Jurnal Sistem dan Informatika Vol. 3 (Pebruari 2007), 47-60

K-Means – Penerapan, Permasalahan dan Metode Terkait

Yudi Agusta, PhD
STMIK STIKOM BALI, Denpasar, Bali
yudi@stikom-bali.ac.id

Abstract: K-Means is a type of unsupervised classification method which partitions data items into one or more clusters. K-Means tries to model a dataset into clusters so that data items in a cluster have similar characteristic and have different characteristics from the other clusters. In this paper, the development of K-Means and problems usually involved when using the method are illustrated. Some related information are also explained including the method for choosing the most appropriate number of clusters, the issue between supervised and unsupervised classification, an extended development of K-Means which using the kernel trick and mixture modelling which is similar to K-Means in terms of the algorithm used. The algorithm of the methods described in this paper are also provided.

Keywords: K-Means, Membership Function, Mixture Modelling, Supervised and Unsupervised.

1. Pendahuluan

Data Clustering merupakan salah satu metode *Data Mining* yang bersifat tanpa arahan (*unsupervised*). Ada dua jenis data clustering yang sering dipergunakan dalam proses pengelompokan data yaitu *hierarchical* (hirarki) data clustering dan *non-hierarchical* (non hirarki) data clustering. *K-Means* merupakan salah satu metode data clustering non hirarki yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih cluster/kelompok. Metode ini mempartisi data ke dalam cluster/kelompok sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu cluster yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam kelompok yang lain. Adapun tujuan dari data clustering ini adalah untuk meminimalisasikan *objective function* yang diset dalam proses clustering, yang pada umumnya berusaha meminimalisasikan variasi di dalam suatu cluster dan memaksimalisasikan variasi antar cluster.

Data clustering menggunakan metode *K-Means* ini secara umum dilakukan dengan algoritma dasar sebagai berikut[6]:

1. Tentukan jumlah cluster
2. Alokasikan data ke dalam cluster secara random
3. Hitung *centroid*/rata-rata dari data yang ada di masing-masing cluster
4. Alokasikan masing-masing data ke *centroid*/rata-rata terdekat
5. Kembali ke Step 3, apabila masih ada data yang berpindah cluster atau apabila perubahan nilai *centroid*, ada yang di atas nilai *threshold* yang ditentukan atau apabila perubahan nilai pada *objective function* yang digunakan di atas nilai *threshold* yang ditentukan

Dalam tulisan ini beberapa hal terkait dengan metode *K-Means* ini berusaha untuk dijelaskan, termasuk di antaranya beberapa pengembangan yang telah dilakukan terhadap *K-Means*, beberapa permasalahan yang harus diperhitungkan dalam menggunakan metode *K-Means* dalam pengelompokan data, ulasan mengenai keberadaan *K-Means* di antara metode pengklasifikasian dengan arahan (*supervised)* dan tanpa arahan (*unsupervised*), ulasan singkat mengenai metode *K-Means* untuk *dataset* yang mempunyai bentuk khusus dan *mixture modelling*, serta algoritma dari metode-metode pengelompokan yang masih digolongkan sebagai pengembangan metode *K-Means*.

2. Perkembangan Penerapan *K-Means*

Beberapa alternatif penerapan *K-Means* dengan beberapa pengembangan teori-teori penghitungan terkait telah diusulkan. Hal ini termasuk pemilihan:
1. *Distance space* untuk menghitung jarak di antara suatu data dan *centroid*[1,3,7,8,9]
2. Metode pengalokasian data kembali ke dalam setiap *cluster*[1,3,6,7,8,9,16]
3. *Objective function* yang digunakan[1,3,6,7,8,9,16]

2.1. *Distance Space* Untuk Menghitung Jarak Antara Data dan *Centroid*

Beberapa *distance space* telah diimplementasikan dalam menghitung jarak (*distance*) antara data dan *centroid* termasuk di antaranya L_1 (*Manhattan/City Block*) *distance space*[9], L_2 (*Euclidean*) *distance space*[3], dan L_p *(Minkowski) distance space*[9]. Jarak antara dua titik x_1 dan x_2 pada *Manhattan/City Block distance space* dihitung dengan menggunakan rumus sebagai berikut[8]:

$$D_{L_1}(x_2, x_1) = \|x_2 - x_1\|_1 = \sum_{j=1}^{p} |x_{2j} - x_{1j}| \quad (1)$$

dimana:

p : Dimensi data
$|\,.\,|$: Nilai absolut

Sedangkan untuk L_2 (*Euclidean*) *distance space*, jarak antara dua titik dihitung menggunakan rumus sebagai berikut[3]:

$$D_{L_2}(x_2, x_1) = \|x_2 - x_1\|_2 = \sqrt{\sum_{j=1}^{p} (x_{2j} - x_{1j})^2} \quad (2)$$

dimana:

p : Dimensi data

L_p (*Minkowski*) *distance space* yang merupakan generalisasi dari beberapa *distance space* yang ada seperti L_1 (*Manhattan/City Block*) dan L_2 (*Euclidean*), juga telah diimplementasikan[9]. Tetapi secara umum *distance space* yang sering digunakan adalah *Manhattan* dan *Euclidean*. *Euclidean* sering digunakan karena penghitungan jarak dalam *distance space* ini merupakan jarak terpendek yang bisa didapatkan antara dua titik yang diperhitungkan, sedangkan *Manhattan* sering digunakan karena kemampuannya dalam mendeteksi keadaan khusus seperti keberadaaan *outliers* dengan lebih baik.

2.2. Metode Pengalokasian Ulang Data ke Dalam Masing-Masing *Cluster*

Secara mendasar, ada dua cara pengalokasian data kembali ke dalam masing-masing cluster pada saat proses iterasi clustering. Kedua cara tersebut adalah pengalokasian dengan cara tegas (hard), dimana data item secara tegas dinyatakan sebagai anggota cluster yang satu dan tidak menjadi anggota cluster lainnya, dan dengan cara fuzzy, dimana masing-masing data item diberikan nilai kemungkinan untuk bisa bergabung ke setiap cluster yang ada. Kedua cara pengalokasian tersebut diakomodasikan pada dua metode *Hard K-Means*[6] dan *Fuzzy K-Means*[3,8,9]. Perbedaan di antara kedua metode ini terletak pada asumsi yang dipakai sebagai dasar pengalokasian.

Hard K-Means

Pengalokasian kembali data ke dalam masing-masing cluster dalam metode *Hard K-Means* didasarkan pada perbandingan jarak antara data dengan *centroid* setiap cluster yang ada. Data dialokasikan ulang secara tegas ke cluster yang mempunyai *centroid* terdekat dengan data tersebut. Pengalokasian ini dapat dirumuskan sebagai berikut[6]:

$$a_{ik} = \begin{cases} 1 & d = \min\{D(x_k, v_i)\} \\ 0 & lainnya \end{cases} \qquad (3)$$

dimana:

a_{ik} : Keanggotaan data ke-k ke cluster ke-i

v_i : Nilai *centroid* cluster ke-i

Fuzzy K-Means

Metode *Fuzzy K-Means* (atau lebih sering disebut sebagai *Fuzzy C-Means*) mengalokasikan kembali data ke dalam masing-masing cluster dengan memanfaatkan teori *Fuzzy*. Teori ini mengeneralisasikan metode pengalokasian yang bersifat tegas (*hard*) seperti yang digunakan pada metode *Hard K-Means*. Dalam metode *Fuzzy K-Means* dipergunakan variabel *membership function*, u_{ik}, yang merujuk pada seberapa besar kemungkinan suatu data bisa menjadi anggota ke dalam suatu cluster. Pada *Fuzzy K-Means* yang diusulkan oleh Bezdek[3], diperkenalkan juga suatu variabel m yang merupakan *weighting exponent* dari *membership function*. Variabel ini dapat mengubah besaran pengaruh dari *membership function*, u_{ik}, dalam proses clustering menggunakan metode *Fuzzy K-Means*. m mempunyai wilayah nilai

m>1. Sampai sekarang ini tidak ada ketentuan yang jelas berapa besar nilai *m* yang optimal dalam melakukan proses optimasi suatu permasalahan clustering. Nilai *m* yang umumnya digunakan adalah 2.

Membership function untuk suatu data ke suatu cluster tertentu dihitung menggunakan rumus sebagai berikut[3,8,9]:

$$u_{ik} = \sum_{j=1}^{c} \left(\frac{D(x_k, v_i)}{D(x_k, v_j)} \right)^{\frac{2}{m-1}} \quad (4)$$

dimana:

u_{ik} : *Membership function* data ke-*k* ke cluster ke-*i*

v_i : Nilai *centroid* cluster ke-*i*

m : *Weighting Exponent*

Membership function, u_{ik}, mempunyai wilayah nilai $0 \leq u_{ik} \leq 1$. Data item yang mempunyai tingkat kemungkinan yang lebih tinggi ke suatu kelompok akan mempunyai nilai *membership function* ke kelompok tersebut yang mendekati angka 1 dan ke kelompok yang lain mendekati angka 0.

