Introduzione al Data Mining

e applicazioni al dominio del Contact Management

Parte III: principali modelli per l'Unsupervised Learning

Andrea Brunello

Università degli Studi di Udine

In collaborazione con dott. Enrico Marzano, CIO Gap srl
progetto Active Contact System

Contenuti

Parte III:

- principali modelli per l'*Unsupervised Learning*:
  - clustering;
  - regole di associazione.

Clustering

Principale metodologia di *Unsupervised Learning*:

- Cerchiamo di raggruppare fra loro istanze simili, senza considerare un determinato attributo come obiettivo;
- i gruppi individuati possono essere:
  - *esclusivi (hard clustering)*
  - *con overlap (fuzzy clustering)*
  - *probabilistici (fuzzy clustering)*
  - *gerarchici*
- il processo di clustering può essere seguito da uno di classificazione:
  - come “conferma” dei cluster;
  - per assegnare ai gruppi nuove istanze.

Clustering (2)

Diverse famiglie di algoritmi di clustering:

- gerarchici;
- basati su centroidi: *k-means(++), k-medoids, FCM*
- basati sulla distribuzione: *Expectation-Maximization*
- basati sulla densità: *DBSCAN*, *OPTICS*
- ...

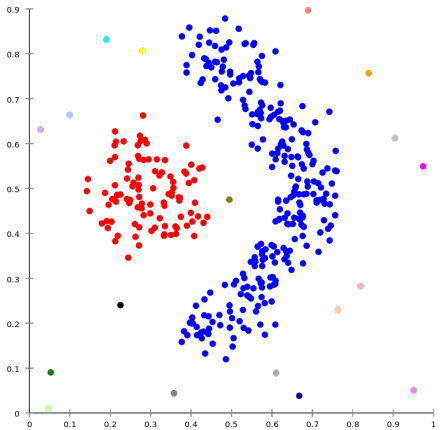
La scelta della tipologia dell'algoritmo riveste grande importanza per quanto riguarda il risultato finale, e dipende anche dalla distribuzione delle istanze.

Clustering gerarchico

Idea - un oggetto è più simile agli oggetti ad esso vicini, rispetto a quelli lontani:

- si parte da un nr. di cluster pari al nr. delle istanze;
- i due cluster più simili fra loro vengono collegati;
- si procede iterativamente collegando di volta in volta fra loro i due cluster più simili (diverse metodologie);
- sino a collegare tutte le istanze in un unico cluster;

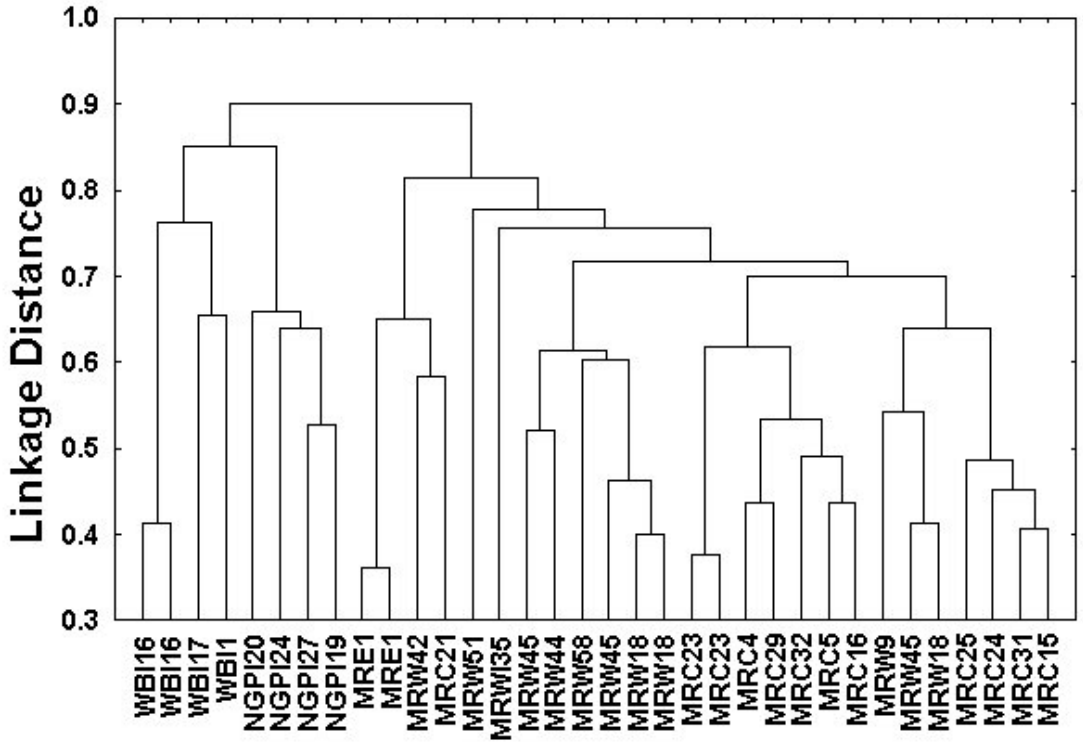
$\rightsquigarrow$ problema relativo agli outliers/rumore.



