Peningkatan Kecepatan Algoritma k-NN Untuk Sistem Pengklasifikasian Kendaraan Bermotor

Improving the Speed of K-NN Algorithm for Vehicle Classification

Wahyono
Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia
E-mail: wahyo@ugm.ac.id

Abstrak

K-Nearest Neighbor (KNN) merupakan algoritma mesin pembelajaran yang memiliki akurasi yang baik meski sangat sederhana untuk diimplementasikan. Namun, salah satu kelemahan algorima KNN adalah kecepatan komputasi yang sangat tergantung pada jumlah dataset yang dimiliki. Penelitian ini mencoba mengusulkan sebuah untuk mengimplementasikan strategi dalam rangka meningkatkan kecepatan algoritma KNN tetapi dengan akurasi yang hampir sama dengan standard KNN. Jika pada standar KNN proses hanya dilakukan dengan menyimpan data latih, yang kemudian akan dibandingkan dengan data uji baru dengan cara menghitung jarak satu persatu, sementara strategi yang diusulkan mencoba mengurangi jumlah data latih dengan strategi clustering, sehingga jumlah data yang akan dibandingkan dengan data uji lebih sedikit. Akibatnya, diharapkan waktu prosesnya menjasi lebih cepat. Strategi yang diusulkan akan diterapkan pada kasus klasifikasi jenis kendaraan berbasis pengolahan citra digital. Untuk menghitung tingkat akurasi dan kecepatan, maka metode yang diusulkan akan dievaluasi menggunakan dataset yang dikumpulkan melalui internet.

Kata kunci: k-nearest neighbor, clustering, klasifikasi jenis kendaraan, mesin pembelajaran

Abstract

K-Nearest Neighbor (KNN) is a machine learning that has a good understanding because it is very easy to implement. However, one of the weaknesses of the KNN algorithm is that speed is very dependent on the number of datasets. This paper presents a strategy to increase the speed of KNN but with the same accuracy as the standard KNN. If the KNN standard process is only done by storing training data, which will then be compared with new test data by calculating the distance one by one, while the proposed strategy tries to calculate the amount of data by clustering strategy, so that the amount of data to be compared with the test data is less . Furthermore, the processing time should be faster. The proposed strategy will be applied in the case of vehicle classification problem. The proposed method will be evaluated in term of accuracy and processing time using data sets collected via the internet.

Keywords: k-nearest neighbor, clustering, vehicle classification, machine learning

1. PENDAHULUAN

Klasifikasi merupakan permasalahan yang fundamental di banyak bidang penelitian khususnya pada bidang computer vision, salah satunya adalah untuk mengklasifikasikan jenis kendaraan berbasis pengolahan citra digital pada sistem transportasi cerdas. Tujuan dari klasifikasi adalah untuk mengklasifikasi sebuah objek yang tidak diketahui ke dalam kelas yang dimodelkan dengan sebuah algoritme pembejalaran mesin berdasarkan data latih. Ada banyak algoritme klasifikasi yang bisa diterapkan seperti antara lain, Bayes classifier, k-nearest neighbor classifier (KNN), support vector machine (SVM), artificial neural networks (ANN), random forests (RF).

Di antara beberapa algoritme pembelajaran mesin, KNN merupakan salah satu algoritme yang banyak digunakan untuk melakukan klasifikasi. KNN sangat sederhana dan mudah untuk dimplementasikan tetapi memiliki tingkat akurasi yang baik [1, 2]. Secara sederhana, KNN hanya menyimpan data latih ke dalam database yang kemudian akan digunakan untuk membandingkan antara tiap data latih dengan data uji dalam rangka melakukan klasifikasi [3] dengan mencari nilai jarak terkecil. Namun sayangnya karena KNN merupakan salah satu algoritme pembelajaran mesin berbasis lazy learning, maka keceparan proses komputasi sangat tergantung dengan jumlah data yang dimikiki. Semakin banyak data maka waktu proses membandingkannya akan semakin lama, sehingga akan sangat sulit mendapatkan proses real-time untuk kasus klasifikasi jenis kendaraan pada sistem transportasi cerdas.

