Generalized Gaussian Distributions for Sequential Data Classification

M. Bicego[1], D. Gonzalez-Jimenez[2], E. Grosso[1], J.L. Alba Castro[2]
[1] *University of Sassari (Italy),* [2] *University of Vigo (Spain)*
{*bicego,grosso*}*@uniss.it,* {*danisub,jalba*}*@gts.tsc.uvigo.es*

Abstract

It has been shown in many different contexts that the Generalized Gaussian (GG) distribution represents a flexible and suitable tool for data modeling. Almost all the reported applications are focused on modeling points (fixed length vectors); a different but crucial scenario, where the employment of the GG has received little attention, is the modeling of sequential data, i.e. variable length vectors. This paper explores this last direction, describing a variant of the well known Hidden Markov Model (HMM) where the emission probability function of each state is represented by a GG. A training strategy based on the Expectation Maximization (E-M) algorithm is presented. Different experiments using both synthetic and real data (EEG signal classification and face recognition) show the suitability of the proposed approach compared with the standard Gaussian HMM.

1. Introduction

The Generalized Gaussian (GG) distribution represents an extension of the standard Gaussian distribution which comprises three parameters: mean, variance (as in Gaussians) and the so-called shape parameter. The latter is a measure of the peakedness of the pdf, and allows the GG to approximate a large class of statistical distributions, including the Gaussian, the Laplacian, and the Uniform distributions. The Generalized Gaussian has been widely used in image processing applications, given that it provides a good approximation for different classes of image-derived features [8, 6, 10, 7, 1, 3, 13, 5]. However, all these applications are focusesd on modeling points (fixed length vectors): a different but crucial scenario, where the employment of the GG has received little attention, is the modeling of sequential data, namely variable length vectors. This paper explores this last direction, merging together the description capability of GG with the effectiveness in dealing with variable length sequences of the Hidden Markov Model (HMM) tool [11]. In particular we describe a variant of the standard HMM, which we call Generalized Gaussian HMM (GG-HMM), where the emission probability function of each state is represented by a GG distribution. Based on the well known Expectation Maximization (E-M) algorithm, we will define a proper training algorithm for GG-HMMs. The usefulness of the proposed approach will be assessed with different synthetic and real world examples: EEG signal classification and face recognition.

2. Fundamentals

This section presents the theoretical background concerning GG and HMMs, mainly to set up notation.

The D-dimensional Generalized Gaussian (GG) distribution is defined as [3]:

$$P_{\boldsymbol{\mu},\beta,\boldsymbol{\Sigma}}(\boldsymbol{x}) = \frac{[\det(\boldsymbol{\Sigma})]^{-1/2}}{[Z(\beta)\,A(\beta)]^{D}} \exp\left(-\left\|\frac{\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{x}-\boldsymbol{\mu})}{A(\beta)}\right\|_{\beta}\right) \quad (1)$$

where β is the so-called *shape* parameter, $\boldsymbol{\mu}$ represents the mean of the distribution, and $\boldsymbol{\Sigma}$ is a symmetric positive definite matrix. $Z(\beta) = \frac{2}{\beta}\Gamma\left(\frac{1}{\beta}\right)$ and $A(\beta) = \sqrt{\frac{\Gamma(1/\beta)}{\Gamma(3/\beta)}}$, with $\Gamma(.)$ the Gamma function. Moreover $\|\boldsymbol{x}\|_{\beta} = \sum_{i=1}^{D} |x_i|^{\beta}$ stands for the l_β norm of vector $\boldsymbol{x}$. The Laplacian, Gaussian and Uniform distributions are special cases of the GG, with $\beta = 1$, $\beta = 2$ and $\beta \to \infty$ respectively.

A Hidden Markov Model (HMM) [11] is composed by the following entities: a set $S = \{S_1, S_2, \cdots, S_N\}$ of (hidden) states; a transition matrix $\mathbf{A} = \{a_{ij}\}$, where $a_{ij} \geq 0$ represents the probability of going from state S_i to state S_j; an emission matrix $\mathbf{B} = \{b(o|S_j)\}$, indicating the probability of emission of symbol o from state S_j; an initial state probability distribution $\boldsymbol{\pi} = \{\pi_i\}$, representing the probability of the first state $\pi_i = P[Q_1 = S_i]$. The standard training phase is carried out by training one model for each class; in the classifica-

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

tion step, then, the unknown sequence is assigned to the class whose model shows the highest likelihood.

