

K-NEAREST NEIGHBOURS

Ali Akbar Septiandri

Universitas Al-Azhar Indonesia

aliakbars@live.com

May 10, 2020

SELAYANG PANDANG

① INSTANCE-BASED LEARNING

② OPTIMASI K-NN

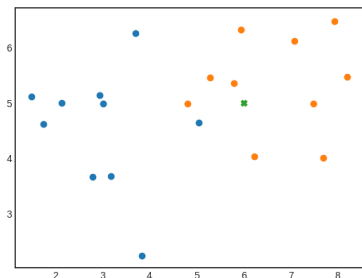
③ EKSTENSI K-NN

BAHAN BACAAN

- ① Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann. (Section 4.7 & 7.1)
- ② “Klasifikasi: k-Nearest Neighbours.” *Cerita Tentang Data*. 31 Agustus 2015. <https://tentangdata.wordpress.com/2015/08/31/klasifikasi-k-nearest-neighbours/>

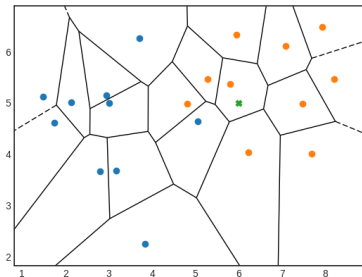
INSTANCE-BASED LEARNING

INTUISI



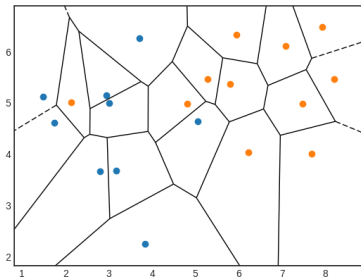
- Terdapat dua variabel: x_1, x_2
- Dua kelas: biru dan jingga
- Apa kelas dari *instance* tanda silang?

KLASIFIKASI NEAREST-NEIGHBOUR



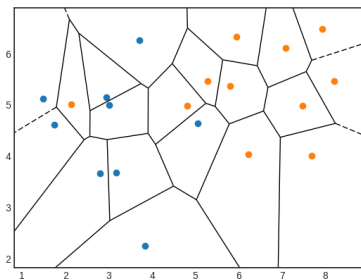
- Cari yang paling mirip, lalu gunakan kelas yang sama!
- *Voronoi tessellation*: membagi region dengan titik yang memiliki jarak yang sama dari dua contoh data latih
- Batas klasifikasi: non-linear

PENCILAN



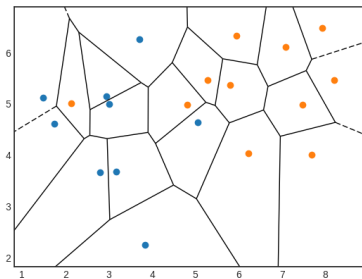
- Sensitif terhadap pencilan

PENCILAN



- Sensitif terhadap pencilan
- Tidak ada $P(y|x)$

PENCILAN



- Sensitif terhadap pencilan
- Tidak ada $P(y|x)$
- Tidak sensitif terhadap *class prior*

*Perbaiki dengan menggunakan lebih dari satu tetangga
(k -tetangga) terdekat!*

ALGORITMA KLASIFIKASI

- Diketahui
 - data latih $\{x_i, y_i\}$
 - x_i : nilai atribut
 - y_i : label kelas
 - *instance* uji x
- Algoritma:
 - 1 Hitung jarak $D(x, x_i)$ untuk semua x_i
 - 2 Pilih k tetangga terdekat dengan labelnya
 - 3 \hat{y} = mayoritas dari label tetangga terdekat

