Introduction au Machine Learning

C5 - Partitionnement et k-moyennes

LAMSADE - Université Paris-Dauphine
lucas.gnecco-heredia@dauphine.psl.eu - C602

* *Remerciement spécial à Florian Yger*

Contexte

Apprentissage non supervisé

Nous avons principalement parlé d'apprentissage supervisé, mais dans l'apprentissage, on retrouve :

- la réduction de dimension
- la détection d'anomalies
- l'estimation de densité
- le partitionnement (clustering)

Partitionnement

Contexte

Dans ce chapitre (et le suivant), on se penche sur un problème d'apprentissage non supervisé.
On dispose de données ($\mathcal{X} = \{x_1, \cdots, x_n\}$ mais d'aucun label et on cherche à en faire émerger de l'information.
Le problème de partitionnement (*clustering* en anglais) consiste à regrouper ensemble les observations se ressemblant.

Problème difficile

- on ne dispose pas de labels (pour vérifier une solution)
- comment définir un *bon* clustering ?

Partitionnement

Contexte

Dans ce chapitre (et le suivant), on se penche sur un problème d'apprentissage non supervisé.
On dispose de données ($\mathcal{X} = \{x_1, \cdots, x_n\}$ mais d'aucun label et on cherche à en faire émerger de l'information.
Le problème de partitionnement (*clustering* en anglais) consiste à regrouper ensemble les observations se ressemblant.

Nombreuses applications

- segmentation de base de clientèle (data exploration)
- résumé pertinent de jeux de données (quantification vectorielle)
  Dans ce cas, une observation sera représentée par son "affinité" avec chaque cluster

Principe général

Résumé

Étant donné un jeu de données, on cherche un partitionnement tel que les données "proches"/"similaires" appartiennent à la même partition/groupe.

Illustration

Principe général

Résumé

Étant donné un jeu de données, on cherche un partitionnement tel que les données "proches" /"similaires" appartiennent à la même partition/groupe.

Illustration

Dans ce chapitre, on va utiliser une approche à base de modèle.

Introduction

Spoiler alert

(en anglais *k-means*)
Aucun lien avec les k-ppv ...

Principe

- Nombre de cluster k fixé a priori
- chaque cluster c est identifié/représenté par la moyenne μ_c des observations qui le composent
- lorsqu'une nouvelle observation arrive, on l'affectera au groupe dont le cluster a la moyenne la plus proche

Introduction

Principe

- Nombre de cluster k fixé a priori
- chaque cluster c est identifié/représenté par la moyenne μ_c des observations qui le composent
- lorsqu'une nouvelle observation arrive, on l'affectera au groupe dont le cluster a la moyenne la plus proche

Comment trouver $\mu_1, \cdots, \mu_k$?

Notations

Pour chaque observation $x_i \in \mathbf{R}^p$, on définit un variable d'affectation

$$r_i = (r_{i1}, \cdots, r_{ip}) \in \mathbf{R}^k$$

avec $r_{ic} = 1$ si x_i est assigné au cluster c (et 0 sinon).
Chaque vecteur d'affectation r_i ne contient qu'un et un seul 1 (tout le reste est nul).
On notera μ l'ensemble des moyennes et r l'ensemble des variables d'affectations.

Problème d'optimisation

On cherche μ et r (les centres des clusters et l'affectation des observations) qui minimisent (en respectant les contraintes sur r)

$$J(\mu, r) = \sum_{i}^{n} \sum_{c=1}^{k} r_{ic} ||x_i - \mu_c||_2^2$$

Problème d'optimisation

On cherche μ et r (les centres des clusters et l'affectation des observations) qui minimisent (en respectant les contraintes sur r)

$$J(\mu, r) = \sum_{i}^{n} \sum_{c=1}^{k} r_{ic} ||x_i - \mu_c||_2^2$$

Cette fonction n'est pas convexe (problème combinatoire) mais on pourrait en trouver un minimum local par optimisation alternée.