2.3. *Objective Function* Yang Digunakan

Objective function yang digunakan khususnya untuk *Hard K-Means* dan *Fuzzy K-Means* ditentukan berdasarkan pada pendekatan yang digunakan dalam poin 2.1. dan poin 2.2. Untuk metode *Hard K-Means*, *objective function* yang digunakan adalah sebagai berikut[6]:

$$J(U,V) = \sum_{k=1}^{N} \sum_{i=1}^{c} a_{ik} D(x_k, v_i)^2 \quad (5)$$

dimana:

N : Jumlah data

c : Jumlah cluster

a_{ik} : Keanggotaan data ke-*k* ke cluster ke-*i*

v_i : Nilai *centroid* cluster ke-*i*

a_{ik} mempunyai nilai 0 atau 1. Apabila suatu data merupakan anggota suatu kelompok maka nilai a_{ik} =1 dan sebaliknya. Untuk metode *Fuzzy K-Means*, *objective function* yang digunakan adalah sebagai berikut[3,8,9]:

$$J(U,V) = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m D(x_k, v_i)^2 \quad (6)$$

dimana:

N : Jumlah data

c : Jumlah cluster

m : *Weighting exponent*
u_{ik} : *Membership function* data ke-*k* ke cluster ke-*i*
v_i : Nilai *centroid* cluster ke-*i*

Di sini u_{ik} bisa mengambil nilai mulai dari 0 sampai 1.

3. Beberapa Permasalahan yang Terkait Dengan *K-Means*

Beberapa permasalahan yang sering muncul pada saat menggunakan metode *K-Means* untuk melakukan pengelompokan data adalah:
1. Ditemukannya beberapa model clustering yang berbeda
2. Pemilihan jumlah cluster yang paling tepat
3. Kegagalan untuk *converge*
4. Pendeteksian *outliers*
5. Bentuk masing-masing cluster
6. Masalah *overlapping*

Keenam permasalahan ini adalah beberapa hal yang perlu diperhatikan pada saat menggunakan *K-Means* dalam mengelompokkan data. Permasalahan 1 umumnya disebabkan oleh perbedaan proses inisialisasi anggota masing-masing cluster. Proses initialisasi yang sering digunakan adalah proses inisialisasi secara random. Dalam suatu studi perbandingan[13], proses inisialisasi secara random mempunyai kecenderungan untuk memberikan hasil yang lebih baik dan independent, walaupun dari segi kecepatan untuk *converge* lebih lambat.

Permasalahan 2 merupakan masalah laten dalam metode *K-Means*. Beberapa pendekatan telah digunakan dalam menentukan jumlah cluster yang paling tepat untuk suatu *dataset* yang dianalisa termasuk di antaranya *Partition Entropy (PE)*[3] dan *GAP Statistics*[15]. Satu hal yang patut diperhatikan mengenai metode-metode ini adalah pendekatan yang digunakan dalam mengembangkan metode-metode tersebut tidak sama dengan pendekatan yang digunakan oleh *K-Means* dalam mempartisi data items ke masing-masing cluster.

Permasalahan kegagalan untuk *converge*, secara teori memungkinkan untuk terjadi dalam kedua metode *K-Means* yang dijelaskan di dalam tulisan ini. Kemungkinan ini akan semakin besar terjadi untuk metode *Hard K-Means*, karena setiap data di dalam *dataset* dialokasikan secara tegas (*hard*) untuk menjadi bagian dari suatu cluster tertentu. Perpindahan suatu data ke suatu cluster tertentu dapat mengubah karakteristik model clustering yang dapat menyebabkan data yang telah dipindahkan tersebut lebih sesuai untuk berada di cluster semula sebelum data tersebut dipindahkan. Demikian juga dengan keadaan sebaliknya. Kejadian seperti ini tentu akan mengakibatkan pemodelan tidak akan berhenti dan kegagalan untuk *converge* akan terjadi. Untuk *Fuzzy K-Means*, walaupun ada, kemungkinan permasalahan ini untuk terjadi sangatlah kecil, karena setiap data diperlengkapi dengan *membership function* (*Fuzzy K-Means*) untuk menjadi anggota cluster yang ditemukan.

Permasalahan keempat merupakan permasalahan umum yang terjadi hampir di setiap metode yang melakukan pemodelan terhadap data. Khusus untuk metode *K-Means* hal ini memang menjadi permasalahan yang cukup menentukan. Beberapa hal yang perlu diperhatikan dalam melakukan pendeteksian *outliers* dalam proses pengelompokan data termasuk bagaimana menentukan apakah suatu data item merupakan *outliers* dari suatu cluster tertentu dan apakah data dalam jumlah kecil yang membentuk suatu cluster tersendiri dapat dianggap sebagai *outliers*. Proses ini memerlukan suatu pendekatan khusus yang berbeda dengan proses pendeteksian *outliers* di dalam suatu *dataset* yang hanya terdiri dari satu populasi yang homogen.