3. Generalized Gaussian HMM

The proposed GG-HMM is defined as a HMM where each state dependent emission probability function is represented by a Generalized Gaussian. Note that although modeling each emission function as a mixture of GGs is also possible (this is typically done for standard Gaussians), we can restrict the formulation to HMMs with just one GG per state. Actually it has been shown [2] that given one HMM using a Mixture of Gaussians in each state, there exists an equivalent (in a likelihood sense) HMM with more states but just one Gaussian per state, and this proof could be easily extended to any kind of mixture. Furthermore, using one GG per state eliminates the problem of choosing the number of components in each Mixture, which is still an open problem, incorporating it in the already present problem of choosing the number of states of the HMM [2].

The proposed training algorithm is based on the well known Expectation Maximization (E-M) algorithm. In particular, given a set of sequences $\{\mathbf{O}_1, \mathbf{O}_2, ..., \mathbf{O}_N\}$, $\mathbf{O}_n = \mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_{T_n}$ and $\mathbf{o}_t \in \Re^D$, the goal is to estimate the best model $\boldsymbol{\lambda}$. In order to simplify the notation, we will provide the re-estimation formulas for $N = 1$ (just one sequence); the generalization to $N > 1$ is straightforward.

Starting from an initial model $\boldsymbol{\lambda}^{(0)}$, the E-M algorithm iteratively repeats two steps: in the first (E-step), the so-called Q function (the expected value of the complete log-likelihood given the current parameter estimates) is evaluated; afterwards, in the M-step, such expectation is maximized in order to find the new values of the parameters. For the HMM, the Q function can be splitted in three independent terms, one containing $\boldsymbol{\pi}$, another related to $\mathbf{A}$ and the third one containing $\mathbf{B}$ [4]. Now, the maximization can be carried out by optimizing each term individually. The re-estimation formulas for $\mathbf{A}$ and $\boldsymbol{\pi}$ do not change with respect to standard HMMs [11], and therefore we will just provide the re-estimation of $\mathbf{B}$. In particular, at each iteration ℓ, the following operations are performed:

E-step: for the calculation of $\mathbf{B}$ in this step we need to evaluate the variable $\gamma_t(i)$, which is defined as the probability of being in state S_i at time t, given the sequence $\mathbf{O}$ and the model estimated in the previous iteration $\boldsymbol{\lambda}^{(\ell-1)}$.

M-step: in this step the new parameters should be estimated by maximizing the following function f:

$$f = \sum_{i=1}^{K} \sum_{t=1}^{T} \gamma_t(i) \log(b_i(\mathbf{o}_t)) \tag{2}$$

In our case $b_i(\mathbf{o}_t)$ is the Generalized Gaussian defined in Equation (1), and the parameters to be estimated are $\beta_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$ for each state S_i. The estimation of the three parameters are obtained by setting to zero the partial derivatives. For the shape parameter β_i we have

$$\begin{aligned}\frac{\partial f}{\partial \beta_i} = 0 = \sum_{t=1}^{T} \gamma_t(i) \left[\sum_{j=1}^{D} \left(\left| \frac{y_{tj}}{A(\beta_i)} \right|^{\beta_i} \log \left| \frac{y_{tj}}{A(\beta_i)} \right| \right) \right. \\ \left. + \frac{1}{2\beta_i} \sum_{j=1}^{D} \left(\left| \frac{y_{tj}}{A(\beta_i)} \right|^{\beta_i} \left(\Psi\left(\frac{1}{\beta_i}\right) - 3\Psi\left(\frac{3}{\beta_i}\right) \right) \right) \right] \\ - \sum_{t=1}^{T} \left[\frac{D\gamma_t(i)}{\beta_i} + \frac{3\gamma_t(i)}{2\beta_i^2} \left(\Psi\left(\frac{1}{\beta_i}\right) - \Psi\left(\frac{3}{\beta_i}\right) \right) \right]\end{aligned} \tag{3}$$

