

# k-Nearest Neighbours & k-Means Clustering

Ali Akbar Septiandri

Universitas Al-Azhar Indonesia

*aliakbars@live.com*

May 5, 2017

# Selayang Pandang

## ① k-Nearest Neighbours

Instance-based Learning  
Extension

## ② k-Means Clustering

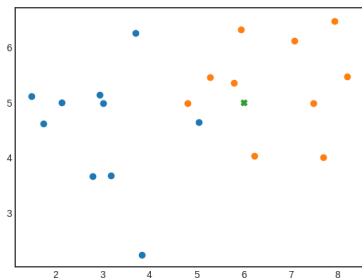
Clustering  
Mengevaluasi Clustering  
Aplikasi

## 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, 4.8, & 7.1)
- ② VanderPlas, J. (2016). Python Data Science Handbook. (In Depth: k-Means Clustering) <http://nbviewer.jupyter.org/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.11-K-Means.ipynb>
- ③ “Klasifikasi: k-Nearest Neighbours.” *Cerita Tentang Data*. 31 Agustus 2015. <https://tentangdata.wordpress.com/2015/08/31/klasifikasi-k-nearest-neighbours/>

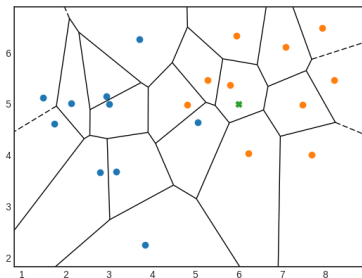
# k-Nearest Neighbours

# Intuisi



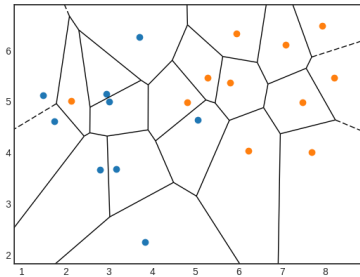
- Terdapat dua variabel:  $x_1, x_2$
- Dua kelas: biru dan merah
- Apa kelas dari *instance* tanda silang?

# Klasifikasi Nearest-Neighbour



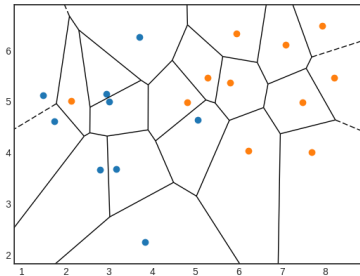
- Cari yang paling mirip, lalu gunakan kelas yang sama!
- *Voronoi tessellation*: membagi region dengan titik yang memiliki jarak yang sama dari dua contoh data latih
- Batas klasifikasi: non-linear

