

Document downloaded from:

<http://hdl.handle.net/10251/121393>

This paper must be cited as:

Belda, J.; Vergara Domínguez, L.; Salazar Afanador, A.; Safont Armero, G. (2018).  
Estimating the Laplacian matrix of Gaussian mixtures for signal processing on graphs.  
Signal Processing. 148:241-249. <https://doi.org/10.1016/j.sigpro.2018.02.017>



The final publication is available at

<http://doi.org/10.1016/j.sigpro.2018.02.017>

Copyright Elsevier

Additional Information

# Estimating the Laplacian matrix of Gaussian mixtures for signal processing on graphs

## 1. Introduction<sup>1</sup>

### 1.1 Background

Signal processing on graphs [1]-[3] is an emerging technique which combines concepts emanating from two largely consolidated areas: discrete signal processing and graph theory. From the perspective of discrete signal processing it leads to a more general definition of a discrete signal by assigning every sample value to the vertex of a graph. Conventional signals will simply be particular cases where sample values are assigned to a uniform (time or space) grid. From the graph theory perspective, new graph transformations can be defined that extend classical discrete signal processing concepts like filtering, prediction and spectral analysis [3].

Let us represent a graph in the form  $G\{V, E, \mathbf{W}\}$ , where  $V$  is a set of  $N$  vertices or nodes,  $E$  is the set of edges connecting the nodes and  $\mathbf{W}$  is the so called adjacency matrix. The generic element  $w_{mn}$  is the weight (assumed real and nonnegative) corresponding to the edge connecting node  $m$  to node  $n$ . A weight equal to zero means that there is no connection from node  $m$  to node  $n$ . We will consider undirected graphs, so  $w_{nm} = w_{mn}$ . Clearly, the graph is characterized by the adjacency matrix although the Laplacian matrix is preferred [4], defined as  $\mathbf{L} = \mathbf{D} - \mathbf{W}$ , where  $\mathbf{D}$  is a diagonal matrix having the generic element  $d_{mm} = \sum_{n=1}^N w_{mn}$ . Preference about  $\mathbf{L}$  is justified by its attractive spectral properties [5].

Therefore, a key aspect in signal processing on graphs is a proper computation of the Laplacian matrix. In specific contexts, it is possible to establish the weights by considering natural relationships between nodes, e.g., spatial proximity, but in a general statistical framework, it is most convenient to devise methods for estimating the Laplacian matrix from training data sets. Thus recent work in signal processing on graphs [6]-[9] has been driven to estimate the so called precision matrix and to use it as the graph Laplacian matrix. These works are related to statistical techniques for the general estimation of covariance or precision matrices [10]-[15]. The rationale under this idea is that the off-diagonal elements of the precision matrix (properly normalized) coincide with the partial correlations between every two elements of the random observation vector  $\mathbf{x} = [x_1 \dots x_N]^T$ . The partial correlation between two random variables  $x_n$  and  $x_m$   $n \neq m$  is defined as the correlation between the residuals obtained by subtracting from  $x_n$  and  $x_m$  their linear least mean square error (LLMSE) estimates obtained from the

---

<sup>1</sup> Throughout the paper we will use the following notation: Matrix  $\mathbf{M}$ ; vector  $\mathbf{v}$ ; scalar  $s$ ; probability density function  $p(\cdot)$ ; probability mass  $P$ ; expectation  $E[\cdot]$ ; any other function  $f(\cdot)$  or  $F(\cdot)$ ;  $k$ -th element of vector  $\mathbf{v}(k)$ ;  $ij$  element of matrix  $m_{ij}$ ;  $k$ -th column of matrix  $\mathbf{m}_k$ ; matrix determinant  $|\mathbf{M}|$ .

rest of variables. Thus, the partial correlation is a measure of the actual degree of linear linking between nodes  $n$  and  $m$ , not affected by the presence of the rest of variables. However, the linear estimator which is implicit in the definition of the partial correlation, is optimum if the underlying multivariate probability density function (pdf)  $p(\mathbf{x})$  is Gaussian. But, if this is not a reasonable assumption, the influence of the rest of variables in  $x_n$  and  $x_m$  is not totally removed and the partial correlation does not properly capture the actual dependence between node  $n$  and node  $m$ . Up to our knowledge, there is a lack of works dealing with the non-Gaussian case in the context of the Laplacian matrix estimation. So this will be the main focus of this contribution.

## 1.2 Statement of the problem

In this paper, we propose a more general definition of the partial correlation to account for the non-Gaussian case. We propose to replace the LLMSE of  $x_n$  and  $x_m$  by their respective conditional means  $E[x_n/\mathbf{x}_{-nm}]$  and  $E[x_m/\mathbf{x}_{-nm}]$ , i.e., their respective (non-linear) LMSE estimates. As these later depend on the specific  $p(\mathbf{x})$ , we will consider a general non-Gaussian model for  $p(\mathbf{x})$  which can fit to a variety of scenarios: a multivariate Gaussian mixture model (GMM)

$$p(\mathbf{x}) = \sum_{k=1}^K P_k \frac{1}{\sqrt{(2\pi)^N |\mathbf{C}_k|}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{b}_k)^T \mathbf{C}_k^{-1}(\mathbf{x}-\mathbf{b}_k)} \quad (1)$$

Where  $P_k$ ,  $\mathbf{C}_k$  and  $\mathbf{b}_k$  are, respectively, the a priori probability, the covariance matrix and the mean value of the  $k$ -th Gaussian mixture component. GMM is the most usual non-Gaussian parametric model due to its good balance between versatility and simplicity. It has been considered in a myriad of statistical and signal processing scenarios (see for example [16]-[23], to mention a few of some recent applications). Our focus will be on showing that better Laplacian matrix estimates can be obtained by assuming a more sophisticated model (GMM) than the multivariate Gaussian model which is implicit in the use of any estimated precision matrix.