Permasalahan kelima adalah menyangkut bentuk cluster yang ditemukan. Tidak seperti metode data clustering lainnya termasuk *Mixture Modelling*[1,7,16], *K-Means* umumnya tidak mengindahkan bentuk dari masing-masing cluster yang mendasari model yang terbentuk, walaupun secara natural masing-masing cluster umumnya berbentuk bundar. Untuk dataset yang diperkirakan mempunyai bentuk yang tidak biasa, beberapa pendekatan perlu untuk diterapkan. Hal ini akan dibahas lebih lanjut dalam Bab 5 dan Bab 6.

Masalah *overlapping* sebagai permasalahan terakhir sering sekali diabaikan karena umumnya masalah ini sulit terdeteksi. Hal ini terjadi untuk metode *Hard K-Means* dan *Fuzzy K-Means*, karena secara teori, metode ini tidak diperlengkapi *feature* untuk mendeteksi apakah di dalam suatu cluster ada cluster lain yang kemungkinan tersembunyi.

4. *Semi-Supervised Classification?*

K-Means merupakan metode data clustering yang digolongkan sebagai metode pengklasifikasian yang bersifat *unsupervised* (tanpa arahan). Pengkategorian metode-metode pengklasifikasian data antara *supervised* dan *unsupervised classification* didasarkan pada adanya *dataset* yang data itemnya sudah sejak awal mempunyai label kelas dan *dataset* yang data itemnya tidak mempunyai label kelas. Untuk data yang sudah mempunyai label kelas, metode pengklasifikasian yang digunakan merupakan metode *supervised classification* dan untuk data yang belum mempunyai label kelas, metode pengklasifikasian yang digunakan adalah metode *unsupervised classification*.

Selain masalah optimasi pengelompokan data ke masing-masing cluster, data clustering juga diasosiasikan dengan permasalahan penentuan jumlah cluster yang paling tepat untuk data yang dianalisa. Untuk kedua jenis *K-Means*, baik *Hard K-Means* dan *Fuzzy K-Means*, yang telah dijelaskan di atas, penentuan jumlah cluster untuk *dataset* yang dianalisa umumnya dilakukan secara *supervised* atau ditentukan dari awal oleh pengguna, walaupun dalam penerapannya ada beberapa metode yang sering dipasangkan dengan metode *K-Means*. Karena secara teori metode penentuan jumlah cluster ini tidak sama dengan metode pengelompokan yang dilakukan oleh *K-Means*, kevalidan jumlah cluster yang dihasilkan umumnya masih dipertanyakan.

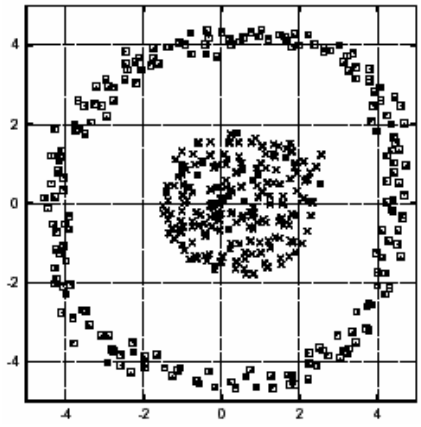
Melihat keadaan dimana pengguna umumnya sering menentukan jumlah cluster sendiri secara terpisah, baik itu dengan menggunakan metode tertentu atau berdasarkan pengalaman,

di sini, kedua metode *K-Means* ini dapat disebut sebagai metode *semi-supervised classification*, karena metode ini mengalokasikan data items ke masing-masing cluster secara unsupervised dan menentukan jumlah cluster yang paling sesuai dengan data yang dianalisa secara supervised.

5. *K-Means* untuk Data yang Mempunyai Bentuk Khusus

Beberapa *dataset* yang mempunyai bentuk tertentu memerlukan suatu metode pemecahan khusus yang disesuaikan dengan keadaan data tersebut. Gambar 1. mengilustrasikan suatu *dataset* yang mempunyai bentuk khusus yang kalau dimodel dengan metode *K-Means*, baik *Hard K-Means* dan *Fuzzy K-Means* akan memberikan hasil yang tidak mewakili keadaan *dataset* tersebut.

