MINI BATCH K-MEANS

Makalah Ini Disusun untuk Memenuhi Tugas Mata Kuliah

Metode Kecerdasan Buatan

Dosen Pengampu : Muhammad Irvan Septiar Musti, S.Si, M.Si

Disusun oleh :

Ghina Rahmah	11190940000053
Fida Suci Rahmani	11190940000027
Rosa Amalia Nursinta	11190940000041
Elviana Saputri	11190940000043
Meissy Astariva Putri	11190940000063

PROGRAM STUDI MATEMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH JAKARTA

2022

KATA PENGANTAR

Assalammu'alaikum Warahmatullahi Wabarakatuh

Puji syukur mari kita panjatkan kepada Allah SWT yang Maha Esa karena atas berkat rahmat dan karunia-Nya kami dapat menyelesaikan penyusunan makalah dengan Materi "Mini Batch K-Means" dapat dikerjakan dengan tepat waktu dan dengan sebaik-baiknya.

Makalah ini merupakan tugas yang harus diselesaikan dalam mata kuliah Metode Kecerdasan Buatan dengan dosen pengampunya adalah Bapak Muhammad Irvan Septiar Musti, S.Si, M.Si. Harapan kami tentunya berharap agar makalah ini dapat membantu dan menambah wawasan bagi para pembaca.

Kami menyadari penuh bahwa dalam penyusunan project ini masih terdapat banyak kekurangan. Kami mengharapkan kritik dan saran yang membangun dari pembaca agar kedepannya kami bisa melakukan perbaikan untuk mendapatkan hasil yang lebih baik. Terakhir, kami berharap semoga laporan ini dapat bermanfaat dan dapat tercapai sesuai dengan yang diharapkan.

Wassalammu'alaikum Warahmatullahi Wabarakatuh

Tangerang Selatan, 19 Juni 2022

BAB I

PENDAHULUAN

A. Latar Belakang

Klasfikasi merupakan pengelompokkan data berdasarkan kesamaan label, contohnya dalam mengklasifikasikan bunga iris, dengan mengelompokkan data yang telah berlabel yaitu pada bunga iris virginiga, versicolor, dan setosa. Sedangkan klasterisasi merupakan pengelompokkan data yang berdasarkan kemiripan data, bisa saja tidak ada labelnya. Kemiripan ini dilihat pada nilai atribut-atributnya (nilai pada kolom-kolomnya).

Kemudian bagaimana menentukan kemiripan data? . Pengukuran jarak data yang dihitung dengan persamaan *Euclidean Distance*

$$d = \sqrt{\sum_{N} (x_i - y_i)^2}$$

Pengukuran jarak dengan menjumlahkan selisih antara x_i atribut data yang pertama dan y_i atribut data yang kedua. Dimana i meyatakan index dari atributnya sampai atribut ke-n.

Algoritma pengelompokan Mini batch K-means adalah algoritma K-means yang dapat digunakan saat mengelompokkan pada himpunan data besar, mini batch k-means menggunakan batch data kecil, acak, ukuran tetap untuk disimpan dalam memori, dan kemudian dengan setiap iterasi, sampel acak dari data dikumpulkan dan digunakan untuk memperbarui cluster. Terkadang kinerjanya lebih baik daripada algoritma K-mean saat bekerja pada himpunan data besar karena tidak memerlukan pengulangan di seluruh himpunan data.

Keuntungan utama menggunakan algoritma Mini-batch K-means adalah mengurangi biaya komputasi untuk menemukan cluster. Ide utama dari mini batch K-means adalah menggunakan batch kecil acak dari kumpulan data dengan ukuran tetap sehingga dapat disimpan dalam memori. Dalam setiap iterasi, sampel acak baru yg diperoleh dari kumpulan data digunakan untuk memperbarui centroid dan ini diulangi hingga konvergen. Setiap mini batch memperbarui centroid menggunakan metode penurunan gradien yang menerapkan learning rate untuk mendapatkan Konvergensi yg lebih cepat.

BAB II

TINJAUAN PUSTAKA

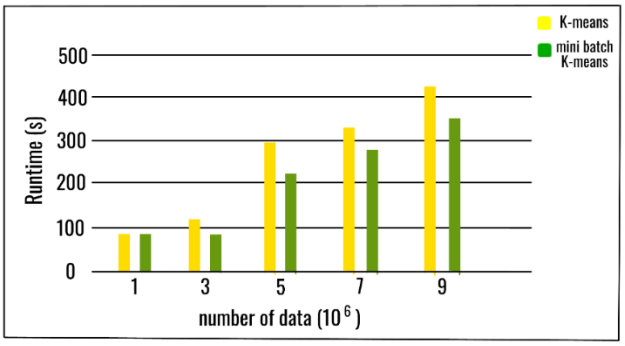
K-Means

K-means adalah salah satu algoritma pengelompokan yang paling populer, terutama karena kinerja waktu yang baik. Dengan bertambahnya ukuran dataset yang dianalisis, waktu komputasi K-means meningkat karena kendalanya membutuhkan seluruh dataset di memori utama. Ada metode yang dapat mengurangi biaya temporal dan spasial dari algoritma, yaitu algoritma Mini batch K-means.