where y_{tj} is the j-th component of the vector $\mathbf{y}_t$, defined by $\mathbf{y}_t = \boldsymbol{\Sigma}_i^{(-1/2)}(\mathbf{o}_t - \boldsymbol{\mu}_i)$. $\Psi(\cdot)$ is the Digamma function, i.e. $\Psi(x) = \Gamma'(x)/\Gamma(x)$. The new value of $\beta_i^{(\ell)}$ is obtained by solving the non-linear equation (3), which is done by means of numerical routines[1].

$\boldsymbol{\mu}_i$ – The mean $\boldsymbol{\mu}_i$ is a D-dimensional vector, the expression for the h-th component $\boldsymbol{\mu}_{ih}$ is:

$$\begin{aligned}\frac{\partial f}{\partial \boldsymbol{\mu}_{ih}} = 0 = \sum_{t=1}^{T} \gamma_t(i) \left(\sum_{j=1}^{D} \eta(\mathbf{o}_{tj}, \boldsymbol{\mu}_{ih}) \times \right. \\ \left. \times \left| \frac{\left(\boldsymbol{\Sigma}_i^{(-1/2)}\right)_{jh} (\mathbf{o}_{th} - \boldsymbol{\mu}_{ih})}{A(\beta_i)} \right|^{\beta_i - 1} \right)\end{aligned} \tag{4}$$

where $(\boldsymbol{M})_{jh}$ is the (jh) entry of the matrix $\boldsymbol{M}$. $\eta(a, b) = 1$ if $a \geq b$ and -1 otherwise. Also in this case, the new parameter $\boldsymbol{\mu}_i^{(\ell)}$ is obtained numerically.

$\boldsymbol{\Sigma}_i$ – For the estimation of $\boldsymbol{\Sigma}_i$ the following analytical expression can be obtained:

$$\boldsymbol{\Sigma}_i^{(\ell)} = \sum_{t=1}^{T} \gamma_t(i)(\mathbf{o}_t - \boldsymbol{\mu}_i)(\mathbf{o}_t - \boldsymbol{\mu}_i)' \tag{5}$$

It is well known that the E-M algorithm is very sensitive to the problem of initialization. In all our experiments, we initialized randomly $\mathbf{A}$ and $\boldsymbol{\pi}$, whereas $\mathbf{B}$ was initialized by clustering. In particular, the set of points derived from unrolling the training sequences was clustered in K clusters (with K the number of states). Afterwards, the data belonging to each cluster was modeled using a Gaussian, whose estimated parameters were used to initialize each GG mean ($\boldsymbol{\mu}_i$) and covariance ($\boldsymbol{\Sigma}_i$), with each β_i consequently set to 2.

[1] In particular, we used the `fzero` function of MATLAB.

Class	A	π	B
1	[0.5 0.5; 0.5 0.5]	[0.5 0.5]	$[\mathcal{N}_{(1,1)}, \mathcal{N}_{(3,1)}]$
2	[0.5 0.5; 0.5 0.5]	[0.5 0.5]	$[\mathcal{N}_{(1.2,1)}, \mathcal{N}_{(3.2,1)}]$

(a)

Class	A	π	B
1	[0.5 0.5; 0.5 0.5]	[0.5 0.5]	$[\mathcal{N}_{(1,1)}, \mathcal{U}_{[2,4]})]$
2	[0.5 0.5; 0.5 0.5]	[0.5 0.5]	$[\mathcal{N}_{(1.1,1)}, \mathcal{U}_{[2.1,4.1]}]$

(b)

Figure 1. Generating HMMs for the synthetic problems. $\mathcal{N}_{(\mu,\sigma)}$ is a Gaussian distribution with mean μ and variance σ; $\mathcal{U}_{[a,b]}$ is an uniform distribution in the interval $[a, b]$.

4. Experimental evaluation

Classification problems with both synthetic and real data have been devised in order to compare the performance of GG emission HMMs (GG-HMMs) with standard Gaussian emission HMMs (G-HMMs).

