Algoritma Pengklusteran Pautan Tunggal

Rito Goejantoro

Program Studi Statistika, FMIPA Universitas Mulawarman
Jl. Barong Tongkok no.5 Kampus Unmul Gn. Kelua Sempaja Samarinda 75119

Abstrak
Analisis kluster adalah teknik yang digunakan untuk menggabungkan objek-objek ke dalam grup atau kluster sedemikian sehingga setiap grup atau kluster homogen terhadap karakteristik tertentu, dan setiap grup harus berbeda dari grup lainnya terhadap karakteristik yang sama. Metode pengelompokan pautan tunggal adalah salah satu metode analisis kluster. Ada beberapa jenis algoritma pengklusteran pautan tunggal. Beberapa modifikasi dapat membuat algoritma menjadi lebih efisien.

Kata Kunci : algoritma, analisis kluster, efisien, pautan tunggal

Pendahuluan
Metode pengklusteran pautan tunggal (single-link clustering method) adalah salah satu metode analisis kluster. Metode ini dikenal dengan nama lain yaitu metode minimum, metode tetangga terdekat (the nearest neighbor method) dan metode keterhubungan (the connectedness method) (Hartigan, 1975).

Metode pautan tunggal ini menggunakan jarak tetangga terdekat untuk mengukur ketakmiripan antara dua grup. Misalkan C_i, C_j dan C_k adalah tiga grup data maka jarak antara C_k dan C_i & C_j adalah minimum $\{D(C_k,C_i), D(C_k,C_j)\}$ dengan $D(C,C')= \min d(x,y)$, $x \in C$ dan $y \in C'$
Analisis kluster mengklasifikasikan objek yaitu responden, produk dll pada himpunan karakteristik yang dipilih.

Permasalahan
Masalah pengklusteran pautan tunggal adalah mengelompokkan n objek sedemikian sehingga setiap grup atau kluster homogen terhadap karakteristik tertentu, dan setiap grup harus berbeda dari grup lainnya terhadap karakteristik yang sama. Dan bagaimana membuat algoritma yang efisien, yang membutuhkan waktu yang lebih singkat dan memerlukan penyimpanan yang kecil ?

Manfaat
Analisis kluster digunakan dalam berbagai bidang seperti ilmu alam, ilmu medis, ekonomi, pemasaran dll. Dalam pemasaran sebagai contoh analisis kluster berguna untuk membentuk dan mendeskripsikan segmen pasar yang berbeda dari survei konsumen. Perusahan asuransi tertarik pada perbedaan kelas dari pelanggan yang potensial sehingga dapat menentukan harga optimal untuk pelayanannya.

Tujuan
Membuat algoritma pengklusteran pautan tunggal yang efisien.

Definisi dasar
Rantai dari objek i ke objek j adalah barisan terurut dari objek o_1, o_2, …, o_t dengan o_1=i dan o_t= j. Tidak ada objek yang muncul lebih dari satu dalam rantai yang diberikan. Kardinalitas t dari rantai didefinisikan sebagai banyaknya objek dalam rantai. Panjang dari rantai adalah jumlah nilai ketakmiripan antara pasangan objek yang berdekatan dalam rantai yang diberikan. Ukuran rantai didefinisikan sebagai nilai ketakmiripan terbesar.

Dua objek i dan j anggota dari kluster pautan tunggal yang sama pada tingkat pengklusteran Δ jika dan hanya jika terdapat rantai berukuran kurang dari atau sama dengan Δ yang menghubungkan kedua objek ini.

Pengklusteran *C* adalah partisi n objek menjadi k himpunan (kluster) yang saling lepas C_1, C_2,.., C_k.Hasil akhir dari analisis kluster pautan tunggal dalam skema pengklusteran hirarki adalah barisan kluster yang berbeda : C_0, C_1,..., C_ω di mana $0 \le \omega \le n-1$. C_0 adalah pengklusteran terlemah di mana ada n kluster yang masing-masing hanya memuat satu objek. Pengklusteran ini tersarang yaitu setiap kluster dalam C_{k+1} adalah kluster dalam C_k atau gabungan dari dua atau lebih kluster dalam C_k.