Stratégie d'optimisation

Pour un ensemble d'affectation r fixés

On minimise facilement par rapport à μ

$$J(\mu, r) = \sum_{i}^{n} \sum_{c=1}^{k} r_{ic} ||x_i - \mu_c||_2^2$$

$$= \sum_{c=1}^{k} \underbrace{\sum_{i}^{n} r_{ic} ||x_i - \mu_c||_2^2}_{J_c(\mu_c)}$$

Stratégie d'optimisation

Pour un ensemble d'affectation r fixés

On minimise facilement par rapport à μ

$$J(\mu, r) = \sum_{i}^{n} \sum_{c=1}^{k} r_{ic} ||x_i - \mu_c||_2^2$$

$$= \sum_{c=1}^{k} \underbrace{\sum_{i}^{n} r_{ic} ||x_i - \mu_c||_2^2}_{J_c(\mu_c) = \sum_{i \ x_i \in C_c} ||x_i - \mu_c||_2^2}$$

Stratégie d'optimisation

Pour un ensemble d'affectation r fixés

On minimise facilement par rapport à μ

$$J(\mu, r) = \sum_{i}^{n} \sum_{c=1}^{k} r_{ic} ||x_i - \mu_c||_2^2$$

$$= \sum_{c=1}^{k} \underbrace{\sum_{i}^{n} r_{ic} ||x_i - \mu_c||_2^2}_{J_c(\mu_c)}$$

μ_c est la moyenne des poids affectés au cluster c (c.-à-d. tels que $r_{ic} = 1$).

Stratégie d'optimisation (suite)

Pour des moyennes μ fixées

On minimise facilement par rapport à r

$$J(\mu, r) = \sum_{i}^{n} \sum_{c=1}^{k} r_{ic} ||x_i - \mu_c||_2^2$$

Pour chaque x_i, un seul terme parmi les suivants est non-nul :
$r_{i1}||x_i - \mu_c||_2^2, \cdots, r_{ik}||x_i - \mu_k||_2^2$
donc, $r_{ic} = 1$ avec $c = \arg\min_j ||x_i - \mu_j||_2^2$

Stratégie d'optimisation (suite)

Pour des moyennes μ fixées

On minimise facilement par rapport à r

$$J(\mu, r) = \sum_{i}^{n} \sum_{c=1}^{k} r_{ic} ||x_i - \mu_c||_2^2$$

Pour chaque x_i, un seul terme parmi les suivants est non-nul :
$r_{i1}||x_i - \mu_c||_2^2, \cdots, r_{ik}||x_i - \mu_k||_2^2$
donc, $r_{ic} = 1$ avec $c = \arg\min_j ||x_i - \mu_j||_2^2$
On affecte x_i au cluster c dont la distance $||x_i - \mu_c||_2^2$ est minimum (c.-à-d. dont la moyenne est la plus proche de x_i).

Stratégie d'optimisation (suite)

Pour des moyennes μ fixées

On minimise facilement par rapport à r

$$J(\mu, r) = \sum_{i}^{n} \sum_{c=1}^{k} r_{ic} ||x_i - \mu_c||_2^2$$

Pour chaque x_i, un seul terme parmi les suivants est non-nul :
$r_{i1}||x_i - \mu_c||_2^2, \cdots, r_{ik}||x_i - \mu_k||_2^2$
donc, $r_{ic} = 1$ avec $c = \arg\min_j ||x_i - \mu_j||_2^2$

En pratique, on va alterner ces deux étapes.

Algorithme

Algorithm 1 k-moyennes (Algo de Lloyd)

Require: k, μ (initialisation aléatoire, par ex. k points du dataset)

while not conv **do**

for i from 1 to n **do**

$r_{ic} = 1$ avec $c = \arg\min_j ||x_i - \mu_j||_2^2$

$r_{icj} = 0$ avec $\forall j \neq c$

end for

for c from 1 to k **do**

$\mu_c = \frac{1}{n_c} \sum_{i/r_{ic}=1} x_i$ (avec $n_c = \sum_{i=1}^{n} r_{ic}$)

end for

end while

Remarques

- Cet algorithme converge (car à chaque étape le coût ne peut que diminuer) vers une solution localement optimale
- En pratique, la convergence est atteinte quand les moyennes μ n'évoluent plus d'une itération à l'autre
- Une autre initialisation pourra potentiellement donner une tout autre solution
- Cet algorithme est une variante de l'algorithme EM (Expectation-Maximization utilisé pour résoudre certains problèmes de maximisation de vraisemblance)
  - étape E - affectation de chaque observation à un cluster
  - étape M - mise à jour de la moyenne des clusters
- Les clusters formés par cette approche sont toujours convexes.