In the next section we formally define a generalized partial correlation coefficient and compare it with other related definitions in the statistical literature. Then in section 3 the generalized partial correlation coefficient is computed assuming a GMM for  $p(\mathbf{x})$ . Firstly, the general method is considered, then an approximate method is proposed to alleviate the computational burden. Finally, in section 4 we include some experiments with simulated and real data to show that the new proposed methods may capture the actual partial correlation coefficients, i.e., may estimate the Laplacian matrix, in a better way than the usual methods based on the precision matrix.

## 2. Generalized partial correlation coefficient GPCC

Let us define  $\mathbf{x}_{-nm}$  as the vector obtained from vector  $\mathbf{x}$  after dropping the elements  $x_n$  and  $x_m$ ,  $L(x_n/\mathbf{x}_{-nm})$  and  $L(x_m/\mathbf{x}_{-nm})$  the corresponding LLMSE estimates,  $r_n = x_n - L(x_n/\mathbf{x}_{-nm})$  and  $r_m = x_m - L(x_m/\mathbf{x}_{-nm})$  the corresponding residuals. The partial correlation coefficient (PCC) is given by

$$\rho_{nm} = \frac{E[(r_n - E[r_n])(r_m - E[r_m])]}{\sqrt{E[(r_n - E[r_n])^2]} \sqrt{E[(r_m - E[r_m])^2]}} \quad (2)$$

One interesting property of  $\rho_{nm}$  is that it can be estimated without explicitly computing the residuals. This is because it can be calculated from the inverse of the covariance matrix, the so called precision matrix, namely

$$\rho_{nm} = -\frac{q_{nm}}{\sqrt{q_{nn}q_{mm}}} \quad , \quad (3)$$

where  $q_{nm}$  is the  $nm$  element of matrix  $\mathbf{Q}$ , being  $\mathbf{Q}^{-1} = \mathbf{C} = E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^T]$ . We propose a generalized partial correlation by substituting in (2) the linear residuals by the residuals obtained after subtracting from  $x_n$  and  $x_m$  their respective conditional means  $E[x_n/\mathbf{x}_{-nm}]$  and  $E[x_m/\mathbf{x}_{-nm}]$ , i.e., their respective (non-linear) LMSE estimates. Let us call  $r_n^G = x_n - E[x_n/\mathbf{x}_{-nm}]$  and  $r_m^G = x_m - E[x_m/\mathbf{x}_{-nm}]$  the generalized residuals, then the generalized partial correlation (GPCC) coefficient is defined as

$$\rho_{nm}^G = \frac{E[(r_n^G - E[r_n^G])(r_m^G - E[r_m^G])]}{\sqrt{E[(r_n^G - E[r_n^G])^2]} \sqrt{E[(r_m^G - E[r_m^G])^2]}} = \frac{E[r_n^G r_m^G]}{\sqrt{E[(r_n^G)^2]} \sqrt{E[(r_m^G)^2]}} \quad , \quad (4)$$

where we have taken into account that the conditional mean is an unbiased estimator, so  $E[r_n^G] = E[r_m^G] = 0$ . GPCC so defined is related to other concepts already proposed to measure nonlinear dependences between two random variables. One is a local dependence function  $H(x_n, x_m)$  [24], which measures the dependence between two random variables conditioned to some (local) values of both. In the context of a random signal on graph, this may inspire a function  $H_{nm}(\mathbf{x}_{-nm})$  to measure the dependence between  $x_n$  and  $x_m$  at the local points  $\mathbf{x}_{-nm}$  in the form

$$H_{nm}(\mathbf{x}_{-nm}) = \frac{E[(x_n - E[x_n/\mathbf{x}_{-nm}])(x_m - E[x_m/\mathbf{x}_{-nm}])]}{\sqrt{E[(x_n - E[x_n/\mathbf{x}_{-nm}])^2]} \sqrt{E[(x_m - E[x_m/\mathbf{x}_{-nm}])^2]}} \quad , \quad (5)$$

where the expectations are computed with respect to the joint pdf  $p(x_n, x_m)$  so that it could be obtained the local measure  $H_{nm}(\mathbf{x}_{-nm})$ . Another related concept is conditional correlation [25],

$$\rho_{nm}(\mathbf{x}_{-nm}) = \frac{E\left[(x_n - E[x_n/\mathbf{x}_{-nm}]/\mathbf{x}_{-nm})(x_m - E[x_m/\mathbf{x}_{-nm}]/\mathbf{x}_{-nm})\right]}{\sqrt{E\left[(x_n - E[x_n/\mathbf{x}_{-nm}])^2/\mathbf{x}_{-nm}\right]}\sqrt{E\left[(x_m - E[x_m/\mathbf{x}_{-nm}])^2/\mathbf{x}_{-nm}\right]}} . \quad (6)$$

The difference with (5) is that the expectations are computed with respect to the conditional joint pdf  $p(x_n, x_m/\mathbf{x}_{-nm})$  instead of  $p(x_n, x_m)$ . However we are interested in a unique number to assign to every edge of the graph. So, some integration of  $H_{nm}(\mathbf{x}_{-nm})$  or  $\rho_{nm}(\mathbf{x}_{-nm})$  with respect to  $\mathbf{x}_{-nm}$  would be necessary. This is implicitly done in  $\rho_{nm}^G$ , because the expectations in (4) are defined with respect to all the involved random variables, i.e., with respect to  $p(x_n, x_m, \mathbf{x}_{-nm}) = p(\mathbf{x})$ .

