Machine Learning

Università Roma Tre
Dipartimento di Ingegneria
Anno Accademico 2021 - 2022

Introduzione al
Clustering

Sommario

- Supervised e Unsupervised Learning
- Introduzione al Clustering
- Algoritmo k-means
- Algoritmo k-means++

Supervised vs. Unsupervised Learning

- Come sappiamo, molti problemi e metodi di Machine Learning rientrano in una delle due seguenti categorie: apprendimento *supervisionato* o *non supervisionato*.
- Gli esempi visti fino ad ora rientrano nel dominio dell'apprendimento supervisionato:
  - In quei casi (linear regression, logistic regression, ecc.) si hanno delle osservazioni che, a fronte di una certa configurazione delle features, ci dicono quale sia la soluzione corretta.

Supervised vs. Unsupervised Learning

- Nel caso non supervisionato ci troviamo in una situazione più impegnativa, nella quale abbiamo le varie osservazioni caratterizzate dai vari valori delle *features*, ma per le quali non abbiamo disponibili le soluzioni.
- In questa situazione, in un certo senso dobbiamo lavorare alla cieca.
- La situazione è definita *unsupervised* proprio perché nei *data points* disponibili ci manca la risposta che può supervisionare la nostra analisi.

Clustering

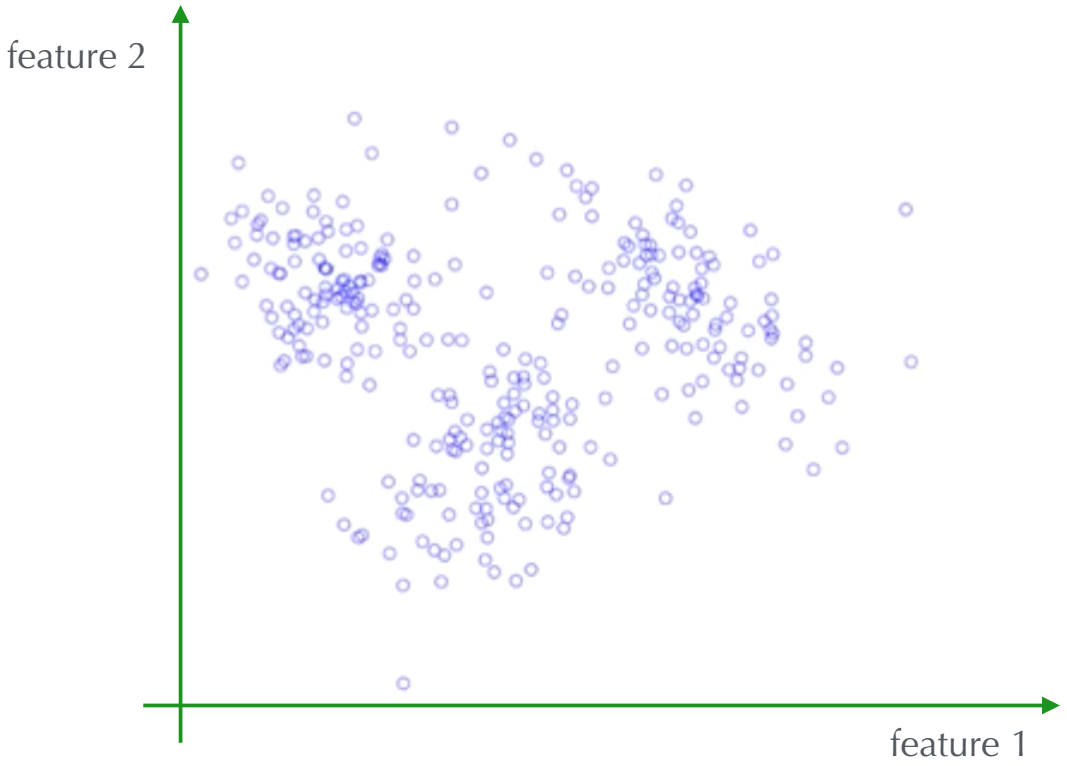
- Dobbiamo chiederci quale tipo di analisi sia possibile in tale contesto.
- Possiamo ad esempio cercare di comprendere le relazioni tra le osservazioni.
- Un approccio che possiamo usare in tali situazioni è quello della *cluster analysis*, o *clustering*.
- L'obiettivo del *clustering* è quello di verificare, date le features in input, se le osservazioni disponibili ricadono all'interno di gruppi relativamente distinti tra di loro.

Clustering

- Il *clustering* è in effetti una delle tecniche più utilizzate per la *exploratory data analysis*.
- In tante discipline, dalle scienze sociali alla biologia alla computer science, gli studiosi cercano di avere delle prime "intuizioni" sui dati di cui dispongono identificando gruppi significativi dei data points:
  - i venditori cercano di identificare cluster di clienti, in base ai loro profili, per migliorare l'attività di marketing (*market segmentation*);
  - i medici cercano di raggruppare i pazienti in base alle loro condizioni cliniche;
  - gli astronomi identificano cluster di stelle in base alla loro prossimità spaziale;
  - ecc. ecc.

