No Makalah : 199

PENINGKATAN AKURASI SHARED NEAREST NEIGHBOR MENGGUNAKAN MULTIPATH COMPONENT DISTANCE

Ricky Eka Putra
Jurusan Teknik Informatika, Fakultas Teknologi Informasi
Institut Teknologi Adhi Tama Surabaya, Surabaya - 60117
rickyeka25@yahoo.com

Abstrak

Shared Nearest Neighbor (SNN) merupakan cara terbaik untuk mengatasi permasalahan besarnya dimensi data dalam proses *clustering*. Salah satu permasalahan yang muncul dalam SNN adalah akurasi pembentukan klaster. Akurasi ini ditentukan oleh algoritma perhitungan kemiripan antar node dalam graph ketetanggaan. Perhitungan kemiripan dalam SNN menggunakan Squared Euclidean Distance (SED). Untuk meningkatkan akurasi, diusulkan sebuah metode untuk mengganti Euclidean distance, yaitu algoritma Multipath Component Distance (MCD). Dari hasil ujicoba komparasi dengan dataset Opinosis antara Euclidean distance dan MCD dalam SNN, diperoleh peningkatan akurasi dari 58,82 % menjadi 76,47 %

Kata kunci : *Shared Nearest Neighbor, Multipath Component Display, temu kembali informasi*

1. Pendahuluan

Nearest neighbor merupakan algoritma pengklasifikasian yang bergantung pada jarak antar *record*. Namun dari beberapa penelitian, diketahui bahwa penggunaan jarak tidak sebanding dengan dimensi data. Semakin besar dimensi, semakin kecil nilai efisiensi dan efektifitas algoritma tersebut. Guha [1] menyatakan bahwa semakin besar dimensi data, maka jarak atau kesamaan diantara titik-titik *data* menjadi semakin sama.

Shared Nearest Neighbor (SNN) merupakan cara terbaik untuk mengatasi permasalahan besarnya dimensi data [2]. Setelah *Nearest Neighbor* ditentukan, algoritma ini mencari nilai kesamaan diantara titik-titik data dengan cara menghitung jumlah *neighbor* yang dimiliki secara bersama-sama (*share*).

SNN membutuhkan sebuah ambang batas untuk menentukan penggabungan atau pemisahan *cluster* data. Bahkan dapat terjadi tidak adanya nilai ambang batas yang sesuai untuk beberapa *dataset*. Pengembangan SNN berbasis kepadatan dikembangkan untuk mengatasi hal ini. Penggunaan *graph* menjadi kekuatan utama dalam penentuan titik-titik representatif. Dari titik-titik ini, nilai kepadatan diketahui. Data dengan nilai kepadatan diatas ambang batas akan dipilih sebagai titik representatif.

Salah satu permasalahan yang muncul dalam SNN berbasis kepadatan adalah akurasi. Tingkat akurasi ini ditentukan oleh algoritma perhitungan kemiripan antar node dalam *graph*. Banyak algoritma yang dikembangkan untuk menghitung kemiripan dan salah satunya adalah Multipath Component Distance (MCD) [3]. Dalam penelitian ini, masih digunakan algoritma utama dari SNN. Implementasi algoritma MCD pada SNN adalah menggantikan algoritma Squared Euclidean Distance (SED). Dengan menggunakan MCD dalam *clustering*, diharapkan mampu meningkatkan akurasi *clustering*.

Pada tulisan ini dibahas bagaimana MCD mampu meningkatkan performa SNN. Bagian II membahas tentang algoritma SNN beserta kompleksitasnya dan algoritma MCD. Pada bagian III dibahas tentang metodologi penelitian yang dipakai dalam penelitian ini, khususnya pada implementasi MCD pada SNN. Sedangkan uji coba komparasi MCD dengan Euclidean distance akan dibahas pada bagian IV.

