Pembelajaran Mesin

Semester Genap Tahun Akademik 2017-2018

Tugas 2: K-Means Clustering

Dosen : Pak Suyanto

Disusun oleh :

Mochamad Yusuf Solihin (1301150020)

IF 39-06

Analisis Masalah

Pada tugas 2.0 ini, diberikan sejumlah data yang tersebar secara acak untuk dikelompokan kepada beberapa kelompok dengan tepat. Setelah data dikelompokan dengan jumlah kelompok yang tepat, selanjutnya diberikan sejumlah data untuk ditest, masuk kedalam kelompok manakan data test tersebut.

Dari masalah yang dipaparkan, perlu dilakukan *Clustering* pada data-data yang tersebar secara acak agar data-data tersebut dapat dikelompokan sesuai dengan kelompoknya masing-masing. Namun yang menjadi masalah adalah berapakah kelompok yang tepat untuk data-data yang tersebar tadi. Maka harus dilakukan pengujian jumlah *Cluster* yang optimum untuk mengelompokan data-data yang tersebar.

Desain

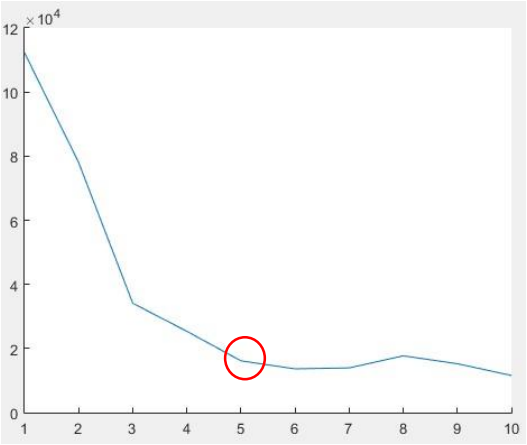
Masalah diatas akan dapat diatasi dengan berbagai metode, seperti *K-Means Clustering, Hierarchical Clustering, dan Self-Organizing Map (SOM)*. Namun pada tugas 2.0 ini, mahasiswa diharuskan menggunakan metode *K-Means Clustering* untuk menyelesaikan masalahnya.

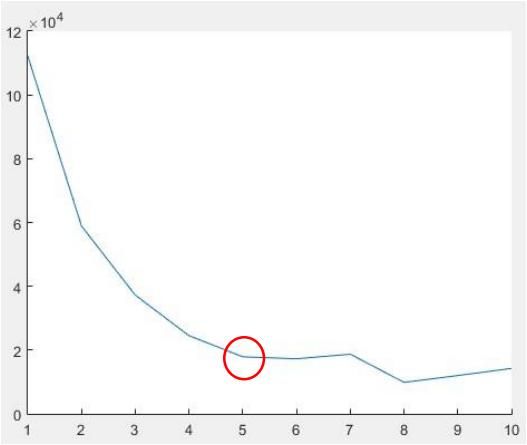
Source code yang dibuat dengan menggunakan bahasa pemrograman Matlab memiliki alur yang cukup mudah untuk dimengerti. Pertama, dibuat terlebih dahulu program SSE untuk menentukan jumlah K terbaik. Kemudian dibuat program K-Means untuk menyelesaikan masalah yang telah dipaparkan.

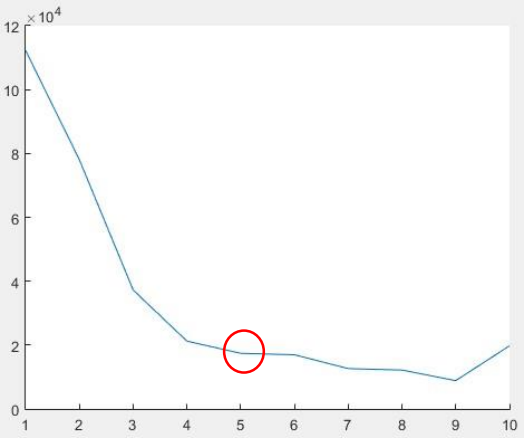
SSE

Pertama, *Import* data_train untuk dilakukan pengujian mendapatkan K terbaik. Kemudian dilakukan perulangan sebanyak 10x untuk menentukan nilai K sebanyak 10 SSE. Looping 1 sampai 10 akan didapatkan nilai SSE untuk setiap K nya. Didalam pengulangan 1 sampai 10 dilakukan pengelompokan (*Clustering*) yang k nya sesuai dengan perulangannya. Misal, saat perulangan ke-2, maka *centroi*nya sebanyak 2 titik. Maka dicari data-data yang paling dekat dengan setiap *Centroid* nya sehingga didapatkan *centroid* yang seimbang (tidak berubah-ubah lagi). Setelah didapatkan centroid yang idela, selanjutnya dilakukan penjumlahan jarak setiap *centroid* dengan anggota *cluster*nya masing-masing. Kemudian setelah didapatkan jumlah setiap *centroid* dengan anggota *cluster*nya,

