# K – NEAREST NEIGHBOR

## *INFORMATION RETRIEVAL*

## (SISTEM TEMU KEMBALI INFORMASI)


**Disusun Oleh :**

| | |
|---|---|
| Alfian Sukma | 081116007 |
| Dian Ramadhan | 081211631003 |
| Bagus Puji Santoso | 081211631061 |
| Tiara Ratna Sari | 081211632014 |
| Ni Made Ayu Karina Wiraswari | 081211633020 |

S1 - Sistem Informasi

**Dosen Pembimbing :**

Badrus Zaman, S. Kom

**UNIVERSITAS AIRLANGGA**

**FAKULTAS SAINS DAN TEKNOLOGI**


**2014**

## 1. Pengertian

Algoritma *k-nearest neighbor* (k-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut.

*K-Nearest Neighbor* berdasarkan konsep '*learning by analog*y'. Data *learning* dideskripsikan dengan atribut numerik *n*-dimensi. Tiap data *learning* merepresentasikan sebuah titik, yang ditandai dengan *c*, dalam ruang *n*-dimensi. Jika sebuah data *query* yang labelnya tidak diketahui diinputkan, maka *K-Nearest Neighbor* akan mencari *k* buah data *learning* yang jaraknya paling dekat dengan data *query* dalam ruang *n*-dimensi. Jarak antara data *query* dengan data *learning* dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data *query* dengan semua titik yang merepresentasikan data *learning* dengan rumus *Euclidean Distance*.

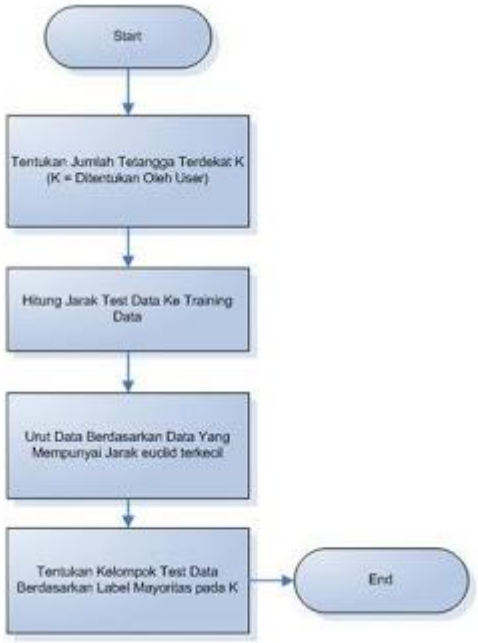
Pada fase *training*, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data *training sample*. Pada fase klasifikasi, fitur – fitur yang sama dihitung untuk *testing data* (klasifikasinya belum diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor *training sample* dihitung, dan sejumlah *k* buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik – titik tersebut.

Nilai *k* yang terbaik untuk algoritma ini tergantung pada data; secara umumnya, nilai *k* yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai *k* yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan cross-validation. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain, $k = 1$) disebut algoritma *nearest neighbor*.

Ketepatan algoritma k-NN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur, agar performa klasifikasi menjadi lebih baik.

*K* buah data *learning* terdekat akan melakukan *voting* untuk menentukan label mayoritas. Label data *query* akan ditentukan berdasarkan label mayoritas dan jika ada lebih dari satu label mayoritas maka label data *query* dapat dipilih secara acak di antara label-label mayoritas yang ada.

Adapun algoritma dari KNN ditunjukan pada flowchart berikut :