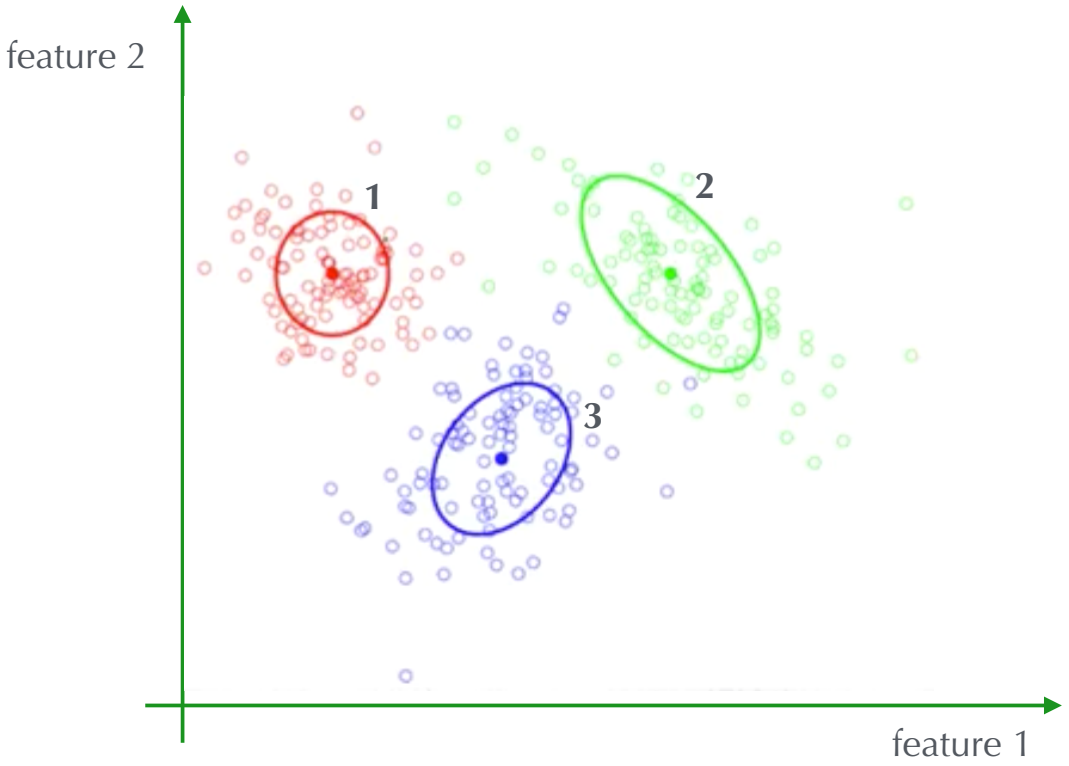
Clustering

- Esempio in due dimensioni: individuare la *cluster structure* solo dagli input:



Clustering

- Ogni cluster è definito dal *centroide* (*cluster center*) e dalla forma (shape/spread):



Clustering

- Ciascuna osservazione $\mathbf{x}_i$ è assegnata al cluster *k* se:
  - Il punteggio (*score*) di $\mathbf{x}_i$ sotto il cluster *k* è migliore rispetto agli altri cluster.
- Per semplicità, spesso si definisce lo *score* come la distanza dal *centroide* del cluster (si ignora lo shape).

k-means Clustering

- L'algoritmo *k-means* assume come *score* proprio la distanza di una osservazione dal *centroide*. Più bassa è la distanza, "migliore" è lo *score*.
- Definizione dei simboli utilizzati nell'esempio che segue:

N : numero delle osservazioni

$\mathbf{x}_i$: osservazione i-esima ($\mathbf{x}_i \in \mathbb{R}^d$)

j : indice dei cluster

μ_j : *centroide* del cluster j

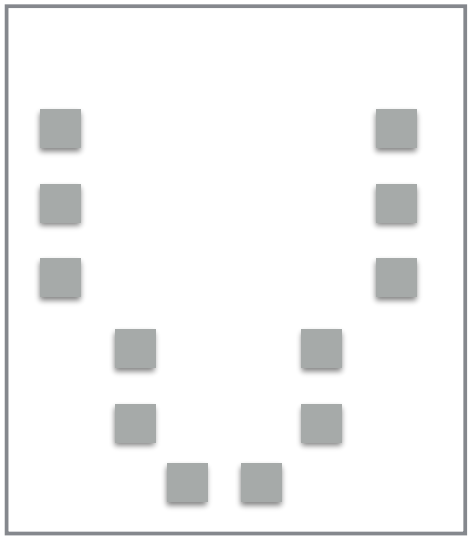
n_j : numero di elementi nel cluster j

z_i : label del cluster a cui appartiene $\mathbf{x}_i$

k : numero dei cluster

k-means Clustering

- Vediamo un esempio di esecuzione dell'algoritmo nel caso in cui i data points siano quelli riportati in figura.
- Supponiamo di scegliere come numero di cluster: k=3