KLASIFIKASI K-NN

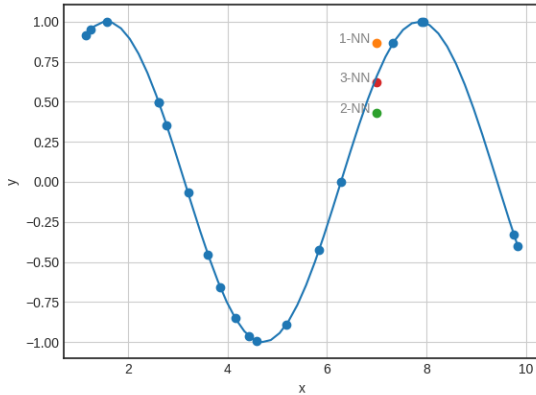


GAMBAR: 7-NN pada data MNIST dengan data uji di paling kanan

ALGORITMA REGRESI

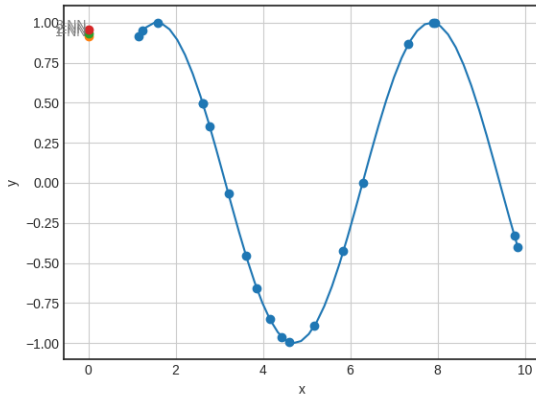
- Diketahui
 - data latih $\{x_i, y_i\}$
 - x_i : nilai atribut
 - y_i : nilai numerik sebenarnya
 - *instance* uji x
- Algoritma:
 - 1 Hitung jarak $D(x, x_i)$ untuk semua x_i
 - 2 Pilih k tetangga terdekat dengan labelnya
 - 3 $\hat{y} = f(x) = \frac{1}{k} \sum_{j=1}^k y_{ij}$ (nilai rata-rata)

REGRESI K-NN



GAMBAR: Interpolasi dengan {1,2,3}-NN

REGRESI K-NN



GAMBAR: Ekstrapolasi dengan $\{1,2,3\}$ -NN

OPTIMASI K-NN

Bagaimana cara memilih nilai k ?

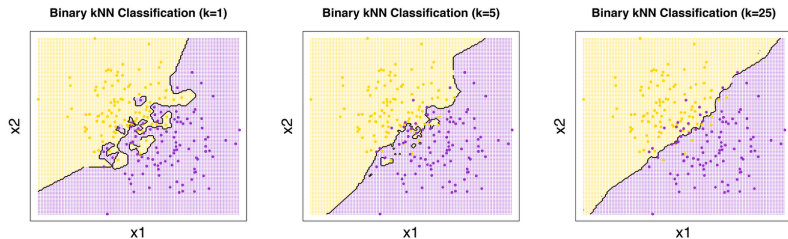
MEMILIH NILAI K

- Nilai yang besar $\rightarrow P(y)$
- Nilai yang kecil \rightarrow terlalu variatif, batas keputusan yang tidak stabil

MEMILIH NILAI k

- Nilai yang besar $\rightarrow P(y)$
- Nilai yang kecil \rightarrow terlalu variatif, batas keputusan yang tidak stabil
- **Solusi:** Gunakan data validasi!

BATAS KEPUTUSAN



GAMBAR: Pengaruh nilai k pada batas keputusan [DeWilde, 2012]

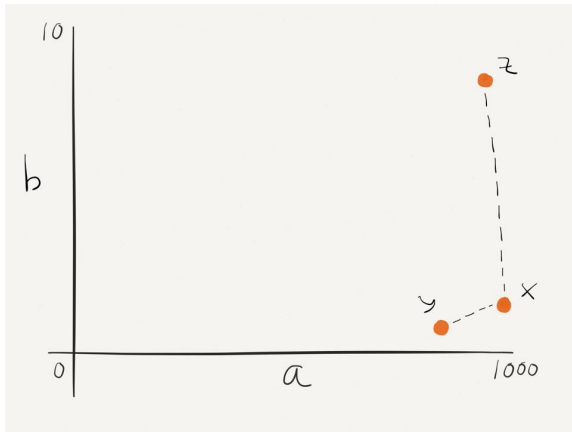
PENGUKURAN JARAK

Minkowski distance (p-norm):

$$d([x_1, x_2, \dots, x_n], [y_1, y_2, \dots, y_n]) = \sqrt[r]{\sum_{i=1}^n |x_i - y_i|^r}$$

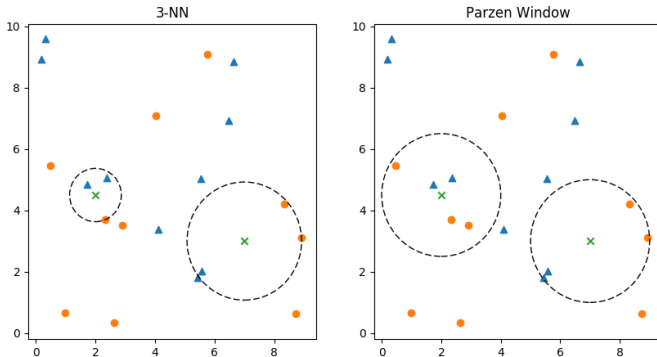
- Hasil seri:
 - ① Gunakan jumlah k ganjil
 - ② Acak, lemparan koin
 - ③ *Prior probability*
 - ④ 1-NN
- *Missing values*: **harus** diganti (*impute*)
- Rentan terhadap perbedaan rentang variabel