4.1. Experiments on Synthetic Data

Here two two-classes problems were tested, using synthetic data drawn from known HMMs. In the first experiment the sequences (of length 100) are generated from the two 2-states HMMs displayed in Fig. 1(a) (the emission function of each state is Gaussian). The number of sequences used to train each class models has been varied, growing from 5 to 50 per class. 100 testing sequences per class have been generated. All the experiments were repeated 50 times, averaging the obtained performances. Fig. 2(left) shows the averaged accuracies (as well as the corresponding standard errors of the mean) for both G-HMM and GG-HMM, when varying the number of training sequences (learning curves). From this figure, we can notice that GG-HMMs perform as well as standard G-HMMs whenever enough training sequences are available (20 sequences seem to be enough). This behavior seems reasonable since in GG-HMMs there exists an additional parameter, and hence more data are needed to obtain accurate estimates of the model.

In the second experiment (same experimental setup), sequences are generated from HMMs shown in Fig. 1(b) (one emission is Uniform). Fig. 2(right) shows the averaged accuracies (with standard errors of the mean) for both G-HMM and GG-HMM, with GG-HMM outperforming its Gaussian counterpart: in this case the Gaussian distribution is not adequate to model the underlying data.

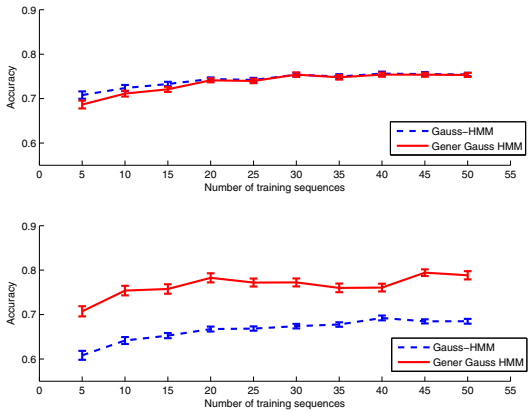

Figure 2. Synthetic experiment: dotted lines represent standard Gaussian HMM, while continuous lines the proposed GG-HMM.

4.2. Experiments on Real Data

Two different real world classification problems are tested: **a)** EEG signal classification, and **b)** face recognition. For all experiments the best number of states of the HMMs was selected using the well known Bayesian Information Criterion (BIC) [12], among values in the range [2-10]. Initialization of GG-HMM training was performed by clustering (as explained in Section 3). The same scheme was applied for G-HMM, in order to focus on the comparison between the two approaches independently of the initial parameters. When dealing with multidimensional data, full covariance matrices were employed. Finally, training algorithms were stopped at the likelihood convergence. We would like to remark that in all experiments a couple of iterations was enough for the proposed GG-HMM training algorithm to converge.

Experiment 1: EEG signal classification. The first experiment aims at distinguishing between alcoholic and non-alcoholic (control) subjects based on their recorded EEG signals[2]. Training (600 sequences) and test (other 600 sequences) sets are already pre-defined. Each sequence contains 256 symbols, each of 64 dimensions (the 64 electrodes of the EEG). In order to compare univariate GG-HMM against G-HMM, we conducted 64 experiments using one channel at a time, finally averaging the 64 obtained accuracies. The results are shown in the first row of Table 1(a) (together with the standard error of the mean). It can be noticed that GG-

[2]See `http://kdd.ics.uci.edu/databases/eeg/eeg.html`

HMM performs slightly better. The second row of Table 1(a) shows the comparison between multivariate GG-HMMs and multivariate G-HMMs. For this experiment the whole set of 64 channels was used, and as can be seen, the use of GG provokes a remarkable improvement.

EEG classification		
Problem	G-HMM	GG-HMM
(1 channel)	59.60% (0.47%)	61.07% (0.46%)
(64 channels)	93.67%	97.50%

(a)

Face recognition		
Problem	G-HMM	GG-HMM
(8 coefficients)	98.54%	98.05%
(2 coefficients)	92.45%	95.62%
(1 coefficient)	75.49%	83.99%

(b)

Table 1. Accuracies of G-HMM and GG-HMM in real world experiments.