Clustering gerarchico (2)

L'output è rappresentato da un *dendogramma*:

- **asse x**: istanze;
- **asse y**: distanza;
- date due istanze, quanto più il loro primo punto di fusione è basso, tanto più sono simili;
- la similarità **NON** dipende dalla vicinanza lungo l'asse delle x!



Clustering basato sui centroidi

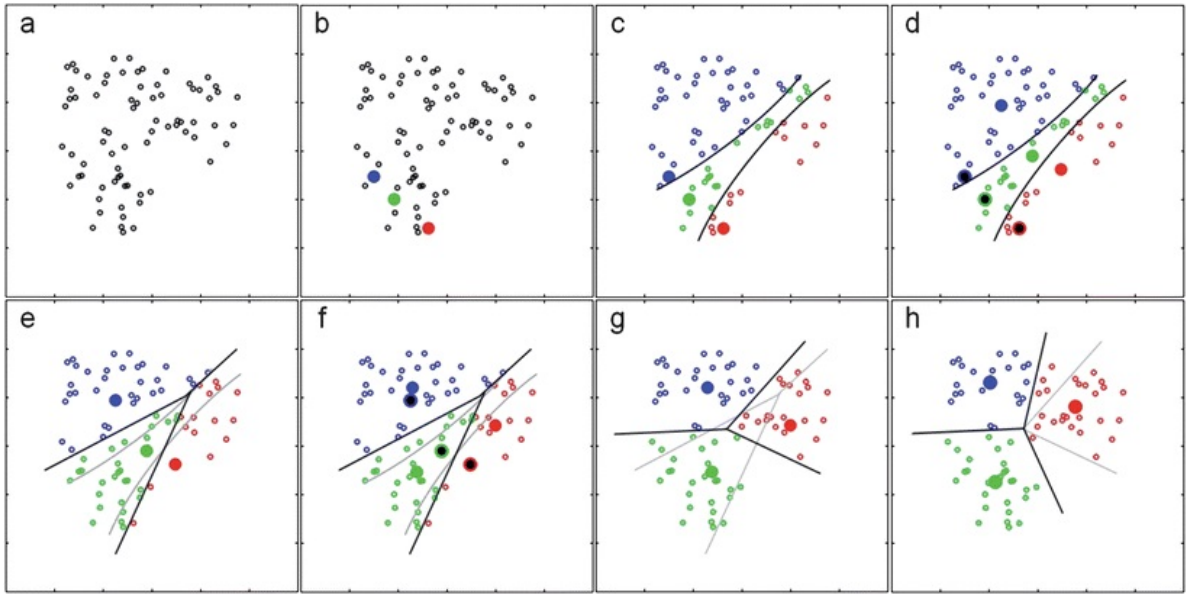
- ciascun cluster viene rappresentato da un *centroide*, elemento che può o meno appartenere al dataset iniziale;
- per gli algoritmi della famiglia *k-means*, il numero di cluster da ricercare deve essere fissato a priori;
- si cercano *k* centri di altrettanti cluster, e si assegnano ad essi le istanze, in modo tale da minimizzare la somma dei quadrati delle distanze;
- problema *NP-hard*, si cercano soluzioni approssimate.

L'algoritmo *k-means*

Uno dei più diffusi algoritmi per il clustering *partizionale*.
Funzionamento:

1. seleziona il numero di cluster da ricercare, k;
2. scegli casualmente k istanze, centri di altrettanti cluster;
3. assegna tutte le istanze-non-centro al punto, fra i k centri, più vicino;
4. calcola sulle istanze di ciascun cluster la media dei diversi attributi, e poni il punto così ottenuto come nuovo centro del cluster;
5. se, alla luce dei nuovi centri, almeno un'istanza cambierebbe cluster di appartenenza, ripeti dal punto 3, altrimenti la situazione è stabile e l'algoritmo termina.

