University of Pisa

SCUOLA DI INGEGNERIA

Corso di Laurea in Artificial Intelligence and Data Engineering

Statistica

Candidato
Marco Parola

Relatore
Prof. Romito

Anno Accademico 2019–2020

Indice

1 Introduction **2**

2 Valutazione numero di cluster **2**

3 Applicazione degli algoritmi di clustering **3**

3.1 K-means 3

3.2 PAM 4

4 Interpretazione cluster e conclusioni **5**

1 Introduction

Il dataset utilizzato per l'analisi dei dati del secondo modulo, scaricabile dalla piattaforma UCI al seguente link: https://archive.ics.uci.edu/ml/machine-learning-databases/00488/ , descrive i post relativi a pagine Facebook di venditori al dettaglio di moda e cosmetici tailandesi.
Numero delle osservazioni: 7050
Numero di fattori: 16
L'analisi viene eseguita su un sottonumero di fattori:

- num_reactions, numero di reazioni che il post ha ricevuto
- num_comments, numero di commenti
- num_shares, numero di condivisioni
- num_likes, numero di 'mi piace'
- num_loves, numero di reazioni 'Love'
- num_wows, numero di reazioni sorprese
- num_hahas, numero di reazioni divertite
- num_sads, numero di reazioni tristi
- num_angrys, numero di reazioni arrabbiate

L'obiettivo di questa analisi è esplorare questi post, in modo da vedere se si riescono ad identificare alcuni trend tra i post, magari riuscendo ad identificare vendite particolarmente vantaggiose collegate a offerte.

2 Valutazione numero di cluster

Dunque vogliamo affrontare un problema di clustering. Come primo passo è necessario identificare il numero di cluster ideale, per creare delle partizioni il più simili tra loro. Tale decisione può essere presa valutando l'andamento della silhouette media, al variare del numero di cluster, che si decide di considerare.

```
library(cluster);
x = read.csv("Live.csv");
x = x[4:12];

as=rep(0,10);
for(k in 2:10){
    cl=kmeans(x,k,nstart=20)$cluster;
    as[k]=mean(silhouette(cl,dist(x))[,3]);
}
plot(as,type="b",pch=20);
```

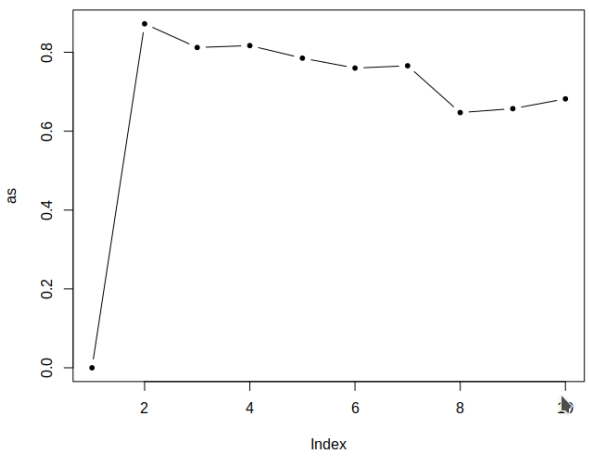

Figura 1: Variazione silhouette all'aumentare

Dal grafico in figura 1 si nota come il valore più alto di silhouette si trova in corrispondenza di 3 cluster, un altro valore interessante, per cui la silhouette assume un valore alto, è 4.

3 Applicazione degli algoritmi di clustering

3.1 K-means

Valutiamo i risultati dell'applicazione dell'algoritmo kmeans, specificando tra i parametri il numero di cluster che vogliamo ottenere.

```
x.km=kmeans(x,3,nstart=20);
x.pca=princomp(scale(x));
plot(x.pca$scores,col=1+x.km$cluster,pch=20);
plot(silhouette(x.km$cluster,dist(x)));
```

