Machine Learning – TP4

Machine Learning - TP4 Analyse en Composantes Principales (PCA), Régression, K-Means

Analyse Composantes Principales

L'Analyse Composantes Principales (PCA) est un outil très important utilisé pour l'extraction de caractéristiques, la compression de données et la visualisation. En principe, ce que nous réalisons avec la PCA est de trouver des motifs qui gouvernent les caractéristiques des données et d'utiliser ces motifs pour créer un nouveau système de coordonnées potentiellement plus représentatif pour nos données. Elle est basée sur des mesures d'analyse statistique des caractéristiques, à savoir la variance et la covariance des caractéristiques. Avec la variance, nous pouvons trouver quelles caractéristiques portent le plus d'informations et avec la covariance, nous pouvons trouver dans quelle mesure les caractéristiques sont indépendantes les unes des autres.

En bref, la PCA trouve le nouveau vecteur de caractéristiques pour les données, puis transforme les données vers ce nouveau vecteur de caractéristiques (composé de caractéristiques orthogonales). La transformation ici signifie que le vecteur de caractéristiques initial est "étiré" et tourné pour correspondre aux nouvelles caractéristiques. En classe, nous avons vu une manière dont la PCA peut être effectuée : en calculant les vecteurs propres et les valeurs propres de la matrice de covariance d'origine, puis en calculant le produit des vecteurs propres avec les données d'origine. Si nous souhaitons effectuer une compression ou une sélection de caractéristiques, nous pouvons omettre certains des vecteurs propres récupérés, ceux qui ont de faibles valeurs propres. Ensuite, nous pouvons également facilement récupérer les données initiales (ou une approximation, si nous n'avons pas conservé tous les vecteurs propres).

Exercice 1

Dans cet exercice, vous êtes invité à implémenter l'algorithme PCA comme décrit dans les diapositives du cours. Les étapes indicatives sont décrites ci-dessous.

Étape 1 : Chargement des données

Nous allons travailler sur le jeu de données Iris, fourni dans les ensembles de données de sklearn. Le jeu de données Iris se compose de 3 types différents d'iris (target) et de la longueur des pétales et des sépales (feature vector). Pour le moment, nous nous intéressons uniquement au vecteur de caractéristiques. Chargez le jeu de données et affichez la forme (nombre de lignes et de colonnes) du vecteur de caractéristiques.

Note : Il est recommandé (mais non obligatoire) de ne conserver que les données correspondant aux classes 1 et 2, pour lesquelles les caractéristiques sont fortement corrélées.

Étape 2 : Préparation des données

Pour calculer les vecteurs propres et les valeurs propres, il est impératif que nous centrions les caractéristiques à zéro (moyenne nulle) et les transformions en variance unitaire. (REMARQUE : dans l'exemple des diapositives du cours, nous n'avons pas effectué l'étape de variance unitaire, ici vous êtes invités à effectuer cela également)

Pour mettre à l'échelle les données en variance unitaire et moyenne nulle, suivez l'une des deux possibilités : Utilisez les formules pour soustraire la moyenne et pour mettre à l'échelle les caractéristiques à la variance unitaire (c'est-à-dire, diviser par l'écart type). Utilisez la classe StandardScaler du module de prétraitement de sklearn, afin de créer le vecteur de caractéristiques mis à l'échelle.

Dans les deux cas, créez deux tableaux stockant les moyennes et les écarts types respectivement.

Étape 3 : Création de la matrice de covariance

Utilisez la méthode numpy.cov(X) pour créer la matrice de covariance. Faites attention au paramètre d'entrée X comme décrit dans l'API !

Étape 4 : Obtention des vecteurs propres et des valeurs propres

Utilisez la méthode np.linalg.eig(matrice de covariance) pour obtenir les vecteurs propres et les valeurs propres de la matrice de covariance. Quel est le premier composant principal et pourquoi ? Combien de composants principaux y a-t-il ? Combien de composants principaux choisiriez-vous de conserver et lesquels (le cas échéant) rejetteriez-vous et pourquoi ?

Créez deux nouveaux vecteurs de caractéristiques, l'un conservant tous les composants principaux, et l'autre ne conservant que ceux que vous considérez comme les plus importants.

Étape 5 : Transformation des données vers les nouvelles dimensions

Transformez les données vers les nouvelles dimensions en utilisant les formules. Faites attention à quand vous devez utiliser la matrice (du vecteur) elle-même ou sa transposée.

Étape 6 : Reconstruction du jeu de données initial

Auparavant, vous avez construit deux versions de jeux de données transformées : l'une utilisant tous les composants principaux et l'autre n'utilisant qu'un sous-ensemble. Implémentez les formules afin d'obtenir les données originales. La reconstruction des données est-elle réussie dans les deux cas ? Pourquoi ?

Étape 7 : Tracé des données transformées

Tracez les données transformées en utilisant les deux premiers composants principaux, et discutez de ce que vous voyez.

