ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 9, Issue 2, March 2020

Identification of Non-Technical Losses by using Gaussian mixture model

Kalyan Pal, Bhavesh Kr Chauhan

Abstract:** Non-technical losses (NTL) identification has been paramount in the last years. However, it is not straightforward to obtain labelled datasets to perform a supervised NTL recognition task. In this paper we first introduce the basic concepts of random variables and the associated distributions. These concepts are then applied to Gaussian random variables and mixture-of-Gaussian random variables. This preface points to the Gaussian mixture model (GMM) is used to fit the practical data such as features selection when assigning of the mixture-of-Gaussian random variables. A Gaussian Mixture Model is a probabilistic model that considers all the data points are created from a mixture of a finite number of Gaussian distributions with unknown parameters. We discuss some main advantages of GMMs in data clustering, among which is the easy way of using them to fit the data of a wide range of features using the EM algorithm. We describe the principle of maximum likelihood and the related EM algorithm for parameter estimation of the GMM in some detail as it is still a widely used method in data clustering. We finally discuss some major weakness of using GMMs in clustering for NTL identification, motivating to introduce the Optimum-Path Forest, a new model of NTL identification.

Keywords: **Non-Technical Losses, GMM, EM Algorithm, Clustering, Optimum-Path Forest.**

I. INTRODUCTION

The Gaussian Mixture Model is a probabilistic model that considers all the data points are created from a mixture of a finite number of Gaussian distributions with unknown parameters [1]. It is possible to imagine such approach as a generalization of k-means clustering by adding the information about the covariance structure of the data, as well as the centres of the latent Gaussians. Data clustering can often be done by using Gaussian mixture models (GMM). Usually, fitted GMMs cluster by assigning query data points to the multivariate normal components that maximize the component posterior probability given the data. This method of assigning a data point to exactly one cluster is called hard clustering [2].

However, GMM clustering is more flexible because it can be viewed as a fuzzy or soft clustering method. Soft clustering methods assign a score to a data point for each cluster. The value of the score indicates the combined strength of the data point to the cluster. As opposed to hard clustering methods, soft clustering methods are flexible in that they can assign a data point to more than one cluster. When clustering with GMMs, the outcome is the posterior probability. Moreover, GMM clustering can produce clusters that have different sizes and correlation structures within them. Because of this, GMM clustering can be more appropriate to use than, e.g. k-means clustering [3].

Like most clustering methods, the number of desired clusters must be specified before fitting the model. The number of clusters indicate the number of components in the GMM. For GMMs, it is best practice to also consider the:

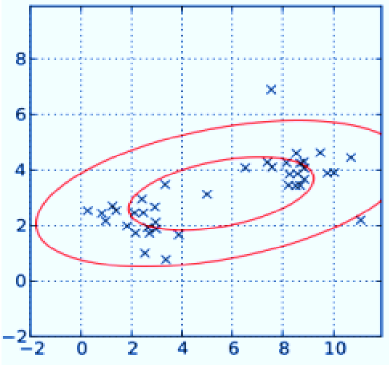
- Component covariance structure: It can specify diagonal or full covariance matrices, or whether all components have the same covariance matrix.
- Initial conditions: The Expectation-Maximization (EM) algorithm fits the GMM. Like the k-means clustering algorithm, EM is responsive to initial conditions and might enter in to a local optimum. One can specify initial cluster assignments for data points or let them be randomly chosen.
- Regularization parameter. When the predictors are more than data points, then estimation stability can be regularized.
- The number of components, K, in a GMM determines number of subpopulations or clusters. A GMM increases in complexity as K increases.

First, if the model is having some hidden, not observable parameters, then GMM should be used. This is because, instead of assigning a flag that the point belongs to certain cluster as in the classical k-Means, this algorithm is assigning a probability to each point to belong to certain cluster. Then, GMM is producing non-convex clusters, which can be controlled with the variance of the distribution. The k-Means is a special case of

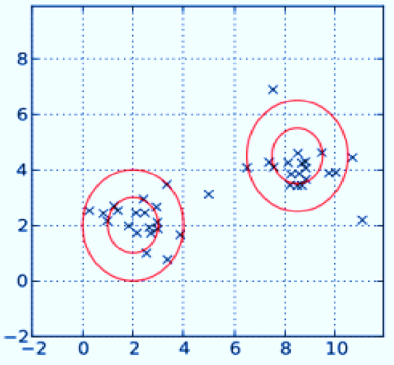
GMM, such that the probability of a one point to belong to a certain cluster is 1, and all other probabilities are 0, and the variance is 1, which a reason why k-Means produces only spherical clusters [3].