### 3. Computing the GPCC of a Gaussian Mixture Model

In the Appendix A, we derive the conditional mean  $E[x_n/\mathbf{x}_{-nm}]$  required to calculate the generalized residuals  $r_n^G$ . From (A11), we can express the generalized residual in the form

$$r_n^G = x_n - E[x_n/\mathbf{x}_{-nm}] = x_n - \mathbf{p}_{-nm}^T \hat{\mathbf{x}}_{n_{-nm}} , \quad (7)$$

where  $\mathbf{p}_{-nm}^T = [P(1/\mathbf{x}_{-nm}) \dots P(K/\mathbf{x}_{-nm})]^T$  and the  $k$ -th element of vector  $\hat{\mathbf{x}}_{n_{-nm}}$  is given by  $\hat{\mathbf{x}}_{n_{-nm}}(k) = \mathbf{c}_{kn_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} (\mathbf{x}_{-nm} - \mathbf{b}_{k_{-nm}}) + b_{kn}$   $k=1 \dots K$ . Let us define the vector  $\mathbf{e}_{n_{-nm}} = \mathbf{x}_n - \hat{\mathbf{x}}_{n_{-nm}}$  whose elements are the individual residuals corresponding to every component of the GMM (we have defined  $\mathbf{x}_n = [x_n \dots x_n]^T = x_n \cdot \mathbf{1}$ ). We can write:

$$r_n^G = x_n - \mathbf{p}_{-nm}^T (\mathbf{x}_n - \mathbf{e}_{n_{-nm}}) = \mathbf{p}_{-nm}^T \mathbf{e}_{n_{-nm}} , \quad (8)$$

because  $\mathbf{p}_{-nm}^T \mathbf{x}_n = x_n \sum_{k=1}^K \mathbf{p}_{-nm}(k) = x_n$ . Now let us compute the correlation corresponding to the generalized residuals

$$E[r_n^G r_m^G] = E\left[(\mathbf{p}_{-nm}^T \mathbf{e}_{n_{-nm}})(\mathbf{p}_{-nm}^T \mathbf{e}_{m_{-nm}})\right] = E\left[\mathbf{p}_{-nm}^T \mathbf{e}_{n_{-nm}} \mathbf{e}_{m_{-nm}}^T \mathbf{p}_{-nm}\right] . \quad (9)$$

Then we can express

$$\rho_{nm}^G = \frac{E[r_n^G r_m^G]}{\sqrt{E[(r_n^G)^2]}\sqrt{E[(r_m^G)^2]}} = \frac{E[\mathbf{p}_{-nm}^T \mathbf{e}_{n_{-nm}} \mathbf{e}_{m_{-nm}}^T \mathbf{p}_{-nm}]}{\sqrt{E[\mathbf{p}_{-nm}^T \mathbf{e}_{n_{-nm}} \mathbf{e}_{n_{-nm}}^T \mathbf{p}_{-nm}]}\sqrt{E[\mathbf{p}_{-nm}^T \mathbf{e}_{m_{-nm}} \mathbf{e}_{m_{-nm}}^T \mathbf{p}_{-nm}]}} . \quad (10)$$

Notice that  $\mathbf{p}_{-nm}$  can be calculated from

$$\mathbf{p}_{-nm}(k) = P(k/\mathbf{x}_{-nm}) = \frac{p(\mathbf{x}_{-nm}/k)P_k}{p(\mathbf{x}_{-nm})} = \frac{p(\mathbf{x}_{-nm}/k)P_k}{\sum_{k'=1}^K P(\mathbf{x}_{-nm}/k')P_{k'}} , \quad (11)$$

where  $p(\mathbf{x}_{-nm}/k)$  can be obtained from equation (A7) . So the statistics in (10) are functions of  $x_n, x_m, \mathbf{x}_{-nm}$  and the GMM parameters, then we could make sample means or numerical integration to get estimates of the expectations required to estimate the GPCCs. However the computational burden will be much higher than the one required to estimate the PCCs from estimates of the precision matrix. We propose in the following a practical solution after considering a simplifying assumption.

We can express from (9)

$$E[r_n^G r_m^G] = E\left[\left(\mathbf{p}_{-nm}^T \mathbf{e}_{n_{-nm}}\right)\left(\mathbf{p}_{-nm}^T \mathbf{e}_{m_{-nm}}\right)\right] = E\left[\sum_{k=1}^K \sum_{k'=1}^K P(k/\mathbf{x}_{-nm})P(k'/\mathbf{x}_{-nm})\mathbf{e}_{n_{-nm}}(k)\mathbf{e}_{m_{-nm}}(k')\right] . \quad (12)$$

