Technology
Arts Sciences
TH Köln

Unsupervised Learning

Von Hochdimensionalen Daten zum Visualisierten Cluster

Cihat Özkurt 11148632

Inhalt

Einleitung

Zuviel Informationen sind nicht immer gut. Gerade dann, wenn man versucht, nur Relevante Informationen rauszufiltern um darin Abhängigkeiten und Muster zu erkennen. Der Weg von einer Tabelle mit vielen Zeilen und Spalten zu einem aussagekräftigen Diagramm ist nichts, was man mal eben schnell erledigt.

Dazu muss man sich die Daten genausten anschauen und abwägen welche Merkmale sich sinnvoll in Relation setzten lassen, wie wir Sie gewichten und ob wir dann daraus Rückschlüsse auf die Gesamte Datenmenge ziehen können.

Ein komplizierter Weg. Aber wenn es darum geht Muster und Strukturen in Daten zu finden gibt es auch andere Wege. Einer ist das Unüberwachte Lernen. Ein über Thema, welches selbst zum Maschinen Lernen gehört und selbst viele Methoden zur Datenanalyse mitbringt.

Wir wollen uns damit beschäftigen, wie man es schafft Hochdimensionale Daten, also Daten mit vielen Merkmalen, sinnvoll abzubilden, um daraufhin Muster in den Daten zu erkennen.

Unsupervised Learning

Unsüpervised Learning ist eine Art des maschinellen Lernens, bei dem in einem Datensatz nach bisher unentdeckten Mustern gesucht wird, ohne dass bereits Labels vorhanden sind.

Vergleichung Unsupervised Learning und Supervised Learning

Gelabelte Daten: Der Hauptunterschied zwischen den beiden Ansätzen ist die Verwendung von gelabelte Datensätzen. Vereinfacht gesagt, werden beim überwachten Lernen gelabelte Daten verwendet, während ein Unüberwachter Lernalgorithmus dies nicht tut.

Ziele: Beim überwachten Lernen besteht das Ziel darin, die Ergebnisse für neue Daten vorherzusagen. Bei einem unüberwachten Lernalgorithmus besteht das Ziel darin, aus großen Mengen neuer Daten Erkenntnisse zu gewinnen.

Anwendungen: Modelle des überwachten Lernens eignen sich unter anderem für die Erkennung von Spam, Stimmungsanalysen, Wettervorhersagen und Preisvorhersagen. Unüberwachtes Lernen hingegen eignet sich hervorragend für die Erkennung von Anomalien, Empfehlungsmaschinen, Kundenpersönlichkeiten usw.

Komplexität: Beim überwachten Lernen handelt es sich um eine einfache Methode des maschinellen Lernens.Beim unüberwachten Lernen benötigen Sie leistungsstarke Tools für die Arbeit mit großen Mengen unklassifizierter Daten.

Unsupervised Learning Aufgaben

Assoziation entdeckt Zusammenhängen zwischen den Merkmalen oder Variablen in den Daten. Meistens wird es in der Warenkorb-Datenanalyse, im Cross-Marketing eingesetzt.

Dimensionalitätsreduktion entfernt redundante und irrelevante Merkmale, ohne dass dabei viel Information verloren geht, denn mehr Eingangsmerkmale machen die Vorhersageaufgabe für unser Modell schwieriger, was das Risiko einer massiven Überanpassung mit sich bringt.

Anomalieerkennung: Hierbei werden abnormale oder ungewöhnliche Muster in den Daten identifiziert, die sich von der normalen Datenverteilung unterscheiden.

Clustering: Bei dieser Art des unüberwachten Lernens gibt es verschiedene Algorithmen, mit denen wir unsere Daten auf der Grundlage von Ähnlichkeiten oder versteckten Mustern in unseren Daten in Clustern zusammenfassen können. Für Customer Segmentation, Bild- und Objekterkennung verwendet Clustering Algorithmen werden. Die belieste Clustering Algorthmen sind: K-mean Clustering,DBSCAN und Hierarchical Clustering.

PCA

Die PCA „Principal Component Analysis“, zu Deutsch Hauptkomponentenanalyse, ist ein Mathematisches Verfahren, welches zur Daten Analyse und Aufbereitung dient.