dilakukan penjumlahan juga untuk didapatkan nilai SSE dari setiap perulangannya. Kemudian ditampilkan grafik nilai SSE terhadap *Cluster* untuk dianalisis nilai K yang terbaik. Cara membaca grafik SSE nya adalah dengan melihat penurunan nilai SSE yang drastis sampai penurunan nilai SSE nya yang tidak drastis. Jika pada K =1 ke K-2 selisihnya besar, makas lanjut ke K=3, lihat lagi selisihnya antara K=2 dengan K=3, jika didapatnya nilai selisihnya kecil, maka nilai K yang optimum nya bisa jadi K=3. Namun dari SSE yang didapatkan, dilakukan sebanyak 3 kali *run* program untuk melihat grafik SSE sebanyak 3 kali agar lebih tepat melihan selisih nilai SSE nya. Dan didapatkan nilai K yang optimum adalah 5.

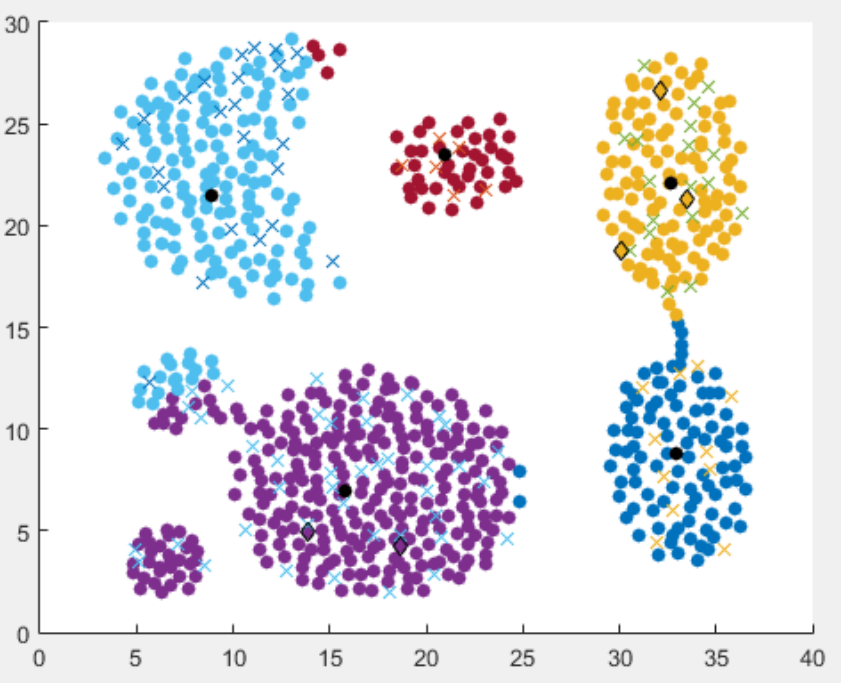






K-Means

Source code untuk *K-Means*, pertama dilakukan *import* data_train dan data_test nya untuk dikelompokan sebanyak K yang optimum. Selanjutnya random titik untuk setiap *Cluster* yang dijadikan *centroid* awal yang nantinya akan diperbaharui sampai titik *centroid* nya tidak berubah lagi. Selanjutnya dilakukan perulangan untuk mencari jarak setiap data dengan *centroid* yang nantinya dimasukan kedalam sebuah tabel. Selanjutnya dari tabel yang sudah terisi dengan jarak dari setiap *centroid* dengan datanya, dicari jarak terdekat dari setiap data dengan *centroidnya* yang selanjutnya dimasukan ke anggota dari *centroid* terdekatnya. Setelah setiap data memiliki *cluster* nya masing-masing, maka setiap *cluster*nya menghitung rata-rata dari jarak *centroid* dengan anggota datanya. Hasil dari rata-ratanya adalah perbaharuan dari *centroid*nya. Kemudian dilakukan perulangan sampai *centroid*nya tidak berubah-berubah lagi. Kemudian setelah *centroid*nya sudah didapatkan dan anggota-anggotanya sudah didapatkan juga, selanjutnya buat *scatter* untuk menampilkan data dengan pengelompokannya.



Selesai pengelompokan, maka data_test dapat diklasifikasikan termasuk kedalam *cluster* yang mana. Pada gambar diatas, titik yang berbentuk x adalah data_test yang telah masuk kedalam masing-masing *cluster.* Bentuk diamond berwarna hitam adalah *centroid- centroid* awal yang didapat dari nilai random.

Evaluasi Eksperimen

Dari hasil yang didapat, masih terdapat kesalahan dalam melakukan penentuan random *cluster* awal yang baik. Dimana titik random *cluster* awal yang baik adalah titik yang tersebar secara merata di sekitar pesebaran data nya. Yang saya baca dari berbagai sumber di internet dan paper, salah satu cara agar mendapat titik random *cluster* awal yang baik adalah dengan menggunakan metode random dengan memperhatikan probabilitas proportional sehingga pesebaran data nya merata ke setiap data yang tersebar. Hanya saja, saya masih kurang faham bagaimana mengimplementasikan kedalam eksperimen yang dilkakuan pada tugas 2.0 ini. Sehingga belum bisa mendapatkan akurasi yang tepat.