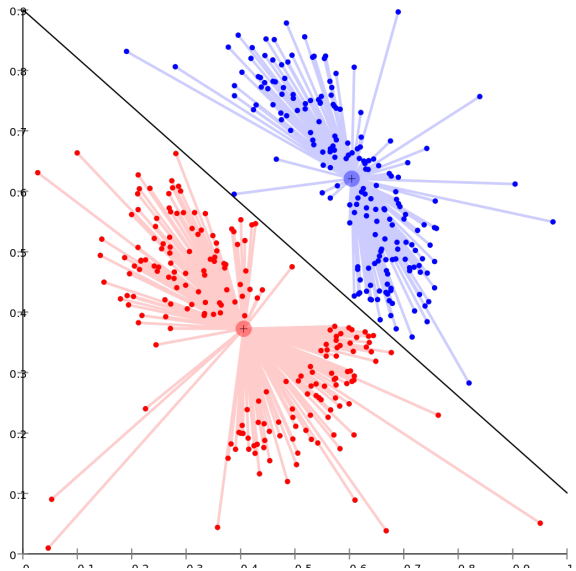
L'algoritmo *k-means* (2)

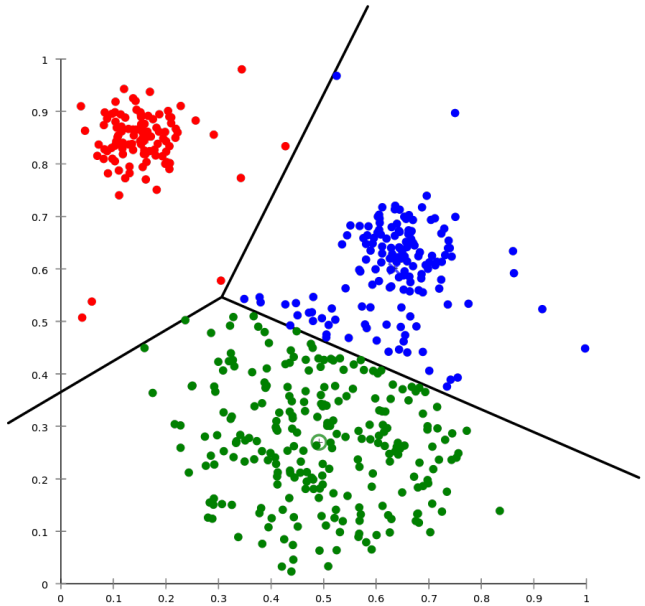


- l'algoritmo non garantisce l'ottimalità del raggruppamento;
- la variante *k-means++* è migliore per velocità ed accuratezza;
- sono da tenere in considerazione aspetti riguardanti la standardizzazione degli attributi;
- è necessario impostare il valore di k, a differenza che in altre metodologie di clustering (es. gerarchico).

Casi di cattiva clusterizzazione

Esempi di cattiva clusterizzazione:





- *prima immagine*: incorretta definizione dei confini fra i cluster;
- *seconda immagine*: impossibilità di catturare la forma dei due cluster (densità);

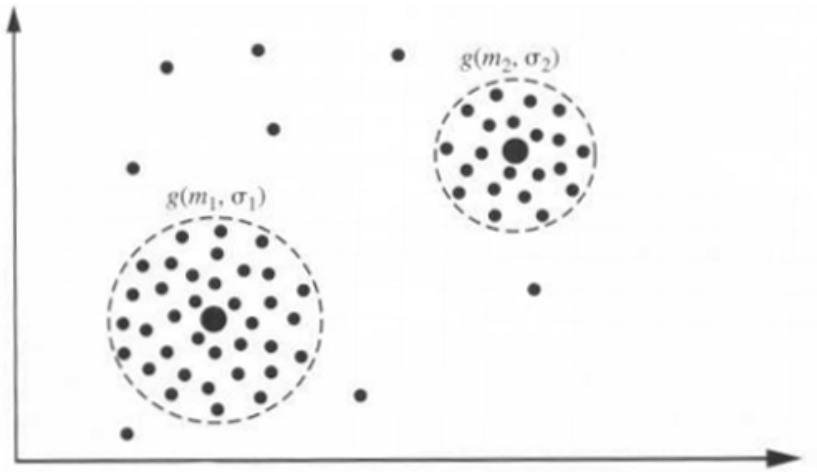
Clustering basato sulla distribuzione

Generalizzazione dell'approccio *k-means*:

- ciascun cluster è rappresentato da una diversa distribuzione di probabilità (consideriamo Gaussiana), ognuna delle quali descrive la distribuzione dei valori per i membri di quel cluster;
- ogni distribuzione fornisce la probabilità che una istanza abbia certi valori per i suoi attributi, supponendo che sia noto a quale cluster appartiene;
- dunque, l'ipotesi di fondo è che le istanze del dataset siano generabili a partire da un mix di modelli probabilistici (*mixture*).

Clustering basato sulla distribuzione (2)

Esempio di un mixture model. In esso sono presenti due cluster, ciascuno dei quali segue una distribuzione Gaussiana con la sua media e la sua deviazione standard:



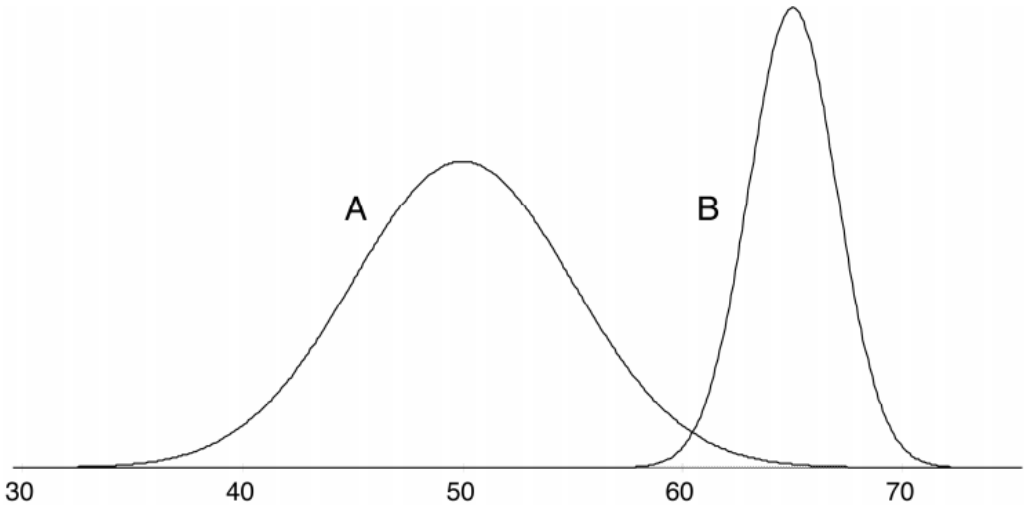
$\rightsquigarrow$ L'obiettivo è, partendo dall'insieme di istanze, trovare le distribuzioni corrispondenti ai cluster, più le probabilità di appartenenza delle istanze ad essi (modello).

Clustering basato sulla distribuzione (3)

Esempio con singola variabile numerica e due distribuzioni gaussiane. Dati etichettati dalle “classi” reali (per comodità, sopra) e modello (sotto):

A	51	B	62	B	64	A	48	A	39	A	51
A	43	A	47	A	51	B	64	B	62	A	48
B	62	A	52	A	52	A	51	B	64	B	64
B	64	B	64	B	62	B	63	A	52	A	42
A	45	A	51	A	49	A	43	B	63	A	48
A	42	B	65	A	48	B	65	B	64	A	41
A	46	A	48	B	62	B	66	A	48		
A	45	A	49	A	43	B	65	B	64		
A	45	A	46	A	40	A	46	A	48		

model



Clustering basato sulla distribuzione (4)

- il modello è dato da due cluster A, B con medie $\mu_A = 50, \mu_B = 65$ e deviazioni standard $\sigma_A = 5, \sigma_B = 2$ (le due *mixture* sono qui formate da un'unica gaussiana);
- le istanze appartengono ad A con probabilità $p_A = 0.6$ ed a B con probabilità $p_B = 1 - p_A$.

$\rightsquigarrow$ date le istanze (senza conoscenza di A e B), vogliamo trovare i 5 parametri: μ_A, μ_B, σ_A, σ_B, p_A.

Algoritmo *E-M*

Problema: non conosciamo né i 5 parametri, né l'effettiva appartenenza delle istanze ai cluster.

