Summary Machine Learning

CLUSTERING

Muh.Ikhsan (H071191049)

---

1. Definisi Clustering

Cluster merupakan kumpulan objek data. Anggota cluster memiliki kemiripan satu sama lain, tetapi berbeda dengan cluster yang lain. ***Clustering*** atau klasterisasi adalah metode pengelompokan data. Menurut Tan, 2006 *clustering* adalah sebuah proses untuk mengelompokan data ke dalam beberapa *cluster* atau kelompok sehingga data dalam satu *cluster* memiliki tingkat kemiripan yang maksimum dan data antar *cluster* memiliki kemiripan yang minimum.

2. Manfaat *Clustering*

a. *Clustering* merupakan metode segmentasi data yang sangat berguna dalam prediksi dan analisa masalah bisnis tertentu. Misalnya Segmentasi pasar, marketing dan pemetaan zonasi wilayah.
b. Identifikasi obyek dalam bidang berbagai bidang seperti computer vision dan image processing.

3. Syarat *Clustering*

a. Skalabilitas, suatu metode *clustering* harus mampu menangani data dalam jumlah yang besar.
b. Kemampuan analisa beragam bentuk data, algoritma klasterisasi harus mampu diimplementasikan pada berbagai macam data seperti data nominal, ordinal, maupun gabungannya.
c. Menemukan *cluster* dengan bentuk yang tidak terduga, banyak algoritma clustering yang menggunakan metode *Euclidean* atau *Manhattan* yang hasilnya berbentuk bulat. Padahal hasil clustering dapat berbentuk aneh dan tidak sama antara satu dengan yang lain
d. Sensitifitas terhadap perubahan input, algorima clustering dengan tingkat sensitifitas rendah dapat menyebabkan perubahan mencolok apabila terdapat perubahan data pada input

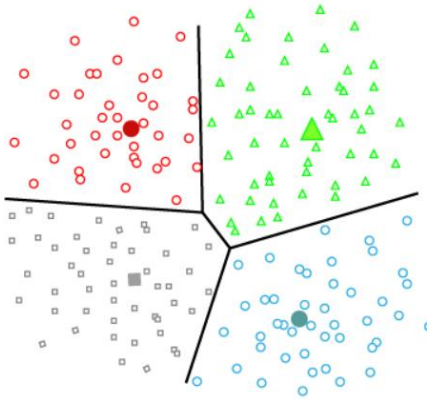
e. Mampu melakukan clustering untuk data dimensi tinggi
f. Hasil dari clustering harus dapat diinterpretasikan dan berguna

4. Metode *Clustering*

Beberapa metode clustering utama dalam machine learning

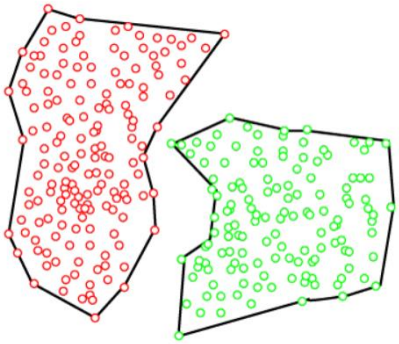
a. Partitioning Clustering,

Jenis pengelompokan yang membagi data menjadi kelompok non-hierarki. Ini juga dikenal sebagai ***metode berbasis centroid***. Contoh paling umum dari partisi clustering adalah algoritma *K-Means Clustering.*



b. Density-Based Clustering

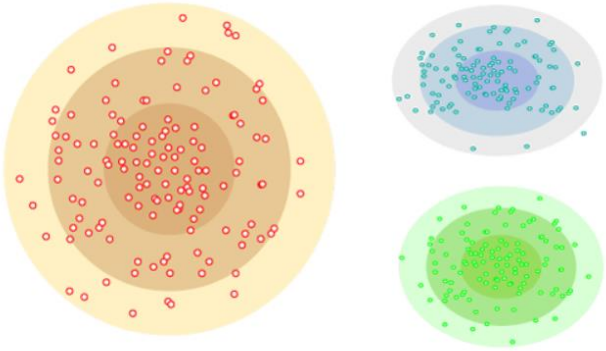
Metode ini menghubungkan daerah yang sangat padat ke dalam cluster, dan distribusi secara sewenang-wenang terbentuk selama daerah padat dapat dihubungkan. Algoritma-algoritma ini dapat menghadapi kesulitan dalam mengelompokkan titik-titik data jika kumpulan data memiliki kepadatan yang bervariasi dan dimensi yang tinggi.