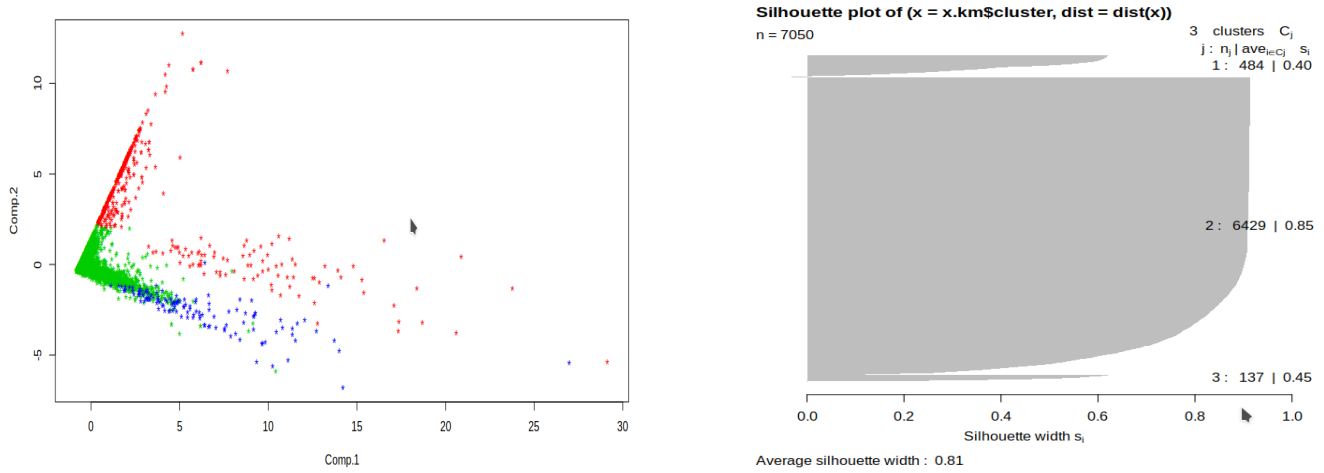

Figura 2: Left: Osservazioni rappresentate sul piano delle copenenti principali. Right: Grafico della silhouette.

Da una prima visualizzazione dei risultati, si nota una disomogeneità nel numero di campioni per ogni cluster.
Un' ipotesi per spiegare questa non omogeneità è l'esistenza, tra i post Facebook, di una maggioranza di post standard, che possiamo attribuire a vendite ordinarie; mentre una minoranza che viene clusterizzata in due gruppi minori, magari l'esordio di due nuovi trend, che prenderanno piede maggiormente col passare de tempo.
Proseguendo la visualizzazione dei risultati, si scopre che i risultati ottenuti sono buoni: una silhouette totale che ammonta a 81%, mentre quella del cluster più numeroso 85% e i due cluster minori 40% e 45%.

```
x.km=kmeans(x,4,nstart=20);
plot(x.pca$scores,col=1+x.km$cluster,pch=20);
plot(silhouette(x.km$cluster,dist(x)));
```

