Report on Homework 1

CS420, Machine Learning, Shikui Tu, Autumn 2017

Zelin Ye 515030910468

1 k-means vs GMM

Basically, k-means employs One-in-K assignment while GMM utilizes soft assignment. It turns out that k-means has bad robust. Therefore, I tend to introduce some of the ideas in GMM into k-means to form a new variant of k-means.

There are two main differences in my variant of k-means algorithm compared with the original:

- I introduce parts of the soft assignment into k-means to make it more robust.
- I affiliate a hyper parameter $Thres$ into original k-means. When the posteriori probablity is large than $Thres$, it can be retained, or it would be 0.

The pseudo-codes of my algorithm can refer to 1.

Algorithm 1: Improvements of k-means

---

input : The number of clusters K

output: $\pi_k, \mu_k, \Sigma_k, (k = 1, 2, ..., K)$

Initialize the means μ_k, covariances Σ_k, mixing coefficients π_k and threshold $Thres$.

Evaluating the initial value of the log likelihood.

while *the convergence criterion of parameters or log likelihood is not satisfied* **do**

E step. Evaluate the responsibilities using the current parameter values:

$$\omega \leftarrow \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum\limits_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

$$\gamma(z_{nk}) = \begin{cases} \omega & \omega > Thres \\ 0 & \omega \leq Thres \end{cases}$$

$$z_n \leftarrow \frac{e^{z_{n_i}}}{\sum\limits_{j=1}^{K} e^{z_{n_j}}}$$

M step. Re-estimate the parameters using the current responsibilities:

$$N_k \leftarrow \sum_{n=1}^{N} \gamma(z_{nk})$$

$$\mu_k^{new} \leftarrow \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$

$$\Sigma_k^{new} \leftarrow \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$$

$$\pi_k^{new} \leftarrow \frac{N_k}{N}$$

Evaluate the log likelihood:

$$\ln p(X|\mu, \Sigma, \pi) \leftarrow \sum_{k=1}^{K} \ln\left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}.$$

return $\pi_k, \mu_k, \Sigma_k, (k = 1, 2, ..., K)$;

With the introduction of probability, my algorithm is somewhat similar to EM algorithm and would be more robust than k-means. Besides, the $Thres$ would eliminate the negligible posteriori probabilities and improve accuracy under the premise of robustness.

Generally speaking, more hyper parameters would result in more tricks, my algorithm is no exception. The choice of $Thres$ may greatly affect the result.

Additionally, my algorithm also encounters the initialization and unknown K problems. But I think such problems do not contradict with my core ideas, and could be resolved by introducing more components (e.g. the idea of RPCL).

2 k-mean vs CL

3 Model Selection of GMM

References

[1] S. P. Bhavnani, D. Muñoz, and A. Bagai, "Data science in healthcare: implications for early career investigators," *Circulation: Cardiovascular Quality and Outcomes*, vol. 9, no. 6, pp. 683–687, 2016.

[2] Y. Liang and A. Kelemen, "Big data science and its applications in health and medical research: Challenges and opportunities," *Austin Journal of Biometrics & Biostatistics*, vol. 7, no. 3, 2016.

[3] S. O' Halloran, S. Maskey, G. McAllister, D. K. Park, and K. Chen, "Data science and political economy: application to financial regulatory structure," *RSF*, 2016.

[4] W. Xiao-fan, "Data science and social network: Big data, small world," *Science and Society*, vol. 1, p. 003, 2014.

[5] C. Vande Kerckhove, *Data science for modeling opinion dynamics on social media*. PhD thesis, UCL-Université Catholique de Louvain, 2017.

[6] "Numpy." http://www.numpy.org/.

[7] "Pandas." http://pandas.pydata.org/.

[8] "matplotlib." https://matplotlib.org/.

[9] "scikit-learn." http://scikit-learn.org/stable/.