# Pencilan



- Sensitif terhadap pencilan

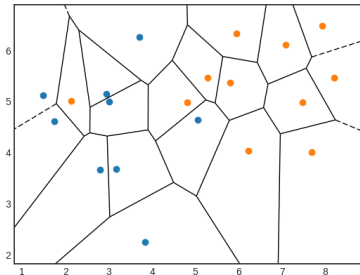
# Pencilan



- 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

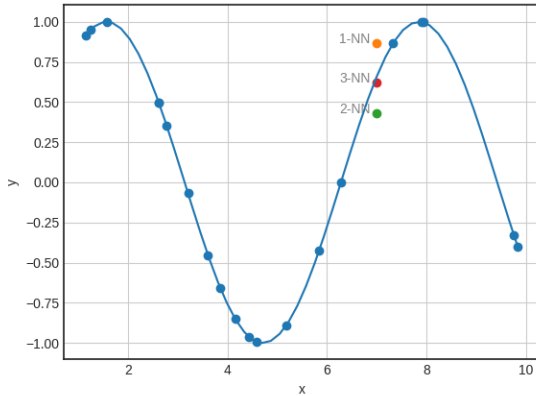


Gambar: 7-NN pada data MNIST dengan data uji di paling kanan

# Algoritma Regresi

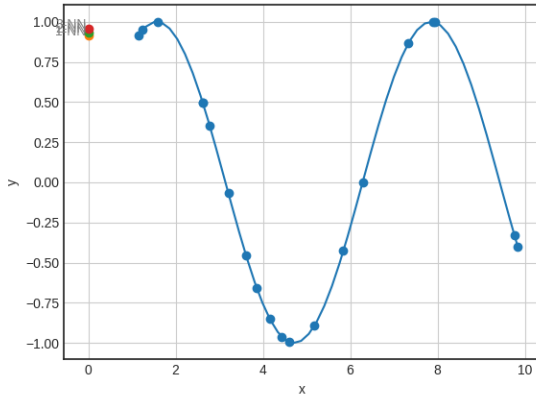
- Diketahui
  - data latih  $\{x_i, y_i\}$ 
    - $x_i$ : nilai atribut
    - $y_i$ : nilai numerik sebenarnya
  - *instance* uji  $x$
- Algoritma:
  - 1 Hitung jarak  $D(x, x_i)$  untuk semua  $x_i$
  - 2 Pilih  $k$  tetangga terdekat dengan labelnya
  - 3  $\hat{y} = f(x) = \frac{1}{k} \sum_{j=1}^k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

# Regresi k-NN



Gambar: Interpolasi dengan  $\{1,2,3\}$ -NN

# Regresi k-NN



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

Bagaimana cara memilih nilai  $k$ ?



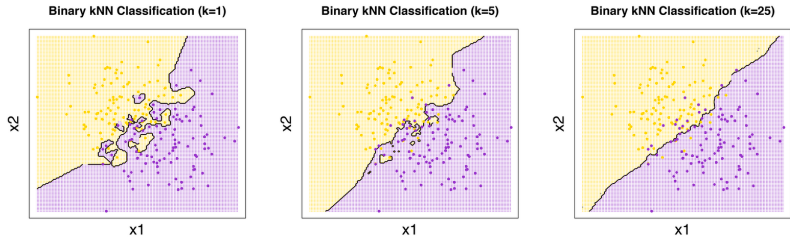
## Memilih Nilai $k$

- 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

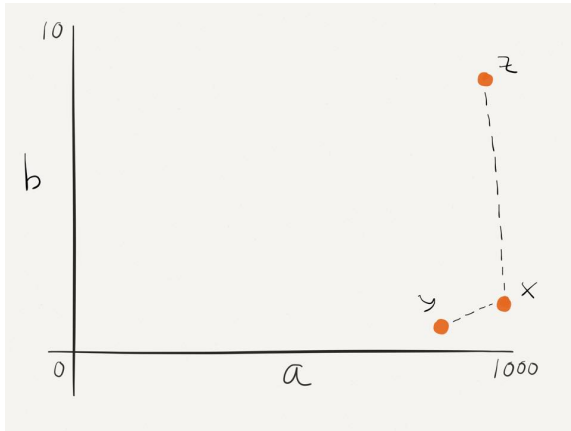
Minkowski distance (p-norm):

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

*Catatan: Lihat kembali salindia minggu ketiga!*

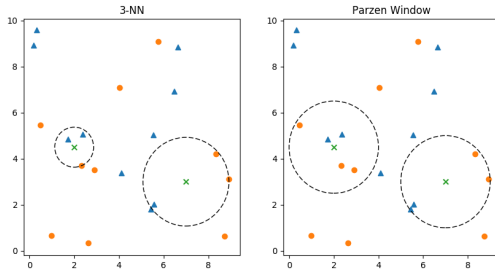
- Hasil seri:
  - ① Gunakan jumlah  $k$  ganjil
  - ② Acak, lemparan koin
  - ③ *Prior probability*
  - ④ 1-NN
- *Missing values*: **harus** diganti (*impute*)
- Rentan terhadap perbedaan rentang variabel

## 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**  $O(nd)$

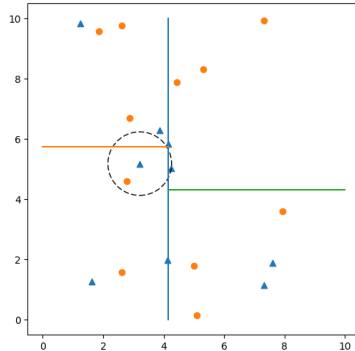


## Mempercepat k-NN

- 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  $h_1 \dots h_k$  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  $h_1 \dots h_k$  yang berbeda