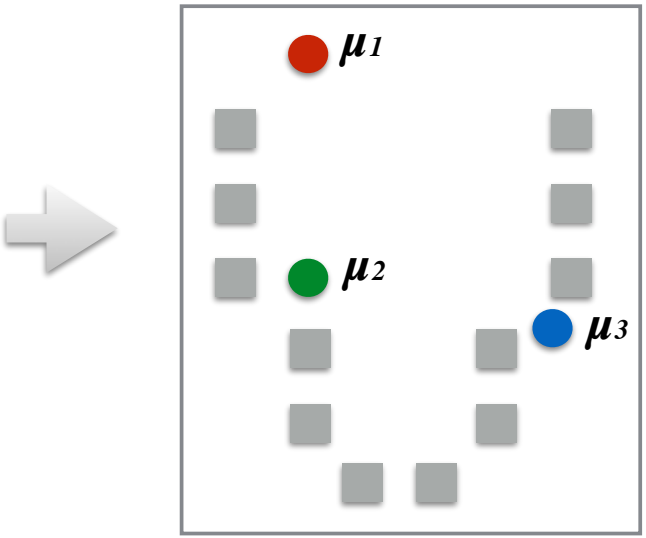


k-means Clustering

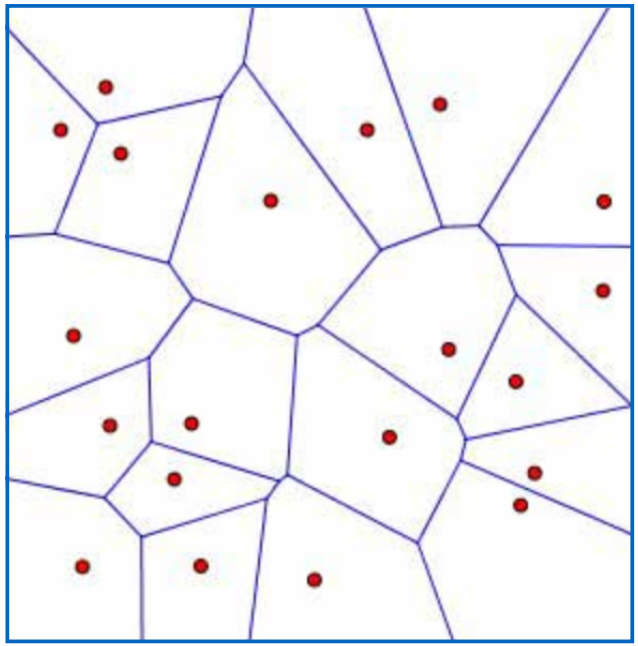
- Scelta del numero di cluster k e inizializzazione dei k centroidi:

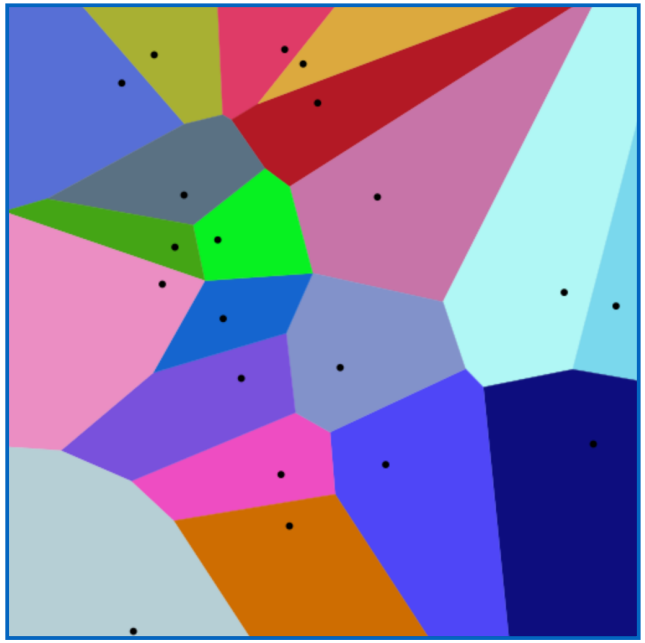
$$\mu_1, \mu_2, \ldots, \mu_k$$

Esempio per k = 3



Voronoi Tesselation

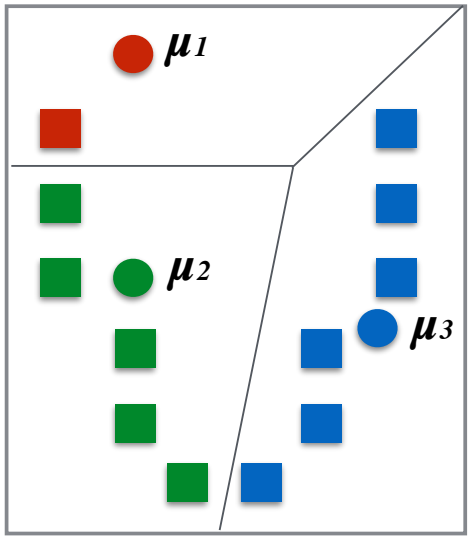




k-means Clustering

- Assegnazione delle osservazioni al più vicino centroide:

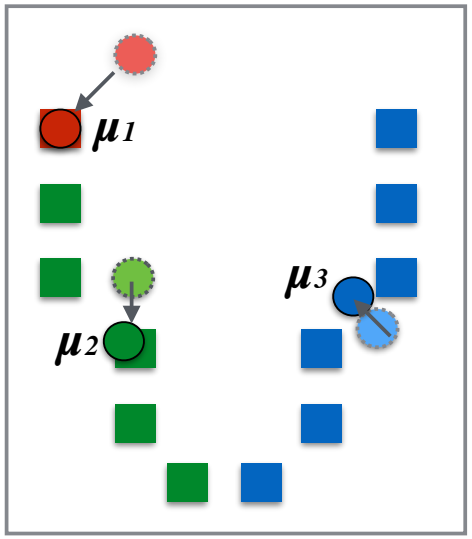
$$z_i \leftarrow \underset{j}{\operatorname{argmin}} \|\mu_j - \mathbf{x}_i\|^2$$



k-Means Clustering

- Si ricalcolano i centroidi come media delle osservazioni assegnate ad ogni cluster:

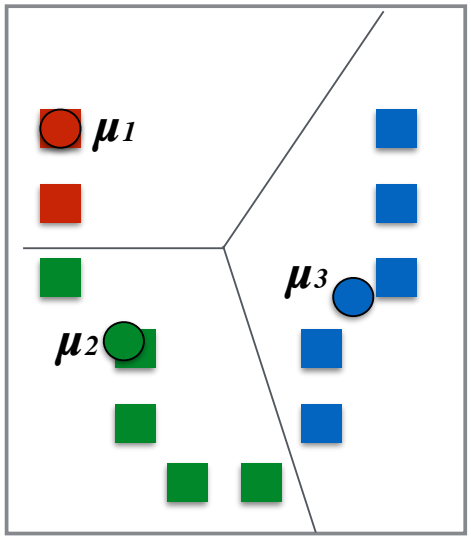
$$\mu_j = \frac{1}{n_j} \sum_{i:\, z_i = j} \mathbf{x}_i$$



k-Means Clustering

- Si riassegnano le osservazioni al centroide più vicino:

$$z_i \leftarrow \underset{j}{\operatorname{argmin}} \|\mu_j - \mathbf{x}_i\|^2$$



… e così via fino al raggiungimento di una cond. di terminazione.

Algoritmo k-means

- L'algoritmo può essere pertanto sintetizzato come segue:

Scegliamo il numero k dei cluster

Inizializziamo i centroidi $\mu_1, \mu_2, \ldots, \mu_k$

while not converged

  for $i = 1, ..., N$

    $z_i \leftarrow \underset{j}{\text{argmin}} \|\mu_j - \mathbf{x}_i\|^2$; assegniamo i data points al cluster center più vicino

  for $j = 1, ..., k$

    $\mu_j = \frac{1}{n_j} \sum_{i:\, z_i = j} \mathbf{x}_i$; aggiorniamo ciascun cluster center come media dei suoi data points

Algoritmo k-means come Coordinate Descent

- Si noti che la formula per il calcolo delle medie:

$$\mu_j = \frac{1}{n_j} \sum_{i:\, z_i = j} \mathbf{x}_i$$

è equivalente alla seguente espressione:

$$\mu_j \leftarrow \operatorname*{argmin}_{\mu} \sum_{i:\, z_i = j} \|\mu - \mathbf{x}_i\|^2$$

Algoritmo k-means come Coordinate Descent

- Abbiamo dunque la seguente versione equivalente dell'algoritmo:

Scegliamo il numero k dei cluster

Inizializziamo i centroidi $\mu_1, \mu_2, \ldots, \mu_k$

while not converged

  for $i = 1, ..., N$

    $z_i \leftarrow \underset{j}{\text{argmin}} \|\mu_j - \mathbf{x}_i\|^2$; assegniamo i data points al cluster center più vicino

  for $j = 1, ..., k$

    $\mu_j \leftarrow \underset{\mu}{\text{argmin}} \sum_{i:\, z_i = j} \|\mu - \mathbf{x}_i\|^2$; calcolo centroidi che minimizzano la somma del

    ; quadrato delle norme per i loro data points

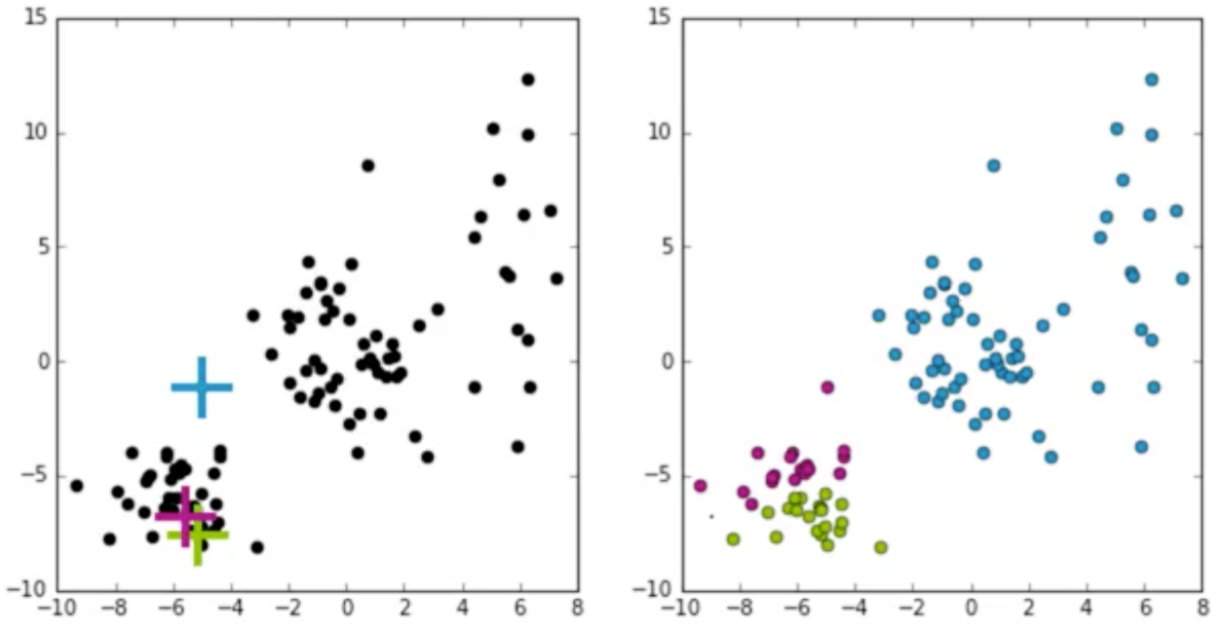
dove si alternano le minimizzazioni (a): z dato μ e (b): μ dato z.