Meskipun KNN telah ditemukan sejak tahun 1950an [4], sampai saat ini penelitian tentang KNN masih banyak dilakukan karena masih ada beberapa peningkatan yang bisa dilakukan terhadap metode standar KNN yaitu mengurangi jumlah data yang terlibat dalam rangka mengurangi kebutuhan memori dan meningkatkan kecepatan prosesnya. Beberapa penelitian untuk meningkatkan algoritme KNN baik dari segi akurasi maupun kecepatan telah banyak dilakukan. Salah satu cara mengurangi kebutuhan memori yang besar adalah dengan menggunakan strategi feature reduction [1], atau dengan cara mengurangi jumlah dimensi pada feature menggunakan metode seleksi dan ekstraksi ciri [5]. Metode lainnya adalah dengan menggunakan strategi boosting yang diusulkan oleh Nicolas [6]. Lebih lanjut, Viswanath [7] mengusulkan k-nearest neighbor mean classifier (k-NNMC) yang mencari nilai rata-rata dari setiap k-nearest neighbour pada tiap kelas yang didapatkan. Hasil klasifikasi kemudian ditentukan berdasarkan pada nilai rata-rata yang terkecil.

Seperti diketahui bahwa, algoritme k-NN berbasis fungsi jarak yang digunakan untuk menghitung perbedaan atau similarity antara dua objek/ciri. Oleh sebab itu pemilihan fungsi jarak sangat penting dalam algoritme KNN. Banyak penelitian menggunakan fungsi jarak Euclidean karena sederhana. Namun sayangnya fungsi jarak Euclidean memiliki kelemahan ketika menghadapi banyak data outlier [3]. Beberapa penelitian mencoba mengganti fungsi jarak Euclidean dengan fungsi jarak berbobot. Hassan [3] mengusulkan fungsi jarak berbobot berdasarkan area di bawah kurva Receiver Operating Characteristics (ROC). Metode ini menghitung bobot untuk fungsi jarak berdasarkan nilai ROC pada tetangga yang nilainya akan dihitung antara data uji dan data latih. Zhou [8] memperkenalkan fungsi jarak berbobot berdasarkan hubungan antara satu data latih dengan data latih lainnya. Sementara Jiang [9], Frank [10] dan Jiang [11] menghitung bobot berdasarkan naïve bayes yang menghasilkan tingkat akurasi yang baik [12]. Beberapa survey terkait peningkatan algoritme KNN bisa ditemukan pada paper yang ditulis oleh Jiang [13].

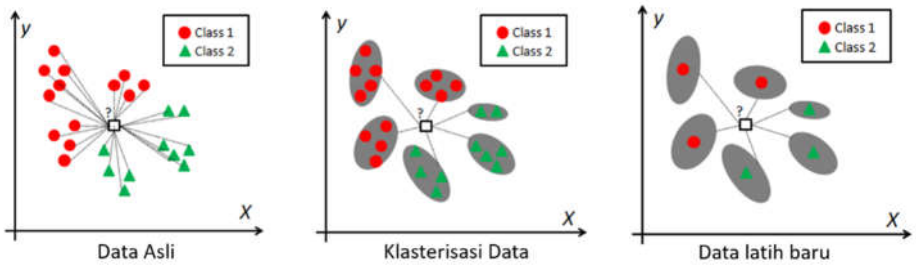
Oleh sebab itu, penelitian ini mencoba mengusulkan sebuah strategi untuk meningkatkan kecepatan algoritma KNN tetapi dengan akurasi yang hampir sama dengan standard KNN. Jika pada standar KNN proses hanya dilakukan dengan menyimpan data latih, yang kemudian akan dibandingkan dengan data uji baru dengan cara menghitung jarak satu persatu, sementara strategi yang diusulkan mencoba mengurangi jumlah data latih dengan strategi clustering untuk mendapatkan c cluster dan beberapa parameternya, yaitu rata-rata dan variance, untuk setiap kelas pada data latih. Sehingga jumlah data yang akan dibandingkan dengan data uji lebih sedikit. Akibatnya, diharapkan waktu prosesnya menjasi lebih cepat. Klasifikasi kemudian dilakukan menggunakan fungsi jarak berbobot yang dihitung berdasarkan nilai rata-rata dan variance tiap kluster kelas.

