Apprentissage automatique (Machine learning)

Méthodes non supervisées

Visseho Adjiwanou, PhD.

15 June 2024

1. Introduction

Comme mentionné précédemment, les méthodes d'apprentissage non supervisées sont utilisées lorsque nous n'avons pas de **variable cible** à estimer ou à prédire, mais que nous voulons comprendre les clusters, les groupes ou les motifs dans les données. Ces méthodes sont souvent utilisées pour l'exploration des données, comme dans les exemples suivants :

1. Lorsqu'on est confronté à un grand corpus de données textuelles, par exemple des archives d'e-mails, des projets de loi du congrès, des discours ou des réponses libres à des sondages, les méthodes d'apprentissage non supervisées sont souvent utilisées pour comprendre et saisir les motifs dans nos données.

2. Étant donné un ensemble de données sur les étudiants et leur comportement au fil du temps (performance académique, notes, résultats aux tests, présence, etc.), on pourrait vouloir comprendre les comportements typiques ainsi que les trajectoires de ces comportements au fil du temps. Les méthodes d'apprentissage non supervisées (clustering) peuvent être appliquées à ces données pour obtenir des "segments" d'étudiants ayant un comportement similaire.

3. Étant donné un ensemble de données sur les publications ou les brevets dans différents domaines, nous pouvons utiliser des méthodes d'apprentissage non supervisées (règles d'association) pour déterminer quelles disciplines ont le plus de collaborations et quelles sont les domaines où les chercheurs ont tendance à publier dans différents domaines.

4. Étant donné un groupe de personnes à haut risque de récidive, le clustering peut être utilisé pour comprendre les différents groupes de personnes au sein de l'ensemble à haut risque, afin de déterminer les programmes d'intervention qui pourraient être nécessaires à créer.

2. Clustering

Le clustering est la technique d'apprentissage non supervisé la plus courante et est utilisée pour regrouper ensemble des points de données qui sont similaires les uns aux autres. L'objectif des méthodes de clustering est de produire une similarité intra-cluster (à l'intérieur) élevée et une similarité inter-cluster (entre clusters) faible.

Les algorithmes de clustering nécessitent généralement une métrique de distance (ou de similarité)[1] pour générer des clusters.

Ils prennent un ensemble de données, une métrique de distance (et parfois des paramètres supplémentaires), et ils génèrent des clusters en fonction de cette métrique de distance. La métrique de distance la plus couramment utilisée est la distance euclidienne, mais d'autres métriques couramment utilisées sont la distance de Manhattan, Minkowski, Chebyshev, cosinus, Hamming, Pearson et Mahalanobis. Souvent, des métriques de similarité spécifiques au domaine peuvent être conçues pour être utilisées dans des problèmes spécifiques. Par exemple, lors de l'exécution des tâches de liaison d'enregistrements , vous pouvez concevoir une métrique de similarité qui compare deux prénoms et leur attribue une similarité élevée (faible distance) s'ils sont tous les deux mappés vers le même nom canonique, de sorte que, par exemple, Sammy et Sam soient mappés vers Samuel.

La plupart des algorithmes de clustering exigent également que l'utilisateur spécifie le **nombre de clusters** (ou un autre paramètre qui détermine indirectement le nombre de clusters) à l'avance en tant que paramètre. Cela est souvent difficile à faire a priori et rend généralement le clustering une tâche itérative et interactive. Un autre aspect du clustering qui le rend interactif est souvent la **difficulté d'évaluer automatiquement la qualité des clusters**.

Bien que diverses métriques analytiques de clustering aient été développées, le meilleur clustering dépend de la tâche et doit donc être évalué par l'utilisateur. Il peut y avoir différents regroupements qui peuvent être générés avec les mêmes données. Vous pouvez imaginer regrouper des histoires similaires de nouvelles en fonction du contenu du sujet, du style d'écriture ou du sentiment. Le bon ensemble de clusters dépend de l'utilisateur et de la tâche qu'il a à accomplir. Le clustering est donc typiquement utilisé pour **explorer les données**, **générer des clusters**, **explorer les clusters**, puis relancer la méthode de clustering avec différents paramètres ou modifier les clusters (en les divisant ou en les fusionnant avec l'ensemble précédent de clusters).