Convergenza di k-means

- In genere k-means converge ad un ottimo locale.
- L'algoritmo è molto sensibile all'inizializzazione dei centroidi.
- Vediamo un esempio:

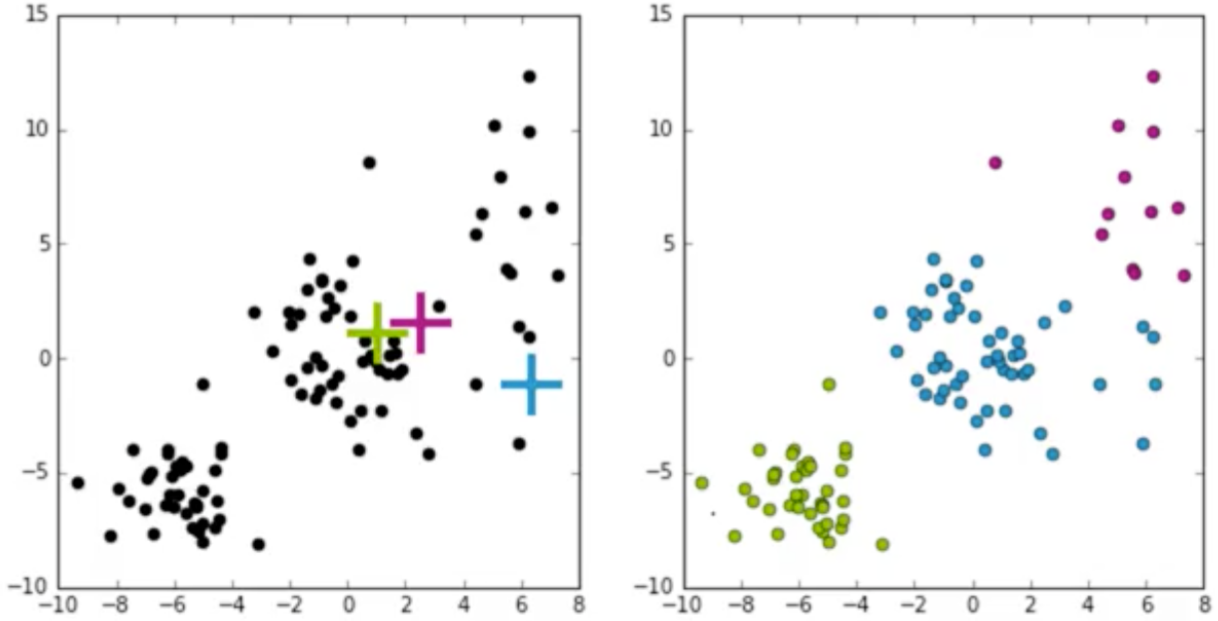
Convergenza di k-means

- Data la scelta dei centroidi iniziali mostrata nella figura a sinistra, si ottiene il risultato mostrato a destra:



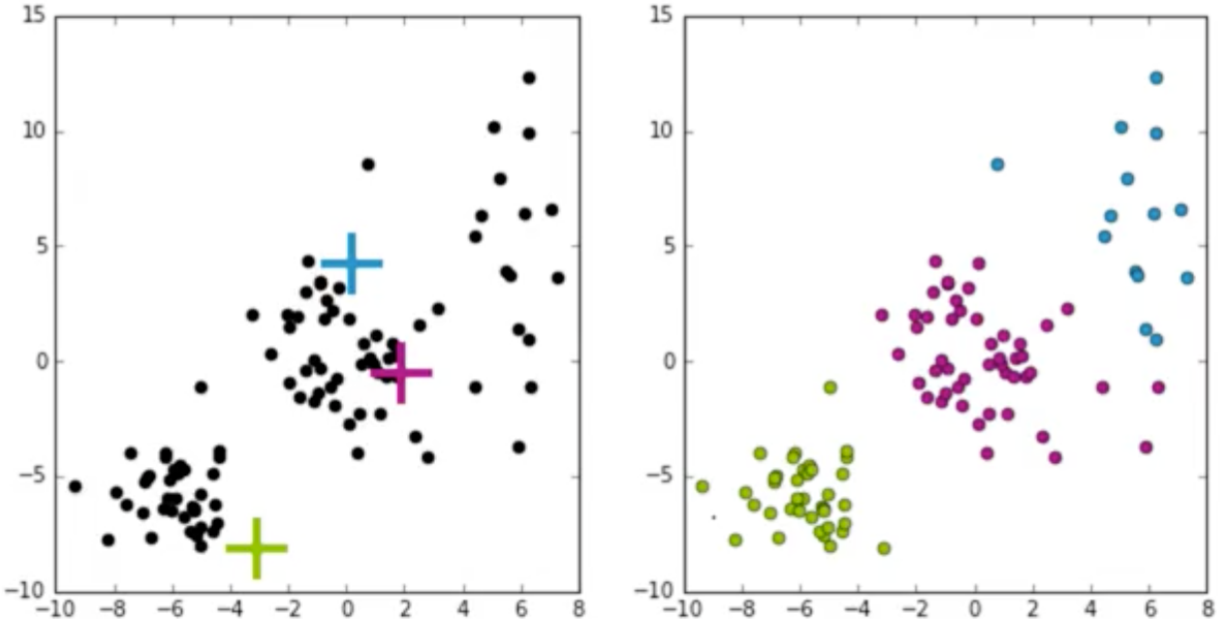
Convergenza di k-means

- Altra scelta dei centroidi iniziali:



Convergenza di k-means

- Altra scelta dei centroidi iniziali:



k-means++

- Come abbiamo visto, l'inizializzazione di k-means è critica ai fini della qualità dell'ottimo locale trovato.
- Ora vediamo k-means++, un metodo che consiste in una particolare inizializzazione dei centroidi che in genere dà buoni risultati.
- Riferimenti:

Arthur, D. e Vassilvitskii, S. "k-means++: the advantages of careful seeding", in *Proc. of the 18th ACM-SIAM Symp. on Discrete Algorithms*, 2007, pp. 1027-1035.

Bahmani, B., Moseley, B., Vattani, A., Kumar, R. e Vassilvitskii, S. "Scalable k-means++", in *Proc. of VLDB*, 2012.

k-means++

- **Smart initialization**:

  1. Scegliere il primo centroide in modo casuale tra tutti i data points.

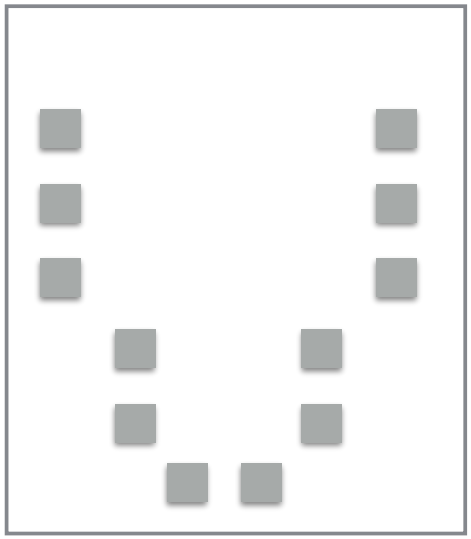
  2. Per ogni osservazione $\mathbf{x}_i$, calcolare la distanza $d(\mathbf{x}_i)$ tra $\mathbf{x}_i$ e il più vicino centroide.

  3. Scegliere il nuovo centroide tra i data point, con la probabilità di $\mathbf{x}_i$ di essere scelto proporzionale a $d(\mathbf{x}_i)^2$, ossia al quadrato della distanza tra $\mathbf{x}_i$ e il centroide più vicino già scelto.

  4. Ripeti gli step 2 e 3 fino ad arrivare a scegliere k centroidi.

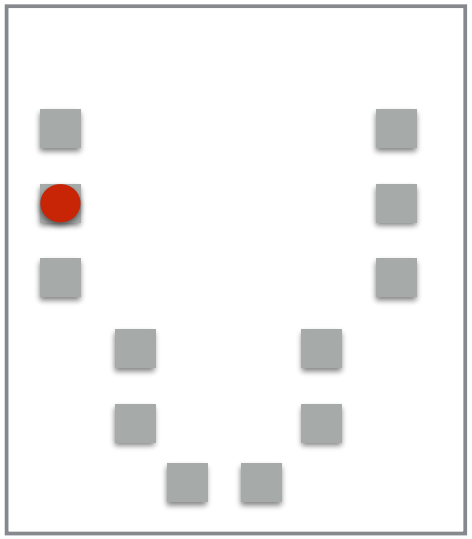
k-means++: esempio

- Vediamo un esempio di inizializzazione con k=3, relativo alle osservazioni in figura:



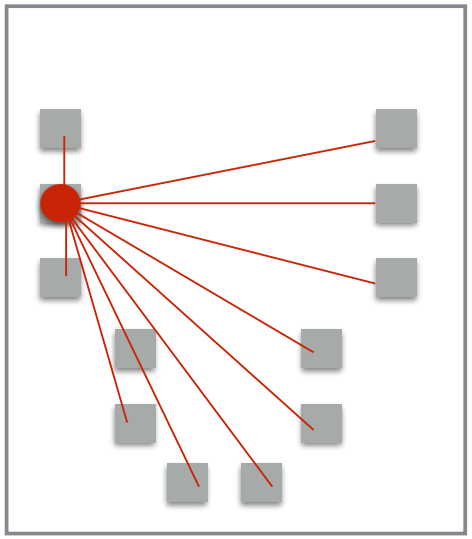
k-means++: esempio

- Scelta random del primo cluster center:



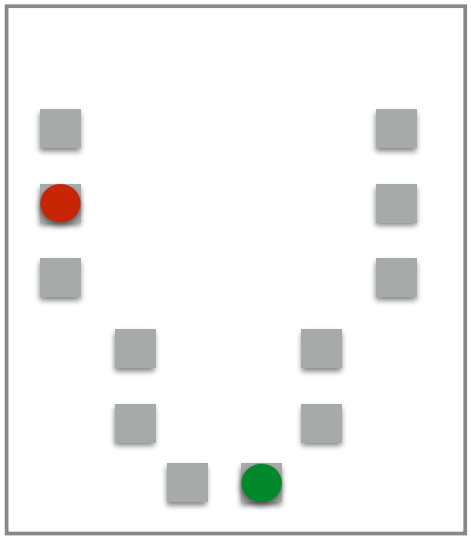
k-means++: esempio

- Scelta del secondo cluster center. Si sceglie il punto con la probabilità maggiore, dove la probabilità è proporzionale a $d(\mathbf{x})^2$. In figura sono mostrate le varie distanze.