2. METODE PENELITIAN

Pada bagian ini akan dijelaskan lebih detil bagaimana metode yang diusulkan diimplementasikan.

2.1 Overview Metode Yang Diusulkan

Pada penelitian ini, usulan strategi untuk meningkatkan metode KNN adalah dengan mengurangi jumlah memori serta jumlah perbandingan yang dibutuhan saat proses pencocokan antara data baru dengan data yang ada pada memori. Strategi yang digunakan adalah dengan memanfaatkan metode clustering untuk mengurangi jumlah data dengan asumsi bahwa data-data yang mirip akan memiliki fitur yang sama sehingga cukup membandingkan dengan salah satu fitur saja. Ilustrasi pengurangan data menggunakan strategi klustering pada metode yang diusulkan ditunjukkan pada Gambar 1.



Gambar 1. Ilustrasi pengurangan data dengan metode clustering

2.2. Pengurangan Jumlah Data

Metode usulan diawali dengan mengurangi jumlah data pada data latih sehingga saat dilakukan perbandingan jumlah operasi perbandingan jadi lebih sedikit sehingga kecepatan proses akan meningkat. Untuk mengatasi masalah ini, strategi pengimplementasian metode klastering diusulkan seperti terlihat pada algoritma 1.

Algoritma 1 Mengurangi jumlah data

Masukan

- Sekumpulan data training $X=[x_1,x_2,...x_n]$, di mana N adalah jumlah data, $x_i=[f_1,f_2,..f_R]$, f_i adalah nilai dari fitire x ke-i, dan R adalah ukuran dimensi data.
- C kelas sesuai dengan jumlah kelas pada data.

Keluaran

- Data training yang sudah berkurang $Q=[q_1,q_2,...q_{MC}]$, di mana M adalah jumlah dimensi.

Proses

1: **for** *i*: 1 **to** *C* **do**
2: cluster semua data yang termasuk pada kelas c_i
3: *M* kelas akan didapatkan
4: **for** *k*: 1 to **M** do
5: Hitung rata-rata kelas q_k, $\mu=[\mu_1, \mu_2, ... \mu_R]$
6: Hitung nilai variansi dari kelas q_k, $\sigma^2=[\sigma^2_1,\sigma^2_2, ... \sigma^2_R]$
7: **Return** data training yang sudah dikurangi $Q = [q_1,q_2,q_3,...,q_{MC}]$, where $q_k=(\mu, \sigma^2)$.

2.2.2 Perhitungan Kemiripan

Pada langkah sebelumnya didapatkan data latih baru yang terdiri dari rata-rata kelas dan variansi dari kelas. Nilai inilah yang akan digunakan untuk menghitung jarak antara data baru dan data latih dalam rangka menghitung nilai kemiripan untuk menentukan data baru masuk ke kelas mana dengan menggunakan persamaan berikut:

$$d(q,y)=\sqrt{\sum_{i=1}^{R} w_i^2\left(\mu_i(q)-f_i(y)\right)^2} \quad (1)$$

di mana q=[$\mu_1, \mu_2, \ldots \mu_R, \sigma^2_1, \sigma^2_2, \ldots \sigma^2_R$] adalah fitur pertama hasil dari pengurangan data pada data latih, $y=[f_1, f_2, \ldots f_R]$ adalah fitur kedua yang akan dicari kemiripannya dengan q, dan w_i $(i=1 \ldots R)$ adalah bobot dari fitur f_i yang didefinisikan dengan formula:

$$w_i = \frac{| SD - \sigma_i |}{SD} \quad (2)$$

Dimana $SD = \sum_{j=1}^{R} \sigma_j$ adalah jumlah standar devisi. Kemudian untuk menentukan citra tertentu masuk ke kelas mana, metode yang digunakan adalah metode *neirest neighbour classifier*.