Interpréter un cluster peut être non trivial : vous pouvez regarder le centroïde d'un cluster, examiner les distributions de fréquence des différentes caractéristiques (et les comparer à la distribution antérieure de chaque caractéristique), ou vous pouvez construire un arbre de décision où la **variable cible est l'ID du cluster** qui peut décrire le cluster en utilisant les caractéristiques de vos données. Un bon exemple d'outil permettant le clustering interactif à partir de données textuelles est Ontogen (**Ontogen?**).

[1] Les métriques de distance sont des formules mathématiques pour calculer la distance entre deux objets. Par exemple, la distance de Manhattan est la distance parcourue par une voiture d'un endroit à un autre dans un système de rues basé sur une grille, tandis que la distance euclidienne (en deux dimensions) est la distance "en ligne droite" entre deux points.

2.1. Clustering par k-means

L'algorithme de clustering le plus couramment utilisé s'appelle k-means, où k définit le nombre de clusters. L'algorithme fonctionne comme suit :

1. Sélectionner k (le nombre de clusters que vous souhaitez générer).
2. Initialiser en sélectionnant k points comme centroïdes des k clusters. Cela est généralement fait en sélectionnant k points de manière uniforme et aléatoire.
3. Assigner à chaque point un cluster en fonction du centroïde le plus proche.
4. Recalculer les centroïdes des clusters en fonction de l'assignation à l'étape (3) comme la moyenne de tous les points de données appartenant à ce cluster.
5. Répéter les étapes (3) et (4) jusqu'à convergence.

L'algorithme s'arrête lorsque les assignations ne changent pas d'une itération à l'autre (Figure (**ref?**)(fig)). Cependant, **les ensembles finaux de clusters dépendent des points de départ**. S'ils sont initialisés différemment, il est possible d'obtenir des clusters différents. Une astuce pratique courante est d'exécuter k-means plusieurs fois, chacune avec différents points de départ (aléatoires). L'algorithme k-means est rapide, simple et facile à utiliser, et c'est souvent le premier algorithme de clustering à essayer pour voir s'il convient à vos besoins. Lorsque les données sont de forme où la moyenne des points de données ne peut pas être calculée, une méthode apparentée appelée K-medoids peut être utilisée (**park2009simple?**).

2.2. L'algorithme espérance-maximisation (EM) dans le contexte du clustering

Vous êtes peut-être familier avec l'algorithme EM dans le contexte de l'imputation des données manquantes. EM est une approche générale pour maximiser la vraisemblance en présence de données incomplètes. Cependant, il est également utilisé comme méthode de clustering où les données manquantes sont les clusters auxquels un point de données appartient. Contrairement à k-means, où chaque point de données est assigné à un seul cluster, EM effectue une assignation douce où chaque point de données reçoit une assignation probabiliste à divers clusters. L'algorithme EM itère jusqu'à ce que les estimations convergent vers une solution (localement) optimale.

L'algorithme EM est assez efficace pour **traiter les valeurs aberrantes** ainsi que les données de haute dimension, comparé à k-means. Cependant, il présente quelques limitations. Tout d'abord, il ne fonctionne pas bien avec un **grand nombre de clusters** ou lorsque certains **clusters contiennent peu d'exemples**. De plus, lorsque la valeur de k est plus grande que le nombre réel de clusters dans les données, EM peut ne pas donner des résultats raisonnables.

