Gede Prasidha Bhawarnawa

IF 2020 - 13520004

UNSUPERVISED LEARNING

Apa itu unsupervised learning?

Unsupervised learning adalah sebuah tipe algoritma pembelajaran mesin yang digunakan untuk menganalisis dan mengelompokkan dataset yang tidak dilabeli. Algoritma ini dibuat dengan tujuan mendeteksi pola-pola pada dataset learning yang tidak terlihat secara kasat mata. Algoritma ini tepat digunakan untuk dataset yang tidak dilabeli oleh manusia. Meskipun begitu, algoritma ini dapat secara akurat mengelompokkan data-data sebagaimana manusia mungkin melakukannya. Umumnya algoritma ini digunakan untuk memecahkan permasalahan analisa data yang bersifat eksploratif (seperti penemuan trend, pola kebiasaan, dll), OCR (optical chracter recognition), dan segmentasi pelanggan atau pengunjung sebuah website atau tempat.

Jelaskan bagaimana cara kerja dari algoritma yang anda implementasikan!

1. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) adalah sebuah algoritma yang mengklusterkan data berdasarkan jarak antar data dan jumlah tetangga atau neighbor antar data. Algoritma ini bekerja dengan memanfaatkan dua buah data input dari pengguna beserta sebuah dataset untuk training. Dua data input tersebut adalah epsilon dan minimum points. Epsilon adalah besar ukuran radius dengan salah satu atribut data sebagai titik pusat yang digunakan untuk mencari “tetangga” dari data tersebut. Lalu, minimum points adalah sebuah jumlah minimum tetangga yang harus dimiliki oleh sebuah data untuk disebut sebagai core point. Terdapat tiga jenis point pada analisa DBSCAN: core point, border point, dan noise. Core point adalah sebuah titik yang memiliki lebih banyak tetangga dari syarat yang didefinisikan dari minimum points. Core point dan tetangganya ini yang kemudian disebut sebagai sebuah cluster. Border point adalah sebuah point data yang tergolong pada sebuah cluster namun tidak memiliki jumlah tetangga yang sama dengan atau lebih dari minimum points. Sementara itu, noise adalah point data yang tidak memiliki tetangga di sekitarnya.
   Algoritma ini bekerja dengan awalnya melakukan pemilihan asal dari semua titik di dalam dataset lalu mencari titik data di sekitarnya yang memenuhi persyaratan menjadi tetangganya (berada dalam jarak di bawah epsilon untuk setiap atributnya). Lalu, bila titik awal tadi merupakan sebuah core point, maka akan diulang proses diatas untuk setiap tetangga dalam radius titik tersebut. Bila titik tersebut merupakan sebuah border point, maka iterasi dan pencarian titik selanjutnya dihentikan dan algoritma akan melanjutkan dengan titik tetangga selanjutnya. Bila semua tetangga telah berhasil diiterasi, maka tahap selanjutnya adalah mengulangi proses ini dari awal namun dengan titik data lainnya yang belum tergolong ke dalam sebuah cluster. Ini dilakukan hingga sebuah data telah berhasil dibuat klusternya.
2. KMeans diimplementasikan dengan cara menerima sebuah dataset dan variabel input k sebagai jumlah kluster yang diinginkan. Ide implementasi dari Kmeans cukup sederhana. Pada awalnya, dicari titik (yang kemudian akan direferensikan sebagai sentroid) sejumlah k. Sentroid ini akan menjadi titik tengah dari klaster yang akan terbentuk, menghasilkan nama KMeans atau K sebagai rata-rata. Pada awal iterasi algoritma, karena komputer belum mengetahui data mana yang berada di tengah, dia akan memilih secara random atau memilih berdasarkan nilai maksimum atau minimum dari dataset yang sudah ada lalu dirata-

rata. Tujuannya adalah diharapkan didapatkan sentroid yang sedekat mungkin dengan mayoritas data pada dataset. Lalu, akan dicari sum of squared errors per sentroid terhadap semua data yang ada di dataset. Squared errors dapat dicari dengan menggunakan jarak Euclidean atau jarak mutlak antara titik data di dataset dengan sentroid. Lalu hasilnya diakumulasi. Jika pada sebuah iterasi didapatkan sentroid dengan sum of squared errors yang lebih kecil maka peran sentroid akan digantikan dengan rata-rata data dengan tingkat error yang lebih rendah. Ini akan diulangi hingga posisi titik dan sentroid di dalam k kluster tidak berubah. Pada titik ini, algoritma dapat berhenti melakukan iterasi dan kluster dapat diplot ke dalam sebuah grafik atau divisualisasikan dengan asumsi jumlah atributnya lebih kecil dari atau sama dengan 3.

3. KMedoids adalah sebuah metode yang mirip dengan metode KMeans, namun perbedaannya terletak pada metode pencarian titik tengah. Bila pada Kmeans menggunakan rata-rata dari semua data pada sebuah cluster, KMedoids menggunakan data anggota cluster tersebut untuk dijadikan titik tengah. Cara kerjanya cukup mirip dengan KMeans. Pertama-tama dicari titik random dari dataset sebanyak K dan jadikan titik tersebut sebagai titik pusat dari K cluster yang ingin dibuat, selanjutnya direferensikan sebagai medoid. Setelah itu asosiasikan titik-titik yang tersisa di dataset tersebut dengan medoid terdekat dan tukar perannya (medoid – data biasa). Bila setelah ditukar, cost atau beban yang ada di sistem berkurang, maka tukaran menjadi permanen. Bila tidak, kembalikan ke mode semula. Berhenti lakukan iterasi saat nilai cost sudah minimum. Cost sebuah data dapat dihitung dengan menjumlahkan nilai dari semua fitur pada data dan menselisihkannya dengan total nilai dari semua fitur pada medoid. Lalu, cost ini akan ditotal untuk semua data yang ada di dalam cluster.