2. Kajian Pustaka

2.1 Shared Nearest Neighbor

SNN sendiri merupakan algoritma *clustering* berbasis *graph* dengan prinsip jika dua titik memiliki kesamaan terhadap titik yang sama banyak, maka kedua titik tersebut memiliki kesamaan satu sama lain, bahkan jika pengukuran kesamaan tidak menunjukkan kesamaan tersebut.

Ide pokok dari algoritma ini adalah mengambil sejumlah titik-titik data untuk menentukan pengukuran kesamaan. Kesamaan dalam algoritma SNN didasarkan pada jumlah tetangga yang dimiliki secara bersama-sama selama kedua obyek

terdapat dalam daftar tetangga terdekat masing-masing.

Pengembangan SNN berbasis kepadatan merupakan solusi dari kekurangan pada algoritma Jarvis Patrick (JP) [4]. Algoritma JP merupakan algoritma yang menggunakan SNN sebagai dasar pengembangannya. Algoritma JP ini menggantikan jarak antara dua titik dengan nilai kesamaan. Sebuah nilai ambang batas kemudian digunakan untuk melakukan sparsifikasi *graph* ketetanggaan. *Cluster* diperoleh dari komponen *graph* ketetanggaan yang masih berhubungan.

Dalam SNN berbasis kepadatan, digunakan penggabungan algoritma kesamaan SNN dengan proses penentuan titik representatif dari algoritma DBSCAN dan CURE.

Karena itulah, SNN berbasis kepadatan dianggap sebagai salah satu algoritma pengelompokan yang efisien karena:

1. Dapat menentukan dan membentuk matriks kesamaan antara titik-titik data berdasaran jumlah ketetanggaan yang dimiliki secara bersama-sama oleh masing-masing titik data tersebut.
2. Membentuk *cluster* dari titik-titik representatif yang ditentukan melalui bobot *graph* ketetanggaan yang dibentuk dari matriks kesamaan.

Secara garis besar, algoritma SNN berbasis kepadatan memiliki empat prosedur utama, yaitu:

1. Perhitungan nilai kesamaan dan pembentukan daftar *k* tetangga terdekat
2. Pembentukan *graph* ketetanggaan dari daftar k tetangga terdekat
3. Menentukan titik-titik representative dari *graph* ketetanggaan.
4. Membentuk *cluster* dari titik-titik representatif.

Implementasi algoritma SNN dalam program adalah sebagai berikut:

SNN $(D, k, toptc, merge)$:

1. Matriks I_{ij} ← Dataset D ;
2. $nrow$ ← jumlah record dataset D ; $ncol$ ← jumlah dimensi dataset D ;
3. k ← jumlah tetangga terdekat ; $toptc$ ← nilai kepadatan ;
4. $merge$ ← nilai bobot ketetanggaan ;
5. **SNN-SIMILARITY** $(I_{ij}, k, nrow, ncol)$;
6. Inisialisasi **vector** O_i ;
7. **SNN-MERGING** $(I_{ij}, nrow, toptc, merge, O_i)$;
8. Return O_i ;

SNN-SIMILARITY $(I_{ij}, k, nrow, ncol)$:

9. inisialisasi **vector** Q_i ;
10. **for** $m = 0$ **to** $(nrow - 1)$ **do**
11. **for** $n = m + 1$ **to** $(nrow - 1)$ **do**
12. **for** $p = 0$ **to** $(ncol - 1)$ **do**
13. $Q_m \leftarrow sqrt((I_{mp} - I_{np}) \times (I_{mp} - I_{np}))$;
14. **push** Q_m;
15. **end for**
16. **end for**
17. **end for**
18. inisialisasi **matriks** G_{ij};
19. **for** $m = 0$ **to** $(nrow - 1)$ **do**
20. **for** $n = 0$ **to** $(k - 1)$ **do**
21. **pop** Q_m;
22. $g_m \leftarrow Q_n$;
23. $g_n \leftarrow n + 1$;
24. **end for**
25. urutkan g_{mn} secara menanjak berdasarkan nilai dalam g_{mn} ;
26. **end for**
27. **SNN-CONSTRUCTION** $(g_{ij}, k, nrow)$;