Notice that  $P(k/\mathbf{x}_{-nm}), P(k'/\mathbf{x}_{-nm})$  are functions of just  $\mathbf{x}_{-nm}$  and that  $\mathbf{e}_{n_{-nm}}(k), \mathbf{e}_{m_{-nm}}(k')$  are prediction errors corresponding to linear predictions from the predicting variables  $\mathbf{x}_{-nm}$ . Hence  $P(k/\mathbf{x}_{-nm}), P(k'/\mathbf{x}_{-nm})$  are uncorrelated with  $\mathbf{e}_{n_{-nm}}(k), \mathbf{e}_{m_{-nm}}(k')$  and we can write:

$$E[r_n^G r_m^G] = \sum_{k=1}^K \sum_{k'=1}^K E\left[P(k/\mathbf{x}_{-nm})P(k'/\mathbf{x}_{-nm})\right] E\left[\mathbf{e}_{n_{-nm}}(k)\mathbf{e}_{m_{-nm}}(k')\right] . \quad (13)$$

But

$$\begin{aligned} E\left[P(k/\mathbf{x}_{-nm})P(k'/\mathbf{x}_{-nm})\right] &= E\left[\frac{p(\mathbf{x}_{-nm}/k)P_k}{p(\mathbf{x}_{-nm})} \frac{p(\mathbf{x}_{-nm}/k')P_{k'}}{p(\mathbf{x}_{-nm})}\right] = \\ &= \int_{-\infty}^{\infty} \frac{p(\mathbf{x}_{-nm}/k)P_k}{p(\mathbf{x}_{-nm})} \frac{p(\mathbf{x}_{-nm}/k')P_{k'}}{p(\mathbf{x}_{-nm})} p(\mathbf{x}_{-nm}) d\mathbf{x}_{-nm} . \\ &= P_k P_{k'} \int_{-\infty}^{\infty} \frac{p(\mathbf{x}_{-nm}/k)p(\mathbf{x}_{-nm}/k')}{p(\mathbf{x}_{-nm})} d\mathbf{x}_{-nm} \end{aligned} \quad (14)$$

The integral in (14) can be simplified if we assume that the GMM components are “well separated” so that  $p(\mathbf{x}_{-nm}/k)p(\mathbf{x}_{-nm}/k') \square 0 \quad k \neq k'$ . This is the most interesting case from the perspective of improving the performance of the PCC as a measure of the pairwise connectivity, as the more separated the components are, the more we are away from Gaussianity. Separation could be defined in different ways, using a variety of distance measures between two normal pdfs (see for example [26] and references there in). However, in our context, “well separation” means that  $p(\mathbf{x}_{-nm}/k)p(\mathbf{x}_{-nm}/k') \square 0$  for all  $\mathbf{x}_{-nm}$  and  $k \neq k'$ , thus in the Appendix B we propose a practical procedure for measuring mixture component separation in terms of the GMM parameters by defining an “ad hoc” index of separation  $0 \leq I_s \leq 1$  where 1 means perfect separation . On the other hand, for  $k = k'$ , the “well separation”

assumption leads to  $\int_{-\infty}^{\infty} \frac{p(\mathbf{x}_{-nm}/k)p(\mathbf{x}_{-nm}/k)}{p(\mathbf{x}_{-nm})} d\mathbf{x}_{-nm} \square \int_{-\infty}^{\infty} \frac{p(\mathbf{x}_{-nm}/k)p(\mathbf{x}_{-nm}/k)}{P_k p(\mathbf{x}_{-nm}/k)} d\mathbf{x}_{-nm} = \frac{1}{P_k}$ , so we can approximate

$$E\left[P(k/\mathbf{x}_{-nm})P(k'/\mathbf{x}_{-nm})\right] = \begin{cases} P_k & k = k' \\ 0 & k \neq k' \end{cases} . \quad (15)$$

Then considering (15) in (13)

$$E\left[r_n^G r_m^G\right] = \sum_{k=1}^K P_k E\left[\mathbf{e}_{n_{-nm}}(k)\mathbf{e}_{m_{-nm}}(k)\right] = \tilde{E}\left[r_n^G r_m^G\right] , \quad (16)$$

which means that the total residual correlation is approximated by the weighted sum of the individual residual correlation due to every GMM component, neglecting the cross-terms between different components. Let us now compute  $E\left[\mathbf{e}_{n_{-nm}}(k)\mathbf{e}_{m_{-nm}}(k)\right]$ , notice that these residuals correspond to the assumption that the  $k$ -th component of the GMM is in force, hence

$$\begin{aligned} E\left[\mathbf{e}_{n_{-nm}}(k)\mathbf{e}_{m_{-nm}}(k)\right] &= E\left[(x_n - \hat{\mathbf{x}}_{n_{-nm}}(k))(x_m - \hat{\mathbf{x}}_{m_{-nm}}(k))\right] = \\ &= R_k(n, m) - E\left[x_n \hat{\mathbf{x}}_{m_{-nm}}(k)\right] - E\left[x_m \hat{\mathbf{x}}_{n_{-nm}}(k)\right] + E\left[\hat{\mathbf{x}}_{n_{-nm}}(k)\hat{\mathbf{x}}_{m_{-nm}}(k)\right] . \end{aligned} \quad (17)$$

Where  $R_k(n, m) = C_k(n, m) + b_{kn}b_{km}$  are the elements of the correlation matrix  $\mathbf{R}_k = \mathbf{C}_k + \mathbf{b}_k\mathbf{b}_k^T$  corresponding to the component  $k$  of the GMM.