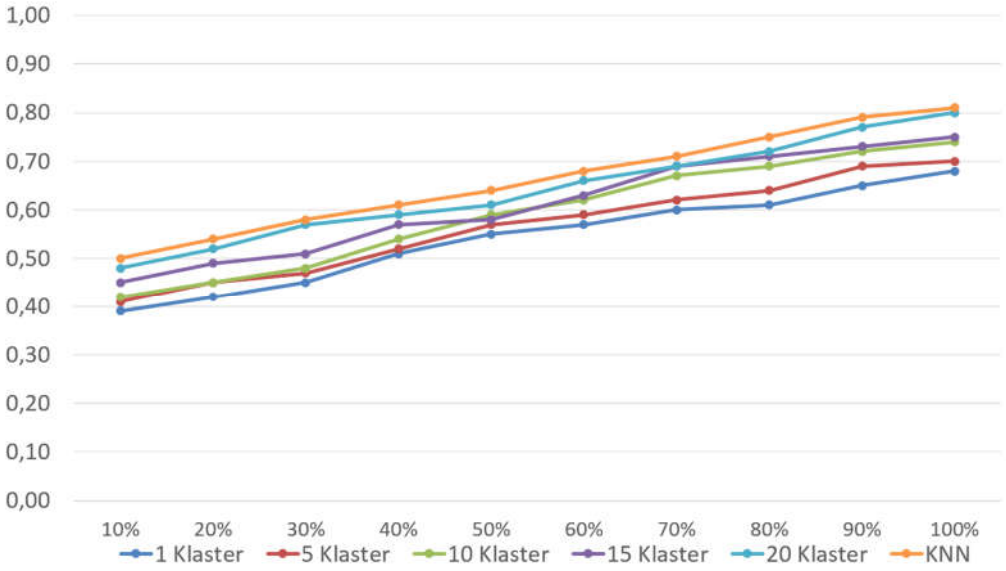
3. HASIL DAN PEMBAHASAN

Usulan metode telah diimplementasikan menggunakan bahasa pemrograman C++ dengan PC Berbasis Windows OS. Dalam implementasi, metode yang diusulkan pertama kali kami mencoba untuk mencari nilai k yang optimal pada proses klustering menggunakan algoritme *k-means*. Nilai k yang optimal ini selanjutnya akan dibandingkan dengan standar KNN pada nilai akurasi dan waktu proses. Nilai akurasi didapatkan dengan menghitung banyaknya jumlah data uji yang diklasifikasikan secara tepat dibagi dengan jumlah keseluruhan data uji yang ada pada dataset.

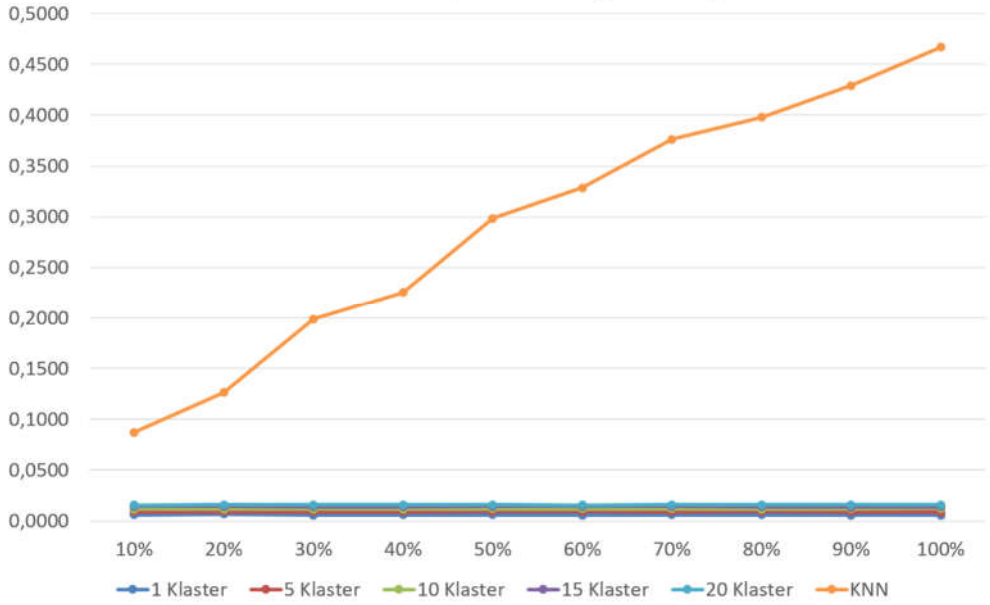
3.1 Karakteristik Dataset dan Ektraksi Fitur

