Software Requirements Specification for : Guassian Mixture Model based on EM algorithm

WONG, Kim Ying

February 6, 2024

Contents

Revision History

Date	Version	Notes
1 Feb 2024	1.0	SRS First Draft

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document, we would consider all values to be unit-free. The rationale behind is that all units used in this document or model depend on the input dataset, while there is no general constraint on the unit of input. Therefore the input and the derived the statistical relation could be any units.

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units (if exists). The choice of symbols was made to be consistent with the statistics literature and with existing documentation for the Gaussian Mixture Model. The symbols are listed in alphabetical order.

symbol	**Description**
M	number of data-points
N	number of predictor variables
K	number of clusters needed
z	latent variable in GMM
p(x)	Gaussian mixture distribution
Σ	covariance
μ	mean
π	mixing coefficient
$N(\mu_k, \Sigma_k)$	Gaussian distribution with mean μ and variance Σ
p(x)	probability density function
$\gamma(z_k)$	responsibility

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
	put an expanded version of your program name here (as appropriate)
TM	Theoretical Model
GMM	Gaussian Mixture Model
EM	Expectation Maximization
GMM-EM	Software Name

Add any other abbreviations or acronyms that you add

1.4 Mathematical Notation

We borrow the convention for mathematical notation from the book Pattern Recognition and Machine Learning.
Typographic Conventions:
We denote vector as a lowercase bold roman letters such as $\mathbf{x}$. Matrices are denoted by uppercase bold roman letters such as $\mathbf{M}$. A single real-valued variable will be denoted as a lowercase roman letter such as x.
We introduce the notation for conditional probability as: $p(Y|X)$ as the probability of Y given X, where X is the marginal probability, or simply called probability of X; and the joint probability is defined as: $p(X, Y)$. The normal distribution with mean μ and covariance Σ is denoted as $N(\mu, \Sigma)$. These notation could be generalized to higher dimension by replacing real-valued numbers with vectors or matrices.

2 Introduction

Mixture model is a probabilistic model that describes the datasets or measurement in a linear combination of some basic distributions. Gaussian mixture model falls into a subset of the mixture model, where Gaussian distribution is used as a basis. In general, almost all continuous density could be approximated as sufficient number of Gaussian mixtures with appropriate mean and covariance.
Therefore, it can be used in various cases in machine learning, such as clustering and density estimation. The GMM-EM aims at implementation of the Gaussian mixture model for clustering fundamentally. This section is divided into 4 parts:

- 2.1 Purpose of document
- 2.2 Scope of requirements
- 2.3 Characteristics of Intended reader
- 2.4 Organization of document

2.1 Purpose of Document

The document serves as a outline for the requirement specification for the GMM-EM. The reader could refer to this document for the general and specific system description. Theoretical model and their assumption will be detailed in this document.

2.2 Scope of Requirements

The GMM-EM would be a general library in implementation of Gaussian mixture model based on the EM algorithm with a fundamental focus in clustering problem. We will limit to deal with any dataset with only real-valued number without any missing value. The software would guarantee a convergence in solution but not the optimality.

2.3 Characteristics of Intended Reader

The intended readers should acquire basic understanding on linear algebra, calculus and statistics, preferably exposure on the basic statistical learning algorithm. Undergraduate introduction courses in these topics should suffix the above requirements.

2.4 Organization of Document

The document consist of 5 parts:

- Section 3 General System Description
- Section 4 Specific System Description

- Section 5 Requirements
- Section 6 Likely Changes and Section 7 Unlikely Changes
- Section 8 Traceability Matrices and Graphs

3 General System Description

This section provides general information about the software GMM-EM. It consists of three parts:

- 3.1 System Context
- 3.2 User Characteristics

3.1 System Context

The software follows the structure as Figure shown. The user is supposed to input a real-valued dataset as a matrix to the GMM-EM. The GMM-EM will give an array which contains the predicted label for each data points in the dataset.