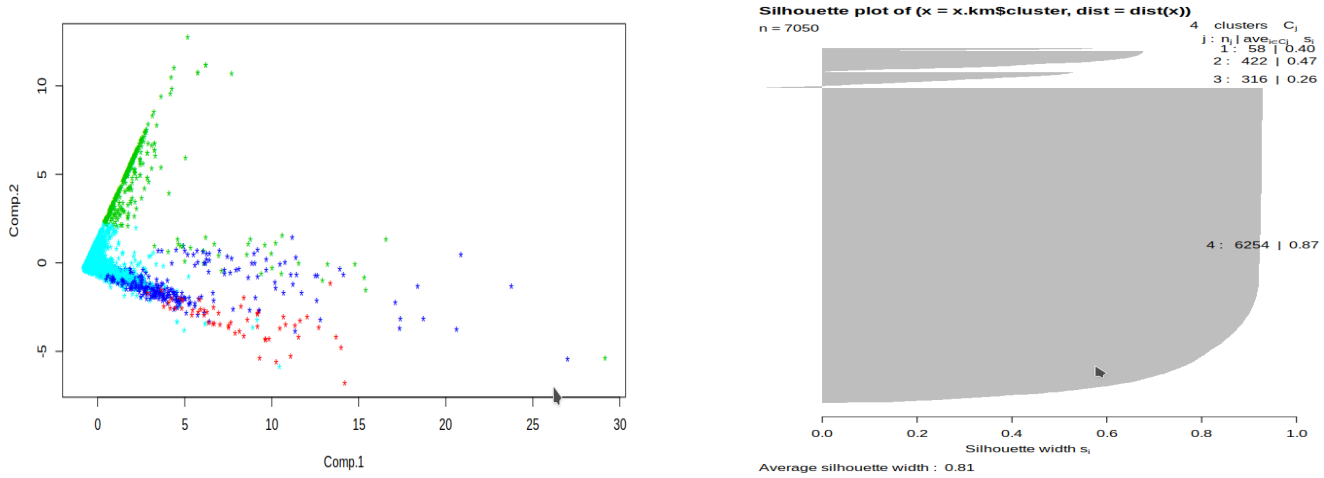

Figura 3: Left: Osservazioni rappresentate sul piano delle copenenti principali. Right: Grafico delle silhouette

Dall'esecuzione dell'algoritmo kmeans che produce 4 cluster, si giunge alla stessa conclusione precedente per quanto riguarda la disomogeneità dei campioni, inoltre la silhouette complessiva non è variata. La differenza con la precedente esecuzione è che la silhouette del cluster più numeroso è aumentata, è presente un terzo cluster minore, con una numerosità davvero bassa di soli 58 osservazioni e la silhouettedi uno dei cluster minori ammonta solamente al 26%. Possiamo concludere che la prima esecuzione dell'algoritmo ha portato risultati migliori, ma vale la pena provare a risolvere il problema con un altro algoritmo di clustering.

3.2 PAM

Un'alternativa all'algoritmo kmeans è l'algoritmo pam, specificando tra i parametri il numero di cluster, in un caso uguale a tre, nell'altro uguale a quattro.

```
pm=pam(x,3);
plot(x.pca$scores,col=pm$cluster,pch=12);
plot(pm);
```

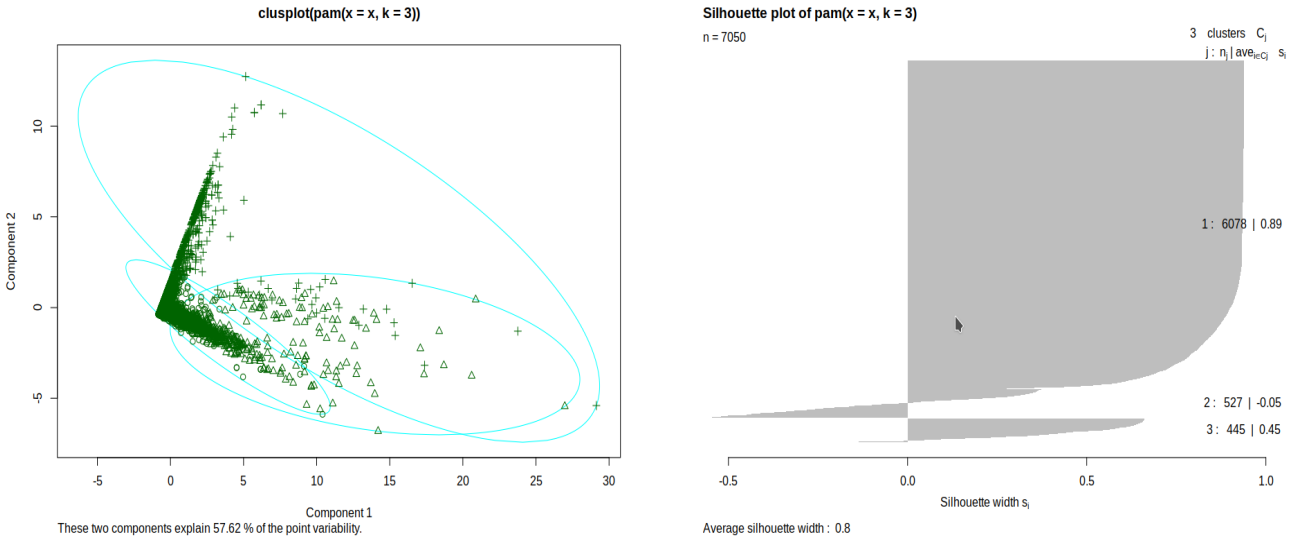

Figura 4: Risultato dell'alg. PAM con 3 cluster

Anche questa esecuzione dell'algoritmo ci conferma l'ipotesi precedente della presenza di un cluster molto numeroso che comprende circa l'85% dell'intera popolazione.
In generale, però, l'algoritmo ha restituito pessimi risultati, nonostante la silhouette complessiva si alta (80%), uno dei cluster minori ha una silhouette negativa, che significa che la somiglianza tra i campioni appartenenti al cluster generato non è alta.

```
pm=pam(x,4);
plot(x.pca$scores,col=pm$cluster,pch=12);
plot(pm);
```