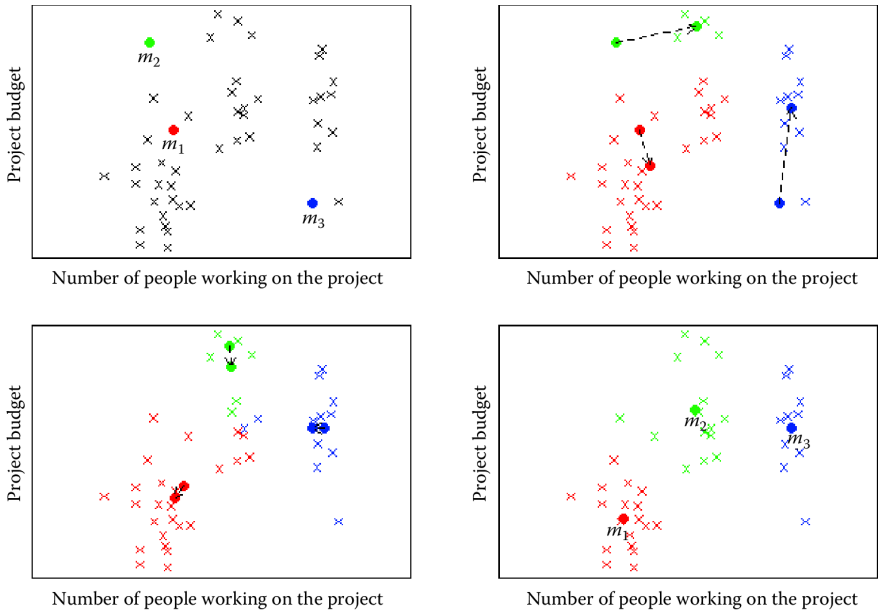

Figure 1: Exemple de clustering par k-means avec $k = 3$. Le panneau en haut à gauche montre la distribution des données et les trois points de départ m_1, m_2, m_3 placés au hasard. En haut à droite, on voit ce qui se passe lors de la première itération. Les moyennes des clusters se déplacent vers des positions plus centrales dans leurs clusters respectifs. Le panneau en bas à gauche montre la deuxième itération. Après six itérations, les moyennes des clusters ont convergé vers leurs destinations finales et le résultat est montré dans le panneau en bas à droite

2.3. Le clustering par Mean Shift

Le clustering par Mean Shift fonctionne en identifiant des régions denses dans les données en définissant une fenêtre autour de chaque point de données et en calculant la moyenne des points de données dans cette fenêtre. Ensuite, il déplace le centre de la fenêtre vers la moyenne et répète l'algorithme jusqu'à convergence. Après chaque itération, la fenêtre se déplace vers une région plus dense de l'ensemble de données. Voici le déroulement de l'algorithme :

1. Fixer une fenêtre autour de chaque point de données (basée sur le paramètre de bande passante qui définit la taille de la fenêtre).
2. Calculer la moyenne des données à l'intérieur de la fenêtre.
3. Déplacer la fenêtre vers la moyenne et répéter jusqu'à convergence.

Mean Shift nécessite un paramètre de bande passante h à régler, qui influence le taux de convergence et le nombre de clusters. Une grande valeur de h peut fusionner des clusters distincts, tandis qu'une petite valeur de h peut conduire à trop de clusters. Mean Shift peut ne pas bien fonctionner dans des dimensions plus élevées car le nombre de maxima locaux est élevé et il peut converger rapidement vers un optimum local.

Une des différences les plus importantes entre Mean Shift et k-means est que k-means fait deux hypothèses générales : le nombre de clusters est déjà connu et les clusters ont une forme sphérique (ou elliptique). Mean Shift ne suppose rien sur le nombre de clusters (mais la valeur de h le détermine indirectement). De plus, Mean Shift peut gérer des clusters de forme arbitraire.

L'algorithme k-means est également sensible aux initialisations, tandis que Mean Shift est assez robuste aux initialisations. Typiquement, Mean Shift est exécuté pour chaque point, ou parfois les points sont sélectionnés de manière uniformément aléatoire. De même, k-means est sensible aux valeurs aberrantes, tandis que Mean Shift l'est moins. Cependant, les avantages de Mean Shift viennent avec un coût : la vitesse. La procédure de k-means est rapide, tandis que Mean Shift est computationnellement lent mais peut être facilement parallélisé.

2.4. Hierarchical clustering

Les méthodes de regroupement que nous avons vues jusqu'à présent, souvent appelées méthodes de partitionnement, produisent un ensemble plat de clusters sans hiérarchie. Parfois, nous voulons générer une hiérarchie de clusters, et les méthodes qui peuvent le faire sont de deux types :

1. Agglomératif (ascendant) : Commencez avec chaque point comme son propre cluster et fusionnez itérativement les clusters les plus proches. Les itérations s'arrêtent soit lorsque

les clusters sont trop éloignés pour être fusionnés (selon un critère de distance prédéfini), soit lorsqu'il y a un nombre suffisant de clusters (selon un seuil prédéfini).