But

$$\begin{aligned} E\left[x_n \hat{\mathbf{x}}_{m_{-nm}}(k)\right] &= E\left[x_n \left(\mathbf{c}_{km_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} (\mathbf{x}_{-nm} - \mathbf{b}_{k_{-nm}}) + b_{km}\right)\right] = \\ &= \mathbf{c}_{km_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} (\mathbf{r}_{kn_{-nm}} - \mathbf{b}_{k_{-nm}} b_{kn}) + b_{kn} b_{km} = \mathbf{c}_{km_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} \mathbf{c}_{kn_{-nm}} + b_{kn} b_{km} , \end{aligned} \quad (18)$$

$$E\left[x_m \hat{\mathbf{x}}_{n_{-nm}}(k)\right] = \mathbf{c}_{kn_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} \mathbf{c}_{km_{-nm}} + b_{kn} b_{km}$$

where vector  $\mathbf{r}_{kn_{-nm}}$  is formed by the elements of the  $n$ -th column of the matrix  $\mathbf{R}_k$  dropping rows  $n$  and  $m$ . And

$$\begin{aligned} E\left[\hat{\mathbf{x}}_{n_{-nm}}(k)\hat{\mathbf{x}}_{m_{-nm}}(k)\right] &= E\left[\left(\mathbf{c}_{kn_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} (\mathbf{x}_{-nm} - \mathbf{b}_{k_{-nm}}) + b_{kn}\right)\left(\mathbf{c}_{km_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} (\mathbf{x}_{-nm} - \mathbf{b}_{k_{-nm}}) + b_{km}\right)\right] = \\ &= \mathbf{c}_{kn_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} E\left[(\mathbf{x}_{-nm} - \mathbf{b}_{k_{-nm}})(\mathbf{x}_{-nm} - \mathbf{b}_{k_{-nm}})^T\right] \mathbf{C}_{k_{-nm}}^{-1} \mathbf{c}_{km_{-nm}} \\ &+ \mathbf{c}_{kn_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} (E[\mathbf{x}_{-nm}] - \mathbf{b}_{k_{-nm}}) b_{km} + \mathbf{c}_{km_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} (E[\mathbf{x}_{-nm}] - \mathbf{b}_{k_{-nm}}) b_{kn} + b_{kn} b_{km} = \\ &= \mathbf{c}_{kn_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} \mathbf{c}_{km_{-nm}} + b_{kn} b_{km} \end{aligned} . \quad (19)$$

So finally

$$\begin{aligned} E\left[\mathbf{e}_{n_{-nm}}(k)\mathbf{e}_{m_{-nm}}(k)\right] &= C_k(n, m) + b_{kn}b_{km} - \mathbf{c}_{km_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} \mathbf{c}_{kn_{-nm}} - b_{kn}b_{km} - \mathbf{c}_{kn_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} \mathbf{c}_{km_{-nm}} - b_{kn}b_{km} \\ &+ \mathbf{c}_{kn_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} \mathbf{c}_{km_{-nm}} + b_{kn}b_{km} = C_k(n, m) - \mathbf{c}_{kn_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} \mathbf{c}_{km_{-nm}} . \end{aligned} \quad (20)$$

In conclusion, given the GMM parameters, we can compute  $\tilde{E}\left[r_n^G r_m^G\right]$  from equations (16)-(20). Then an approximate GPCC (aGPCC) is defined as

$$\rho_{nm}^{aG} = \frac{\tilde{E} \begin{bmatrix} r_n^G & r_m^G \\ r_n^G & r_m^G \end{bmatrix}}{\sqrt{\tilde{E} \left[ (r_n^G)^2 \right]} \sqrt{\tilde{E} \left[ (r_m^G)^2 \right]}} = \frac{\sum_{k=1}^K P_k \left( C_k(n, m) - \mathbf{c}_{kn-nm}^T \mathbf{C}_{k-nm}^{-1} \mathbf{c}_{km-nm} \right)}{\sqrt{\sum_{k=1}^K P_k \left( C_k(n, n) - \mathbf{c}_{kn-nm}^T \mathbf{C}_{k-nm}^{-1} \mathbf{c}_{kn-nm} \right)} \sqrt{\sum_{k=1}^K P_k \left( C_k(m, m) - \mathbf{c}_{km-nm}^T \mathbf{C}_{k-nm}^{-1} \mathbf{c}_{km-nm} \right)}} \quad . (21)$$

Figure 1 shows the Directed Acyclic Graph (DAG) of the Bayesian Network (BN) corresponding to a GMM. These kind of diagrams help to understand the role and dependences of the different variables and parameters, and to define learning algorithms (see for example [27]). Thus, in figure 1a we represent the compact model which factorizes the jointed pdf of the observed ( $\mathbf{x}$ ) and the latent ( $k$ ) variables, as the product of the prior by the conditional pdf of the observed variables  $p(\mathbf{x}, k) = p(\mathbf{x}/k)P_k$ . Also in figure 1a (rectangular boxes), we include the parameters which are to be estimated for a complete characterization of the BN. In figure 1b we have split the compact graph to make explicit the dependences involved in the computations of the partial correlations. Then variables  $x_n, x_m$  are separated so that the jointed pdf is now factorized in the form  $p(\mathbf{x}, k) = p(x_1, x_2/\mathbf{x}_{-nm}, k)p(\mathbf{x}_{-nm}/k)P_k$ . The parameters indicated in the rectangular boxes of figure 1b are obtained from the ones of figure 1a. Moreover, we have included at the bottom of figure 1, the values of the DAG respectively involved in the calculation of GPCC and aGPCC.

