Réduction de dimensionnalité et clustering

Module 7

Objectifs

Objectifs

- utiliser les bonnes méthodes de réduction de dimensionnalité
- faire du clustering avec k-means
- projeter des variables catégorielles dans des embeddings

PCA

PCA

- réduire les dimensions tout en conservant la variance
- trouve une nouvelle base aux données et ordonne les axes par variance
- sélectionner les axes dans l'ordre pour conserver le plus de variance

Explication visuelle

http://setosa.io/ev/principal-component-analysis/

Utilisation classique

Réduire le nombre de dimensions en gardant $x\%$ de la variance. x souvent égal à 99.

t-SNE

t-SNE

- approche probabiliste
- map au mieux des distributions de distances dans l'espace haut-dimensionnel et dans l'espace bas-dimensionnel
- très utile pour visualiser des données en très hautes dimensions

t-SNE

- approche probabiliste
- map au mieux des distributions de distances dans l'espace haut-dimensionnel et dans l'espace bas-dimensionnel
- très utile pour visualiser des données en très hautes dimensions

t-SNE

- calcul d'une distribution P des similarités des objets en haute dimension
- calcul d'une distribution Q de similarités des objets en basse dimension
- minimisation de la KL divergence de Q par rapport à P

Sensibilité aux paramètres

Attention, t-SNE est très sensible au choix de ses paramètres :

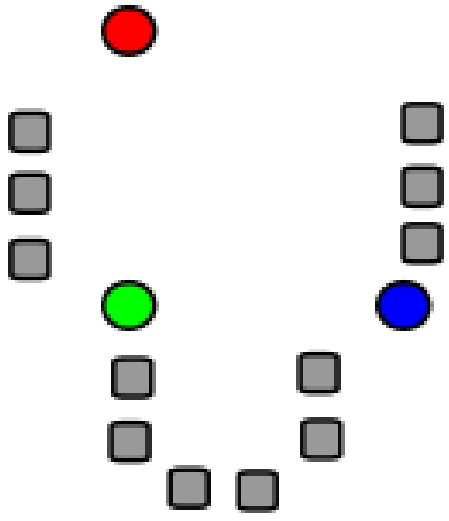
https://distill.pub/2016/misread-tsne/

k-means

k-means

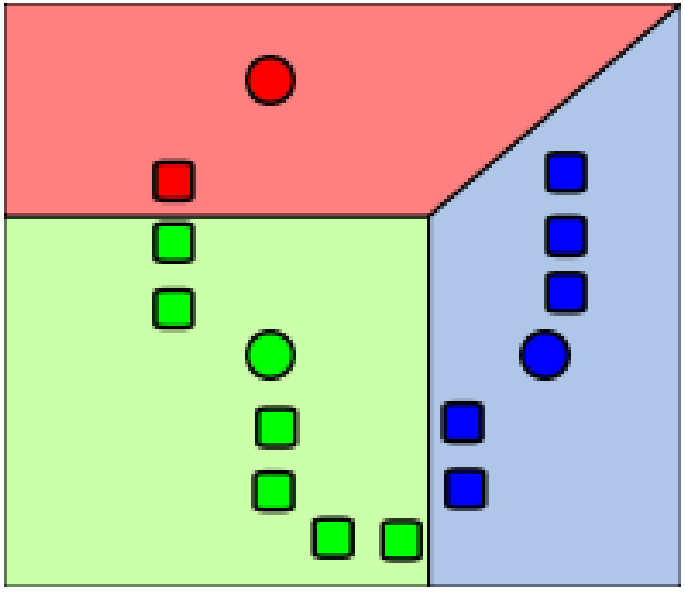
- algo incoutournable en clustering
- sert aussi de base à d'autres algorithmes
- sert aussi de préprocessing

k-means — 1re étape



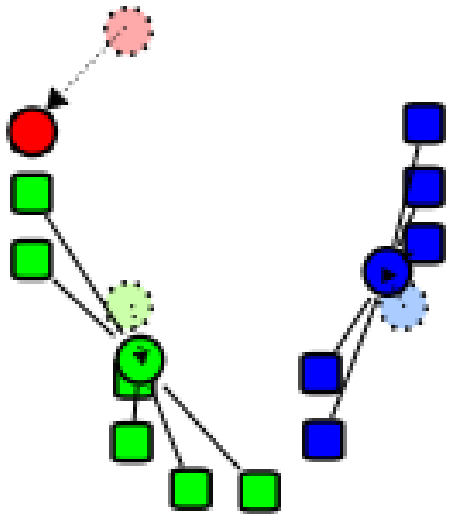
Décider de l'emplacement de k centroïds.