II. THE GMM MODEL

This Model is composed of K multivariate normal density components, where $K \neq 0$,is a positive integer. Each component has a d-dimensional mean (d is a positive integer), d-by-d covariance matrix, and a mixing proportion j. It determines the proportion of the density composed by component$j$, $j = 1, \dots, K$. One hint that data might follow a mixture model is that the data looks multimodal, i.e. there is more than one "peak" in the distribution of data. Trying to fit a multimodal distribution with a unimodal (one "peak") will generally give a poor fit, as shown in the example below. Since many simple distributions are unimodal, an obvious way to model a multimodal distribution would be to assume that it is generated by multiple unimodal distributions. For various theoretical reasons, Gaussian distribution is the most commonly used distribution in modeling real world clustering data. Thus, it makes intuitive sense of modeling multimodal data as a mixture of many unimodal Gaussian distributions. Moreover, GMMs maintain many of the theoretical and computational benefits of Gaussian models, that make them practical for efficiently modeling very large datasets.



Fit with one Gaussian distribution



Fit with GMM with 2 components

A Gaussian mixture model is identified by two types of values, the mixture component weights and the component means and variances/covariances. For a Gaussian mixture model with K components, the kthcomponent has a mean of $\mu_k$and variance of $\sigma_k$for the univariate case and a mean of$\vec{\mu}_k$and covariance matrix of $\sum k$for the multivariate case. The mixture component weights are defined as $\phi_k$for component C_k, with the constraint that $\sum_{i=1}^{K} \phi_i = 1$so that the total probability distribution normalizes to 1. If the component weights aren't learned, they can be viewed as a hypothetical distribution over components such that $p(x \text{ generated by component } C_k) = \phi_k$ [4]. If they are instead learned, they are the true estimates of the component probabilities given the data.

One dimensional Model

$$p(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(x \mid \mu_i, \sigma_i)$$

$$\mathcal{N}(x \mid \mu_i, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2} \right)$$

$$\sum_{i=1}^{K} \phi_i = 1$$

Multi-dimensional Model

$$p(\vec{x}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i)$$

$$\mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^K |\Sigma_i|}} \exp\left(-\frac{1}{2} (\vec{x} - \vec{\mu}_i)^{\mathrm{T}} \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i) \right)$$

$$\sum_{i=1}^{K} \phi_i = 1$$

GMM allows for mixed membership of points to clusters; which is one of the significance of its covariance structure. In k-means, a point belongs to one and only one cluster, whereas in GMM a point belongs to each cluster to a different degree; which is based on the probability of the point being generated from each cluster's (multivariate) normal distribution, with cluster center as the distribution's mean and cluster covariance as its covariance [5].

A huge amount of research has gone into finding ways of constraining GMMs to increase their evaluation speed and to optimize the trade-off between their flexibility and the amount of training data required to avoid over fitting. Both inside and outside the pattern recognition domain, the GMM is commonly used for modelling the data and for statistical classification. GMMs are well known for their ability to represent arbitrarily complex distributions with multiple modes. If the GMM parameters are discriminatively learned after they have been generatively trained by EM to maximize its probability of generating the observed features in the training data, the accuracy in pattern recognition can be drastically improved. The accuracy can also be improved by increasing (or concatenating) the input features with sequential or bottleneck features generated using neural networks [6].

III. EXPECTATION MAXIMIZATION

Expectation maximization is the technique most commonly used to estimate the mixture model parameters if the number of components K is known. Models are typically learned by using maximum likelihood estimation techniques in frequents probability theory, which seek to maximize the probability, or likelihood, of the observed data given by the model parameters. Unfortunately, finding the maximum likelihood solution for mixture models by differentiating the log likelihood and solving for 0 is usually analytically impossible.

Expectation maximization (EM) is a numerical technique for maximum likelihood estimation, and is usually used when closed form expressions for updating the model parameters can be calculated. Expectation maximization is an iterative algorithm and has the convenient property that the maximum likelihood of the data strictly increases with each subsequent iteration, meaning it is guaranteed to approach a local maximum point.

There are two steps in mixture models of Expectation maximization. The first step, known as the Expectation, or E, step, consists of calculating the expectation of the component assignments C_k for each data point $x_i \in X$ given the model parameters ϕ_k, $\mu_k$and σ_k.The second step is known as the Maximization, or M, step, which consists of maximizing the expectations calculated in the E step with respect to the model parameters. This step consists of updating the values ϕ_k, μ_k and σ_k [7].