With the aim of facilitating the understanding and the implementation, we include in the following a pseudocode description of the algorithms to estimate the GPCCs and aGPCCs. It can be appreciated the great simplification of aGPCC estimation, as no sample estimates are required (the loop  $l=1 \dots L$  is not necessary).

<p><b>Algorithm 1: Computation of the GPCC</b></p> <p>1: <b>Input:</b> Training set <math>\{\mathbf{x}_{(l)}\} \quad l=1 \dots L</math></p> <p>2: <b>Estimate GMM parameters</b> <math>\mathbf{b}_k, \mathbf{C}_k, P_k, k=1 \dots K</math> from the training set (EM algorithm)</p> <p>3: <b>for</b> <math>n=1 \dots N-1</math></p> <p>4: <b>for</b> <math>m=n+1 \dots N</math></p> <p>5: <b>for</b> <math>l=1 \dots L</math></p> <p>6: <b>Compute</b> <math>\mathbf{p}_{(l)-nm}^T = \left[ P(1/\mathbf{x}_{(l)-nm}) \dots P(K/\mathbf{x}_{(l)-nm}) \right]^T</math>, eq. (11) and (A7)</p> <p>7: <b>Compute</b> <math>\hat{\mathbf{x}}_{(l)n-nm}</math> and <math>\hat{\mathbf{x}}_{(l)m-nm}</math>, eq. (A11)</p> <p>8: <b>Compute</b> <math>r_{(l)n}^G</math> and <math>r_{(l)m}^G</math>, eq. (7)</p> <p>9: <b>end for</b></p> <p>10: <b>Estimate</b> <math>\rho_{nm}^G</math> (sample estimates in eq. (10))</p> <p>11: <b>Compute</b> <math>\rho_{nm}^G = \rho_{nm}^G</math></p> <p>11: <b>end for</b></p> <p>12: <b>end for</b></p> <p>13: <b>Output:</b> <math>\rho_{nm}^G \quad n=1 \dots N \quad m=1 \dots M</math></p>	<p><b>Algorithm 2: Computation of the aGPCC</b></p> <p>1: <b>Input:</b> Training set <math>\{\mathbf{x}_{(l)}\} \quad l=1 \dots L</math></p> <p>2: <b>Estimate GMM parameters</b> <math>\mathbf{b}_k, \mathbf{C}_k, P_k, k=1 \dots K</math> from the training set (EM algorithm)</p> <p>3: <b>for</b> <math>n=1 \dots N-1</math></p> <p>4: <b>for</b> <math>m=n+1 \dots N</math></p> <p>5: <b>Estimate</b> <math>\rho_{nm}^G</math>, eq. (21)</p> <p>6: <b>Compute</b> <math>\rho_{nm}^G = \rho_{nm}^G</math></p> <p>7: <b>end for</b></p> <p>8: <b>end for</b></p> <p>9: <b>Output:</b> <math>\rho_{nm}^G \quad n=1 \dots N \quad m=1 \dots M</math></p>
---	---



Finally, table I indicates the computational complexity of the different methods in terms of the observation dimension  $N$ , the number of Gaussian components  $K$  and the size  $L$  of the training set. Only higher-order terms have been retained, which essentially relates to the computation of the required inverses. Thus PCC is  $O(N^3)$  because we must compute the inverse of the covariance matrix. However, aGPCC is  $O(N^5)$  as the computations must be done for every pair  $nm$ . For a given  $N$ , the computational complexity of aGPCC depends linearly on  $K$ , as the computations must be done for every GMM component, while  $L$  has no influence. Finally GPCC is  $O(N^5)$  as the computations are also required for every pair  $nm$ . For a given  $N$ , the computational complexity of aGPCC depends linearly on  $K$ , but also on  $L$ , as the computations must be done for every member of the training set. This dependence on  $L$  is the main responsible of the increased computational complexity of GPCC with respect to aGPCC.

## 4. Experiments

### 4.1 Simulations

Let us consider that  $p(\mathbf{x})$  is a mixture of two Gaussian components having identical covariance matrices and priors, but opposite constant mean values, i.e.,

$$p(\mathbf{x}) = 0.5 \frac{1}{\sqrt{(2\pi)^N |\mathbf{C}|}} \left[ e^{-\frac{1}{2}(\mathbf{x}-b\mathbf{1})^T \mathbf{C}^{-1}(\mathbf{x}-b\mathbf{1})} + e^{-\frac{1}{2}(\mathbf{x}+b\mathbf{1})^T \mathbf{C}^{-1}(\mathbf{x}+b\mathbf{1})} \right], \quad (22)$$

where  $\mathbf{1} = [1 \dots 1]^T$ . First we have computed the Laplacian matrix in the form