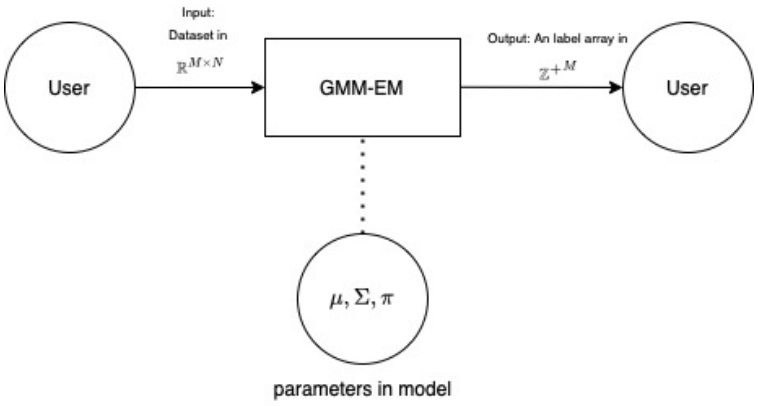

Figure 1: System Context

- User Responsibilities:
  - Provides a real-valued number of dataset without missing value in a matrix form
  - Specify the number of clusters needed (optional)
- GMM-EM Responsibilities:
  - Detect any wrong data type in input like containing string
  - Return an array of label for each data point to which cluster it should belong.
  - Identify the number of clusters needed if users do not provide

3.2 User Characteristics

The user should acquire basic programming skill and be able to perform data preprocessing for a dataset. It is also preferable that users know some common libraries in handling different data formats such as csv or json.

4 Specific System Description

- Section 4.1 Problem Description
- Section 4.2 Solution Characteristics Specification

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, definitions and finally the instance models.

4.1 Problem Description

GMM-EM is intended to implement a library on Gaussian Mixture model which is specifically used in clustering. The software tackle a general clustering problem for different datasets

4.1.1 Terminology and Definitions

In this section, we would explain the terminology used in our theoretical model.

- Likelihood function: The degree of similarity of the observed dataset similar to what we guess about the data points
- mixture of Gaussian: data points where are from different Gaussian distributions

4.1.2 Goal Statements

Given the a dataset without missing value and infinity value which perferably designed for clustering or classification , the goal statements are:

GS1: Predict which cluster should each data point belongs to

GS2: Detect how many clusters should use if number of cluster are not specified

4.2 Solution Characteristics Specification

This section illustrate the theory which GMM-EM fundamentally relied on. We start from statistical assumption, to statistical theorem and definitions and the provide more details on the mathematical foundation about the training process for GMM-EM The instance models that govern are presented in Subsection **??**. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.3 Scope Decisions

The GMM-EM will fundamentally focus on the real-valued dataset which is designed for clustering or classification. Other GMM features or application such as density estimation may not be provided in our library.

4.3.1 Assumptions

A1: We assume properties of probability density function holds here, which include non-negativity and normalization

A2: We assume the dataset could be approximated by mixtures of Gaussian.

A3: We assume the dataset is real-valued without any missing value.

4.3.2 Theoretical Models

In this section, we will introduce the theoretical foundation for the GMM-EM. It fundamentally relied on the fact the data could always approximated by a mixture of Gaussian. The mathematical formulation is listed below.

RefName: **TM:GMM**

Label: mixture of Gaussian

Equation: $p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k N(x|\mu_\mathbf{k}, \mathbf{\Sigma_k})$ with $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$

Description: The above equation gives the mathematical formulation on GMM. It states that for a Gaussian mixture distribution, it could be written as the linear combination of Gaussian distribution with a certain covariance $\mathbf{\Sigma_k}$ and mean $\mu_\mathbf{k}$. From the normalization and non-negativity of the probability density function, the condition for mixing coefficient is imposed as above. It is a direct result from our basic assumption (A1).

Notes: None.

Source: Pattern Recognition and Machine Learning by Christopher M. Bishop

Ref. By: GD2, GD1

Preconditions for TM:GMM: None

Derivation for TM:GMM: Not Applicable

RefName: **TM:BT**

Label: Bayes' Theorem

Equation: $P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)}$

Description: The above equation gives the mathematical foundation on Bayesian Statistics. This define the Log likelihood function which will be maximized in parameter training process.

Notes: None.

Source: Pattern Recognition and Machine Learning by Christopher M. Bishop

Ref. By: GD2, GD1

Preconditions for TM:BT: None

Derivation for TM:BT: Not Applicable

4.3.3 General Definitions

We will define the Log Likelihood function and responsibilities in this section. The maximization in Log likelihood function will be key for training the GMM-EM.