- *EM* è un algoritmo di raffinamento iterativo che può essere usato per trovare le stime dei parametri;
- può essere visto come un'estensione di $k - means$ operante *fuzzy clustering*;
- esistono varianti in cui il numero di cluster viene determinato automaticamente;
- non viene garantita l'ottimalità del risultato.

Algoritmo *E-M* (2)

Funzionamento:

1. imposta casualmente i valori per μ_A, μ_B, σ_A, σ_B, p_A;
2. itera:
   - 2.1 **Expectation step** - calcola la (densità di) probabilità di appartenenza ai cluster A, B per ogni istanza x_i, cioè $Pr(x_i \in A)$ e $Pr(x_i \in B)$, a partire dai parametri (assegnamento "soft" delle istanze ai cluster);
   - 2.2 **Maximization step** - stima, in maniera pesata in base alle probabilità di appartenenza delle istanze ai cluster, i parametri.

$\rightsquigarrow$ dunque, ad ogni passo *E-M* riassegna gli oggetti tenendo conto del modello dato dai parametri;
gli oggetti assegnati vengono quindi usati per generare nuove stime dei parametri.

Algoritmo E-M (3): dettagli

Possiamo calcolare $Pr(x_i \in A)$ a partire dai parametri:

$$Pr(x_i \in A) = Pr(A|x_i) = \frac{Pr(x_i|A) * p_A}{Pr(x_i)}$$

dove:

$$Pr(x_i|A) = \frac{1}{\sigma_A \sqrt{2\pi}} e^{-\frac{(x_i - \mu_A)^2}{2\sigma_A^2}}$$

ossia la funzione di densità di probabilità nel caso della distribuzione normale.

Non conosciamo $Pr(x_i)$ (*probabilità di un'istanza dati i cluster*); tuttavia, $Pr(A|x_i) + Pr(B|x_i) = 1$, dunque:

$$\frac{Pr(x_i|A) * p_A + Pr(x_i|B) * p_B}{Pr(x_i)} = 1$$

Algoritmo E-M (4): dettagli

I parametri possono essere invece ricalcolati a partire dalle (densità di) probabilità come segue:

$$\mu_a = \frac{\sum_i Pr(x_i \in A) * x_i}{\sum_i Pr(x_i \in A)}$$

$$\sigma_a = \sqrt{\frac{\sum_i Pr(x_i \in A) * (x_i - \mu_A)^2}{\sum_i Pr(x_i \in A)}}$$

$$p_A = \frac{\sum_i Pr(x_i \in A)}{numero\_totale\_istanze}$$

Algoritmo *E-M* (5)

Quando terminare?

- *EM* converge verso un punto fisso;
- concetto di verosimiglianza globale (*likelihood*), calcolata ad ogni iterazione:
  - misura della bontà del clustering;
  - aumenta ad ogni iterazione, fino ad un massimo locale;
  - intuitivamente, quanto è verosimile che il modello descriva/generi il dataset originario.
- l'algoritmo si arresta quando la *likelihood* rimane invariata (o miglioramento trascurabile).

Algoritmo *E-M* (6)

Come calcolare la *likelihood*?

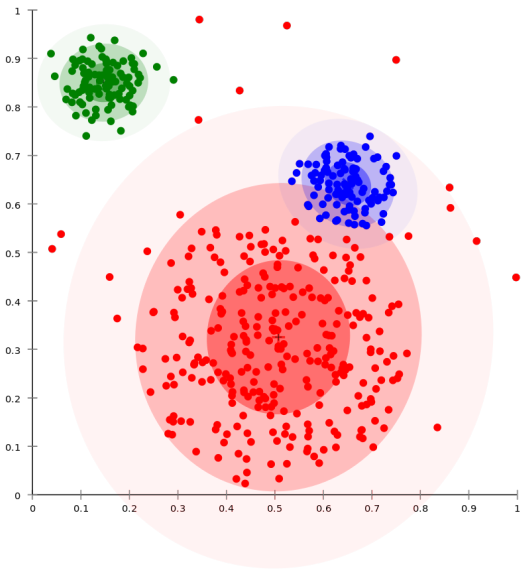
$$\mathcal{L} = \prod_i Pr(x_i) = \prod_i (Pr(x_i|A) * p_A + Pr(x_i|B) * p_B)$$