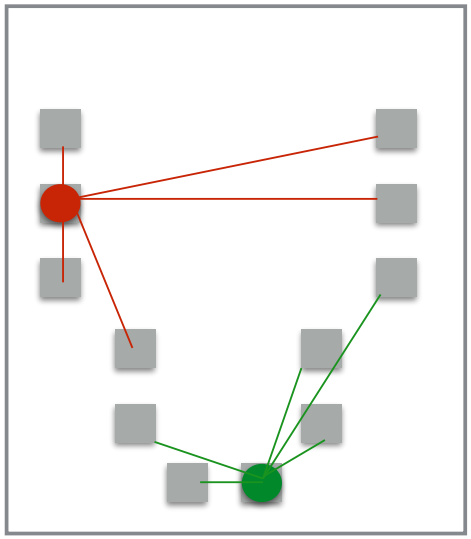
k-means++: esempio

- Supponiamo che venga scelto il secondo cluster center in verde:



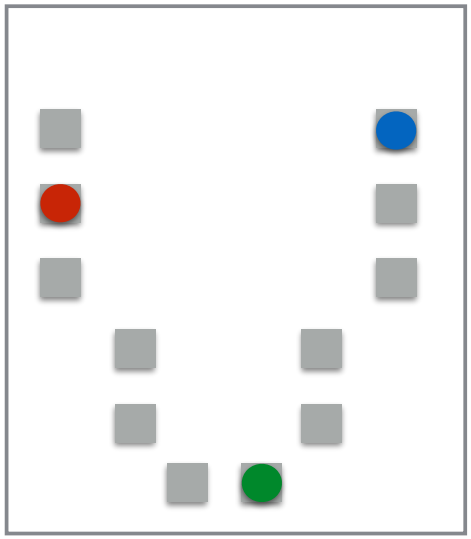
k-means++: esempio

- Scelta dell'ultimo cluster center. Di nuovo, si sceglie il punto con la probabilità maggiore, dove la probabilità è proporzionale a $d(\mathbf{x}_i)^2$, quadrato della distanza tra il punto i e il più vicino centroide:



k-means++: esempio

- Supponiamo che il cluster center scelto sia quello in blu. I tre centroidi scelti sono quelli con cui inizializziamo l'algoritmo k-means.



k-means++: pros & cons

- Eseguire k-means++ per individuare i centroidi iniziali è certamente più oneroso computazionalmente rispetto alla scelta random dei suddetti centroidi.
- Per contro, l'esecuzione di k-means con l'inizializzazione di k-means++ è spesso più efficiente, nel senso che converge in genere più rapidamente.
- In generale possiamo dire che k-means++ tende a migliorare la qualità dell'ottimo locale trovato e diminuire il tempo di esecuzione.

Cluster Heterogeneity

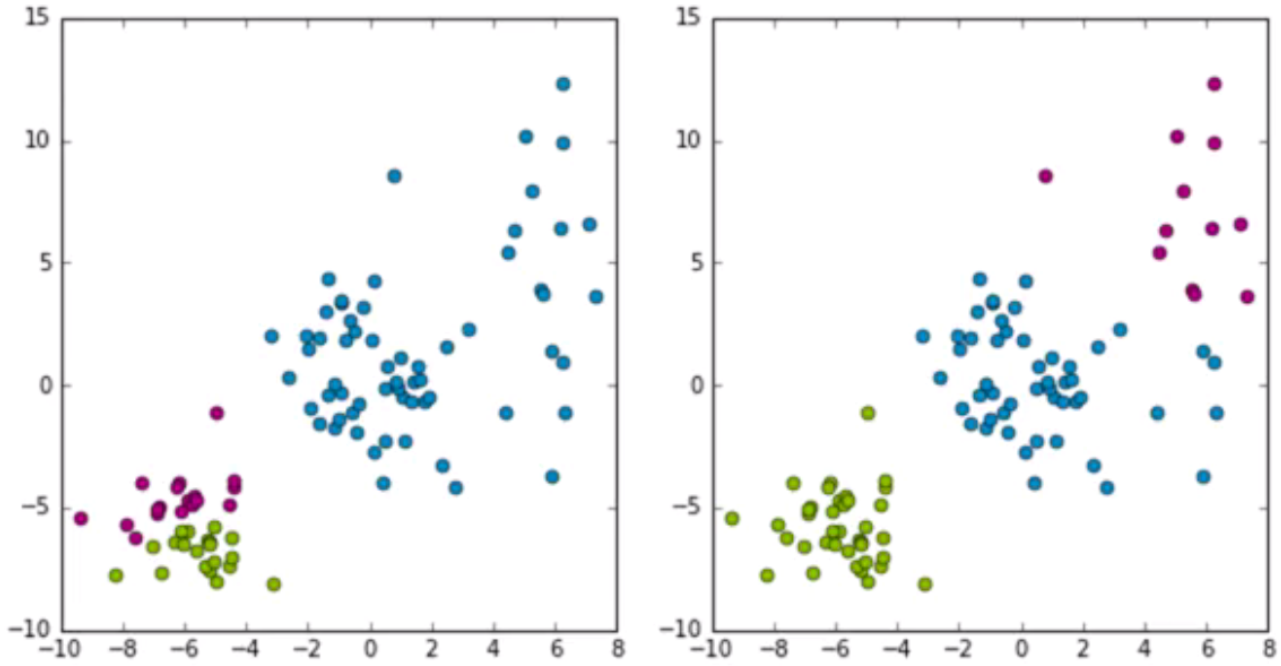
- L'algoritmo k-means cerca di minimizzare la somma dei quadrati delle distanze (*distortion*):