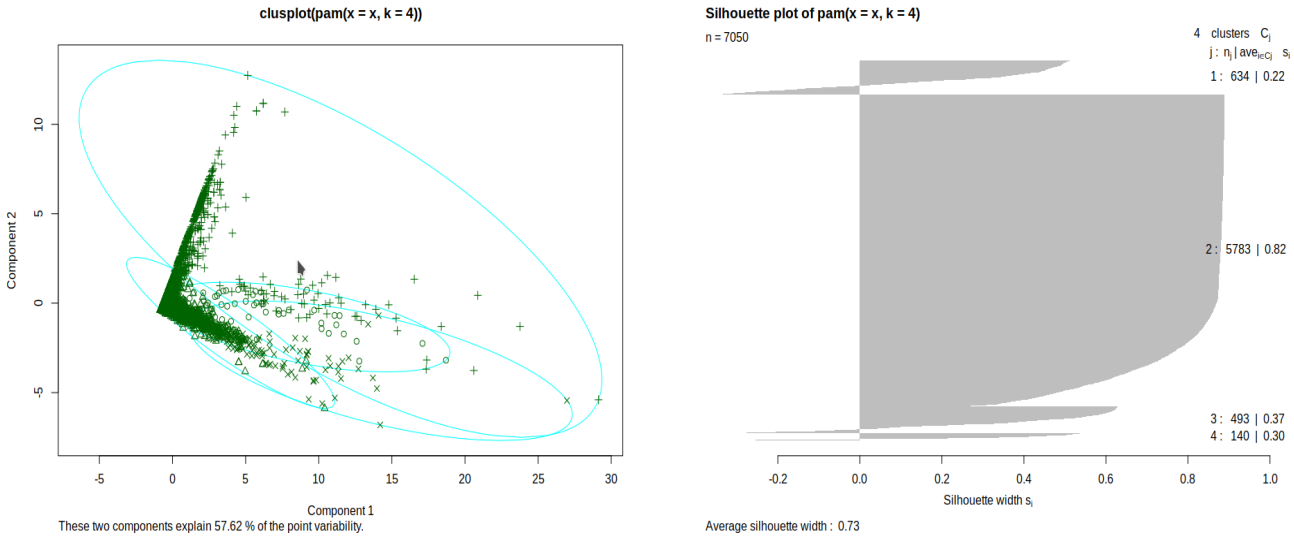

Figura 5: Risultato dell'alg. PAM con 4 cluster

Questa seconda esecuzione dell'algoritmo pam, restitusce 4 cluster, in cui quello più numeroso possiede un numero di campioni un pochino inferiore rispetto alle precedenti esecuzioni, di conseguenza troviamo 3 cluster minori più abbondandti.
La silhouette complessiva è diminuita fino al 73%, inoltre valutando le silhouette dei singoli cluster minori, si notano valori bassi: 30%, 22% e 37%.

