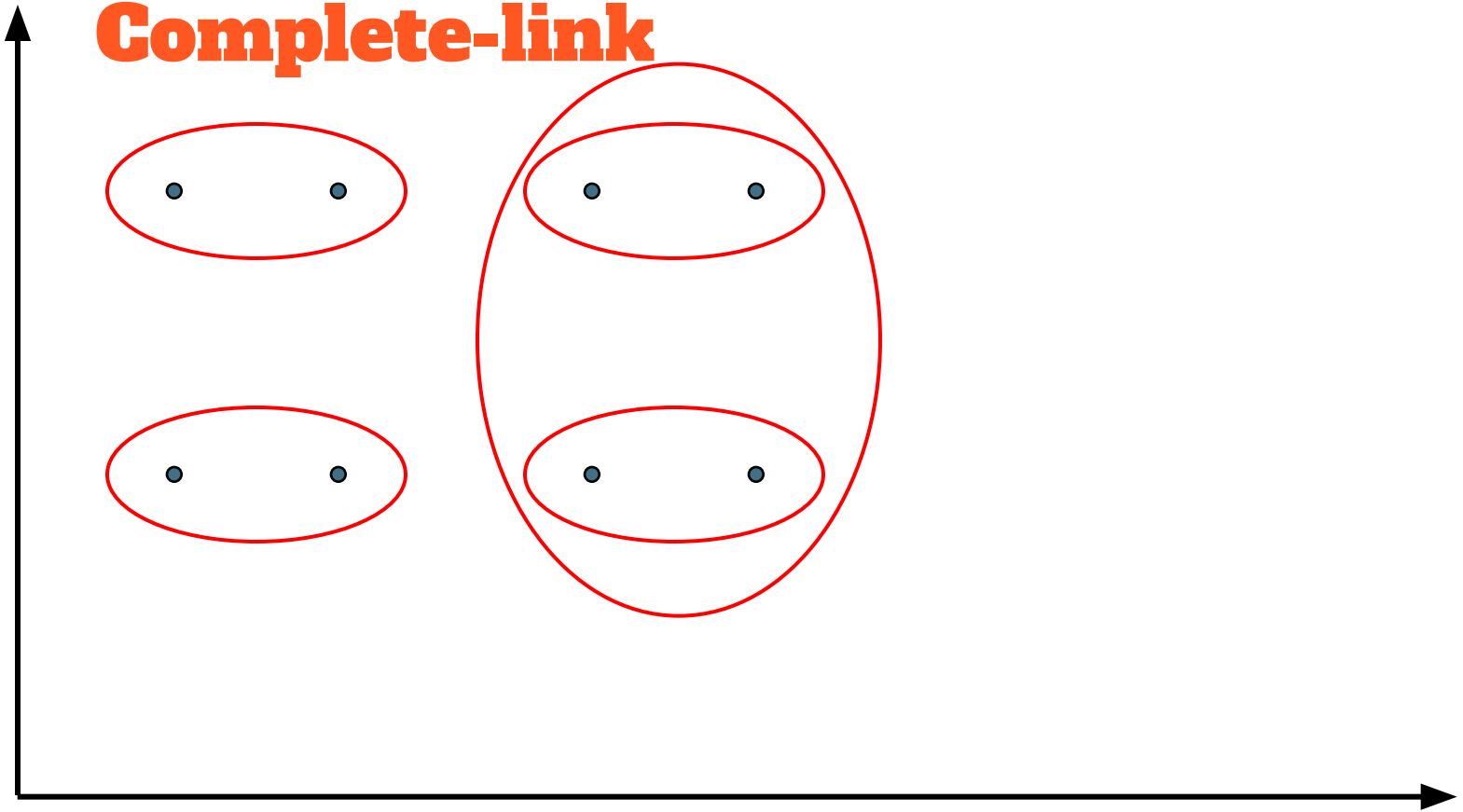
Complete-link



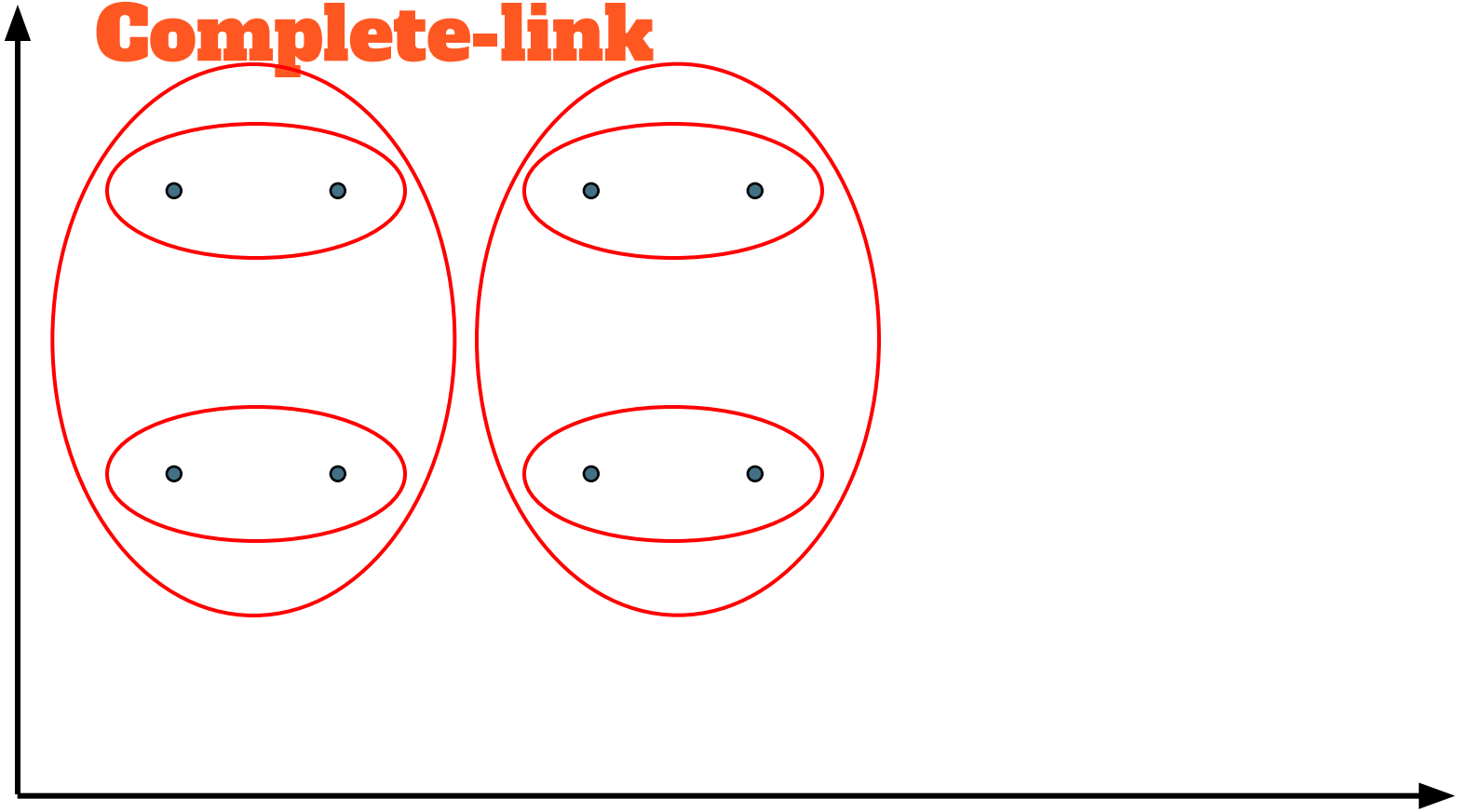
Complete-link

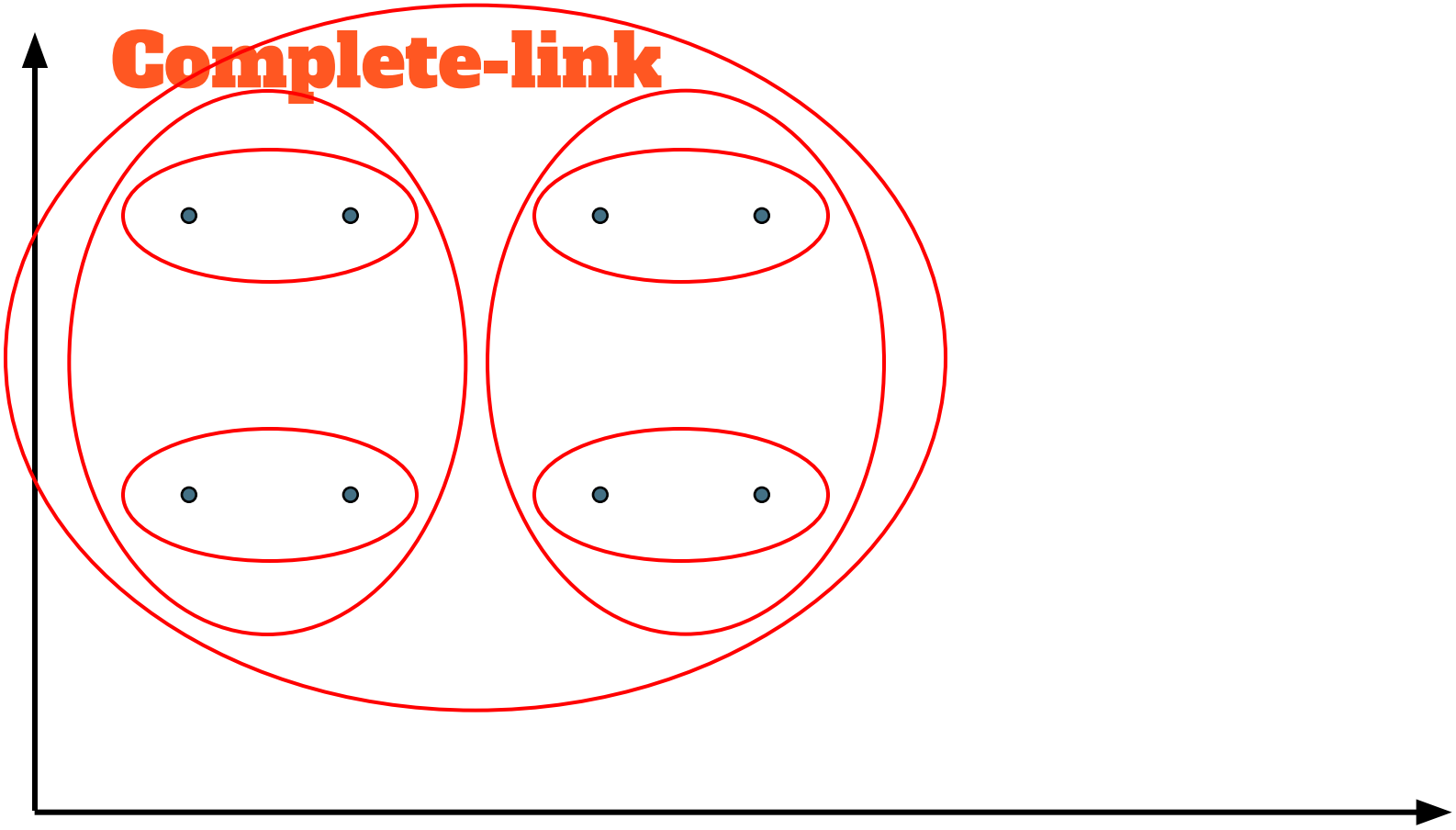


Complete-link



Complete-link




Complete-link

Clustering jerárquico partitivo

Top-down

Aplicar recursivamente un algoritmo de clustering partitivo

- Rápido
- Fácil de implementar
- Fácil de manipular
- Voraz (mínimos locales)

LDA (Latent Dirichlet Allocation)

- Cada objeto tiene una probabilidad de pertenecer a un cluster
- Cada cluster es una distribución sobre características
- Cada característica tiene una probabilidad de haber sido generada en un cluster

Clusters = causas latentes (p.ej., temas de documentos)

Características = fenómenos observables (p.ej., palabras de documentos)

Es un modelo generativo, especialmente adecuado para modelar resultados de procesos que entendemos como generativos

Evaluación

Un experto de dominio **interpreta** los clusters y encuentra información valiosa

¿Cómo mostrar el contenido de los clusters?

- Centroides (medoides)
- Resumen de características
- Características más distintivas de cada cluster
- Aplicar un algoritmo de aprendizaje automático interpretable (Decision Tree)

Evaluación intrínseca

Coeficiente Silhouette

Mide la semejanza de cada objeto al cluster al que se asigna (cohesión), comparada con otros clusters (separación).