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

# k-Means Clustering

# Clustering

- *Unsupervised learning*
- Subpopulasi apa yang ada dalam data?
- Apa kesamaan dari elemen di tiap subpopulasi?
- Bisa digunakan untuk menemukan pencila

# Jenis-jenis Clustering

- Tujuan:
  - Monothetic: *common property*
  - Polythetic: kemiripan data dengan pengukuran jarak
- Irisan:
  - Hard clustering
  - Soft clustering
- Flat vs hierarchical



# Metode Clustering

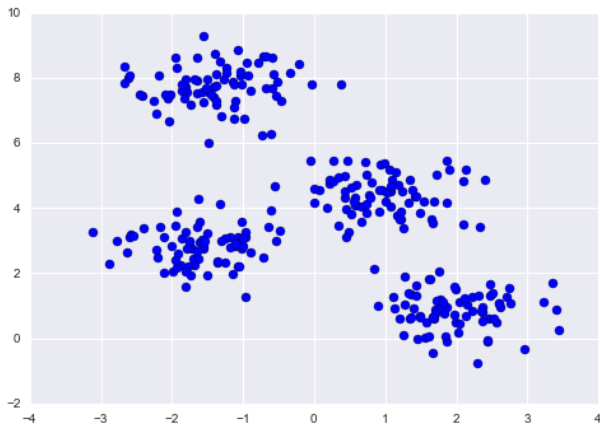
Metode *clustering* yang akan dibahas dalam kuliah ini:

- K-D Trees: *monothetic, hard boundaries, hierarchical*
- k-Means: *polythetic, hard boundaries, flat*
- Gaussian mixtures (EM algorithm): *polythetic, soft boundaries, flat*
- Agglomerative clustering: *polythetic, hard boundaries, hierarchical*

## k-Means

- Jumlah  $k$  ditentukan dari awal
- Tidak memerlukan label
- Menggunakan *centroid*, i.e. rata-rata nilai dari objek yang masuk dalam *cluster* tersebut
- Mencari *centroid* terdekat dari tiap objek

## Contoh Data



Gambar: Contoh data dalam 2D [VanderPlas, 2016]

## Hasil k-Means



Gambar: Setelah algoritma k-Means dijalankan [VanderPlas, 2016]

# Algoritma: Expectation-Maximization

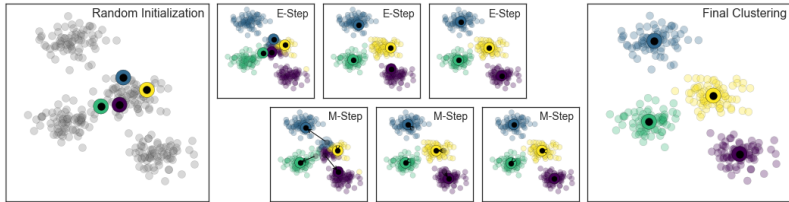
- 1 Inisialisasi k *centroid* secara acak
- 2 Ulangi hingga konvergen
  - A. E-step: Masukkan tiap titik/objek ke *centroid* terdekat

$$\arg \min_j D(x_i, c_j)$$

- B. M-step: Ubah nilai *centroid* menjadi rata-rata dari tiap titik/objek

$$c_j(a) = \frac{1}{n_j} \sum_{x_i \rightarrow c_j} x_i(a), \text{ for } a = 1..d$$

# Visualisasi EM



**Gambar:** Konvergensi kluster tercapai hanya dalam tiga iterasi  
[VanderPlas, 2016]

Perhatikan bahwa algoritma ini sangat bergantung  
pada inisialisasi *centroid*!

## Properti dari k-Means

- Meminimalkan jarak agregat intra-kluster

$$V = \sum_j \sum_{x_i \rightarrow c_j} D(c_j, x_i)^2$$

- Konvergensi ke **minimum lokal**
- Poin yang berdekatan mungkin masuk ke kluster yang berbeda





Berapa nilai  $k$  yang optimal?