SNN-MERGING $(N_{ij}, nrow, toptc, merge, O_i)$:

28. inisialisasi **vector** $density_i$;
29. **for** $m = 0$ **to** $(nrow - 1)$ **do**
30. **for** $n = 0$ **to** $(nrow - 1)$ **do**
31. **if** $N_{mn} \geq (merge \times nrow)$ **then**
32. $density_m \leftarrow density_m + 1$;
33. **end for**
34. **end for**
35. **for** $m = 0$ **to** $(nrow - 1)$ **do**
36. **if** $density_m \geq (toptc \times nrow)$ **then**
37. **if** O_m belum memiliki tanda *cluster* **then**
38. tandai O_m sebagai *cluster* baru ;
39. **for** $n = 0$ **to** $(nrow - 1)$ **do**
40. **if** $N_{mn} \geq (merge \times nrow)$
41. **and** (O_n belum memiliki tanda *cluster*) **then**
42. masukkan O_n kedalam *cluster* yang sama ;
43. **end for**
44. **end if**
45. **end for**

SNN-CONSTRUCTION $(g_{ij}, k, nrow)$:

46. inisialisasi **matriks** g_{ij} ;
47. **for** $m = 0$ **to** $(nrow - 1)$ **do**
48. **for** $n = 0$ **to** $(nrow - 1)$ **do**
49. **for** $p = 0$ **to** $(k - 1)$ **do**
50. **if** $g_{mp} = g_{np}$ **then**
51. $N_{mn} \leftarrow N_{mn} + 1$;
52. **continue** ;
53. **end if**
54. **end for**
55. **end for**
56. **end for**

Perhitungan similarity pada SNN-SIMILARITY menggunakan Squared Euclidean Distance (SED) seperti yang terlihat pada baris 10

sampai dengan 17. Pada penelitian ini, untuk dapat meningkatkan performa *clustering*, diusulkan sebuah metode yaitu Multipath Component Distance (MCD), yang akan dibahas pada sub-bab 2.2, untuk menggantikan Euclidean distance.

2.2 Multipath Component Distance

Sering kali kita jumpai geometri berbasis model kanal stokastik dengan menggunakan konsep *scattering clusters* yang berisi sejumlah komponen *multipath* stokastik yang beragam.

Masalah praktis yang utama dari model-model ini parametrisation *cluster* secara akurat, maka secara otomatis mengidentifikasi *cluster* dari data pengukuran dan ekstrak karakteristik mereka.

Titik awalnya adalah sejumlah besar data estimasi dari kanal parametrik multidimensi. Adapun parameter yang muncul dalam *cluster* pada kumpulan dari *multipath components* (MPCs) dengan parameter yang *similar*, seperti *delay, angles-of-arrival* (AoA) dan *angles-of-departure* (AoD).

Proses *clustering* dapat diperoleh dengan inspeksi visual, yang menjadi sangat rumit pengukurannya jika data yang digunakan besar. Akhir-akhir ini, pendekatan heuristik semi otomatis berdasar pada estimasi parametrik *windowed clustering* telah dikenalkan.

MCD telah digunakan dengan algoritma *clustering* pohon hirarki untuk mengidentifikasi *scattering clusters*.

Data masukan untuk algoritma *clustering* SNN adalah berupa MPCs yang dinyatakan dalam sebuah array $L\ x\ P$, dimana L adalah jumlah MPCs dan P adalah jumlah parameter kanal yang terestimasi. Secara khusus, dimensi dari P adalah *delay* (τ), *azimuth* dan *eleveation* AoA $(\varphi_{AoA}, \theta_{AoA})$, dan *azimuth* dan *eleveation* AoD $(\varphi_{AoD}, \theta_{AoD})$. Setiap MPC ditugaskan ke *cluster centroid* dengan jarak terkecil. Algoritma secara iteratif ini mengoptimalkan posisi *centroid* dalam rangka untuk meminimalkan total jarak dari setiap MPC ke centroid. Adapun total jarak tersebut dapat dihitung melalui Persamaan 1.

$$D = \sum_{l=1}^{L} d\left(x_l, c_{x_l}\right) \quad (1)$$

dimana x_l menunjukkan vektor parameter dari MPC ke-l. Sedangkan c_{x_l} menunjukkan parameter-parameter dari *cluster centroid* terdekat ke MPC ke-l, dan $d(.)$ menunjukkan fungsi jarak antara dua titik dalam parameter ruang.