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \quad w_{nm} = |\rho_{nm}^G| \quad d_{nn} = \sum_{m=1}^N |\rho_{nm}^G|, \quad (23)$$

where  $\rho_{nm}^G$  has been calculated from (10) considering the true values of the parameters of the two-component GMM. The expectations in (10) were calculated by using numerical integration. Then, the obtained matrix is considered as the true Laplacian. In figure 2 and figure 3, we show the normalized error (defined as the quotient of the Frobenious norm of the difference between the true and the estimated Laplacian matrices, divided by the Frobenious norm of the true Laplacian) corresponding to respectively replacing in (23) the true  $\rho_{nm}^G$  by  $\hat{\rho}_{nm}$  (PCC, eq.(3) with sample estimates of the precision matrix),  $\hat{\rho}_{nm}^{aG}$  (aGPCC, eq. (21), with EM estimates of the GMM parameters) and  $\hat{\rho}_{nm}^G$  (GPCC, eq. (10) with EM estimates of the GMM parameters and sample estimates of the expectations). Then, in figure 2 we show the normalized estimation error for varying separation ( $b$  in (22)) of the two GMM components. The observation dimension was  $N = 20$ , the training size 5000, and the covariance in (22)  $C(n,n)=1 \quad C(n,m)=0.5 \quad n \neq m$ . We see that for  $b = 0$  all the methods give similar error, actually this corresponds to the Gaussian case. When  $b$  increases GPCC is the best in approximating the true

Laplacian. Moreover, as expected, for small  $b$ , PCC yields less error than aGPCC, but after  $b=1,5$ , aGPCC outperforms PCC and almost equates GPCC for  $b \geq 2$ . This is because of the “well separation” assumption considered in the derivation of aGPCC. In figure 3, we show the variation of the estimation error with increasing size of the training set. The GMM component separation was  $b=5$ , the observation size and the covariance matrices were the same than before. We can see that GPCC yields always the best approximation to the true Laplacian, and that for a training size greater than 1600, aGPCC outperforms PCC while being reasonably close to GPCC. A general conclusion of both figures is that as PCC departs from the GPCC performance, then aGPCC better approximates GPCC. Hence, to some extent both methods may be considered a complementary pair of methods for efficient estimation of the partial correlation. Figure 4 shows the normalized error for an increasing number  $K$  of GMM components (from 2 to 10). The training size was 5000 in all cases; that is why the error increases with the number of parameters to be estimated. As in the previous experiments  $N=20$  and  $C_k(n,n)=1$   $C_k(n,m)=0.5$   $n \neq m$  for all  $k$ . The mean value vectors corresponding to the GMM components were uniformly selected between  $\mathbf{b}_1 = -5(K-1)\mathbf{1}$  and  $\mathbf{b}_K = 5(K-1)\mathbf{1}$  so that separation between the components is always a multiple of 5. We observe again the improved performance of aGPCC with respect to PCC for every value of  $k$ . Finally, figure 5 shows the computational time required by the different methods to estimate the Laplacian matrix for varying dimension  $N$  of the observation vector, 2 GMM components,  $b=5$ , training set size  $L=1000$  and the covariance matrices were the same than before. This illustrates the dramatical reduction (some 3 orders of magnitude) of aGPCC respect to GPCC. On the other hand aGPCC is some 2 orders of magnitude above PCC, but this is not so relevant, as the required computational time is reasonably small:  $\square$  0.04 seconds for  $N=4$  to  $\square$  5 seconds for  $N=50$ .

#### 4.2 A real data experiment

We have applied the different methods for estimating the Laplacian matrix to a real data experiment. The framework is that of monitoring the sleep of a patient having some possible disorders like apnea or epilepsy [28]. These disorders are characterized by the regular appearance of an abnormal sleep stage, usually termed as “arousal”. The patient is monitored by 19 channels of EEG recordings. Every signal channel is segmented in intervals of 1-3 seconds and a given feature is computed from every interval and averaged in epochs of 25 seconds. Associated to every epoch, we form an observation vector  $\mathbf{x}$  with the feature extracted from all the channels (the same for all of them), thus  $N = 19$ . A total of 1000 epochs were considered, so this is the size of the training set.

Broadly, the feature vectors so obtained correspond to two different classes: normal sleep or abnormal sleep. Hence, in a first approximation, the overall feature pdf may be considered as a two-component GMM, whose parameters can be estimated using the EM algorithm. Results are shown in table II for 6 different patients having different levels of disorders. We have separately considered two different features. The first one is “amplitude” which correspond to the maximum amplitude in the corresponding

interval, the second one is the “alfa-slow-index” (ASI) which is the ratio of power in the alpha band (8.0–11 Hz) to the combined power in the delta (0.5–3.5 Hz) and theta (3.5–8.0 Hz) bands. For every patient and every feature we have computed the normalized error (as previously defined) of the Laplacian matrix estimates of aGPCC and PCC with respect to the estimates obtained with GPCC. Notice, that the Laplacian matrix is actually a measure of connectivity between the features associated to every channel so, improving its estimation can be relevant in the area of brain connectivity analysis and classification [29]. We have also computed the index of separation  $I_s$  as defined in Appendix B. We can see in table II that aGPCC method is always better than PCC when using the amplitude feature. In that case  $I_s = 1$ , so the “well separation” assumption holds. However,  $I_s$  decreases when using the ASI feature, this produces that in most patients PCC is better than aGPCC for the estimation of the Laplacian matrix.

## 5. Conclusions

We have proposed new methods for estimating the Laplacian matrix of undirected graphs. By assuming that the underlying pdf is a mixture of Gaussians, the new methods capture the connectivity of the graph more completely than the precision matrix. The normalized elements of this matrix are the PCC, a measure of conditional pairwise linear dependence. We propose to compute the GPCC, thus capturing the conditional non-linear dependences implicit in any non-Gaussian model. The GPCC can be estimated from estimates of the GMM parameters and sample estimates or numerical integration. To overcome the computational burden of this later, we have also proposed aGPCC which is an approximate solution under the assumption of well separated components of the GMM.