Illustration - pas à pas

That's all folks

Limites et perspectives

- le résultat de l'algorithme dépend du choix de k

Illustration

- sensibilité aux valeurs aberrantes (des données très éloignées pourraient potentiellement se trouver dans son propre cluster)
- problème non convexe et minima locaux
  mais pour résoudre ce problème, on peut lancer plusieurs fois l'algorithme (à partir d'initialisations différentes) et garder le meilleur résultat

Perspectives

- extension possible à d'autres géométries (en changeant la distance utilisée et en adaptant le calcul de moyenne)

Application numérique - TD

On dispose des données bivarées suivantes :

	x1	x2
1	-1	0
2	-2	0
3	-1	1
4	-2	1
5	1	0
6	2	0
7	1	-1
8	2	-1

- Représenter le jeu de données
- Appliquer l'algorithme des k-moyennes (avec $k = 2$ et en initialisant $\mu_1 = x_2$ et $\mu_2 = x_4$) en explicitant chaque étape jusqu'à convergence
- Représenter sur le premier graphique les différentes étapes de l'algorithme

Introduction

(en anglais *Gaussian Mixture Model - GMM*)
On considère un modèle génératif pour les données, en supposant qu'il existe k clusters, chacune d'eux ayant une densité de probabilité particulière.

Principe

- Nombre de clusters k fixé a priori
- chaque cluster c est identifié/représenté par sa moyenne μ_c et sa covariance Σ_c

Introduction

On considère un modèle génératif pour les données, en supposant qu'il existe k clusters, chacune d'eux ayant une densité de probabilité particulière.

Principe

- Nombre de clusters k fixé a priori
- chaque cluster c est identifié/représenté par sa moyenne μ_c et sa covariance Σ_c

Comment trouver $\mu_1, \cdots, \mu_k$ *et* $\Sigma_1, \cdots, \Sigma_k$?

Modèle

On fait l'hypothèse, que selon le modèle génératif, chaque observation a été générée de la façon suivante :

1. choix du cluster d'appartenance $z \in \{1, \cdots, k\}$
   $Z \sim$ multinomiale$(\pi_1, \cdots, \pi_k)$ (généralisation de la loi binomiale pour plusieurs catégories)
2. génération de l'observation en elle-même
   $X|Z = z \sim p(x|z)$ (en faisant le choix d'une distribution gaussienne $\mathcal{N}(\mu_k, \Sigma_k)$)

Cela revient à considérer

$$P(X, Z) = \underbrace{P(X|Z)}_{\mathcal{N}(\mu_k, \Sigma_k)} \underbrace{P(Z)}_{\pi_k}$$

Quelques remarques

Dans les faits

- on observe seulement X (et jamais (X, Z)) (on est dans un cadre non supervisé)
- Z joue le rôle d'une variable d'affectation probabiliste

Le modèle de mélange de gaussiennes et un modèle à variables latentes (les variables d'affectations Z)

Variables d'affectation

Lorsqu'on observe $X = x$, alors d'après le modèle, $p(Z = z|X = x) = \frac{p(x,z)}{p(x)}$ On parle d'affectation souple (*soft assignement*)

Variables d'affectation

Lorsqu'on observe $X = x$, alors d'après le modèle,
$p(Z = z|X = x) = \frac{p(x,z)}{p(x)}$ On parle d'affectation souple (*soft assignement*)
Une affectation stricte (*hard assignement*) serait
$z^\star = \arg\max_l p(Z = l|X = x)$

Variables d'affectation

Lorsqu'on observe $X = x$, alors d'après le modèle,
$p(Z = z|X = x) = \frac{p(x,z)}{p(x)}$ On parle d'affectation souple (*soft assignement*)

Si on connaît le modèle, l'affectation (et donc le partitionnement) est triviale.
On va donc chercher les paramètres du modèle à partir des données.

Modèle

Paramètres

On fait donc l'hypothèse que les données ont étés échantillonnées à partir d'un mélange de gaussiennes. de paramètres :

- probabilité (a priori) d'appartenance à un cluster : $\pi = (\pi_1, \cdots, \pi_k)$
- moyennes des clusters : $\mu = (\mu_1, \cdots, \mu_k)$
- covariances des clusters : $\Sigma = (\Sigma_1, \cdots, \Sigma_k)$

Pour chaque observation x, on a :

$$p(x) = \sum_{z=1}^{k} p(x, z) = \sum_{z=1}^{k} \pi_z \mathcal{N}(x|\mu_z, \Sigma_z)$$

Vraisemblance

Pour un ensemble d'observations (i.i.d.), on peut écrire la vraisemblance du modèle :

$$\begin{aligned}\mathcal{L}(\pi, \mu, \Sigma) &= \prod_{i=1}^{n} p(x_i) \\ &= \prod_{i=1}^{n} \sum_{z=1}^{k} \pi_z \mathcal{N}(x_i | \mu_z, \Sigma_z)\end{aligned}$$

ou son logarithme :

$$\begin{aligned}J(\pi, \mu, \Sigma) &= \log(\mathcal{L}(\pi, \mu, \Sigma)) \\ &= \sum_{i=1}^{n} \log \left(\sum_{z=1}^{k} \pi_z \mathcal{N}(x_i | \mu_z, \Sigma_z) \right)\end{aligned}$$

Estimation du maximum de vraisemblance

Remarque

Ici encore, pas d'expression analytique de la solution (et même pire, pas d'unicité de la solution) mais un algorithme pour trouver une solution.

