Gaussian Mixture Models & Hierarchical Clustering

Ali Akbar Septiandri

December 9, 2017

untuk Astra Graphia IT

Daftar Isi

Bahan Bacaan

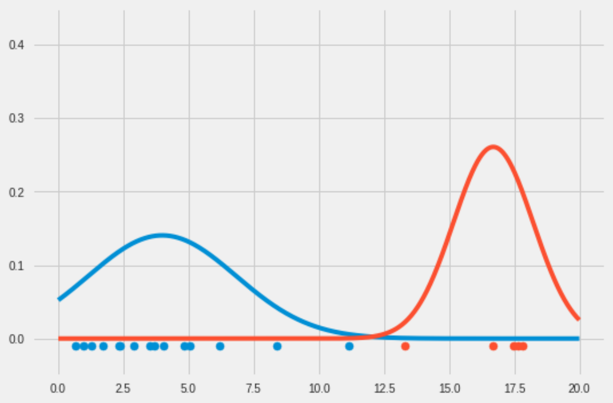
1. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann. (Section 9.3)
2. VanderPlas, J. (2016). Python Data Science Handbook. (In Depth: Gaussian Mixture Models) `http://nbviewer.jupyter.org/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.12-Gaussian-Mixtures.ipynb`
3. Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1). Springer, Berlin: Springer series in statistics. (Section 14.3.12)

Gaussian Mixture Models

Mixture Models

- Pendekatan probabilistik untuk *clustering*
- Setiap klaster adalah model generatif, e.g. Gaussian atau multinomial
- Menggunakan parameter
- Didasarkan pada algoritma Expectation Maximisation (EM)

Mixture Models 1D



Bagaimana kalau kita tidak tahu kelasnya?

Expectation Maximisation (EM)

1. Inisialisasi dengan dua Gaussians secara acak (μ_a, σ_a^2), (μ_b, σ_b^2)
2. Ulangi hingga konvergen
   a. **E-step**: Apakah x_i terlihat masuk ke a atau b, i.e. $P(a|x_i)$?[1]

   $$a_i = P(a|x_i) = \frac{P(x_i|a)P(a)}{P(x_i)}$$

   $$b_i = P(b|x_i) = 1 - a_i$$

   b. **M-step**: Perbaiki nilai (μ_a, σ_a^2), (μ_b, σ_b^2)

   $$\mu_a = \frac{a_1x_1 + a_2x_2 + ... + a_nx_n}{a_1 + a_2 + ... + a_n}$$

   $$\sigma_a^2 = \frac{a_1(x_1 - \mu_a)^2 + ... + a_n(x_n - \mu_a)^2}{a_1 + a_2 + ... + a_n}$$

[1] Bayes' rule!

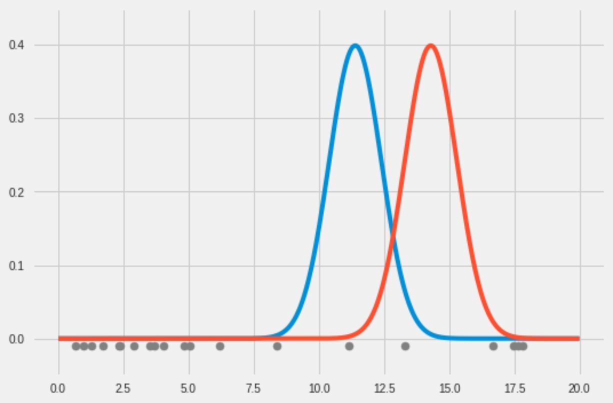
Prior dari Bayes' Rule

- Bisa dibuat tetap, atau
- Dibuat berubah-ubah, i.e.

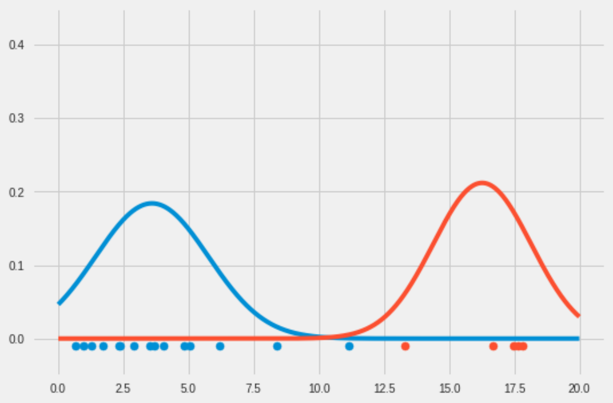
$$P(a) = \frac{a_1 + a_2 + ... + a_n}{n}$$

$$P(b) = 1 - P(a)$$

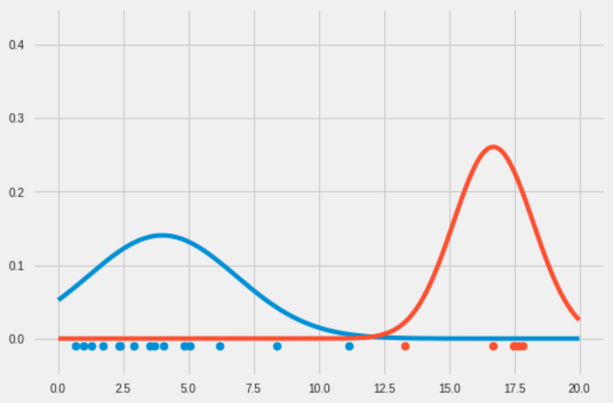
Mixture Models 1D



Mixture Models 1D



Mixture Models 1D



Berapa nilai K?