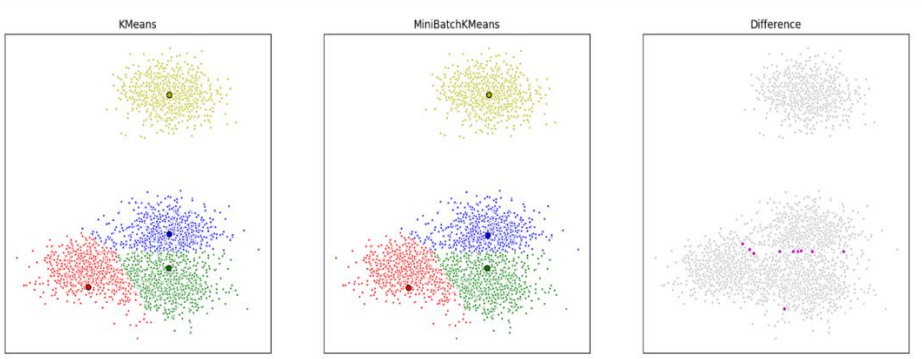
Mini Batch K-Means

Algoritma pengelompokan Mini batch K-means adalah algoritma K-means yang dapat digunakan saat mengelompokkan pada himpunan data besar, mini batch k-means menggunakan batch data kecil, acak, ukuran tetap untuk disimpan dalam memori, dan kemudian dengan setiap iterasi, sampel acak dari data dikumpulkan dan digunakan untuk memperbarui cluster. Ide utama algoritma Mini Batch K-means adalah menggunakan kumpulan data acak kecil dengan ukuran tetap, sehingga dapat disimpan dalam memori. Setiap iterasi sampel acak baru dari dataset diperoleh dan digunakan untuk memperbarui cluster dan ini diulang sampai konvergensi. Setiap batch mini memperbarui cluster menggunakan kombinasi cembung dari nilai prototipe dan data, menerapkan tingkat pembelajaran yang menurun dengan jumlah iterasi. Laju pembelajaran ini adalah kebalikan dari jumlah data yang ditugaskan ke cluster selama proses. Dengan bertambahnya jumlah iterasi, efek data baru berkurang, sehingga konvergensi dapat dideteksi ketika tidak ada perubahan dalam cluster yang terjadi pada beberapa iterasi berturut-turut.

Hasil empiris menunjukkan bahwa mini batch k-means dapat memperoleh penghematan waktu komputasi yang substansial dengan mengorbankan beberapa kehilangan kualitas cluster, tetapi tidak studi ekstensif tentang algoritma telah dilakukan untuk mengukur bagaimana karakteristik dataset, seperti jumlah cluster atau ukurannya, mempengaruhi kualitas partisi.



Berdasarkan grafik diatas, algoritma mini batch k-means lebih cepat dibandingkan algoritma k-means. Algoritma mini batch k-means mengambil kumpulan dataset kecil yang dipilih secara acak untuk setiap iterasi. Setiap data dalam batch ditugaskan ke cluster, tergantung pada lokasi centroid cluster sebelumnya. Kemudian memperbarui lokasi centroid cluster berdasarkan poin baru dari batch. Mini batch k-means lebih cepat tetapi memberikan hasil yang sedikit berbeda dari K-means batch normal.



Berdasarkan gambar diatas, ketika jumlah cluster dan jumlah data meningkat, penghematan relatif dalam waktu komputasi juga meningkat. Penghematan waktu komputasi lebih terlihat hanya ketika jumlah cluster sangat besar. Pengaruh ukuran batch dalam waktu komputasi juga lebih jelas ketika jumlah cluster lebih besar. Dapat disimpulkan bahwa, meningkatkan jumlah cluster, mengurangi kesamaan solusi K-means batch mini dengan solusi K-means. Meskipun partisi berkurang dengan bertambahnya jumlah cluster, fungsi tujuan tidak menurun pada tingkat yang sama. Ini berarti bahwa partisi akhir berbeda, tetapi lebih dekat kualitasnya. kekurangan : jumlah data (ukuran/size) pada tiap batchnya tidak boleh lebih kecil dari banyaknya klaster, dan Kelebihan: Keuntungan utama menggunakan algoritma Mini-batch K-means adalah mengurangi biaya komputasi untuk menemukan cluster dan Terkadang kinerjanya lebih baik daripada algoritma K-mean saat bekerja pada himpunan data besar karena tidak memerlukan pengulangan di seluruh himpunan data sehingga konvergen lebih cepat.