The entire iterative process repeats until the algorithm converges, giving a maximum likelihood estimate. Intuitively, the algorithm works because knowing the component assignment $C_k$for each $x_i$makes solving for ϕ_k, μ_k and $\sigma_k$easy, while knowing ϕ_k, μ_k and σ_k makes inferring $p(C_k | x_i)$easy. The expectation step corresponds to the latter case while the maximization step corresponds to the former. Thus, by alternating between which values are assumed fixed, or known, maximum likelihood estimates of the non-fixed values can be calculated in an efficient manner.

The expectation maximization algorithm for Gaussian mixture models starts with an initialization step, which assigns model parameters to reasonable values based on the data. Then, the model iterates over the Expectation (E) and Maximization (M) steps until the parameters' estimates converge, i.e. for all parameters θ_t at iteration$t$, $|\theta_t - \theta_{t-1}| \leq \epsilon$ for some user-defined toleranceϵ.

The EM algorithm for a univariate Gaussian mixture model with K components is described below. A variable denoted $\hat{\theta}$ denotes an estimate for the valueθ. All equations can be derived algebraically by solving for each parameter as outlined in the section above titled EM for Gaussian Mixture Models.
Initialization Step:

- Randomly assign samples without replacement from the dataset $X = \{x_1, \ldots, x_N\}$ to the component mean estimates $\hat{\mu}_1, \ldots, \hat{\mu}_K$. E.g. for $K = 3$ and $N = 100$, set $\hat{\mu}_1 = x_{45}$, $\hat{\mu}_2 = x_{32}$, $\hat{\mu}_2 = x_{10}$

- Set all component variance estimates to the sample variance $\hat{\sigma}_1^2, \ldots, \hat{\sigma}_K^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$ where $\bar{x}$ is the sample mean $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$

- Set all component distribution prior estimates to the uniform distribution $\hat{\phi}_1, \ldots, \hat{\phi}_K = \frac{1}{K}$

Expectation (E) Step:

Calculate $\forall i, k$:

$$\hat{\gamma}_{ik} = \frac{\hat{\phi}_k \mathcal{N}(x_i \mid \hat{\mu}_k, \hat{\sigma}_k}{\sum_{j=1}^{K} \hat{\phi}_j \mathcal{N}(x_i \mid \hat{\mu}_j, \hat{\sigma}_j)})$$

$\hat{\gamma}_{ik}$ is the probability that x_i is generated by component C_k. Thus, $\hat{\gamma}_{ik} = p(C_k | x_i, \hat{\phi}, \hat{\mu}, \hat{\sigma})$

Maximization (M) Step:

Using the $\hat{\gamma}_{ik}$ calculated in the Expectation step, calculate in the following order $\forall k$:

$$\hat{\phi}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N}$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} x_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$$

$$\hat{\sigma}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} (x_i - \hat{\mu}_k)^2}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$$

Once the EM algorithm has run to completion, the fitted model can be used to perform various forms of inference such as clustering.A similar procedure adopted for k-means is also used here. After estimating the parameters of each cluster of Gaussian distribution using E-M algorithm, the cluster's centre (mean parameter) is labelled with the same label of its nearest neighbour, and then such label is propagated to all elements that fall in the very same cluster. But when cluster sequence information is considered, the GMM is no longer a good model as it contains no sequence information [8].

Given a univariate model's parameters, the probability that a data point $x$belongs to component $C_i$is calculated using Bayes' Theorem:

$$p(C_i \mid x) = \frac{p(x, C_i)}{p(x)} = \frac{p(C_i)p(x \mid C_i)}{\sum_{j=1}^{K} p(C_j)p(x \mid C_j)} = \frac{\phi_i \mathcal{N}(x \mid \mu_i, \sigma_i)}{\sum_{j=1}^{K} \phi_j \mathcal{N}(x \mid \mu_j, \sigma_j)}$$

IV. LIMITATIONS OF GMM

Considering the problem of non-technical losses identification, two problems are usually faced: i) it is not straightforward to design a labelled dataset for such purposes, and ii) it is difficult to build a balanced dataset, since the number of irregular consumers is often lower than regular consumers. The main reason for that concerns the problem of associating to a given consumer the thief label, since some legal problems are usually associated with that task [9].

Despite all their advantages, GMMs have a serious shortcoming. That is, GMMs are statistically inefficient for modelling data that lie on or near a nonlinear manifold in the data space. For example, modelling the set of points that lie very close to the surface of a sphere only requires a few parameters using an appropriate model class, but it requires a very large number of diagonal Gaussians or a fairly large number of full-covariance Gaussians [10].