Dataset klasifikasi kendaraan bermotor terdiri dari lima kelas yaitu bus, motor, sedan, truck, dan vans, seperti terlihat pada Gambar 2. Kami mengumpulkan dataset dari internet sebagai 200 citra untuk setiap kelas dengan ukuran 64×64 piksel. Pengumpulan dataset dilakukan menggunakan tool *Google Image Search* dengan mengetikkan kata kunci sesuai dengan kelas yang ingin dicari. Citra yang seharusnya diambil idealnya adalah citra dengan arah *bird-view* namun karena sulitnya mendapatkan data ini, maka data yang dikumpulkan adalah citra dengan arah *front-view*. Dataset kemudian dibagi menjadi kelas pelatihan dan pengujian dengan komposisi 75% dan 25%. Untuk setiap data citra latih, HOG feature [14] dengan ukuran cell 8x8, ukuran blok 4x4, dan jumlah bin sama dengan 8 akan diekstraksi.

Gambar 2. Contoh sample citra kendaraan bermotor (bus, motor, sedan, truck, dan vans)



Gambar 3. Hasil akurasi pada data dengan beberapa klaster



Gambar 4. Waktu proses metode usulan versus standar KNN.

3.2 Pengaruh Nilai Jumlah Klaster

Pertama, dilakukan pengujian terhadap jumlah klaster yang efektif dalam pengurangan jumlah data pada proses KNN. Dilakukan ujicoba dengan menggunakan beberapa nilai *k* pada metode klastering *k-means* yaitu k= {1, 5, 10, 15, dan 20} yang kemudian dibandingkan dengan standar KNN.

Gambar 3 dan 4 menunjukan akurasi dan waktu proses untuk standar KNN dan metode yang diusulkan. Jika dilihat pada Gambar 3, nilai akurasi mengalami kenaikan saat jumlah data latih yang digunakan semakin besar baik untuk KNN maupun metode yang diusulkan. Selain itu kami juga menemukan bahwa semakin banyak jumlah klaster yang digunakan, akurasi sistem cenderung mengalami kenaikan dengan selisih yang kecil, namun di sisi lain waktu proses yang dibutuhkan juga jadi lebih lama. Evaluasi dengan menggunakan 100% data latih, metode yang diusulkan dengan jumlah kluster 20 menghasilkan akurasi yang paling tinggi dibandingkan

dengan jumlah kluster lainnya yaitu sebesar 79%. Meski demikin jumlah ini masih lebih kecil dibandingan dengan standar KNN yang menghasilkan akurasi sebesar 80%. Jika dilihat selisih yang dihasilkan hanya 1%, tetapi jika melihat pada hasil waktu proses, metode yang diusulkan jauh lebih cepat dibandingkan standar KNN.

Selain itu, berdasarkan pada Gambar 4 dapat dilihat bahwa menambahkan jumlah data latih tidak memberikan pengaruh terhadap waktu proses, di mana waktu proses yang dibutuhkan bersifat konstan. Sebaliknya pada standar KNN, kenaikan jumlah data akan menyebabkan waktu proses menjadi lebih lama.