k-means — 2e étape



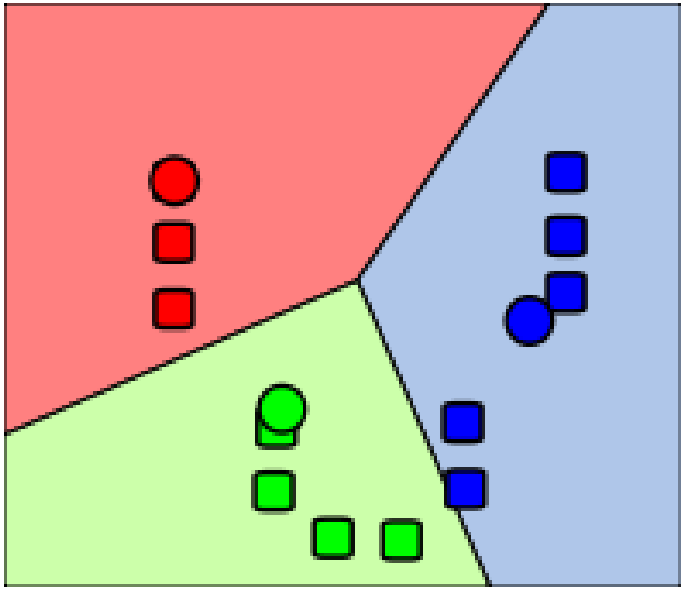
À chaque point, attribuer le cluster le plus proche.

k-means — 3e étape



Ajuster les centroïdes pour qu'ils soient les barycentres des points.

k-means — 4e étape



Répéter les étapes 2 et 3 jusqu'à convergence.

k-means — principaux paramètres

- k : nombre de clusters.
- centroïdes de départ : principalement deux stratégies :
  - hasard
  - points existants
  - k-means++ :
    1. choisir un point dans les données avec une distribution uniforme
    2. calculer une distance des autres points à ce point
    3. utiliser une distribution biaisée par la distance
    4. recommencer 2 et 3 jusqu'à avoir k points

  $\rightarrow$ Choisir par redémarrages multiples la meilleure option
- iterations : moins important

Embeddings

Idée

Projection des catégories de variables catégorielles dans un espace à n dimensions.

Pour les mots, $n \in [\![50, 1000]\!]$

Utilité

- utilisé en surcouche des one-hot encodings dans les réseaux
- initialisés aléatoirement
- entrainés comme le reste du réseau
- converge vers un espace riche

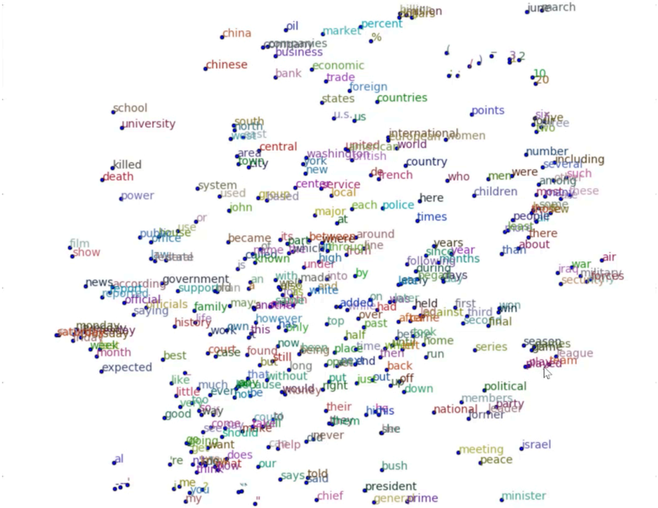
Problèmes de l'encodage one-hot

- très volumineux
- deux mots proches ne partageront rien de commun dans l'input

Embeddings

Pour pallier ces problèmes, projection dans un espace restreint.

Embeddings



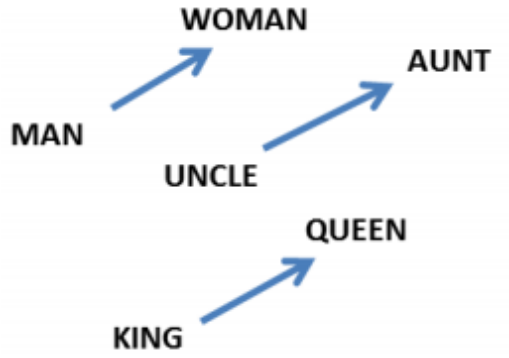
china
oil
market
percent
companies
business
chinese
bank
economic
trade
foreign
states
countries
school
university
points
central
washington
international
world
country
number
including
killed
death
power
system
john
french
who
children
police
here
times
people
film
show
became
around
years
months
about
war
air
news
government
official
family
saying
week
month
expected
best
history
series
political
national
party
good
meeting
peace
israel
bush
president
minister
chief

Embeddings

Espace appris riche et requêtable :

- trouver les symboles les plus proches dans l'espace
- résoudre des analogies : woman - man = aunt - ?

→ Énorme aide pour le réseau



Embeddings

Applicable à toutes les variables catégorielles :

- utilisateurs
- pays
- types de carte SIM
- types d'appels
- ...

Pas limité aux mots !

Conclusion

Conclusion

- méthodes gobale et « plus locale » de réduction de dimensionnalité
- clustering avec k-means, initialisé correctement et cross-validé sur son nombre de clusters
- embeddings pour gérer les variables catégorielles