Number	GD1
Label	**Responsibility**
Equation	$\gamma(z_k) = \frac{\pi_k N(\mathbf{x}\|\mu_\mathbf{k},\mathbf{\Sigma_k})}{\sum_{j=1}^{K} \pi_k N(\mathbf{x}\|\mu_\mathbf{j},\mathbf{\Sigma_j})}$
Description	The responsibility is defined as conditional probability of a latent variable $\mathbf{z}$ given that we know about $\mathbf{x}$. We could under this as given that a data point, how probable it would be in a certain cluster (which could be interpreted as one-hot encoding of $\mathbf{z}$) The mathematical symbols are defined in Section 1.2
Source	Pattern Recognition and Machine Learning by Christopher M. Bishop
Ref. By	IM1

Number	GD2
Label	**Log likelihood function**
Equation	$\ln p(\mathbf{X}\|\pi,\mu,\mathbf{\Sigma}) = \sum_{n=1}^{N} ln \sum_{k=1}^{K} \pi_k N(\mathbf{x}\|\mu_\mathbf{k},\mathbf{\Sigma_k})$
Description	Likelihood function is a direct result from given by the expression of Gaussian mixture with the Bayes' Theorem. We impose the log scale to the likelihood and obtain the log likelihood function. The mathematical symbols are defined in Section 1.2
Source	Pattern Recognition and Machine Learning by Christopher M. Bishop
Ref. By	IM1

4.3.4 Instance Models

We will outline the EM part in GMM-EM. Instead of giving details in implementation, we will provide a rather abstract but mathematically non-rigorous treatment in this section.

Number	IM1
Label	**EM Algorithm**
Input	Initialized $\pi_{\mathbf{k}}, \mu_{\mathbf{k}}, \mathbf{\Sigma_k}$ based on the input dataset $\mathbf{X} \in \mathbb{R}^{M \times N}$
Output	Best parameter in $\pi_{\mathbf{k}}, \mu_{\mathbf{k}}, \mathbf{\Sigma_k}$ and from this we could obtain the label array $\in \mathbb{Z}^{+\mathbb{M}}$ from this.
Description	EM algorithm aims at maximize the likelihood function with respect to parameters. The algorithm Iterative update on $\pi_{\mathbf{k}}, \mu_{\mathbf{k}}, \mathbf{\Sigma_k}$ based on the responsibility and log likelihood calculated until convergence in $\pi_{\mathbf{k}}, \mu_{\mathbf{k}}, \mathbf{\Sigma_k}$ or log likelihood.
Sources	Pattern Recognition and Machine Learning by Christopher M. Bishop

A rough derivation of EM Algorithm

To avoid over-complexity, instead of giving the rigorous derivation, we outline the idea here. The idea is to guess the parameters and update to the new set of parameters which maximize the log likelihood. The implementation of update detail is ignored here but the idea is basically taken the derivative for likelihood function and simplify the expression with the concept of responsibility.

4.3.5 Input Data Constraints

Each value x in the dataset should satisfied $x \in \mathbb{R}$.

4.3.6 Properties of a Correct Solution

The correct solution should satisfy the convergence of EM algorithm. The solution should converge to an at least local optimal for the parameters in model that the algorithm would no longer update.

5 Requirements

The function requirements and nonfunctional requirements for GMM-EM are defined in this section

5.1 Functional Requirements

R1: Real-valued dataset without missing value should be given. The dataset should be suitable for clustering or classification for a meaningful result.

R2: Convergence of solution is guaranteed (IM1)

R3: We verified the model with existing library with some simple dataset as test case.

R4: The output will be an array specific the which cluster data points should belong to. (IM1)

Every IM should map to at least one requirement, but not every requirement has to map to a corresponding IM.

5.2 Nonfunctional Requirements

NFR1: **Accuracy** We do not guarantee an optimal clustering for our GMM-EM,specially when user input a dataset which is not intended to perform clustering or classification. The model still provides output but may not be applicable in this context. In the general clustering or classification case, other existing library of GMM will be used as a benchmark for the accuracy. The result from GMM-EM shall be comparable to these existing library.

NFR2: **Usability** We will give a detailed description of function call, class in the library, which should include the at least function/class attributes, class method and function/class arguments taken to facilitate the user.