3.3 Evaluasi Keseluruhan

Secara keseluruhan didapatkan bahwa meski standar KNN masih menghasilkan akurasi yang lebih baik dibandingkan versi usulan dengan selisih akurasi 1%, tetapi waktu proses metode usulan jauh lebih cepat dibandingkan dengan standar KNN. Usulan peningkatan metode dengan jumlah cluster sebanyak 20 membutuhkan waktu proses 40x lebih dibandingkan dengan standar KNN dengan akurasi yang tidak jauh berbeda. Sehingga metode ini cocok diimplementasikan pada aplikasi real-time intelligent transportation system.

4. KESIMPULAN DAN SARAN

Peningkatan kecepatan KNN berhasil dilakukan dengan menggunakan strategi clustering dalam rangka mengurangi jumlah memori. Metode yang diusulkan diimplementasi pada kasus klasifikasi kendaraan bermotor. Dibandingkan dengan standar KNN, metode yang diusulkan dengan jumlah klaster 20 memiliki keceparan 40x lebih cepat dengan perbedaan akurasi hanya 1%. Metode ini cocok diimplementasikan pada aplikasi realtime seperti pada ITS.

UCAPAN TERIMA KASIH

Penelitian ini terlaksana karena dana hibah penelitian program pasca sarjana Departemen Ilmu Komputer dan Elektronika, FMIPA UGM Tahun 2018 dan 2019.

DAFTAR PUSTAKA

[1] Babu, V. S. dan Viswanath, P., "Rough-fuzzy weighted k-nearest leader classifier for large data sets," *Pattern Recognition*, vol. 42, no. 2009, pp.1719–1731, 2009.

[2] Dasarathy, B. V. "Data mining tasks and methods: Classification: Nearest-neighbor approaches," in *Handbook of data mining and knowledge discovery*. New York: Oxford University Press, pp. 288–298, 2002.

[3] Hassan, M. R., Hossain, M. M., Bailey, J., dan Ramamohanarao, K. "Improving k-nearest neighbor classification with distance functions based on receiver operating characteristics," in W. Daelemans et al. (Eds.): ECML PKDD 2008, Part I, *LNAI* 5211, pp. 489-504. 2008.

[4] Fix, E. dan Hodges, Jr. J.L., "Discriminatory Analysis: Non-parametric Discrimination: Small Sample Performance," *Report* No. 11, USAF School of Aviation Medicine, Randolph Field, Texas, 1952.

[5] Duda, R. O., Hart, P. E., dan Stork, D. G., *Pattern Classification*, 2nd ed. John Wiley & Sons: A Wiley-interscience Publication, 2000.

[6] Nicolas, G.-P. dan Domingo, O.-B., "Boosting k-nearest neighbor classifier by means of input space rejection," *Expert systems with application* 36, pp. 10570-10582, 2009.

[7] Viswanath, P., dan Sarma, T. H., "An improvement to k-nearest neighbor classifier," *IEEE Recent Advances in Intelligent Computational Systems*, pp, 227-231, 2011.

[8] Zhou, C. Y., dan Chen, Y. Q., "Improving nearest neighbor classification with cam weighted distance," *Pattern recognition* Vol. 39, pp. 635-645, 2006.

[9] Jiang, Z. H. S. J. "Instance cloning local naive bayes," *In Proceedings of the Eighteenth Canadian Conference on Artificial Intelligence*, pages 280–291. Springer, 2005.

[10] Frank, H. M. P. B. "Locally weighted naive bayes," *In Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 249–256. Morgan Kaufmann, 2003.
[11] Jiang, L., Zhang, H., dan Cai., Z., "Dynamic k-nearest-neighbor naive bayes with attribute weighted," *In Proceedings of the 3rd International Conference on Fuzzy Systems and Knowledge Discovery*, pages 365–368. Springer, 2006.
[12] Manocha, S., dan Girolami, M. A., "An empirical analysis of the probabilitics k-nearest neighbor classifier," *Pattern Recognition Letter* 28, pp. 1818-1824, 2007.
[13] Jiang, L., Cai, Z., Wang, D., dan Jiang, S., "Survey of improving k-nearest neighbor for classification," *FSKD*, 2007.
[14] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, ICPR, volume 2, pages 886–893, INRIA Rhˆone-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.