MCD memungkinkan mengabungkan parameter yang datang dalam unit yang berbeda. Untuk *angular data* diberikan Persamaan 2.

$$\mathrm{MCDAoA/AoD,ij} = \frac{1}{2}\left| \begin{pmatrix} \sin(\theta_i)\cos(\varphi_i) \\ \sin(\theta_i)\sin(\varphi_i) \\ \cos(\theta_i) \end{pmatrix} - \begin{pmatrix} \sin(\theta_j)\cos(\varphi_j) \\ \sin(\theta_j)\sin(\varphi_j) \\ \cos(\theta_j) \end{pmatrix} \right| \quad (2)$$

Penghitungan MCD untuk AoA dan AoD dilakukan secara terpisah dengan menggunakan Persamaan 2. Sedangkan untuk jarak *delay* diperoleh dari Persamaan 3.

$$\mathrm{MCD}_{\tau,\mathrm{ij}} = \zeta . \frac{\left|\tau_i - \tau_j\right|}{\Delta\tau_{\max}} . \frac{\tau_{std}}{\Delta\tau_{\max}} \quad (3)$$

dengan $\Delta\tau_{\max} = \max_{i,j}\left\{\left|\tau_i - \tau_j\right|\right\}$, τ_{std} menjadi standar deviasi dari *delay* sedangkan ζ menjadi faktor skala *delay* yang cocok untuk memberikan *delay* yang lebih "penting" ketika diperlukan. Hal ini mempunyai efek keuntungan ketika melakukan *clustering* pada *real-world data*. Jika tidak menunjukkan sebaliknya, kita memilih $\zeta = 1$. Lalu, kita menskalakan jarak *delay* dengan *normalised delay spread*.

Hasil dari pengukuran jariak ini dapat dilihat pada Persamaan 4.

$$\mathrm{MCD}_{\mathrm{ij}} = \sqrt{\left\|\mathrm{MCD}_{\mathrm{AoA,ij}}\right\|^2 + \left\|\mathrm{MCD}_{\mathrm{AoD,ij}}\right\|^2 + \mathrm{MCD}_{\tau,\mathrm{ij}}^2} \quad (4)$$

yang dapat diartikan sebagai jari-jari dari sebuah (*hyper-*)*sphere* dalam ruang jarak parameter multipath yang dinormalisasi. Kita menggunakan jarak ini untuk *joint clustering*.

3. Metodologi Penelitian

Pada SNN, perhitungan kemiripan antar dokumen menggunakan algoritma Squared Euclidean Distance (SED). Untuk meningkatkan performa klasterisasi, diusulkan untuk mengganti algoritma SED dengan algoritma Multipath Component Distance (MCD).

3.1 Preprocessing

Pada tahap ini dilakukan pendaftaran semua dokumen dan term. Dokumen diambil dari dataset Opinosis versi 1.0. Dataset ini berisi kalimat yang diekstrak dari review pada topik tertentu. Ada 51 topik dalam dataset ini dan ada hampir 100 kalimat per topik. Untuk setiap dokumen dilakukan penghapusan stopword.

Setelah stopword dihilangkan, dilakukan pembobotan menggunakan metode *term frequency-inverse document frequency* (TF-IDF). *Term frequency* (TF) mencari bobot term dalam setiap dokumen, sedangkan *document frequency* (DF)

digunakan untuk mencari bobot term dalam seluruh dokumen. *Inverse document frequency* (IDF) dicari dari jumlah dokumen dibagi dengan bobot DF.

$$idf = \log\left(\frac{N}{df}\right)$$

(5)

Setelah diketahui bobot masing-masing dokumen berdasarkan *term*-nya, langkah selanjutnya adalah proses SNN itu sendiri seperti yang dijelaskan pada sub-bab 3.2.