- Model probabilistik $\rightarrow$ *maximum likelihood*

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} \sum_{k=1}^{K} P(x_i|k)P(k)$$

$$\mathcal{L} = \log P(x_i, ..., x_n) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} P(x_i|k)P(k)$$

Berapa nilai K?

- Model probabilistik $\rightarrow$ *maximum likelihood*

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} \sum_{k=1}^{K} P(x_i|k)P(k)$$

$$\mathcal{L} = \log P(x_i, ..., x_n) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} P(x_i|k)P(k)$$

- $\mathcal{L}$ bisa dimaksimalkan dengan membuat $K = n \rightarrow$ *overfitting*!

Berapa nilai K?

- Model probabilistik $\rightarrow$ *maximum likelihood*

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} \sum_{k=1}^{K} P(x_i|k)P(k)$$

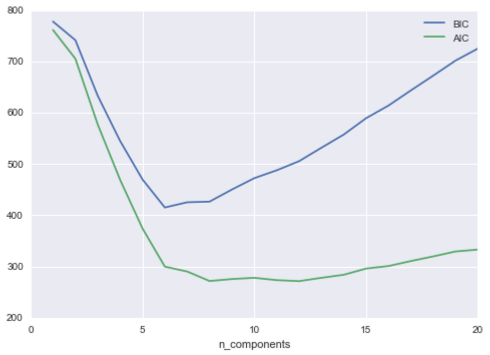
$$\mathcal{L} = \log P(x_i, ..., x_n) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} P(x_i|k)P(k)$$

- $\mathcal{L}$ bisa dimaksimalkan dengan membuat $K = n \rightarrow$ *overfitting*!
- Occam's razor
  - Bayes. Inf Criterion (BIC): $\max_p(\mathcal{L} - \frac{1}{2} p \log n)$
  - Akaike Inf Criterion (AIC): $\min_p(2p - \mathcal{L})$

dengan $\mathcal{L}$ adalah *log likelihood* dan p adalah jumlah parameter

Tenang, sudah ada di scikit-learn!

AIC dan BIC



Gambar 1: Nilai terbaik adalah saat `n_components` antara 8-12 [VanderPlas, 2016]

Hierarchical Clustering

Memilih Nilai K

- Tidak ada algoritma yang bisa memilih nilai K secara langsung

Memilih Nilai K

- Tidak ada algoritma yang bisa memilih nilai K secara langsung
- Memilih $K \sim$ pertanyaan *granularity*

Memilih Nilai K

- Tidak ada algoritma yang bisa memilih nilai K secara langsung
- Memilih $K \sim$ pertanyaan *granularity*
- Bagaimana kalau kita membuat hierarki alih-alih menentukan satu nilai K?

Hierarki Klaster

- Semakin bawah, semakin granular
- Strategi
    - *top-down*: satu klaster besar, bagi secara rekursif
    - *bottom-up*: dari *singletons*, gabung dengan kriteria tertentu

Hierarchical K-means

- *Top-down*, nilai K ditentukan di awal, bagi secara rekursif
- Setiap rekursi menjadi semakin lebih cepat karena semakin sedikit data yang dimasukkan klaster
- *Greedy*, ada kemungkinan titik yang berdekatan tidak ada klaster yang sama

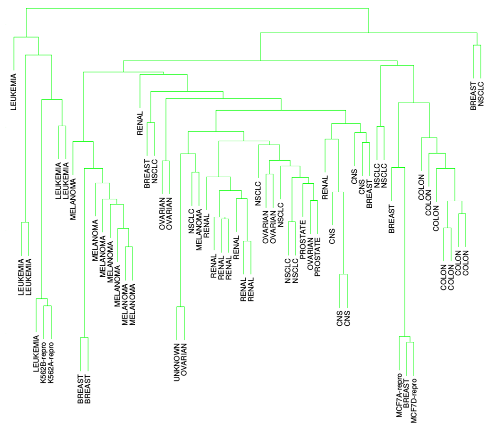
Agglomerative Clustering

1. Mulai dari sejumlah C dengan n *singletons*
2. Ulangi hingga menjadi satu klaster
   a. Cari sepasang klaster terdekat $\min_{i,j} D(c_i, c_j)$
   b. Gabungkan c_i, c_j menjadi satu klaster c_{i+j}
   c. Buang c_i, c_j dari C, masukkan c_{i+j}

Agglomerative Clustering

- *Bottom-up*, setiap poin yang berdekatan akan ada dalam satu klaster
- Menghasilkan **dendogram**
- Perlu mendefinisikan metode pengukuran jarak antarklaster