Erstm

Genutzt wird es, um mehrdimensionale Daten auf wenige Hauptkomponenten zu Projizieren. Mit den reduzierten Daten kann man dann einfacher weiterarbeiten.

Wie viele Hauptkomponenten man am Ende braucht hängt vom Anwendungsfall ab. Man kann die Daten auf 3 Hauptkomponenten reduzieren, um Dichtebasierte Analysen vorzunehmen. Bei 2 Hauptkomponenten kann man die Daten Clustern oder Visualisieren. Oder die Dimensionen werden für weitere Rechenoperationen Reduziert, um weniger Rechenleistung in Anspruch zu nehmen.

Ziel der PCA ist es die Dimensionen zu reduzieren und dabei Möglichst viel Informationen über die Daten zu behalten. Der Algorithmus selbst kann nicht entscheiden welche Informationen wichtig sind oder nicht. Die Hauptkomponenten, die er berechnet, werden aufgrund der Größten Varianz ausgewählt.

Grundlagen

Datenvorbereitung

Das Problem an der PCA ist, dass wir die Daten gut kennen sollten. Ausreißer können das Ergebnis verfälschen. Und die am ende Ausgegebenen Hauptkomponenten sind keiner bestimmten Dimension zuordbar, sie sind eine Zusammenführung der am stärksten Variierenden Merkmale. Die Hauptkomponenten müssen am Ende also neu interpretiert werden. Wichtig für die PCA ist eine Datenvorbereitung. Sind die Zahlen

Hauptkomponenten

Die Hauptkomponenten, englisch „Prinzipal Components“, werden mit „PC1, PC2, PC3 etc.“ durchnummeriert. Alle Hauptkomponenten sind der Größe nach sortiert. Die Sortierung ist abhängig von der erklärten Varianz der Hauptkomponenten. PC1 ist die erste Hauptkomponente und gibt den Vektor mit der Größten Varianz der Daten an. PC2 entsprechend den mit der zweitgrößten Varianz und so weiter. Um den Anteil der erklärten Varianz graphisch darzustellen nutzt man den Scree-Plot.

Scree Plot

Der Scree Plot ist ein Graph

Mathematischer Ansatz

PCA in Python

Natürlich kann man die PCA in Python Schritt für Schritt selbst durchführen. Die nötigen Rechenoperationen kann man mit Numpy durchführen. (s.308f)

Um diesen Aufwand zu umgehen kann man stattdessen die Klasse PCA nutzen. Diese befindet sich in der Scikit-learn Bibliothek im decomposition Modul. Nachdem man das Modul importiert hat, kann man von der Klasse PCA eine Instanz erzeugen und hat die Möglichkeit anzugeben auf wie viele Hauptkomponenten die Daten projiziert werden sollen. Außerdem hat man die Möglichkeit anzugeben wieviel Prozent der Varianz erhalten bleiben soll. Die PCA berechnet dann selbstständig die Anzahl der Hauptkomponenten aus, die benötigt werden, um den gewählten Anteil der Varianz abzubilden.

K-Means Algorithmus

K-means ist ein unüberwachter Lernalgorithmus. Sein Ziel ist es, herauszufinden, wie sich verschiedene Punkte gruppieren. Die Intuition hinter diesem mathematischen Modell ist, dass ähnliche Datenpunkte näher beieinander liegen. K-means versucht dann, verschiedene k-Punkte, so genannte Zentroide, zu bestimmen, die sich im Zentrum (geringster kumulativer Abstand) von anderen Punkten derselben Klasse befinden, aber weiter entfernt von Punkten einer anderen Klasse.

Wie K means algortihm funktionert? Die Schritten des Algorithms

Schritt 1: Auswahl von K zufälligen Punkten als Clusterzentren, den sogenannten Zentroiden.

Schritt 2: Zuweisung jedes Datenpunkte zum nächstgelegenen Cluster durch Anwendung der euklidischen Distanz (d.h. Berechnung der Distanz zu jedem Zentroid).

Schritt 3: Identifizierung neuer Zentroide durch Ermittlung des Durchschnitts der zugewiesenen Punkte

Schritt 4: Wiederholen Sie Schritt 2 und Schritt 3, bis Konvergenz erreicht ist.