3.2 SNN

Tahap awal SNN adalah menghitung *similarity* antar dokumen. Perhitungan ini menggunakan metode MCD dengan algoritma yang teah dijelaskan pada sub-bab 2.2. Nilai kesamaan ini dicari dengan mengukur jarak antar dokumen. Pada sinyal MIMO, terdapat delay, angle of arrival dan angle of departure. Nilai-nilai ini pada dokumen diubah menjadi: delay dianggap 0; angle of arrival = angle of departure. Sehingga rumusan akhir MCD menjadi:

$$\mathrm{MCD}_{ij} = \sqrt{2\left\|\mathrm{MCD}_{\mathrm{AoD},ij}\right\|^2}$$

(6)

Setelah jarak antar dokumen diketahui, dilakukan pembentukan graph ketetanggaan berdasarkan jumlah *k* tetangga terdekat yang telah ditentukan sebelumnya, menentukan titik representative dari graph ketetanggaan, dan pada akhirnya dibuat klaster sesuai dengan bobot ketetanggaan *topic* dan nilai *merge* yang telah ditentukan sebelumnya.

4. Hasil dan Pembahasan

Tabel 1. memberikan hasil ujicoba dari 2 SNN dengan 2 metode perhitungan jarak yang berbeda, yaitu SED dan MCD. Parameter dalam SNN secara manual diset *k=30, topic=0.02,* dan *merge=0.3*.

Tabel 1. Hasil ujicoba SNN-SED dan SNN-MCD

SNN	Jumlah Klaster	Akurasi (%)
SED	12	58,82
MCD	7	76,47

Untuk SNN SED, dari 51 dokumen dengan jumlah tetangga terdekat (k) adalah 30, bobot kepadatan (topic) 0,02 dan bobot ketetanggaan (merge) 0,3, terbentuk 12 klaster dengan akurasi 58,82 %. Sedangkan untuk SNN MCD, jumlah klaster yang terbentuk adalah 7 klaster dengan tingkat akurasi 76,47 %.

Dari hasil tersebut, sesuai dengan hipotesis bahwa MCD mampu meningkatkan akurasi SNN, yang dalam hal ini menggunakan dataset Opinosis, sebesar 17,65 % dari SED.

5. Kesimpulan

Pada tulisan ini, diajukan sebuah metode MCD untuk meningkatkan akurasi SNN. Adapun dalam penelitian ini, metode MCD ini dibandingkan dengan metode SED untuk mencari metode pengukuran jarak yang lebih baik. Dari hasil uji coba dapat diambil kesimpulan bahwa MCD mampu meningkatkan akurasi SNN sebesar 17,65 % dari SED. Sehingga metode MCD dapat digunakan untuk meningkatkan akurasi SNN.

Daftar Pustaka:

[1] Guha, S., Rastogi, R., Shim, K., 1997, *CURE: An Efficient Clustering Algorithm for Large Database*, Korea Advanced Institute of Science and Technology, Taejon.

[2] Ertoz, L., Steinbach, M., Kumar, V., 2003, *Finding Topics in Collections of Documents: A Shared Nearest Neighbor Approach*, Journal of Clustering and Information Retrieval, hal. 83-104.

[3] Czink, N., Cera, P., Salo, J., Bonek, E., Nuutinen, JP., Ylitalo, J., 2006, *Improving Clustering Performance using Multipath Component Distance*, Electronic Letters, Vol. 42, No. 1.

[4] Jarvis, RA., Patrick, EA., 1973, *Clustering Using a Similarity Measure Based on Shared Nearest Neighbors*, IEEE Transactions on Computers, Vol. C-22, No. 11.