c. Distribution Model-Based Clustering

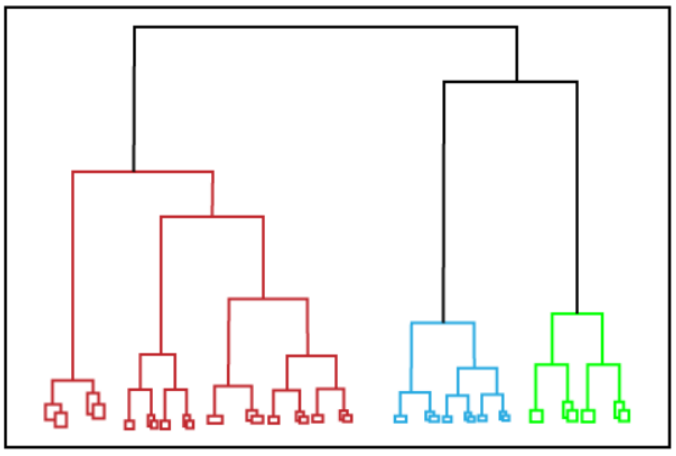
Dalam metode pengelompokan berbasis model distribusi, data dibagi berdasarkan probabilitas bagaimana suatu kumpulan data termasuk dalam distribusi tertentu. Pengelompokan dilakukan dengan mengasumsikan

beberapa distribusi umumnya *Distribusi Gaussian*. Contoh dari tipe ini adalah algoritma *Expectation-Maximization Clustering.*



d. *Hierarchical Clustering*

Pengelompokan hierarki dapat digunakan sebagai alternatif untuk pengelompokan yang dipartisi karena tidak ada persyaratan untuk menentukan jumlah cluster yang akan dibuat. Dalam teknik ini, dataset dibagi menjadi cluster untuk membuat struktur seperti pohon, yang juga disebut dendrogram. Pengamatan atau sejumlah cluster dapat dipilih dengan memotong pohon pada tingkat yang benar. Contoh paling umum dari metode ini adalah algoritma *Agglomerative Hierarchical*.



e. *Fuzzy Clustering*

Fuzzy clustering adalah jenis metode lunak di mana objek data mungkin milik lebih dari satu kelompok atau cluster. Algoritma *Fuzzy C-means* adalah contoh dari tipe clustering ini.

5. Algoritma K-Means

K-means merupakan algoritma clustering. K-means Clustering adalah salah satu "unsupervised machine learning algorithms" yang paling sederhana dan populer. K-Means Clustering adalah suatu metode penganalisaan data atau metode Data Mining yang melakukan proses pemodelan tanpa supervisi (unsupervised) dan merupakan salah satu metode yang melakukan pengelompokan data dengan sistem partisi.

Data clustering menggunakan metode K-Means Clustering ini secara umum dilakukan dengan algoritma dasar sebagai berikut:

a. Tentukan jumlah cluster.
b. Tentukan centroid awal dari tiap cluster *(Using Lower and Upper bounds or Randomly)*.
c. Alokasikan data dengan jumlah cluster yang ditentukan berdasarkan jarak terdekan dengan centroid (menggunakan rumus *Ecludien distance space* atau *Manhattan / City Block distance space*).
d. Setelah cluster dan anggotanya terbentuk, hitung mean tiap cluster dan jadikan sebagai centroid baru.
e. Kembali ke-3 apabila masih terdapat perpindahan data dari satu cluster ke cluster yang lain, atau apabila perubahan pada nilai centroid masih diatas nilai threshold yang ditentukan.

6. Algoritma K-Medoids

K-Medoids atau Partitioning Around Method (PAM) adalah metode cluster non hirarki yang merupakan varian dari metode K-Means. K-Medoids hadir untuk mengatasi kelemahan K-Means yang sensitif terhadap outlier karena suatu objek dengan suatu nilai yang besar mungkin secara substansial menyimpang dari distribusi data (Jiawei & Kamber, 2006). K-Medoids menggunakan metode pengelompokan partisi untuk mengelompokan sekumpulan n objek menjadi sejumlah k cluster. Algoritma ini menggunakan objek pada kumpulan objek yang mewakili sebuah cluster. Objek yang mewakili sebuah cluster disebut dengan medoids.