$$\text{costo\_k\_means} = \sum_{j=1}^{k} \sum_{i:\, z_i = j} \|\mu_j - \mathbf{x}_i\|^2$$

- Come abbiamo visto, in genere l'algoritmo trova un minimo locale.

Cluster Heterogeneity

- Confrontiamo i seguenti due risultati: la figura a destra è sicuramente migliore. La figura a sinistra è più “eterogenea”.

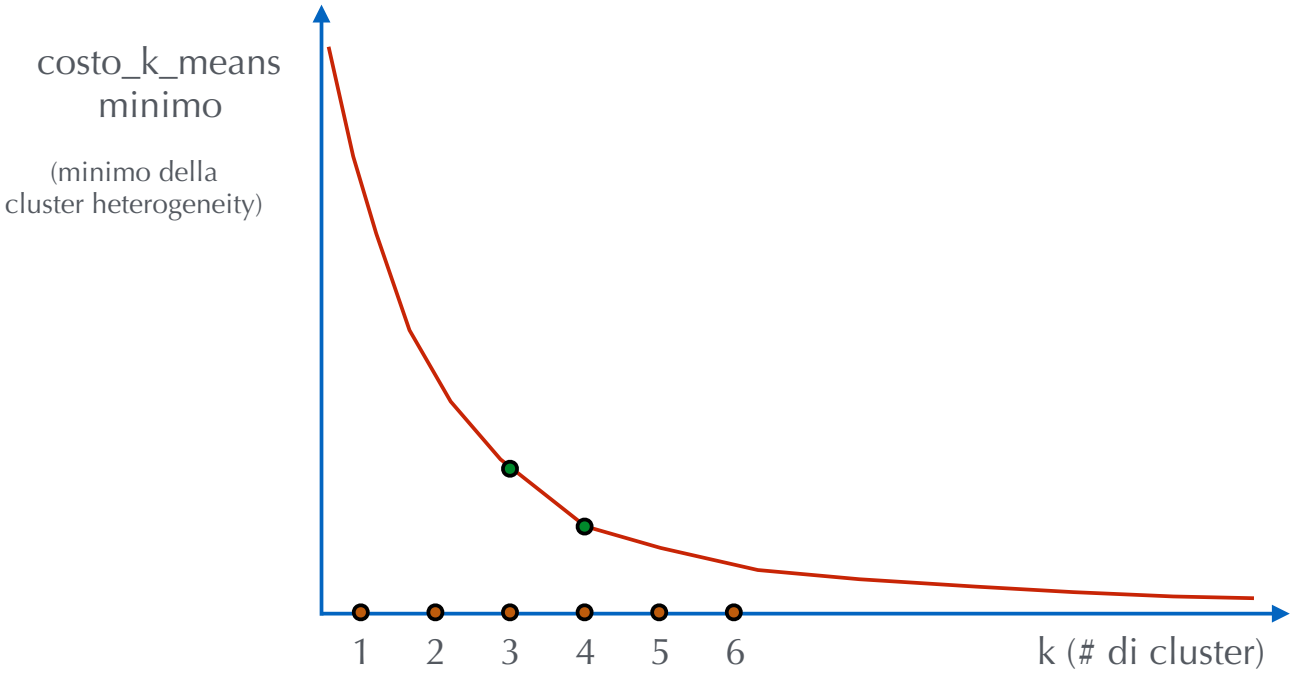


Cosa accade al crescere di k

- Consideriamo il caso estremo k = N:
  - Significa che ogni cluster center è un data point.
  - Il costo (heterogeneity) è uguale a zero.
- Il costo (heterogeneity) decresce al crescere di k.

Scelta del numero di cluster k

- "Elbow Method": Un'euristica usata è quella di scegliere un punto che si trova nel "gomito" della curva:



Riferimenti

- Watt, J., Borhani, R., Katsaggelos, A.K. *Machine Learning Refined*, 2nd edition, Cambridge University Press, 2020.
- James, G., Witten, D., Hastie, T., Tibishirani, R. *An Introduction to Statistical Learning*, Springer, 2013.
- Ross, S.M. *Probabilità e Statistica per l'Ingegneria e le Scienze*, Apogeo, 3a edizione, 2015.
- *Machine Learning: Clustering & retrieval*, University of Washington - Coursera, 2017.
- Flach, P. *Machine Learning - The Art and Science of Algorithms that Make Sense of Data*, Cambridge University Press, 2012.
- Murphy, K.P. *Machine Learning - A Probabilistic Approach*, The MIT Press, 2012.