Bandingkan ketiga algoritma tersebut, kemudian tuliskan kelebihan dan kelemahannya!

1. Kelebihan dari algoritma DBSCAN adalah tidak membutuhkan nilai predetermined untuk jumlah cluster. Alasannya adalah DBSCAN dapat mengidentifikasi sendiri jumlah klaster yang ada sesuai dengan nilai epsilon dan minimum points yang di feed ke dalam algoritmanya. Selain itu, algoritma ini juga dapat dengan mudah mengklasifikasikan noise atau outlier dari data, sehingga mudah juga untuk membuang data-data tersebut jika dikira tidak relevan. Terakhir, algoritma ini dapat menemukan klaster dengan berbagai bentuk dan ukuran karena berbeda dengan KMeans atau KMedoids, dia menggunakan sistem neighboring sehingga dapat menangkap data-datanya yang lebih tersebar ketimbang KMeans atau KMedoids yang bergantung pada satu titik tunggal sebagai referensi.
   Kekurangan dari algoritma DBSCAN adalah kebutuhan untuk penyesuaian nilai epsilon dan minimum points sehingga untuk data-data yang berdesimal tinggi dapat menimbulkan kesulitan bagi programmer untuk mendeteksi nilai epsilon dan minimum points yang tepat. Selain itu, algoritma DBSCAN juga memiliki kesulitan mendeteksi klaster pada dataset yang densitasnya tidak merata atau jika fiturnya terlalu banyak karena dapat memperlambat waktu komputasi.
2. Kelebihan dari algoritma KMeans adalah relatif mudah untuk diimplementasikan karena hanya membutuhkan informasi nilai max dan min dari data pada awalnya lalu dapat menggunakan nilai Euclidean distance untuk mencari sum of squared errors. Karena pendekatannya menggunakan rata-rata dari nilai maksimum dan minimum, pada mayoritas data, ini dekat dengan median sehingga penyesuaian dari iterasi pertama ke iterasi selanjutnya tidak terlalu ekstrim dan mempercepat waktu komputasi. Model ini juga mudah mendeteksi bentuk klaster yng bervariasi dan mudah untuk diskala ulang untuk dataset yang

lebih besar atau lebih kecil. Penambahan data baru ke dalam training set juga tidak terlalu menyulitkan proses modeling karena algoritmanya yang sederhana.
Kelemahan dari algoritma ini adalah mudah dipengaruhi oleh outlier, mengingat penggunaan nilai maksimum dan minimum untuk sentroid awal. Sehingga dibutuhkan pembersihan data terlebih dahulu sebelum menggunakan algoritma ini. Selain itu, dibutuhkan intervensi dari programmer untuk menentukan jumlah klaster yang dibutuhkan. Terakhir, algoritma ini juga kesulitan dalam mengklasterkan data yang terlalu bervariasi densitasnya dan memiliki terlalu banyak fitur atau atribut (multi dimensi)

3. Kelebihan dari algoritma KMedoids adalah kemudahan untuk diimplementasi karena konsep algoritmanya yang sederhana, relatif cepat karena butuh sedikit langkah dan iterasi (alasannya medoid sudah berada di dalam klaster jadi penyesuaian minimal dengan asumsi nilai k mencukupi), serta lebih insensitif terhadap data outlier atau noise dibandingkan metode partisi lainnya (kecuali DBSCAN).
   Kekurangan dari algoritma ini adalah sulit untuk mendeteksi klaster data yang berbentuk non-spherical karena jika dibandingkan dengan DBSCAN yang berfokus pada konektivitas antardata, KMedoids berfokus pada kepadatan data yang mungkin meninggalkan beberapa data yang sebenarnya terkoneksi namun diidentifikasi sebagai outlier. Selain itu, ada kemungkinan hasil klaster dalam pengulangan algoritma dengan jumlah k yang sama berbeda. Alasannya adalah titik medoid yang dipilih pertama kali adalah acak.

Jelaskan penerapan dari algoritma unsupervised di dunia nyata!

Salah satu contoh paling populer dalam penggunaan algoritma unsupervised adalah di recommendation system. Recommendation system adalah sistem yang merekomendasikan barang baru sesuai dengan track record di masa lalu. Track record ini jumlahnya banyak dengan contoh sumbernya berupa click stream data di website yang mencatat apa yang kita klik, apa yang kita suka, kapan kita mengklik item tersebut, dan lain-lain. Lalu, unsupervised learning berperan disini sebagai pencari pola sehingga dapat diklasifikasikan pola atau kebiasaan sehingga sistem dapat merekomendasikan item yang sesuai. Tentu saja sistem ini memiliki bias namun dengan bertambahnya jumlah data di dalam dataset, akurasi sistem tersebut dapat bertambah dan barang yang direkomendasikan bisa semakin cocok dengan kebutuhan atau keinginan pengguna sehingga memakmurkan bisnis atau visitor dari website tersebut.