Untuk keperluan seperti itu, beberapa peneliti[5,10,11] telah mengusulkan pengembangan metode *K-Means* yang secara khusus memanfaatkan kernel trik, dimana *data space* untuk data awal di-mapping ke *feature space* yang berdimensi tinggi. Beberapa hal yang perlu diperhatikan dalam pengembangan metode *K-Means* dengan kernel trik ini adalah bahwa data pada *feature space* tidak lagi dapat didefinisikan secara eksplisit, sehingga penghitungan nilai *membership function* dan *centroid* tidak dapat dilakukan secara langsung. Beberapa trik penghitungan telah diusulkan dalam menurunkan nilai kedua variabel yang diperlukan tersebut[5,10,11]. Dengan penerapan trik perhitungan terhadap kedua variabel tersebut, *objective function* yang digunakan dalam menilai apakah suatu proses pengelompokan sudah *converge* atau tidak juga akan berubah.



Gambar 1. Salah Satu *Dataset* yang Mempunyai Bentuk Khusus

6. Mixture Modelling

Mixture modelling[1,7,16] merupakan salah satu jenis data clustering dimana dalam pemodelannya, data dalam suatu kelompok diasumsikan terdistribusi sesuai dengan salah satu jenis distribusi statistik yang ada. *Mixture Modelling* merupakan metode yang mempunyai cara optimasi yang sama dengan *K-Means* melalui proses optimization and maximization. Berbeda dengan metode *Hard K-Means* dan *Fuzzy K-Means*, perbandingan

jumlah data yang tercakup di dalam masing-masing cluster juga mempengaruhi hasil akhir dari proses data clustering. Perbandingan jumlah data yang terdapat di dalam masing-masing cluster sering diistilahkan dengan nama *relative abundance*.

Distribusi statistik yang paling sering digunakan dalam data clustering menggunakan metode *mixture modelling* adalah distribusi *Gaussian*/Normal. Disamping karena kemudahan penurunan berbagai rumus yang diperlukan, kecenderungan umum yang ada pada saat melakukan observasi adalah bahwa data yang didapatkan umumnya dalam keadaan terdistribusi secara normal. Berbeda dengan *K-Means*, *distance space* yang digunakan di dalam *mixture modelling* berbasis distribusi Gaussian/Normal adalah *Mahalanobis distance space*. *Mahalanobis distance space* yang sering dikaitkan dengan distribusi *multivariate Gaussian* menghitung jarak dengan rumus sebagai berikut[1,7]:

$$D_{Mahalanobis}(x_2, x_1) = \|x_2 - x_1\|_{Mahalanobis} = (x_2 - x_1)^T \Sigma^{-1} (x_2 - x_1) \qquad (7)$$

dimana:

$(.)^T$: *Transpose* dari sebuah matriks
$(.)^{-1}$: *Inverse* dari sebuah matriks
Σ : *Variance Covariance* matriks

Proses pengalokasian kembali data ke masing-masing cluster menggunakan metode *mixture modeling* umumnya sama dengan proses pengalokasian menggunakan metode *Fuzzy K-Means*. Perbedaannya terletak pada cara penghitungan nilai keanggotaan (*Fuzzy K-Means*) dan nilai probabilitas data (*Mixture Modelling*) untuk menjadi anggota suatu cluster. Penghitungan nilai kemungkinan suatu data untuk menjadi anggota suatu cluster dilakukan dengan menghitung nilai probabilitas data tersebut untuk berada pada suatu cluster dikalikan dengan *relative abundance* dari cluster yang bersangkutan seperti berikut ini[1,7.16]:

$$p_{ik} = \pi_i \times f_i(x_k | \vec{\theta}_i) \qquad (8)$$

dimana:

p_{ik} : Probabilitas data ke-*k* menjadi anggota cluster ke-*i*
π_i : *Relative abundances* cluster ke-*i*
$f_i(x_k | \vec{\theta}_i)$: Distribusi probabilitas cluster ke-*i*
$\vec{\theta}_i$: Parameter yang tercakup di dalam distribusi yang diasumsikan untuk cluster ke-*i*

Dalam *Mixture Modelling*, pemilihan jumlah cluster umumnya dilakukan dengan metode yang secara teori sama dengan metode yang digunakan untuk mendefinisikan karakteristik masing-masing cluster. Kedua kegiatan baik pendefinisian karakteristik masing-masing cluster dan pemilihan jumlah cluster yang paling tepat juga dilakukan secara simultan. Beberapa teori yang sering digunakan sebagai dasar teori dalam metode *Mixture Modelling* adalah *Penalised Maximum Likelihood* yang umumnya memasangkan metode *Maximum Likelihood* untuk mendefinisikan karakteristik masing-masing cluster dan metode pemilihan jumlah cluster seperti *Akaike Information Criterion* (AIC)[2] atau *Schwarz's Bayesian*

Information Criterion (BIC)[14]. Beberapa metode berbasis teori *Bayesian* juga sering digunakan seperti *Markov Chain Monte Carlo* (MCMC)[12] dan *Minimum Message Length* (MML)[1,16]. *AutoClass*[4], salah satu perangkat lunak *Mixture Modelling*, mengasumsikan suatu model *mixture* sebagai suatu model *Bayesian Networks*, dimana dalam pendefinisian karakter masing-masing cluster, perangkat lunak tersebut menggunakan metode *Maximum A Posteriori* (MAP).