NFR3: **Maintainability** A documentation for the library will be created to describe the design of the program, including algorithms, function call, class and other possible implementation details. Future development could be facilitated based by the documentation.

NFR4: **Portability** The GMM-EM should be a cross-platform library. The library should be able to operated in Windows 7+ and MacOS 10+ with the corresponding compiler or interpreter.

5.3 Rationale

To reduce the complexity and retain the software to a specific domain in machine learning tasks, the scope of GMM-EM would be focused on the clustering in real-valued dataset.

6 Likely Changes

LC1: The input dataset will be in form of matrix, but the actual data structure implemented depends on the programming languages. For example, pandas dataframe will be used in python or vector will be used in C++.

7 Unlikely Changes

LC2: EM algorithm as the training algorithm is unlikely changed as it is widely adapted in GMM.

8 Traceability Matrices and Graphs

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed.

	R1	R2	R3	R4
IM1		X		X

Table 1: Traceability Matrix Showing the Connections Between Requirements and Instance Models

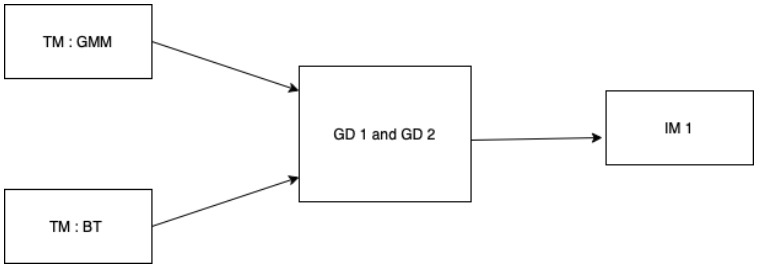

Figure 2: Traceability Graph

9 Values of Auxiliary Constants

References

Author Author. System requirements specification. https://github.com/..., 2019.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL http://citeseer.ist.psu.edu/428727.html.

David L. Parnas. On the criteria to be used in decomposing systems into modules. *Comm. ACM*, 15(2):1053–1058, December 1972.

David L. Parnas. Designing software for ease of extension and contraction. In *ICSE '78: Proceedings of the 3rd international conference on Software engineering*, pages 264–277, Piscataway, NJ, USA, 1978. IEEE Press. ISBN none.

David L. Parnas and P.C. Clements. A rational design process: How and why to fake it. *IEEE Transactions on Software Engineering*, 12(2):251–257, February 1986.

D.L. Parnas, P.C. Clement, and D. M. Weiss. The modular structure of complex systems. In *International Conference on Software Engineering*, pages 408–419, 1984.

James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.

W. Spencer Smith. Systematic development of requirements documentation for general purpose scientific computing software. In *Proceedings of the 14th IEEE International Requirements Engineering Conference, RE 2006*, pages 209–218, Minneapolis / St. Paul, Minnesota, 2006. URL http://www.ifi.unizh.ch/req/events/RE06/.

W. Spencer Smith and Nirmitha Koothoor. A document-driven method for certifying scientific computing software for use in nuclear safety analysis. *Nuclear Engineering and Technology*, 48(2):404–418, April 2016. ISSN 1738-5733. doi: http://dx.doi.org/10.1016/j.net.2015.11.008. URL http://www.sciencedirect.com/science/article/pii/S1738573315002582.

W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ågerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

W. Spencer Smith, Lei Lai, and Ridha Khedri. Requirements analysis for engineering computation: A systematic approach for improving software reliability. *Reliable Computing, Special Issue on Reliable Engineering Computation*, 13(1):83–107, February 2007.

W. Spencer Smith, John McCutchan, and Jacques Carette. Commonality analysis of families of physical models for use in scientific computing. In *Proceedings of the First International Workshop on Software Engineering for Computational Science and Engineering (SECSE 2008)*, Leipzig, Germany, May 2008. In conjunction with the 30th International Conference on Software Engineering (ICSE). URL http://www.cse.msstate.edu/~SECSE08/schedule.htm. 8 pp.

W. Spencer Smith, John McCutchan, and Jacques Carette. Commonality analysis for a family of material models. Technical Report CAS-17-01-SS, McMaster University, Department of Computing and Software, 2017.