Stratégie d'optimisation

Si on connaissait l'affectation de chaque observation aux clusters, on pourrait facilement estimer les paramètres des clusters :

$$\mu_{MV} = \frac{1}{n} \sum_i x_i$$

$$\Sigma_{MV} = \frac{1}{n} \sum_i (x_i - \mu_{MV})(x_i - \mu_{MV})^\top$$

Stratégie d'optimisation (suite)

On note γ_i^j, la responsabilité que le cluster j a dans la génération de l'observation x_i :

$$\begin{aligned}\gamma_i^j &= p(Z = j | X = x_j) \\ &= \frac{P(Z = j, X = x_i)}{P(X = x_i)} \\ &= \frac{\pi_j \mathcal{N}(x_i | \mu_j, \Sigma_j)}{\sum_{c=1}^{k} \pi_j \mathcal{N}(x_i | \mu_j, \Sigma_j)}\end{aligned}$$

Commentaires sur l'optimisation

Remarque

- Le vecteur $\gamma_i = (\gamma_i^1, \cdots, \gamma_i^k)$ est une affectation souple de x_i aux clusters (que l'on peut interpréter comme une probabilité d'appartenance)
- x_i "appartient" à plusieurs clusters à proportions différentes
- Le "nombre de points affectés à c" sera donc noté :
  $n_c = \sum_{i=1}^{n} \gamma_i^c$

Algorithme

Algorithm 2 Mélange de gaussiennes

Require: k, μ, Σ, π (initialisation)

while not conv **do**

Etape (E)

for i from 1 to n **do**

for j from 1 to k **do**

evaluation de la responsabilité de chaque cluster pour chaque observation

$$\gamma_i^j = \frac{\pi_j \mathcal{N}(x_i|\mu_j,\Sigma_j)}{\sum_{c=1}^{k} \pi_c \mathcal{N}(x_i|\mu_c,\Sigma_c)}$$

end for

end for

Etape (M) (à suivre)

end while

Algorithme

Algorithm 3 Mélange de gaussiennes

Require: k, μ, Σ, π (initialisation)

while not conv **do**

Etape (E)

Etape (M)

for c from 1 to k **do**

estimation des paramètres (avec les responsabilités)

$$\mu_c = \frac{1}{n_c} \sum_{i=1}^{n} \gamma_i^c x_i$$

$$\Sigma_c = \frac{1}{n_c} \sum_{i=1}^{n} \gamma_i^c (x_i - \mu_c)(x_i - \mu_c)^\top$$

$$\pi_c = \frac{n_c}{n}$$

end for

end while

Algorithme

Algorithm 4 Mélange de gaussiennes

Require: k, μ, Σ, π (initialisation)

while not conv **do**

Etape (E)

Etape (M)

for c from 1 to k **do**

calculs de moyennes et covariance pondérées

$$\mu_c = \frac{1}{n_c} \sum_{i=1}^{n} \gamma_i^c x_i$$

$$\Sigma_c = \frac{1}{n_c} \sum_{i=1}^{n} \gamma_i^c (x_i - \mu_c)(x_i - \mu_c)^\top$$

$$\pi_c = \frac{n_c}{n}$$

end for

end while

Lien entre k-moyennes et mélanges de gaussiennes

Remarque

Le problème des k-moyennes est un problème de mélange de gaussiennes où :

- la covariance de chaque cluster est fixée à $\sigma \mathbf{I}_p$
- une affectation stricte est utilisée

Limites

- le mélange de gaussiennes souffre des mêmes limites que les k-moyennes (dépendance au choix k, sensibilité à l'initialisation et aux valeurs aberrantes)

Conclusion globale

- le clustering permet de résumer un jeu de données / regrouper des observations similaires,
- ce type d'approches étant non supervisée, sa validation est difficile, mais il existe des métriques pour comparer des clustering
  - stabilité des clusters (notamment pour choisir l'hyperparamètre du modèle)
  - des indices de séparabilité des clusters, d'homogénéité des clusters ...
- pour les deux approches considérées (k-moyennes et sa généralisation, le mélange de gaussiennes) le choix de k est critique