Skema pengklusteran hirarki selalu dapat digambarkan sebagai dendogram yaitu diagram seperti pohon di mana n objek dinyatakan oleh ujung ranting. Barisan penggabungan kluster dinyatakan dengan menghubungkan ranting ke cabang dan akhirnya menjadi batang tunggal. Sebagai contoh, dendogram untuk skema

pengklusteran hirarki yang diberikan dalam tabel 1 ditunjukkan dalam gambar 1.

Perhitungan yang diperlukan algoritma dinyatakan dengan notasi O(f(n)) yang berarti bahwa untuk $n \rightarrow \infty$ waktu yang diperlukan proporsional dengan f(n)

Algoritma

Gan (2007) mengklasifikasikan algoritma pengklusteran pautan tunggal menjadi 5 jenis :

1. Algoritma keterhubungan
2. Algoritma berdasarkan transformasi ultrametrik
3. Algoritma estimasi kepadatan peluang
4. Algoritma aglomeratif
5. Algoritma berdasarkan pohon pembangun minimum.

Algoritma aglomeratif untuk analisis kluster pautan tunggal adalah yang paling terkenal. Algoritmanya adalah sebagai berikut :

Algoritma 1 :

a. Misalkan $C=C_0$ adalah kluster terlemah, di mana setiap kluster C_k hanya memuat satu objek k

b. Menentukan pasangan kluster (C_a,C_b) yang jaraknya paling kecil (paling sedikit ketakmiripannya)

c. Menggabungkan kluster C_a dan C_b menjadi satu kluster C_a ($C_a \cup C_b \rightarrow C_a$) dan membuang C_b dari *C*. Nilai Δ disimpan sebagai tingkat pengklusteran saat kluster baru terbentuk.

d. Mengulangi langkah b dan c untuk semua pasangan kluster (jika ada) yang mempunyai jarak terkecil yang sama.

e. Menghitung kembali jarak kluster yang baru C_a dengan kluster lainnya.

f. Mengulangi langkah b, c dan d sampai hanya sampai terbentuk satu kluster

Algoritma umum ini dapat diimplementasikan dalam banyak cara. Salah satunya dengan menggunakan matriks n x n dari jarak antara semua pasangan objek. Bila pasangan objek i dan j digabungkan, baris dan kolom i dan j yang bersesuaian dihilangkan dari matriks, dan baris dan kolom i' yang baru untuk kluster yang baru ditambahkan. Jadi dalam langkah c, dimensi baris dan kolom matriks berkurang satu setiap kali ada penggabungan.

Matriks jarak cxc dari c kluster yang ada sekarang harus diperiksa untuk menentukan jarak terkecil. Nilai c mula-mula adalah n dan berkurang satu bila ada penggabungan. Karena itu total banyaknya koefisien jarak yang harus diperhatikan adalah $O(n^3)$.

Perhitungan dapat dikurangi dengan menggunakan kenyataan bahwa pada saat memperbarui matriks jarak dalam langkah d, jarak antara dua objek (kluster) tidak dipengaruhi oleh penggabungan yang hanya melibatkan kluster lainnya. Jadi seluruh matriks tidak perlu dicari dalam langkah b tetapi hanya beberapa perubahan lokal yang dibuat sebagai akibat penggabungan. Sebagai contoh, bila dua kluster C_a dan C_b digabungkan, daftar tetangga terdekat setiap kluster yang harus diperbarui hanya untuk kluster yang tetangga terdekatnya kluster a atau kluster b.

Algoritma juga dapat diperbaiki dengan menetapkan semua pasangan kluster terdekat dalam langkah b, daripada hanya satu pasangan kluster dengan jarak minimum. Semua pasangan dapat digabungkan secara bersama-sama dalam langkah c. Dengan modifikasi ini algoritma akan mendekati $O(n^2)$.