BAB III

HASIL DAN PEMBAHASAN

Mini Batch K-Means Use Case and Python Code

Studi kasus menggunakan algoritma ‘MiniBatchKMeans’ yang tersedia pada modul ‘sklearn’.

```
[1]  import numpy as np

     from sklearn.cluster import MiniBatchKMeans, KMeans
     from sklearn.metrics.pairwise import pairwise_distances_argmin
     from sklearn.datasets import make_blobs
```

```
[2]  # Load data in X
     batch_size = 45
     centers = [[1, 1], [-2, -1], [1, -2], [1, 9]]
     n_clusters = len(centers)
     X, labels_true = make_blobs(n_samples = 3000,
                                 centers = centers,
                                 cluster_std = 0.9)

     # perform the mini batch K-means
     mbk = MiniBatchKMeans(init ='k-means++', n_clusters = 4,
                           batch_size = batch_size, n_init = 10,
                           max_no_improvement = 10, verbose = 0)

     mbk.fit(X)
     mbk_means_cluster_centers = np.sort(mbk.cluster_centers_, axis = 0)
     mbk_means_labels = pairwise_distances_argmin(X, mbk_means_cluster_centers)
```

Dataset yang digunakan dalam kasus ini yaitu data ‘make_blobs’ yang didapat dari modul ‘sklearn’. dengan banyaknya data yaitu 3000 data yang dibagi ke dalam beberapa batch dimana untuk setiap batchnya berisikan 45 data dan akan dikelompokkan ke dalam 4 cluster yaitu cluster 0, 1, 2, dan 3.

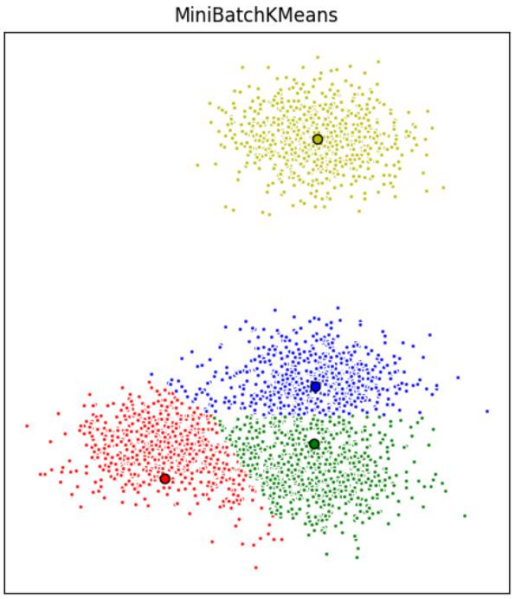
```
[10] # print the labels of each data
     print(mbk_means_labels)

     [3 2 1 ... 3 3 1]
```

Sehingga diperoleh data yang sudah diklasterisasi sebagai berikut :

```
[11] data_x = pd.DataFrame(X)
     label_data = pd.DataFrame(mbk_means_labels)
     ClusterData = pd.concat([data_x, label_data], axis=1)
     ClusterData.columns = ['x1', 'x2', 'Cluster']
     ClusterData
```

	x1	x2	Cluster
0	0.570762	8.911408	3
1	1.020149	1.363858	2
2	0.862305	-0.936075	1
3	-2.047109	-3.514982	0
4	2.582506	8.920657	3
...	...	...	...
2995	0.309157	0.696489	2
2996	0.959725	-3.357591	1
2997	1.317246	9.064186	3
2998	0.681219	7.891238	3
2999	0.148094	-2.510426	1

3000 rows × 3 columns



Interpretasi :

Berdasarkan jarak antar centroid pada visualisasi di atas, terlihat bahwa kelompok biru, hijau, dan merah saling berdekatan, ini berarti bahwa data pada kelompok biru, hijau, dan merah kemungkinan memiliki karakteristik yang tidak jauh berbeda. Sedangkan kelompok kuning berada jauh dari kelompok biru, hijau, dan merah, maka ini menunjukkan bahwa data pada kelompok kuning memiliki karakteristik yang cukup jauh berbeda dibanding ketiga kelompok lainnya.