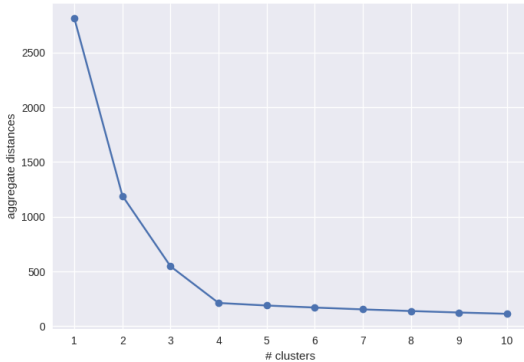
## Menentukan Nilai k

- Gunakan label kelas, e.g. 10 untuk MNIST
- Gunakan  $V$  untuk menggambarkan *scree plot*

$$V = \sum_j \sum_{x_i \rightarrow c_j} D(c_j, x_i)^2$$

lalu gunakan *elbow method*, i.e. nilainya dapat dicari dengan menggunakan nilai optimal turunan kedua

# Scree Plot



**Gambar:** Secara visual, scree plot menunjukkan nilai optimal  $k = 4$

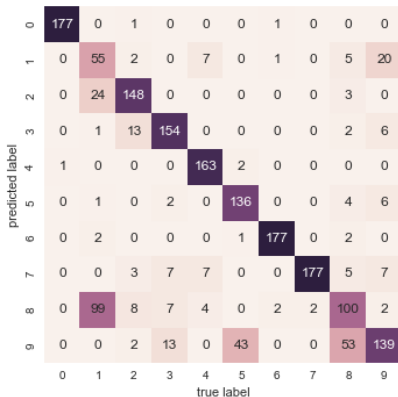
# Evaluasi Klaster

- Ekstrinsik: untuk mengerjakan tugas lain
  - Representasi gambar dengan fitur berupa klaster
  - Menemukan pencilan
- Intrinsik: untuk diri sendiri
  - Memahami data – deskriptif
  - Klaster  $\sim$  kelas, e.g. MNIST  $\rightarrow$  10 klaster
  - Perbandingan pasangan data dari klaster oleh manusia

## Evaluasi Intrinsik: Kluster $\sim$ Kelas

- Kluster  $c_1, c_2, \dots, c_K$
- Kelas  $R_1, R_2, \dots, R_N$
- Cocokkan  $R_i$  dengan  $c_j$ , hitung akurasi atau F1
  - Bagaimana jika  $N \neq K$ ?
  - Ada banyak cara, paling mudah dengan pendekatan *greedy*

## Contoh Evaluasi Intrinsik



Gambar: Confusion matrix dari MNIST clustering [VanderPlas, 2016]

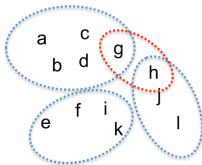
## Contoh Evaluasi Intrinsik

	G1	G2	G3	G4	G5	G6
C1	1	7	0	1	4	0
C2	0	0	0	0	2	7
C3	0	0	2	0	0	0
C4	3	1	0	0	1	0

Gambar: Klaster karakter dalam Julius Caesar

## Evaluasi Intrinsik: Perbandingan Antarpasangan

- Pasangan  $x_i, x_j$  apakah seharusnya berada dalam kluster yang sama?
- Hitung error, akurasi, F1
  - FN: pasangan  $x_i, x_j$  yang harusnya cocok, tapi berada dalam kluster yang lain (e,h)
  - FP: pasangan  $x_i, x_j$  yang harus tidak cocok, tapi berada dalam kluster yang sama (c,d)





# Aplikasi Clustering

- Representasi gambar: *bag of cluster id* atau fitur lain (lihat [Coates, 2012])
- Kompresi gambar (lihat [VanderPlas, 2016])
- *Dimensionality reduction*

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/>



Jake VanderPlas (2016)

In Depth: k-Means Clustering

<http://nbviewer.jupyter.org/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.11-K-Means.ipynb>

# Referensi



Adam Coates & Andrew Y. Ng (2012)

Learning feature representations with k-means.

Neural networks: Tricks of the trade (pp. 561-580). Springer Berlin Heidelberg.

Terima kasih