Algoritma yang lebih efisien dapat diperoleh dari kenyataan bahwa hanya perubahan lokal dalam matriks ketakmiripan dari penggabungan dua kluster. Jika dimulai dengan dendogram yang terdiri dari objek tunggal dan membentuk dendogram akhir secara rekursif dengan menambahkan n-1 objek yang tersisa satu demi satu dengan urutan sembarang. Algoritmanya sebagai berikut :

Algoritma 2 :

a. Dimulai dengan : $1 \rightarrow \Pi_1$, $\infty \rightarrow \Lambda_1$,$1 \rightarrow m$ (Π dan Λ merupakan penyajian pointer dari dendogram)
b. Membuat : $m+1 \rightarrow \Pi_{m+1}$ dan $\infty \rightarrow \Lambda_{m+1}$. Membuat : $d_{i,m+1} \rightarrow M_i$ untuk i=1,2,…,m
c. Membuat $1 \rightarrow i$
d. Jika $\Lambda_i \geq M_i$ maka $\min\{ M_{\Pi_i}, \Lambda_i \} \rightarrow M_{\Pi_i}$, $M_i \rightarrow \Lambda_i$ dan $m+1 \rightarrow \Pi_i$. Jika tidak ($\Lambda_i < M_i$) : $\min\{ M_{\Pi_i}, M_i \} \rightarrow M_{\Pi_i}$
e. Jika $i < m$ maka $i+1 \rightarrow i$ dan ke langkah d; jika tidak lanjutkan
f. $1 \rightarrow i$
g. Jika $\Lambda_i \geq \Lambda_{\Pi_i}$ maka $m+1 \rightarrow \Pi_i$
h. Jika $i < m$ maka $i+1 \rightarrow i$ dan ke langkah d; jika tidak lanjutkan
i. Jika $m < n$ maka $n+1 \rightarrow n$ dan ke langkah b; jika tidak berhenti.

Dalam algoritma di atas Π dan Λ merupakan penyajian pointer dari dendogram. Λi adalah tingkat pengklusteran di mana objek i bukan objek terakhir yang terdaftar dalam klusternya dan Πi adalah objek terakhir dalam kluster di mana objek i bergabung. Penyajian pointer adalah sepasang fungsi:

$\Pi : \{1,2,\ldots,n\} \rightarrow \{1,2,\ldots,n\}$ dan

$\Lambda : \Pi(\{1,2,\ldots,n\}) \rightarrow [0,\infty]$

yang mempunyai sifat berikut :

$\Pi(n) = n$, $\Pi(i) > i$ untuk $i < n$

$\Lambda(n) = \infty$, $\Lambda(\Pi(i)) > \Lambda(i)$ untuk $i < n$

di mana n banyaknya objek
Larik M digunakan untuk penyimpan antara. Perhitungan untuk melakukan algoritma adalah O(n2). Algoritma ini mempunyai sifat bahwa hanya satu baris matriks ketakmiripan yang diperlukan selama iterasi pada langkah b. Jadi matriks ketakmiripan n x n tidak perlu ditempatkan dalam penyimpan akses cepat selama eksekusi. Hal ini memberikan keuntungan besar bila n besar.

Ilustrasi
Sebagai ilustrasi, misalkan diketahui matriks ketakmiripan antara 5 objek :

$$D = \{d_{ik}\} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & 2 & 8 & 0 \end{bmatrix} \end{array}$$

Karena jarak min (d_{ik}) = d_{53} = 2 maka objek 5 dan 3 digabungkan sehingga membentuk kluster {3,5}. Selanjutnya perlu dicari jarak kluster {3,5} dengan objek 1, 2 dan 4. Jarak terkecil adalah :

$d_{(35),1} = \min \{d_{31}, d_{51}\} = \min \{3,11\} = 3$

$d_{(35),2} = \min \{d_{32}, d_{52}\} = \min \{7,10\} = 7$

$d_{(35),4} = \min \{d_{34}, d_{54}\} = \min \{9,8\} = 8$

Menghilangkan baris dan kolom D yang bersesuaian dengan objek 3 dan 5 dan menambahkan baris dan kolom untuk kluster {3,5} sehingga diperoleh matriks jarak

$$\begin{array}{c} \\ \{3,5\} \\ 1 \\ 2 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} \{3,5\} & 1 & 2 & 4 \end{array} \\ \begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 7 & 9 & 0 & \\ 8 & 6 & 5 & 0 \end{bmatrix} \end{array}$$