Fehlermetriks von K Means

1. Elbow-Methode1

Die Elbow-Methode gibt uns eine Vorstellung davon, wie viele k (Gruppe) Cluster auf der Grundlage der Summe der quadratischen Abstände (SSE) zwischen den Datenpunkten und den Zentren der ihnen zugeordneten Cluster sinnvoll wären. Wir wählen k an dem Punkt aus, an dem die SSE beginnt, sich abzuflachen und einen Ellbogen zu bilden. Wir werten SSE(Inertia) für verschiedene Werte von k aus, um zu sehen, wo die Kurve einen Ellbogen bildet und abflacht.

Inertia (SSE) Formula: Summe der quadratischen Abstände

$$\sum_{i=1}^{N}(x_i - C_k)^2$$

- C = centroids
- X = einen Datenpunkt im Datensatz

2. Silhouetten-Analyse

Mit der Silhouettenanalyse kann der Grad der Trennung zwischen den Clustern bestimmt werden.

Berechnung Sie den Koeffizienten

$$s = \frac{b - a}{max(a, b)}$$

- Berechnen Sie den durchschnittlichen Abstand zu allen Datenpunkten im gleichen Cluster (ai).
- Berechnen Sie den durchschnittlichen Abstand zu allen Datenpunkten im nächstgelegenen Cluster (bi).

- Der Koeffizient kann Werte im Intervall [-1, 1] annehmen.
- Wenn er 0 ist -> liegt die Stichprobe sehr nahe an den benachbarten Clustern.
- Ist er 1 -> ist die Stichprobe weit von den benachbarten Clustern entfernt.

Hyperparameters von K-means

init (default = “k-means++“)

init Parameter ist eine Möglichkeit, die Initialisierungsmethode für die Zentroide im K-Means-Algorithmus anzugeben. "k-means++" ist eine verbesserte Initialisierungsmethode, die dazu dient, eine bessere Anfangspositionierung der Zentroide zu erreichen.

n_init (default = 10)

Hyperparameter gibt an, wie oft der K-Means-Algorithmus mit verschiedenen zufälligen Initialisierungen der Zentroide durchgeführt wird. Jede Initialisierung kann zu leicht

unterschiedlichen Ergebnissen führen, und der beste Lauf (mit der niedrigsten Inertia) wird als Endergebnis ausgewählt.

max_iter(default=300)

Maximale Anzahl von Iterationen des K-means-Algorithmus für einen einzigen Durchlauf.

Fazit

Das Kmeans-Clustering ist einer der beliebtesten Clustering-Algorithmen und wird in der Regel von Praktikern bei der Lösung von Clustering-Aufgaben zuerst angewendet, um eine Vorstellung von der Struktur des Datensatzes zu erhalten. Das Ziel von kmeans ist es, Datenpunkte in verschiedene, sich nicht überschneidende Untergruppen zu gruppieren. Es leistet sehr gute Arbeit, wenn die Cluster eine Art Kugelform haben. Es leidet jedoch, wenn die geometrische Form der Cluster von der Kugelform abweicht. Außerdem lernt es die Anzahl der Cluster nicht aus den Daten, sondern muss im Voraus festgelegt werden. Um ein guter Praktiker zu sein, ist es gut, die Annahmen zu kennen, die den Algorithmen/Methoden zugrunde liegen, damit Sie eine ziemlich gute Vorstellung von den Stärken und Schwächen der einzelnen Methoden haben. Dies hilft Ihnen bei der Entscheidung, wann und unter welchen Umständen Sie welche Methode einsetzen sollten. In diesem Beitrag haben wir sowohl die Stärken und Schwächen als auch einige Bewertungsmethoden im Zusammenhang mit kmeans behandelt.

Anhang

1. https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning
2. https://av-eks-blogoptimized.s3.amazonaws.com/62725cluster0
3. https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning
4. https://www.simplilearn.com/tutorials/machine-learning-tutorial/supervised-and-unsupervised-learning
5. https://www.altexsoft.com/blog/semi-supervised-learning/
6. https://www.analyticsvidhya.com/blog/2020/10/a-simple-explanation-of-k-means-clustering/
7. https://www.kaggle.com/
8. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
9. https://medium.com/data-folks-indonesia/step-by-step-to-understanding-k-means-clustering-and-implementation-with-sklearn-b55803f519d6