Si el valor es bajo o negativo, el número de clusters puede ser inadecuado

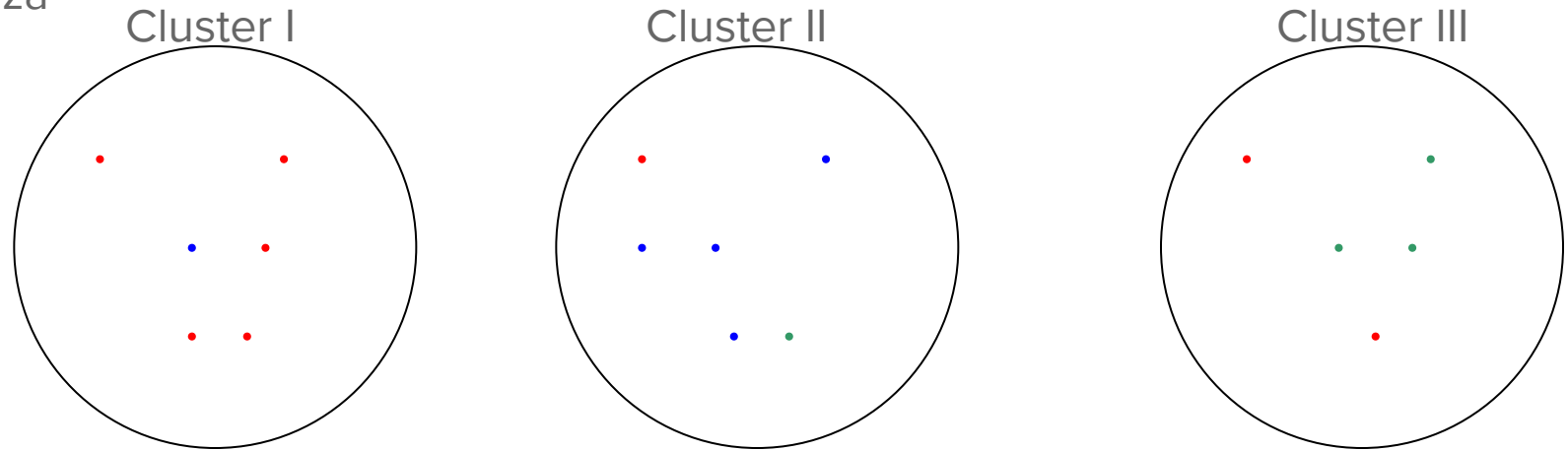
Evaluación con clases

Si los objetos tienen alguna etiqueta, observamos su distribución en los clusters

- Homogeneidad: cada cluster contiene sólo miembros de una clase
- Completitud: todos los miembros de una clase están en el mismo cluster
- V-measure: media harmónica de los anteriores
- Adjusted Rand index: semejanza entre las etiquetas originales y las asignadas
- Información Mútua entre etiquetas originales y asignadas
- Matriz de confusión!

Evaluación con clases

Pureza



Cluster I: Pureza = 1/6 (max(5, 1, 0)) = 5/6

Cluster II: Pureza = 1/6 (max(1, 4, 1)) = 4/6

Cluster III: Pureza = 1/5 (max(2, 0, 3)) = 3/5

Evaluación con testigos

1. Se seleccionan aleatoriamente pares de objetos del dataset
2. Un experto del dominio decide si tienen que estar en el mismo cluster o en diferentes clusters
3. Observamos el grado de acuerdo entre cada solución y los testigos

1. Se seleccionan aleatoriamente objetos del dataset
2. Se los etiqueta
3. Se observa cómo se distribuyen en el dataset

Indicadores de malas soluciones

En general, las malas soluciones se deben a malas características

- Una clase muy grande y el resto mucho más chicas ➔ la mayoría de objetos son no diferenciables con esas características o distancia
- Clases con uno o pocos elementos ➔ el número de clases es demasiado grande para el dataset
- Clusters con las mismas características, poco distinguibles
- Soluciones muy diferentes con diferentes inicializaciones, número de clusters

Clustering no es clasificación

No vamos a obtener clases bien diferenciadas, sino más bien mucho ruido

Es fuertemente sensible a las características de los objetos, a los parámetros, a los outliers

La mayor parte de aproximaciones son muy inestables

La primera aproximación suele ser inservible, hay que refinar características e iterar

Aplicaciones

- Segmentación de clientes, usuarios... para marketing personalizado
- Encontrar temas ➜ topic detection
- Imágenes de los mismos objetos ➜ gatitos, tumores (imágenes médicas), tormentas (imágenes satelitales), plagas (imágenes de cultivos)
- Agrupamiento de productos
- Detección de anomalías
- Taxonomías de plantas y otros organismos
- Detección de clases con significados semejantes