Jarak terkecil antara pasangan kluster sekarang adalah $d_{(35)1}$ = 3 sehingga kluster {1} dan kluster {3,5} digabungkan menjadi kluster {1,3,5}.
Selanjutnya dihitung jarak kluster {1,3,5} dengan 2 dan 4

$d_{(135)2} = \min \{d_{(35)2}, d_{12}\} = \min\{7,9\} = 7$

$d_{(135)4} = \min \{d_{(35)4}, d_{14}\} = \min\{8,69\} = 6$

sehingga diperoleh matriks jarak baru :

$$\begin{array}{c} \\ \{1,3,5\} \\ 2 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} \{1,3,5\} & 2 & 4 \end{array} \\ \begin{bmatrix} 0 & & \\ 7 & 0 & \\ 6 & 5 & 0 \end{bmatrix} \end{array}$$

Jarak terkecil antara kluster adalah d_{42} = 5 sehingga objek 4 dan 2 digabungkan menjadi kluster {2,4}.. Sekarang ada dua kluster {1,3,5} dan {2,4}, jarak terkecilnya adalah :
$d_{(135)(24)} = \min \{ d_{(135)2}, d_{(135)4} \} = \min \{7,6\} = 6$

Matriks jarak terakhir menjadi

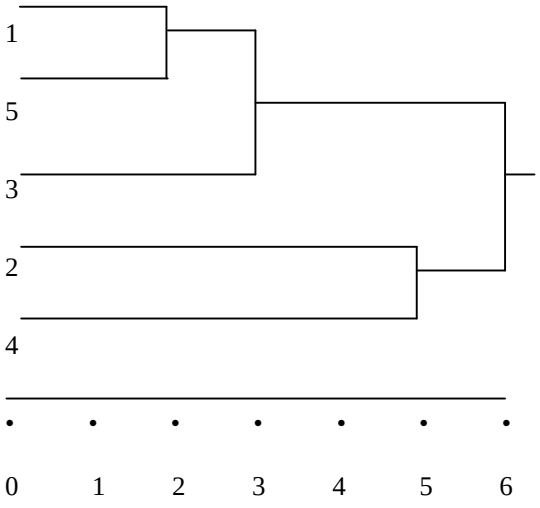
$$\begin{array}{c} \\ \{1,3,5\} \\ \{2,4\} \end{array} \begin{array}{c} \begin{array}{cc} \{1,3,5\} & \{2,4\} \end{array} \\ \begin{bmatrix} 0 & \\ 6 & 0 \end{bmatrix} \end{array}$$

Akhirnya kluster {1,3,5} dan kluster {2,4} digabungkan sehingga terbentuk satu kluster.

Tabel 1 Skema pengklusteran pautan tunggal

K	Δ_K	Pengklusteran C_K
0	0	{1},{2},{3},{4},{5}
1	2	{1,5},{2},{3},{4}
2	3	{1,3,5},{2},{4}
3	5	{1,3,5},{2,4}
4	6	{1,3,5,2,4}

Berikut ini adalah dendogram pengklusteran hirarki :



Gambar 1 Dendogram

Kesimpulan

Dari ilustrasi di atas tidak mudah melakukan analisis kluster secara manual untuk jumlah objek yang sedikit apalagi untuk objek yang banyak sehingga diperlukan bantuan komputer. Algoritma 1 dapat dimodifikasi agar menjadi lebih efisien. Algoritma 2 lebih efisien daripada algoritma 1.

Daftar Pustaka

Gan, Guojun, 2007 Data Clustering, Theory, Algorithms, and Applications, SIAM, Philadelphia

Gordon, A.D., 1999 Classification, Chapman & Hall, New York

Hartigan, J. A.,1975 Clustering Algorithms, Wiley, New York

Sharma, Subhash, 1996 Applied Multivariate Techniques, Wiley, New York