Dendogram



Gambar 2: Dendogram dari *agglomerative clustering* dengan *average linkage* untuk data *human tumor microarray* [Friedman, et al., 2001]

Pengukuran Jarak Antarklaster

- **Single link**: $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  Jarak antara elemen terdekat dari kedua klaster

Pengukuran Jarak Antarklaster

- **Single link**: $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  Jarak antara elemen terdekat dari kedua klaster
- **Complete link**: $D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  Jarak antara pasangan elemen terjauh dari kedua klaster

Pengukuran Jarak Antarklaster

- **Single link**: $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  Jarak antara elemen terdekat dari kedua klaster
- **Complete link**: $D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  Jarak antara pasangan elemen terjauh dari kedua klaster
- **Average link**: $D(c_1, c_2) = \frac{1}{|c_1|} \frac{1}{|c_2|} \sum_{x_1 \in c_1} \sum_{x_2 \in c_2} D(x_1, x_2)$
  Rata-rata dari jarak setiap pasangan antarklaster

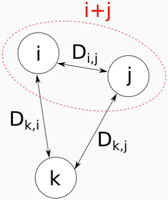
Pengukuran Jarak Antarklaster

- **Single link**: $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  Jarak antara elemen terdekat dari kedua klaster
- **Complete link**: $D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  Jarak antara pasangan elemen terjauh dari kedua klaster
- **Average link**: $D(c_1, c_2) = \frac{1}{|c_1|} \frac{1}{|c_2|} \sum_{x_1 \in c_1} \sum_{x_2 \in c_2} D(x_1, x_2)$
  Rata-rata dari jarak setiap pasangan antarklaster
- **Centroids**: $D(c_1, c_2) = D\left(\left(\frac{1}{|c_1|} \sum_{x \in c_1} \mathbf{x}\right), \left(\frac{1}{|c_2|} \sum_{x \in c_2} \mathbf{x}\right)\right)$
  Jarak antara *centroids* dari kedua klaster

- **Single link**: $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  Jarak antara elemen terdekat dari kedua klaster
- **Complete link**: $D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  Jarak antara pasangan elemen terjauh dari kedua klaster
- **Average link**: $D(c_1, c_2) = \frac{1}{|c_1|} \frac{1}{|c_2|} \sum_{x_1 \in c_1} \sum_{x_2 \in c_2} D(x_1, x_2)$
  Rata-rata dari jarak setiap pasangan antarklaster
- **Centroids**: $D(c_1, c_2) = D\left(\left(\frac{1}{|c_1|} \sum_{x \in c_1} \mathbf{x}\right), \left(\frac{1}{|c_2|} \sum_{x \in c_2} \mathbf{x}\right)\right)$
  Jarak antara *centroids* dari kedua klaster
- **Ward's method**: $TD_{c_1 \cup c_2} = \sum_{x \in c_1 \cup c_2} D(x, \mu_{c_1 \cup c_2})^2$
  Perubahan total jarak dengan *centroids* yang dihasilkan

Lance-Williams Algorithm

1. $D_{i,j}$ = jarak antara semua pasangan x_i dan x_j antara dua klaster
2. Untuk N iterasi:
   a. $i, j = \arg\min D_{i,j}$, i.e. pasangan klaster terdekat
   b. tambahkan klaster $i + j$, buang klaster i dan j
   c. untuk setiap sisa klaster k:
   $D_{k,i+j} = \alpha_i D_{k,i} + \alpha_j D_{k,j} + \beta D_{i,j} + \gamma |D_{k,i} - D_{k,j}|$



Lance-Williams Algorithm

$$D_{k,i+j} = \alpha_i D_{k,i} + \alpha_j D_{k,j} + \beta D_{i,j} + \gamma |D_{k,i} - D_{k,j}|$$

Metode	α_i	α_j	β	γ
Single linkage	0.5	0.5	0	-0.5
Complete linkage	0.5	0.5	0	0.5
Group average	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
Weighted group average	0.5	0.5	0	0
Centroid	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$\frac{-n_i \cdot n_j}{(n_i+n_j)^2}$	0
Ward	$\frac{n_i+n_k}{(n_i+n_j+n_k)}$	$\frac{n_j+n_k}{(n_i+n_j+n_k)}$	$\frac{-n_k}{n_i+n_j+n_k}$	0

Single link:

$D_{k,i+j} = \frac{1}{2}(D_{k,i} + D_{k,j} - |D_{k,i} - D_{k,j}|) = \min(D_{k,i}, D_{k,j})$

Salindia ini dibuat dengan
sangat dipengaruhi oleh Lavrenko (2014)

Referensi

Jake VanderPlas (2016)
In Depth: Gaussian Mixture Models
http://nbviewer.jupyter.org/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.12-Gaussian-Mixtures.ipynb

J. Friedman, T. Hastie, & R. Tibshirani (2001)
The Elements of Statistical Learning (Vol. 1)
Springer, Berlin: Springer series in statistics.

Terima kasih