2. Divisif (descendant) : Commencez avec un seul cluster et créez des divisions de manière récursive.

Typiquement, le regroupement agglomératif est plus souvent utilisé que le regroupement divisif. Une raison en est qu'il est significativement plus rapide, bien que les deux soient généralement plus lents que les méthodes de partition directe telles que k-means et EM. Un autre désavantage de ces méthodes est qu'elles sont gourmandes : un point de données qui est incorrectement assigné au cluster "incorrect" lors d'une division ou fusion antérieure ne peut pas être réassigné plus tard.

2.5. Spectral clustering

La figure (**ref?**)(fig) montre les clusters que k-means générerait sur l'ensemble de données figurant dans la figure. Il est évident que les clusters produits ne sont pas ceux que vous voudriez, et c'est l'un des inconvénients des méthodes telles que k-means. Deux points éloignés l'un de l'autre seront placés dans des clusters différents même s'il existe d'autres points de données créant un "chemin" entre eux. La méthode de clustering spectral corrige ce problème en regroupant des données qui sont connectées mais pas nécessairement compactes ou regroupées dans des frontières convexes. Les méthodes de clustering spectral fonctionnent en représentant les données sous forme de graphe (ou réseau), où les points de données sont des nœuds dans le graphe et les arêtes (connexions entre les nœuds) représentent la similarité entre les deux points de données.

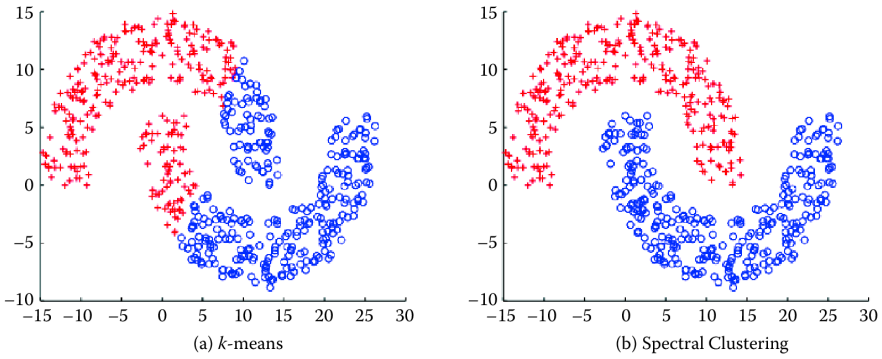

Figure 2: Le même ensemble de données peut produire des clusters radicalement différents : (a) k-means ; (b) clustering spectral

L'algorithme fonctionne comme suit :

1. Calculer une matrice de similarité à partir des données. Cela implique de déterminer une fonction de distance par paire (en utilisant l'une des fonctions de distance que nous avons décrites précédemment).

2. Avec cette matrice, nous pouvons maintenant effectuer un partitionnement de graphe, où les composantes de graphe connectées sont interprétées comme des clusters. Le graphe doit être partitionné de manière à ce que les arêtes connectant différents clusters aient des poids faibles et les arêtes à l'intérieur du même cluster aient des valeurs élevées.

3. Nous pouvons maintenant partitionner ces données représentées par la matrice de similarité de différentes manières. Une méthode courante est d'utiliser la méthode des coupes normalisées. Une autre façon est de calculer un Laplacien de graphe à partir de la matrice de similarité.

4. Calculer les vecteurs propres et les valeurs propres du Laplacien.

5. Les k vecteurs propres sont utilisés comme données de substitution pour l'ensemble de données d'origine, et ils sont entrés dans le clustering k-means pour produire des affectations de cluster pour chaque point de données d'origine.

Le clustering spectral est en général beaucoup meilleur que k-means en termes de performance de clustering, mais beaucoup plus lent à exécuter en pratique. Pour les problèmes à grande échelle, k-means est un algorithme de clustering préféré en raison de son efficacité et de sa rapidité.