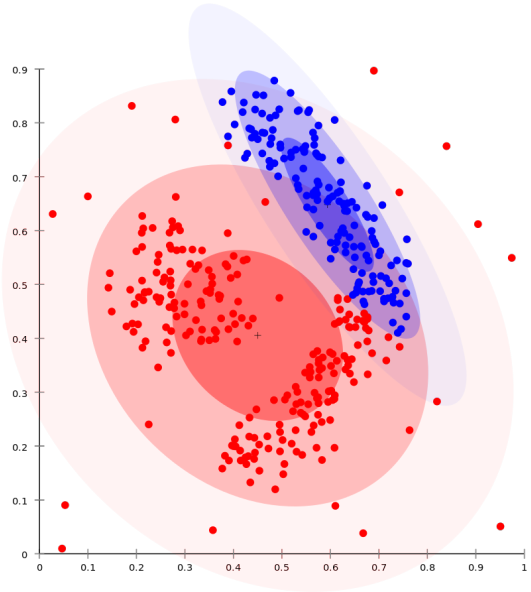
In pratica si calcola il logaritmo della verosimiglianza.

$\rightsquigarrow$ la *likelihood* può essere anche utilizzata per confrontare la bontà di diversi risultati di clustering.

Casi di clusterizzazione

Esempi di clusterizzazione con *EM*:





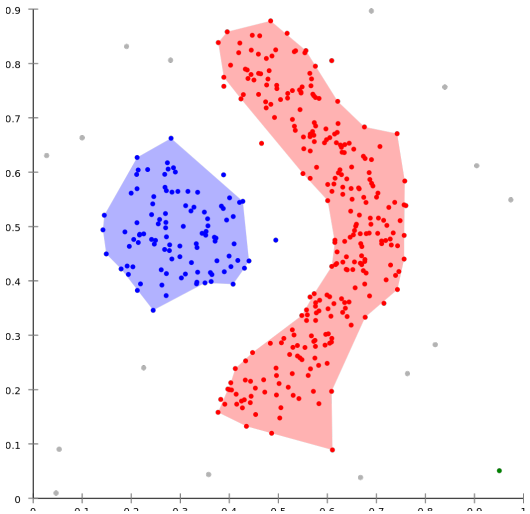
- *prima immagine*: buona clusterizzazione su dati con distribuzione gaussiana;
- *seconda immagine*: impossibilità di catturare la forma dei due cluster (densità).

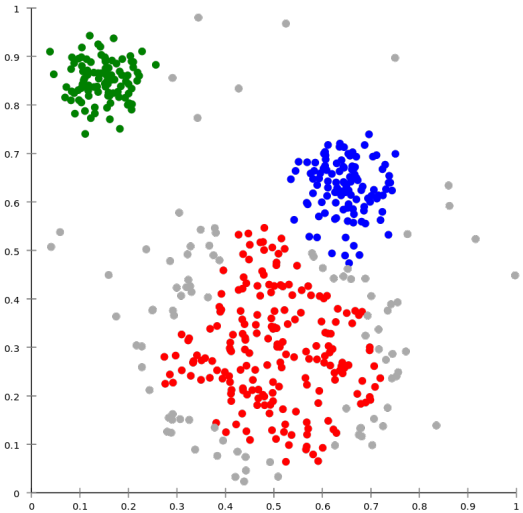
Clustering basato sulla densità

- Un cluster è inteso come una regione densa di punti, separata da regioni a bassa densità dalle altre regioni a elevata densità;
- i punti nelle regioni a bassa densità sono considerati rumore o punti di confine.

Due algoritmi principali:

- *DBSCAN*: assume cluster aventi densità simile;
- *OPTICS*: variante in grado di trattare densità diverse.





Regole di associazione

Simili alle classification rules, con le differenze:

- si vogliono mettere in luce generiche relazioni fra gli attributi;
- la loro parte destra può fare riferimento ad uno o più attributi;
- ciascuna regola opera in maniera indipendente.

$$Temperatura = fredda \rightarrow Umidita = normale;$$

$$Umidita = normale \wedge Vento = false \rightarrow Si\_gioca = si;$$

$$Condizioni = soleggiato \wedge Si\_gioca = no \rightarrow Umidita = alta;$$

$$Vento = false \wedge Si\_gioca = no \rightarrow Condizioni = soleggiato \wedge Umidita = alta;$$

Riferimenti

I. H. Witten, E. Frank, M. A. Hall: *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 3rd edition, 2011.

G. James, D. Witten, T. Hastie, R. Tibshirani: *An Introduction to Statistical Learning with Applications in R*. Springer, 2013.