One of the main problems of GMMs is that they assume a Gaussian data distribution which is not valid for all cases. Another point is that, in several situations, the parameters of the Gaussian probability density function used for data modelling are unknown and need to be estimated by using a separate algorithm such as Expectation-Maximization (EM). Trying to avoid the high computational cost of SVMs, which cannot hold efficiency and effectiveness in large datasets, due to their expensive training phase, some authors have used GMMs for selecting the best samples to compose SVMs training data, however, the assumption of Gaussian data distribution cannot be held for all datasets [11].

V. OPTIMUM-PATH FOREST

Recently, a novel framework was introduced for graph-based classifiers that reduce the pattern recognition problem to an optimum-path forest computation (OPF) in the feature space induced by that graph. This kind of classifier does not interpret the classification task as a hyperplane optimization problem, but rather as a combinatorial optimum-path computation based on certain key samples (prototypes) to the remaining nodes. Every prototype is considered as a root for its optimum-path tree, and each node is classified according to the strength of its connection to the prototype, which defines a discrete optimal partition (influence region) of the feature space. OPF-based classifiers have certain advantages: 1) they are free of parameters, 2) they do not assume any special shape or separability of the feature space, and 3) they run the training phase faster so real-time applications for fraud detection in electrical systems are feasible [12].

VI. CONCLUSION

The main difficulty in learning GMM from unlabelled data is that it is one usually does not know which points came from which latent component. The Expectation-Maximization (E-M) algorithm is usually implemented by GMM to fine-tune each Gaussian mixture, which basically aims at learning the mean and covariance of each model, as well as its weight to be used in the mixture computation. Finally, test samples are classified by associating them to the most likely Gaussian distribution. Since it might be difficult to learn Gaussian mixture models from unlabelled data, E-M tries to circumvent this problem by an iterative process that assumes initial random components and, for each point, it estimates the probability of being part of each component of the model. Then, one takes the parameters to maximize the probability of the data given those assignments. This process is repeated until it met some convergence criterion. Therefore, other types of model such as Optimum-Path Forest, which can capture better properties of features, are expected to work better than GMMs for acoustic modelling of features. The new models should more effectively exploit information embedded in a large window of frames of features than GMM.

REFERENCES

[1] Scikit-learn developers, User guide, 2016. Available at http:// scikit-learn.org/dev/user_guide.html.

[2] Patrick Glauner, Andre Boechat, Lautaro Dolberg, Radu State, Franck Bettinger,Y Rangoni and Diogo Duarte, “Large-Scale Detection of Non-Technical Losses in Imbalanced Data Sets”, arXiv:1602.08350v1 [cs.LG], 26 Feb 2016.

[3] M. Nazari, A. Sayadiyan, and S.M. Valiollahzadeh, “Probabilistic SVM/GMM classifier for speaker-independent vowel recognition in continues speech,” ACM Computing Research Repository, vol. abs/0812.2411, 2008.

[4] D. Yu and L. Deng, “Automatic Speech Recognition”, Signals and Communication Technology, Springer-Verlag London 2015.

[5] J. P. Papa, Aparecido N.Marana, Andre A. Spadotto, Rodrigo C. Guido, Alexandre X. Falcao, “Robust and Fast Vowel Recognition Using Optimum-Path Forest”, IEEE, 2010.

[6] C. Ramos, J. P. Papa, Leandro Aparecido Passos Junior, Douglas Rodrigues, “Unsupervised Non-Technical Losses Identification Through Optimum-Path Forest”, Electric Power Systems Research, Research Gate, June 2016.

[7] J. Guerrero, C. Leon, I. Monedero, F. Biscarri, J. Biscarri, Improving knowledge-based systems with statistical techniques, text mining, and neural networks for non-technical loss detection, Knowledge-Based Systems 71, 2014.

[8] S. Pan, T. Morris, U. Adhikari, “Developing a hybrid intrusion detection system using data mining for power systems”, IEEE Transactions on Smart Grid 6, 2015.

[9] K. Yap, S. Tiong, J. Nagi, J. Koh, F. Nagi, Comparison of supervised learning techniques for non-technical loss detection in power utility, International Review on Computers and Software 7, 2012.

[10] C. Ramos, A. Souza, R. Nakamura, J. Papa, Electrical consumers data clustering through optimum-path forest, in: 2011 16th International Conference on Intelligent System Application to Power Systems, pp. 1–4.

[11] D.A. Reynolds and R.C. Rose, “Robust text-independent speaker identification using Gaussian mixture speaker models,” IEEE Trans. on Speech and Audio Proc., vol. 3, no. 1, pp. 72– 83, 1995.

[12] J. P. Papa, A. Falcao, and C. Suzuki, “Supervised pattern classification based on optimum-path forest,” Int. J. Imag. Syst. Technol., vol. 19, no. 2, pp. 120–131, 2009.