2.6. Analyse en composantes principales

L'analyse en composantes principales est une autre méthode non supervisée utilisée pour trouver des motifs et des structures dans les données. Contrairement aux méthodes de regroupement, la sortie n'est pas un ensemble de clusters mais un ensemble de composantes principales qui sont des combinaisons linéaires des variables d'origine. PCA est généralement utilisée lorsque vous avez un grand nombre de variables et que vous souhaitez en réduire le nombre pour pouvoir les analyser. Cette approche est souvent appelée réduction de dimension. Elle génère des dimensions linéairement non corrélées qui peuvent être utilisées pour comprendre la structure sous-jacente des données. En termes mathématiques, étant donné un ensemble de données en n dimensions, PCA vise à trouver un sous-espace linéaire de dimension d inférieure à n tel que les points de données se trouvent principalement sur ce sous-espace linéaire.

PCA est liée à plusieurs autres méthodes que vous connaissez peut-être déjà. L'échelle multidimensionnelle, l'analyse factorielle et l'analyse en composantes indépendantes diffèrent de PCA par les hypothèses qu'elles font, mais elles sont souvent utilisées à des fins similaires de réduction de dimension et de découverte de la structure sous-jacente dans un ensemble de données.

2.7. Règles d'association

Les règles d'association sont une méthode d'analyse différente qui provient de la communauté de l'exploration de données et des bases de données, principalement axée sur la

recherche d'associations fréquentes entre une collection d'items. Cette méthode est parfois appelée "analyse du panier de marché", car c'était le domaine d'application initial des règles d'association. L'objectif est de trouver des associations d'items qui se produisent ensemble plus fréquemment que ce à quoi on s'attendrait au hasard. L'exemple classique (probablement un mythe) est "les hommes qui vont au magasin pour acheter des couches ont tendance à acheter aussi de la bière en même temps". Ce type d'analyse serait effectué en appliquant les règles d'association à un ensemble de données d'achats de supermarché. Pour les scientifiques sociaux, cette méthode peut être utilisée sur des données contenant les services sociaux que des individus ont reçus dans le passé afin de déterminer quels types de services "co-occurrent" chez les personnes et d'offrir ces services de manière proactive aux personnes dans le besoin.

Les règles d'association prennent la forme $X_1, X_2, X_3 \Rightarrow Y$ avec un support S et une confiance C, ce qui signifie que lorsque une transaction contient les items X_1, X_2, X_3 $C\%$ du temps, elle contient également l'item Y et il y a au moins $S\%$ des transactions où l'antécédent est vrai. Cela est utile dans les cas où l'on veut trouver des motifs à la fois fréquents et statistiquement significatifs, en spécifiant des seuils pour le support S et la confiance C.

Le support et la confiance sont des métriques utiles pour générer des règles mais souvent pas suffisantes. Une autre métrique importante utilisée pour générer des règles (ou réduire le nombre de motifs faux générés) est le lift. Le lift est simplement estimé par le rapport de la probabilité conjointe de deux items, x et y, au produit de leurs probabilités individuelles : $P(x, y)/[P(x)P(y)]$. Si les deux items sont statistiquement indépendants, alors $P(x, y) = P(x)P(y)$, correspondant à un lift de 1. Notez que l'anticorrélation produit des valeurs de lift inférieures à 1, ce qui est également un motif intéressant, correspondant à des items mutuellement exclusifs qui se produisent rarement ensemble.

Les algorithmes de règles d'association fonctionnent comme suit : étant donné un ensemble de transactions (lignes) et d'items pour cette transaction :

1. Trouver toutes les combinaisons d'items dans un ensemble de transactions qui se produisent avec une fréquence minimale spécifiée. Ces combinaisons sont appelées ensembles d'items fréquents.

2. Générer des règles d'association qui expriment la co-occurrence des items au sein des ensembles d'items fréquents.

Pour nos besoins, les méthodes de règles d'association sont un moyen efficace de prendre un panier de caractéristiques (par exemple, les domaines de publication d'un chercheur, différentes organisations où un individu a travaillé au cours de sa carrière, toutes les villes ou quartiers où quelqu'un a pu vivre) et de trouver des motifs de co-occurrence. Cela peut sembler trivial, mais à mesure que les ensembles de données et le nombre de caractéristiques augmentent, cela devient coûteux sur le plan computationnel, et les algorithmes de fouille de règles d'association offrent un moyen rapide et efficace de le faire.