Salah satu metode dari sekian banyak metode yang sering digunakan dalam mengevaluasi jumlah cluster adalah *Schwarz's Bayesian Information Criterion* (BIC) yang dalam proses optimasinya menggunakan rumus sebagai berikut[14]:

$$BIC = L + \frac{N_p}{2} \log N \tag{9}$$

dimana:

L : *Negative log* dari *likelihood function* untuk model yang didapat

N_p : Jumlah *free parameter* yang diestimasi

N : Jumlah data

Likelihood function untuk sudah model mixture umumnya didefinisikan dengan rumus sebagai berikut:

$$L = \sum_{k=1}^{N} \sum_{i=1}^{c} \pi_i \times f_i(x_k | \vec{\theta}_i) \tag{10}$$

dimana:

N : Jumlah data

c : Jumlah cluster

π_i : *Relative abundances* cluster ke-*i*

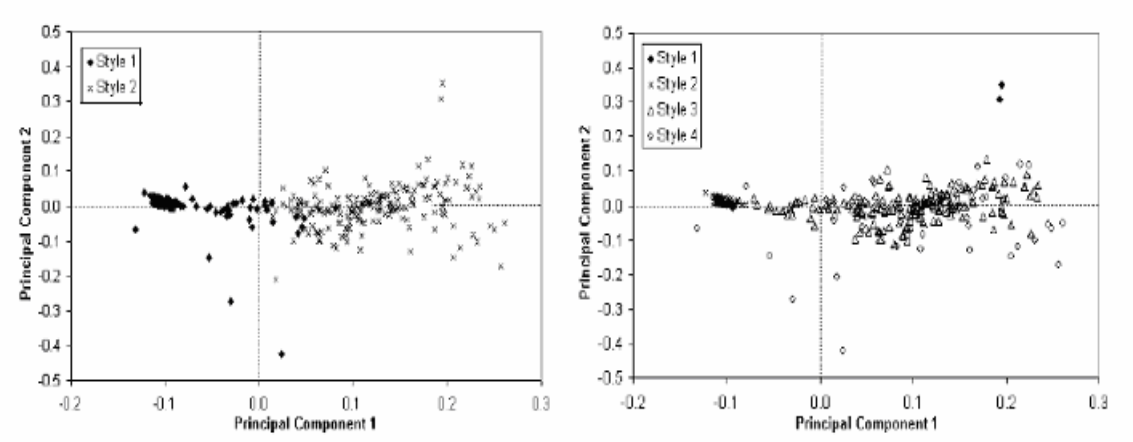
$f_i(x_k | \vec{\theta}_i)$: Distribusi probabilitas cluster ke-*i*

$\vec{\theta}_i$: Parameter yang tercakup di dalam distribusi yang diasumsikan untuk cluster ke-*i*

Beberapa kelebihan dari *Mixture Modelling* dari *K-Means* adalah adanya pengembangan metode penentuan jumlah cluster yang paling sesuai untuk suatu data tertentu yang secara teori sama dengan proses pengalokasian data item ke masing-masing cluster. Kelebihan lainnya adalah *Mixture Modelling* mempunyai kemampuan untuk mendeteksi keberadaan suatu cluster yang overlap dengan cluster yang lain. Distribusi statistik yang diterapkan di dalam *Mixture Modelling* mempunyai kelebihan dalam menangani masalah overlapping ini. Beberapa perlengkapan juga memungkinkan untuk ditambahkan dalam mengakomodasi pendeteksian *outliers* ataupun menangani bentuk-bentuk cluster yang tidak normal[1].

Gambar 2. menunjukkan perbandingan antara *K-Means* dan *Mixture Modelling* dalam memodel USA's equity dan bond funds data. Plotting yang ditunjukkan dalam gambar merupakan plotting dari nilai principal component data yang dianalisa. Dalam pemodelan ini *K-Means* dipasangkan dengan Partition Entropy (PE)[3] dalam menentukan jumlah cluster yang dianalisa, sedangkan *Mixture Modelling* mengaplikasikan prinsip Minimum Message

Length (MML)[1,16]. Dalam pemodelan ini, PE menghasilkan 2 cluster sedangkan MML menghasilkan 4 clusters.