Experiment 2: Face Recognition. The second application regards face recognition, with videos acquired from 24 subjects in two different sessions (380 images in average per video). In a given sequence and for every frame, 50 facial landmarks were automatically detected; 8 Gabor filters (2 scales and 4 orientations) were used to extract feature vectors from each landmark. For each frame, the sequence to be used by the HMM was obtained by scanning the landmarks in a predefined order, similarly to what is done in [9]. Due to the large dimensionality of the data set, accuracies were computed using holdout cross validation: half of the set (randomly chosen) was used for training the HMMs, one for each subject, whereas the remaining part was used for testing. Classification accuracies using the 8 coefficient feature vectors are displayed on the first row of Table 1(b), showing that both methods perform almost perfectly. In order to increase the difficulty of the classification problem, we also considered just 1 or 2 coefficients from each feature vector. Results are shown in Table 1(b), clearly demonstrating that GG-HMM outperforms the standard G-HMM (specially with the most difficult task, i.e. 1 coefficient).

It has been shown in [7] that each Gabor coefficient can be accurately modeled by an univariate Generalized Gaussian, and that the values of the shape parameter do vary depending on the specific coefficient. Since our multivariate GG employs the same β independently of the dimension (i.e. the coefficient), the multidimensional modeling may not be as accurate as desired. This may justify why GG-HMMs do not perform better than G-HMMs in the 8 coefficient test, and also explain why G-HMM's performance gets closer to that of GG-HMM when using 2 coefficients. Performing a statistical test, prior to employ the GG-HMM, to measure the matching between data and GGD would prevent such situations. Moreover, currently, we are investigating how to extend the formulation in order to consider the case in which one different β is used per dimension.

5. Conclusions

This paper has explored the use of multivariate Generalized Gaussians (GG) as emission probability functions for HMMs, leading to the so-called GG-HMM. Based on the E-M algorithm, we derived the formulation of the training process, and assessed the effectiveness of the devised approach in synthetic and real problems (both one-dimensional and multivariate), with promising results.

References

[1] Y. Bazi, L. Bruzzone, and F. Melgani. Image thresholding based on em algorithm and the generalized gaussian distribution. *Pat. Rec.*, 40:619–634, 2007.

[2] M. Bicego, V. Murino, and M. Figueiredo. A sequential pruning strategy for the selection of the number of states in Hidden Markov Models. *PRL*, 24(9–10):1395–1407, 2003.

[3] L. Boubchir and M. Fadili. Multivariate statistical modeling of images with the curvelet transform. In *Proc. IEEE Conf. on Signal Processing and Its Applications*, pages 747–750, 2005.

[4] O. Cappe, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models*. Springer, 2005.

[5] D. Cho and T. Bui. Multivariate statistical modeling for image denoising using wavelet transforms. *Signal Processing: Image Communication*, 20(1):77–89, 2005.

[6] M. Do and M. Vetterli. Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance. *IEEE TIP*, 11:146–158, 2002.

[7] D. Gonzalez-Jimenez, F. Perez-Gonzalez, P. Comesaña, L. Perez-Freire, and J.L. Alba-Castro. Modeling gabor coefficients via generalized gaussian distributions for face recognition. In *Proc. ICIP*, 2007.

[8] J. Hernandez, M. Amado, and F. Perez-Gonzalez. Dct domain watermarking techniques for still images: Detector performance analysis and a new structure. *IEEE TIP*, 9(1):55–68, 2000.

[9] V. Kohir and U. Desai. Face recognition using DCT-HMM approach. In *Proc. Workshop on Advances in Facial Image Analysis and Recognition Technology*, 1998.

[10] P. Moulin and J. Liu. Analysis of multiresolution image denoising schemes using generalized gaussian and complexity prior. *IEEE Trans. on Information Theory*, 45:909–919, 1999.

[11] L. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. of IEEE*, 77(2):257–286, 1989.

[12] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[13] E. Simoncelli and E. Adelson. Noise removal via bayesian wavelet coring. In *Proc. ICIP*, 1996.