PERBEDAAN RENTANG



GAMBAR: Perbedaan rentang variabel bisa mengacaukan klasifikasi k-NN [Wibisono, 2015]

k-NN vs PARZEN WINDOWS



GAMBAR: Perbedaan radius klasifikasi pada k-NN dan Parzen Windows

PROS & CONS

- Pros:
 - Tidak ada asumsi terhadap data, non-parametrik
 - *Asymptotically correct*
- Cons:
 - Harus mengganti nilai yang hilang
 - Sensitif terhadap kelas pencilan (data latih yang salah dilabeli)
 - Sensitif terhadap atribut yang irelevan
 - **Mahal secara komputasi $O(nd)$**

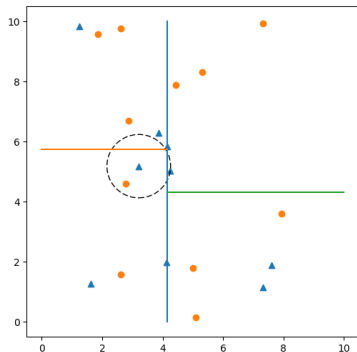
EKSTENSI K-NN

MEMPERCEPAT K-NN

- Pelatihan: $O(d)$, tetapi pengujian: $O(nd)$
- Mengurangi d : *dimensionality reduction*
- Mengurangi n : jangan bandingkan dengan **semua** data latih, i.e. cari $m \ll n$
 - ① K-D trees
 - ② Locality-sensitive hashing (LSH)
 - ③ Inverted lists

K-D TREES

Pilih dimensi secara acak, cari mediannya, pisahkan data, ulangi

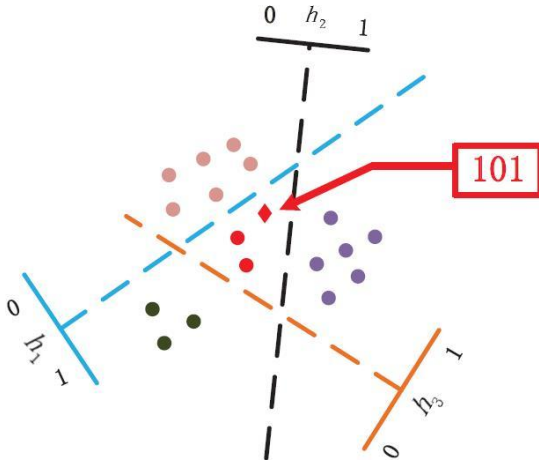


GAMBAR: 3-NN dari semua data berbeda dengan 3-NN yang berada pada region yang sama

LOCALITY-SENSITIVE HASHING (LSH)

- Hyperplanes acak $h_1...h_k$ yang membagi ruang menjadi 2^k region
- Bandingkan x hanya dengan data latih dalam region yang sama: lakukan *dot-product* \rightarrow *hash-code*
- Ada kemungkinan tetangga dekat yang terlewat: ulangi lagi dengan $h_1...h_k$ yang berbeda

LOCALITY-SENSITIVE HASHING



GAMBAR: Menghasilkan *hash-code* dari *hyperplanes* [Li et al., 2017]

INVERTED LISTS

- Jika datanya berupa *bag-of-words*, matriksnya akan *sparse*
- Ide: buat daftar dokumen per atribut

INVERTED LISTS

D1: “send us your password”

(s)

D2: “send us your review” (h)

D3: “review your password” (h)

D4: “review us” (s)

D5: “send your password” (s)

D6: “send us your account” (s)

Dokumen baru: “account
review”

send $\rightarrow \{1, 2, 5, 6\}$

your $\rightarrow \{1, 2, 3, 5, 6\}$

review $\rightarrow \{3, 4\}$

account $\rightarrow \{6\}$

password $\rightarrow \{1, 3, 5\}$

Salindia ini dibuat dengan
sangat dipengaruhi oleh Lavrenko (2014)

REFERENSI



Burton DeWilde (26 Oktober 2012)

Classification of Hand-written Digits (3)

<http://bdewilde.github.io/blog/blogger/2012/10/26/classification-of-hand-written-digits-3/>



Okiriza Wibisono (16 September 2015)

kNN: Perhitungan Jarak, serta Batasan dan Keunggulan

<https://tentangdata.wordpress.com/2015/09/16/knn-perhitungan-jarak-serta-keunggulan-dan-batasan/>



Haisheng Li et al. (2017)

Feature Matching of Multi-view 3D Models Based on Hash Binary Encoding

Neural Network World. 27. 95-105.

Terima kasih