Gambar 2. Perbandingan Hasil Pemodelan Funds Data Antara *K-Means* dan *Mixture Modelling*

Seperti yang ditunjukkan dalam gambar, *K-Means* umumnya membagi data di bagian tengah tanpa memikirkan komposisi dan keadaan data yang dimodel, sedangkan *Mixture Modelling* dengan MML membagi data dengan menyesuaikan pada keadaan data dengan melihat sebaran dan distribusi data yang dianalisa.

7. Algoritma *K-Means*

Hard K-Means

Metode *Hard K-Means* melakukan proses clustering dengan mengikuti algoritma sebagai berikut[6]:

a. Tentukan jumlah cluster
b. Alokasikan data sesuai dengan jumlah cluster yang ditentukan
c. Hitung nilai *centroid* masing-masing cluster
d. Alokasikan masing-masing data ke *centroid* terdekat
e. Kembali ke Step c. apabila masih terdapat perpindahan data dari satu cluster ke cluster yang lain, atau apabila perubahan pada nilai *centroid* masih di atas nilai *threshold* yang ditentukan, atau apabila perubahan pada nilai *objective function* masih di atas nilai *threshold* yang ditentukan.

Untuk menghitung *centroid* cluster ke-*i*, v_i, digunakan rumus sebagai berikut:

$$v_{ij} = \frac{\sum_{k=1}^{N_i} x_{kj}}{N_i} \tag{11}$$

dimana:

N_i : Jumlah data yang menjadi anggota cluster ke-*i*

Untuk penghitungan *membership function* digunakan rumus pada persamaan (3).

Fuzzy K-Means

Metode *Fuzzy K-Means* melakukan proses clustering dengan mengikuti algoritma sebagai berikut[3,8,9]:

a. Tentukan jumlah cluster
b. Alokasikan data sesuai dengan jumlah cluster yang ditentukan
c. Hitung nilai *centroid* dari masing-masing cluster
d. Hitung nilai *membership function* masing-masing data ke masing-masing cluster
e. Kembali ke Step c. apabila perubahan nilai *membership function* masih di atas nilai *threshold* yang ditentukan, atau apabila perubahan pada nilai *centroid* masih di atas nilai *threshold* yang ditentukan, atau apabila perubahan pada nilai *objective function* masih di atas nilai *threshold* yang ditentukan.

Untuk menghitung *centroid* cluster ke-*i*, v_i, digunakan rumus sebagai berikut:

$$v_{ij} = \frac{\sum_{k=1}^{N} (u_{ik})^m x_{kj}}{\sum_{k=1}^{N} (u_{ik})^m} \qquad (12)$$

dimana:

N : Jumlah data
m : *Weighting exponent*
u_{ik} : *Membership function* data ke-*k* ke cluster ke-*i*

Sedangkan untuk menghitung *membership function* data ke-*k* ke cluster ke-*i* digunakan rumus pada persamaan (4).

Mixture Modelling

Berbagai algoritma memungkinkan untuk digunakan dalam memecahkan proses optimasi *mixture modelling* termasuk di antaranya *random search*, *simulated annealing*, *Markov Chain Monte Carlo* (MCMC) maupun algoritma genetika. Untuk makalah ini, dipaparkan metode *random search* yang memberikan nilai jumlah cluster secara random di awal setiap proses optimasi. Algoritma yang digunakan adalah sebagai berikut[1]:

a. Tentukan jumlah cluster
b. Alokasikan data secara *random* ke masing-masing cluster yang telah ditentukan
    1. Hitung *means* (sama dengan *centroid* pada *K-Means*) dari masing-masing cluster
    2. Hitung standar deviasi/*variance covariance* dari masing-masing cluster
    3. Hitung nilai probabilitas masing-masing data ke masing-masing cluster
    4. Kembali ke Step b.1, apabila perubahan nilai probabilitas masih di atas nilai *threshold* yang ditentukan, atau apabila perubahan pada nilai *centroid* masih di atas nilai *threshold* yang ditentukan, atau apabila perubahan pada nilai *objective function* masih di atas nilai *threshold* yang ditentukan.
c. Kembali ke Step a. apabila masih ada jumlah cluster yang ingin dianalisa.