Régression Linéaire (optionnel)

Maintenant, nous allons comparer trois algorithmes de régression linéaire que nous avons vus lors de la session précédente : la régression linéaire (moindres carrés ordinaires), la régression Ridge et la régression Lasso. Dans scikit-learn, vous pouvez les trouver dans le module sklearn.linear_model.

Exercice 2

Chargez le jeu de données California étendu à partir du module mglearn datasets. Ce jeu de données décrit comment les prix des maisons varient en fonction de différentes caractéristiques. Divisez le jeu de données en ensembles d'entraînement et de test en utilisant la méthode train_test_split. Entraînez un modèle de régression linéaire, un modèle Lasso et un modèle Ridge sur les données d'entraînement et fournissez les scores d'entraînement et de test.

Ensuite, faites varier le paramètre alpha (si applicable) parmi les valeurs {0.01, 0.1, 1, 10} et observez comment les scores changent. Expliquez pourquoi vous pensez que cela se produit et quel modèle est préférable pour le jeu de données donné

Clustering avec K-Means

Dans cet exercice, nous allons explorer l'utilisation de l'algorithme K-Means pour la segmentation de données.

Exercice 3

Étape 1 : Chargement des données

Chargez un jeu de données approprié pour la segmentation en clusters. Vous pouvez utiliser un ensemble de données disponible dans scikit-learn ou un jeu de données personnalisé. (Vous pouvez utilisez le jeu de données iris de scikit-learn.)

Étape 2 : Prétraitement des données

Si nécessaire, effectuez un prétraitement des données tel que la mise à l'échelle des caractéristiques.

Étape 3 : Sélection du nombre de clusters

Pour sélectionner le nombre optimal de clusters vous pouvez utiliser des techniques telles que Elbow Method ou la Silhouette Method. Si vous utilisez le jeu de données Iris vous pouvez commencer au nombre de 3 clusters égal au nombre de classes dans le dataset.

Étape 4 : Application de K-Means

Appliquez l'algorithme K-Means sur les données en utilisant le nombre optimal de clusters sélectionné à l'étape précédente.

Étape 5 : Visualisation des clusters

Visualisez les clusters obtenus en utilisant des graphiques appropriés. Vous pouvez également explorer différentes méthodes de visualisation telles que la réduction de dimensionnalité (PCA) pour visualiser les clusters dans un espace de dimension réduit.

Étape 6 : Interprétation des clusters

Interprétez les clusters obtenus et discutez des résultats. Analysez les caractéristiques des clusters et identifiez les tendances ou les structures intéressantes dans les données.

Remarque : Vous pouvez répéter les étapes 3 à 6 en ajustant différents paramètres de l'algorithme K-Means (comme le nombre de clusters ou les initialisations) pour explorer comment ils affectent la qualité des clusters et les interprétations obtenues.

Clustering avec DBSCAN

Exercice 4

Nous allons explorer l'algorithme de clustering DBSCAN (Density-Based Spatial Clustering of Applications with Noise). L'objectif est de comprendre comment DBSCAN fonctionne et comment il peut être utilisé pour découvrir des groupes dans un jeu de données.

1. Commencez par importer la classe DBSCAN de scikit-learn et chargez un jeu de données approprié pour le clustering. Vous pouvez utiliser un jeu de données synthétique ou un ensemble de données réel, tel que Iris ou Wine, selon votre préférence.

2. Appliquez DBSCAN sur le jeu de données en ajustant les paramètres nécessaires, tels que epsilon (ε) et min_samples. Essayez différents réglages de ces paramètres et observez comment ils affectent les résultats du clustering.

3. Évaluez les performances du clustering en utilisant des métriques telles que la silhouette score ou en visualisant les clusters obtenus. Discutez des résultats et de la qualité des clusters trouvés.

   Infos : *Le score de silhouette (silhouette score) est une mesure utilisée pour évaluer la qualité de regroupement (clustering) d'un ensemble de données. Il fournit une indication de la cohésion et de la séparation des clusters dans l'espace des caractéristiques. Le score de silhouette varie entre -1 et 1. Un score de silhouette élevé indique que les clusters sont bien séparés, avec des points bien regroupés à l'intérieur de chaque cluster et une bonne distance entre les clusters. Un score de silhouette proche de zéro indique des clusters chevauchants ou des clusters de forme irrégulière.*

4. Explorez la sensibilité de DBSCAN aux valeurs aberrantes (outliers) en introduisant intentionnellement des valeurs aberrantes dans le jeu de données. Observez comment DBSCAN réagit à ces valeurs aberrantes et comment cela affecte les clusters obtenus.

5. Comparez DBSCAN à K-Means, en termes de performance et de capacité à détecter des clusters de formes et de densités différentes. Que concluez-vous ?

6. Résumez vos observations et conclusions sur l'utilisation de DBSCAN pour le clustering et discutez de ses avantages et inconvénients par rapport à d'autres algorithmes de clustering.

Note : Assurez-vous d'expliquer chaque étape de votre processus et de fournir des visualisations ou des métriques appropriées pour appuyer vos conclusions.