4 Interpretazione cluster e conclusioni

L'esecuzione che ha riportato i mogliori risultati è la prima: kmenas, specificando 3 come numero di cluster in output. Per provare a spiegare i cluster generati, calcolo le medie degli attributi tra i campioni appartenenti allo stesso cluster; di seguito vengono riportati solo gli attributi più interessanti.

```
love = rep(0,3); nlove = rep(0,3); react = rep(0,3); nreact = rep(0,3);
ahah = rep(0,3); nahah = rep(0,3); like = rep(0,3); nlike = rep(0,3);
wow = rep(0,3); nwow = rep(0,3); comment = rep(0,3); ncomment = rep(0,3);

for(i in 1:length(x[,1])){
    love[x.km$cluster[i]] = love[x.km$cluster[i]] + x$num_loves[i];
    nlove[x.km$cluster[i]] = nlove[x.km$cluster[i]] + 1;
    react[x.km$cluster[i]] = react[x.km$cluster[i]] + x$num_reactions[i];
    nreact[x.km$cluster[i]] = nreact[x.km$cluster[i]] + 1;
    ahah[x.km$cluster[i]] = ahah[x.km$cluster[i]] + x$num_hahas[i];
    nahah[x.km$cluster[i]] = nahah[x.km$cluster[i]] + 1;
    like[x.km$cluster[i]] = like[x.km$cluster[i]] + x$num_likes[i];
    nlike[x.km$cluster[i]] = nlike[x.km$cluster[i]] + 1;
    wow[x.km$cluster[i]] = wow[x.km$cluster[i]] + x$num_wows[i];
    nwow[x.km$cluster[i]] = nwow[x.km$cluster[i]] + 1;
    comment[x.km$cluster[i]] = comment[x.km$cluster[i]] + x$num_comments[i];
    ncomment[x.km$cluster[i]] = ncomment[x.km$cluster[i]] + 1;
}
```

```
> love[1]/nlove[1];  love[2]/nlove[2];  love[3]/nlove[3];
[1] 8.335511
[1] 95.07299
[1] 47.77479
>
> react[1]/ nreact[1];  react[2]/ nreact[2];  react[3]/nreact[3];
[1] 113
[1] 486.7372
[1] 1713.153
>
> ahah[1]/nahah[1];  ahah[2]/nahah[2];  ahah[3]/nahah[3];
[1] 0.409706
[1] 7.394161
[1] 2.609504
>
> like[1]/nlike[1];  like[2]/nlike[2];  like[3]/nlike[3];
[1] 103.4124
[1] 376.073
[1] 1652.26
>
> wow[1]/nwow[1];  wow[2]/nwow[2];  wow[3]/nwow[3];
[1] 0.5787836
[1] 4.80292
[1] 9.733471
>
> comment[1]/ncomment[1];  comment[2]/ncomment[2];  comment[3]/ncomment[3];
[1] 108.951
[1] 5309.591
[1] 317.8719
```

Figura 6: Valutazione medie attributi

Analizzando le medie dei valori ottenuti per cluster si nota in prima battuta che il cluster più numeroso (indice 1) ha valori sempre più bassi per tutti gli attributi. Inoltre notiamo come ogni post di uno dei cluster minori (indice 3), abbia ricevuto 1652 'mi piace' in media, ma relativamente pochi commenti (317) considerando i mi piace ottenuti.
Mentre i posti dell'altro cluster minore (indice 2) hanno ottenuto meno like, ma sono stati commentati in media circa 5300 volte, inoltre hanno ricevuto in media 95 reazioni 'love', contro il cluster minore di indice 3 che ne ha ricevute circa 47.
Possiamo spiegare i valori ottenuti nel seguente modo:

- Il cluster numeroso (indice 1) identifica i post relativi a vendite ordinarie, come avevamo supposto all'inizio.
- Il fatto che uno dei cluster minori (indice 3) abbia ricevuto tanti mi piace, ma pochi commenti significa che è stato visto da tanti utenti, ma pochi si sono effettiavamente soffermati a leggere, dunque potremmo classificarlo

come post sponsorizzati, ovvero post che sono comparsi maggiormente nella home degli utenti e che hanno lasciato un mi piace che in realtà è poco significativo, perchè non si è interessato.

- Il secondo cluster minore (indice 2) può essere attribuibile a vendite interessanti, ma non sponsorizzate, magari che riguardavano promozioni, perchè hanno ricevuto meno mi piace, ma molte più reazioni 'love', duqnue traspare un forte sentimento positivo. Le vendite dei post, che costituiscono questo cluster, sono stati molto oggetto di discussione nei commenti, in quanto gli utenti hanno voluto ricevere informazioni in più sul prodotto o sulla possibile offerta.

In generale una delle debolezze dei due algoritmi di clustering, utilizzati nell'analisi, è di produrre dei cluster dalla forma sferica; per migliorare i risultati e creare cluster, i cui elementi sono più simili tra loro, può essere opportuno utilizzare algoritmi che non generino cluster dalla forma concava.