Tahapan-tahapan K-Medoids:

a. Tentukan k (jumlah cluster) yang diinginkan.
b. Pilih secara acak medoid awal sebanyak k dari n data.
c. Hitung jarak masing-masing obyek ke medoid sementara dengan *euclidean distance*, kemudian tandai jarak terdekat obyek ke medoid dan hitung totalnya.
d. Lakukan iterasi medoid.
e. Hitung total simpangan (S).
f. Jika a adalah jumlah jarak terdekat antara obyek ke medoid awal, dan b adalah jumlah jarak terdekat antara obyek ke medoid baru, maka total simpangan adalah S = b — a
g. Jika S < 0, maka tukar obyek dengan data untuk membentuk sekumpulan k baru sebagai medoid.
h. Ulangi langkah 3 sampai 5 dan hentikan jika sudah tidak terjadi perubahan anggota medoid.

7. Algoritma Hirarki Agglomerative

Agglomerative (metode penggabungan) adalah strategi pengelompokan hirarki yang dimulai dengan setiap objek dalam satu cluster yang terpisah kemudian membentuk cluster yang semakin membesar. Jadi, banyaknya cluster awal adalah sama dengan banyaknya objek.

Dalam agglomerative method, teknik pengelompokan yang paling dikenal adalah:

a. Single linkage (jarak terdekat atau tautan tunggal)
   Teknik yang menggabungkan cluster-cluster menurut jarak antara anggota-anggota terdekat di antara dua cluster.
b. Average linkage (jarak rata-rata atau tautan rata-rata)
   Teknik yang menggabungkan cluster-cluster menurut jarak rata-rata pasangan anggota masing-masing pada himpunan antara dua cluster.
c. Complete linkage (jarak terjauh atau tautan lengkap)
   Teknik yang menggabungkan cluster-cluster menurut jarak antara anggota-anggota terjauh di antara dua cluster

Langkah-langkah algoritma agglomerative

1) Hitung matriks jarak, Ada berbagai macam jenis jarak, namun jarak yang sering digunakan adalah Euclidean.

2) Gabungkan dua cluster terdekat, Jika jarak objek a dengan b memiliki nilai jarak paling kecil dibandingkan jarak antar objek lainnya dalam matriks jarak Euclidean, maka gabungan dua cluster pada tahap pertama adalah d_ab.
3) Perbarui matriks jarak sesuai dengan teknik pengelompokan agglomerative method, Jika d_ab adalah jarak terdekat dari matriks jarak Euclidean.

8. Algoritma DBSCAN

Algoritma Density-based Spatial Clustering of Application with Noise (DBSCAN) merupakan metode clustering yang berbasis kepadatan (density-based) dari posisi amatan data dengan prinsip mengelompokkan data yang relatif berdekatan. DBSCAN sering diterapkan pada data yang banyak mengandung noise, hal ini dikarenakan DBSCAN tidak akan memasukkan data yang dianggap noise kedalam cluster manapun.

Dalam proses pembuatan cluster menggunakan DBSCAN sebuah data akan dikelompokkan dengan tetangganya. Sepasang amatan dikatakan bertetangga apabila jarak antara dua amatan tersebut kurang dari sama dengan nilai epsilon. Secara sederhana cara kerja DBSCAN adalah sebagai berikut :

a. Tentukan nilai minPts dan epsilon (eps) yang akan digunakan.
b. Pilih data awal "p" secara acak.
c. Hitung jarak antara data "p" terhadap semua data menggunakan *Euclidian distance*.
d. Ambil semua amatan yang *density-reachable* dengan amatan "*p*".
e. Jika amatan yang memenuhi nilai *epsilon* lebih dari jumlah minimal amatan dalam satu gerombol maka amatan "*p*" dikategorikan sebagai *core points* dan gerombol terbentuk.
f. Jika amatan "p" adalah border points dan tidak ada amatan yang density-reachable dengan amatan "*p*", maka lanjutkan pada amatan lainnya.
g. Ulangi langkah 3 sampai 6 hingga semua amatan diproses.