## 2. Pembahasan

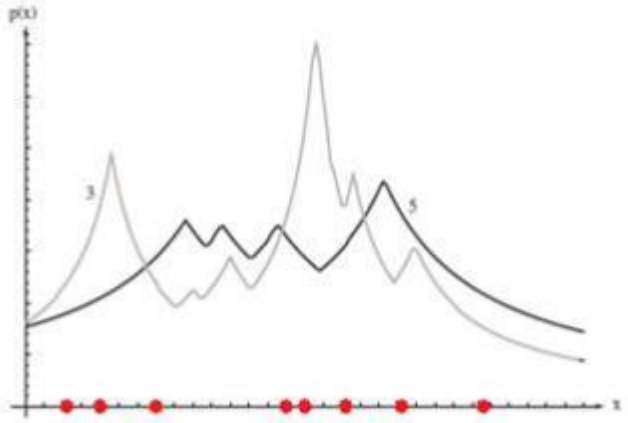
a. Diberikan 2 buah titik P dan Q dalam sebuah ruang vektor *n*-dimensi dengan P($p_1$, $p_2$,..., $p_n$) dan Q($q_1$, $q_2$,...,$q_n$), maka jarak antara P dan Q dapat diukur dengan menggunakan persamaan *Euclidean Distance* sebagai berikut:

$$D_{euc}(P,Q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

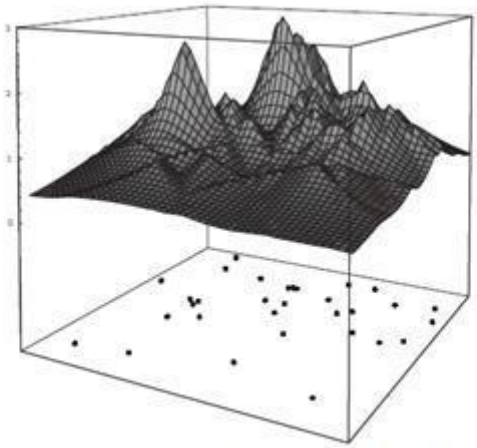
dimana P dan Q adalah titik pada ruang vektor *n* dimensi sedangkan p*i* dan q*i* adalah besaran skalar untuk dimensi ke *i* dalam ruang vektor *n* dimensi.

b. Sebagai contoh, untuk mengestimasi *p(x)* dari *n training sample* dapat memusatkan pada sebuah sel disekitar *x* dan membiarkannya tumbuh hingga meliputi *k samples*. *Samples* tersebut adalah KNN dari *x*. Jika densitasnya tinggi di dekat *x*, maka sel akan berukuran relatif kecil yang berarti memiliki resolusi yang baik. Jika densitas rendah, sel akan tumbuh

lebih besar, tetapi akan berhenti setelah memasuki wilayah yang memiliki densitas tinggi. Pada Gambar 2.13 dan Gambar 2.14 ditampilkan estimasi densitas satu dimensi dan dua dimensi dengan KNN [11].



**Gambar 2.13.** Delapan titik dalam satu dimensi dan estimasi densitas KNN dengan $k = 3$ dan $k = 5$.



**Gambar 2.14.** KNN mengestimasi densitas dua dimensi dengan $k = 5$

Nilai *k* yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai *k* yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Nilai *k* yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus dimana klasifikasi diprediksikan berdasarkan *training data* yang paling dekat (dengan kata lain, *k = 1*) disebut algoritma *nearest neighbor*.

Ketepatan algoritma KNN sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau jika bobot fitur tersebut tidak setara dengan

relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik.

KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap *training data* yang memiliki banyak *noise* dan efektif apabila *training data*-nya besar. Sedangkan, kelemahan KNN adalah KNN perlu menentukan nilai dari parameter *k* (jumlah dari tetangga terdekat), *training* berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap *query instance* pada keseluruhan *training sample*.

Terdapat beberapa jenis algoritma pencarian tetangga terdekat, diantaranya:

- Linear scan
- Pohon kd
- Pohon Balltree
- Pohon metrik
- Locally-sensitive hashing (LSH)

Algoritma k-NN ini memiliki konsistensi yang kuat. Ketika jumlah data mendekati tak hingga, algoritma ini menjamin *error rate* yang tidak lebih dari dua kali *Bayes error rate* (*error rate* minimum untuk distribusi data tertentu).

## 3. Kelebihan dan Kekurangan

- Kelebihan

  KNN memiliki beberapa kelebihan yaitu bahwa dia tangguh terhadap training data yang *noisy* dan efektif apabila data latih nya besar.