Dengan asumsi bahwa data terdistribusi secara normal, *means* cluster ke-*i*, μ_i, dihitung dengan menggunakan rumus sama dengan metode *Fuzzy K-Means* dengan u_{ik} merupakan nilai probabilitas data tersebut termasuk di dalam cluster ke-*i*. Sedangkan standar deviasi/*variance covariance* cluster ke-*i*, σ_i / Σ_i, dihitung dengan menggunakan rumus sebagai berikut:

$$\sigma_i = \sqrt{\frac{\sum_{k=1}^{N} (x_k - \mu_i)^2}{N-1}} \tag{13}$$

$$\Sigma_i = \frac{\sum_{k=1}^{N} (x_k - \mu_i)(x_k - \mu_i)^T}{N-1} \tag{14}$$

dimana:

N : Jumlah data

μ_i : *Means* cluster ke-*i*

sedangkan untuk menghitung nilai probabilitas data ke-*k* ke cluster ke-*i* digunakan rumus penghitungan probabilitas seperti pada persamaan (8).

8. Kesimpulan

Dari pembahasan yang telah diuraikan di dalam tulisan ini beberapa kesimpulan bisa didapatkan, termasuk:

1. Ada beberapa perkembangan penerapan yang telah diimplementasikan terhadap metode *K-Means* termasuk pemilihan *distance space*, cara pengalokasian ulang data ke cluster dan *objective function* yang digunakan. *K-Means* juga telah dikembangkan untuk bisa memodel *dataset* yang mempunyai bentuk khusus dengan memanfaatkan kernel trik.
2. Ada beberapa permasalahan yang perlu untuk diperhatikan dalam menggunakan metode *K-Means* termasuk model *clustering* yang berbeda-beda, pemilihan model yang paling tepat untuk *dataset* yang dianalisa, kegagalan untuk *converge*, pendeteksian *outliers*, bentuk masing-masing cluster dan permasalahan *overlapping*.

Daftar Pustaka

[1] Agusta, Y. (2004). Minimum Message Length Mixture Modelling for Uncorrelated and Correlated Continuous Data Applied to Mutual Funds Classification, *Ph.D. Thesis*, School of Computer Science and Software Engineering, Monash University, Clayton, 3800 Australia.

[2] Akaike, H. (1974). A New Look At The Statististical Model Identification, *IEEE Transactions on Automatic Control* AC-19(6): 716-723.

[3] Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algoritmss*, Plenum Press, New York.

[4] Cheeseman, P. and Stutz, J. (1996). Bayesian Classification (AutoClass): Theory and Results, in U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (eds), *Advances in Knowledge Discovery and Data Mining*, AAAI Press/MIT Press, Cambridge, MA, pp. 153-180.

[5] Girolami, M. (2002). Mercel Kernel Based Clustering in Feature Space, IEEE Transactions on Neural Networks, Vol. 13, No. 3, pp. 761-766.

[6] MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1: 281-297.

[7] McLachlan, G. J. and Peel, D. (2000). *Finite Mixture Models*, John Wiley and Sons, New York.

[8] Miyamoto, S. and Agusta, Y. (1995). An Efficient Algorithm for L1 Fuzzy c-Means and its Termination, *Control and Cybernetics* 24(4): 422-436.

[9] Miyamoto, S. and Agusta, Y. (1995). Algorithms for L1 and Lp Fuzzy C-Means and Their Convergence, in C. Hayashi, N. Oshumi, K. Yajima, Y. Tanaka, H. H. Bock and Y. Baba (eds), *Data Science, Classification, and Related Methods*, Springer-Verlag, Tokyo, Japan, pp. 295-302.

[10] Miyamoto S. and Nakayama, Y. (2003). Algorithms of Hard C-Means Clustering Using Kernel Functions in Support Vector Machines, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 7, No. 1, pp. 19–24.

[11] Miyamoto, S. and Suizu, D. (2003). Fuzzy C-Means Clustering Using Kernel Functions in Support Vector Machines, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 7, No. 1, pp. 25–30.

[12] Neal, R. M. (1991). Bayesian Mixture Modeling by Monte Carlo Simulation, *Technical Report CRG-TR-91-2*, Department of Computer Science, University of Toronto, Toronto, Canada.

[13] Pena, J. M., Lozano, J. A. and Larranaga, P. (1999). An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Lett.,* 20:1027-1040.

[14] Schwarz, G. (1978). Estimating the Dimension of a Model, *The Annals of Statistics* 6: 461 – 464.

[15] Tibshirani, R., Walter, G. and Hastie, T. (2000). Estimating the Number of Clusters in a Dataset using the Gap Statistics, *Technical Report 208*, Department of Statistics, Stanford University, Standford, CA 94305, USA.

[16] Wallace, C. S. and Boulton, D. M. (1968). An Information Measure for Classification, *Computer Journal* 11(2): 185-194.