- Kelemahan

  Sedangkan kelemahan dari KNN adalah :

  1. KNN perlu menentukan nilai dari parameter K (jumlah dari tetangga terdekat)
  2. Pembelajaran berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil yang terbaik

3. Biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap sample uji pada keseluruhan sample latih

## 4. Contoh Soal

Terdiri dari 2 atribut dengan skala kuantitatif yaitu X1 dan X2 serta 2 kelas yaitu baik dan buruk. Jika terdapat data baru dengan nilai X1=3 dan X2=7

| X1 | X2 | Y |
|---|---|---|
| 7 | 7 | Buruk |
| 7 | 4 | Buruk |
| 3 | 4 | Baik |
| 1 | 4 | Baik |

Langkah – langkah :

1. Tentukan parameter K = jumlah tetangga terdekat. Misalkan ditetapkan K = 3
2. Hitung jarak antara data baru dengan semua data training

| X1 | X2 | Kuadrat jarak dengan data baru (3,7) |
|---|---|---|
| 7 | 7 | $(7-3)^2 + (7-7)^2 = 16$ |
| 7 | 4 | $(7-3)^2 + (4-7)^2 = 25$ |
| 3 | 4 | $(3-3)^2 + (4-7)^2 = 9$ |
| 1 | 4 | $(1-3)^2 + (4-7)^2 = 13$ |

3. Urutkan jarak tersebut dan tetapkan tetangga terdekat berdasarkan jarak minimum ke-K

| X1 | X2 | Kuadrat jarak dengan data baru (3,7) | Peringkat Jarak minimum | Termasuk 3 tetangga terdekat |
|---|---|---|---|---|
| 7 | 7 | $(7-3)^2 + (7-7)^2 = 16$ | 3 | Ya |
| 7 | 4 | $(7-3)^2 + (4-7)^2 = 25$ | 4 | Tidak |
| 3 | 4 | $(3-3)^2 + (4-7)^2 = 9$ | 1 | Ya |
| 1 | 4 | $(1-3)^2 + (4-7)^2 = 13$ | 2 | Ya |

4. Periksa kelas dari tetangga terdekat

| X1 | X2 | Kuadrat jarak dengan data baru (3,7) | Peringkat Jarak minimum | Termasuk 3 tetangga terdekat | Y = kelas tetangga terdekat |
|---|---|---|---|---|---|
| 7 | 7 | $(7-3)^2 + (7-7)^2 = 16$ | 3 | Ya | Buruk |
| 7 | 4 | $(7-3)^2 + (4-7)^2 = 25$ | 4 | Tidak | - |
| 3 | 4 | $(3-3)^2 + (4-7)^2 = 9$ | 1 | Ya | Baik |
| 1 | 4 | $(1-3)^2 + (4-7)^2 = 13$ | 2 | Ya | Baik |

# DAFTAR PUSTAKA


Anonim. 2013. *KNN*. http://id.wikipedia.org/wiki/KNN. Diakses pada tanggal 16 April 2014.

Boedy, Cged. 2012. *Pengertian, Kelebihan, dan Keurangan K-nearest Neighbor (K-NN). http://cgeduntuksemua.blogspot.com/2012/03/pengertian-kelebihan-dan-kekurangan-k.html. Diakses pada tanggal 16 April 2014.*

Ecatatan. 2013. *K-Nearest Neighbor*. http://ecatatan.wordpress.com/2013/05/22/k-nearest-neighbor/. Diakses pada tanggal 16 April 2014.

Muliadinata, Saban. 2012. *Algoritma K-Nearest Neighbor (KNN)*. http://sharewy .blogspot.com/2013 /04/algoritma-k-nearest-neighbor-knn_16.html. Diakses pada tanggal 16 April 2014.

Yofianto, Evan. 2010. *Buku TA : K-Nearest Neighbor (KNN)*. http://kuliahinformatika. wordpress. com/2010/02/13/buku-ta-k-nearest-neighbor-knn/. Diakses pada tanggal 16 April 2014.