As the multivariate GMM is able to reasonably fit any non-Gaussian multivariate pdf, the proposed methods may have general applicability in any signal processing on graph problem where a Laplacian matrix is to be learned from training signals.

Future research may be devised by considering more sophisticated non-Gaussian models, as those ones based on mixtures of independent component analyzers [30], [31]. Another issue of interest is the extension of the proposed methods to the complex case. The complex GMM has been considered elsewhere (see for example [22], [23] and references there in). However, while defining a complex GPCC is straightforward, the definition of a complex Laplacian is not so obvious. Depending on it, the well-known properties of a real Laplacian matrix (e.g., semidefinite positive, minimum eigenvalue equal to zero, ...) could be preserved or not. Moreover, the meaning of the imaginary part (or the phase) of the complex weight assigned to an edge in the signal graph requires some interpretation.

## Appendix A: Derivation of the conditional mean

We start by

$$E[x_n/\mathbf{x}_{-nm}] = \int x_n p(x_n/\mathbf{x}_{-nm}) dx_n = \frac{1}{p(\mathbf{x}_{-nm})} \int x_n p(\mathbf{x}_m) dx_n \quad . \quad (\text{A1})$$

Where  $\mathbf{x}_{-m}$  is the vector formed by all the elements of  $\mathbf{x}$  except  $x_m$  and  $p(\mathbf{x}_{-m})$  is the corresponding marginal. This later can be calculated by integrating  $p(\mathbf{x})$  in (1) over  $x_m$ , that is

$$p(\mathbf{x}_{-m}) = \int_{x_m} p(\mathbf{x}) dx_m = \sum_{k=1}^K P_k \int_{x_m} \frac{1}{\sqrt{(2\pi)^N |\mathbf{C}_k|}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{b}_k)^T \mathbf{C}_k^{-1}(\mathbf{x}-\mathbf{b}_k)} dx_m \quad . \quad (\text{A2})$$

But any marginal of a multivariate Gaussian is also multivariate Gaussian. The marginal mean vector is obtained by dropping the  $m$ -th element from the mean vector  $\mathbf{b}_k$ . Similarly, the marginal covariance matrix is obtained from  $\mathbf{C}_k$  by dropping the  $m$ -th row and the  $m$ -th column. Let us respectively call  $\mathbf{b}_{k-m}$  and  $\mathbf{C}_{k-m}$  to the marginal mean and covariance. Then we can write:

$$p(\mathbf{x}_{-m}) = \sum_{k=1}^K P_k \frac{1}{\sqrt{(2\pi)^{N-1} |\mathbf{C}_{k-m}|}} e^{-\frac{1}{2}(\mathbf{x}_{-m}-\mathbf{b}_{k-m})^T \mathbf{C}_{k-m}^{-1}(\mathbf{x}_{-m}-\mathbf{b}_{k-m})} \quad . \quad (\text{A3})$$

And substituting in (7)

$$E[x_n/\mathbf{x}_{-nm}] = \frac{1}{p(\mathbf{x}_{-nm})} \sum_{k=1}^K P_k \int x_n \frac{1}{\sqrt{(2\pi)^{N-1} |\mathbf{C}_{k-m}|}} e^{-\frac{1}{2}(\mathbf{x}_{-m}-\mathbf{b}_{k-m})^T \mathbf{C}_{k-m}^{-1}(\mathbf{x}_{-m}-\mathbf{b}_{k-m})} dx_n \quad . \quad (\text{A4})$$

Now, let us define the multivariate Gaussian pdf

$$p_k(\mathbf{x}_{-m}) = p_k(x_n, \mathbf{x}_{-nm}) = \frac{1}{\sqrt{(2\pi)^{N-1} |\mathbf{C}_{k-m}|}} e^{-\frac{1}{2}(\mathbf{x}_{-m}-\mathbf{b}_{k-m})^T \mathbf{C}_{k-m}^{-1}(\mathbf{x}_{-m}-\mathbf{b}_{k-m})} \quad . \quad (\text{A5})$$

In (10) we have to compute

$$\int_{x_n} x_n p_k(x_n, \mathbf{x}_{-nm}) dx_n = p_k(\mathbf{x}_{-nm}) \int_{x_n} x_n p_k(x_n/\mathbf{x}_{-nm}) dx_n \quad , \quad (\text{A6})$$

but

$$p_k(\mathbf{x}_{-nm}) = \frac{1}{\sqrt{(2\pi)^{N-2} |\mathbf{C}_{k-nm}|}} e^{-\frac{1}{2}(\mathbf{x}_{-nm}-\mathbf{b}_{k-nm})^T \mathbf{C}_{k-nm}^{-1}(\mathbf{x}_{-nm}-\mathbf{b}_{k-nm})} \quad . \quad (\text{A7})$$

Where  $\mathbf{b}_{k_{-nm}}$  is obtained from  $\mathbf{b}_k$  by dropping the  $n$ -th and  $m$ -th elements, and  $\mathbf{C}_{k_{-nm}}$  is obtained by removing the  $n$ -th and  $m$ -th rows and columns from  $\mathbf{C}_k$ .

The last integral in (12) coincides with the conditional mean of  $x_n$  with respect to  $\mathbf{x}_{-nm}$  assuming the multivariate Gaussian model (11), i.e., assuming that  $\mathbf{x}$  has been generated by the  $k$ -th mixture component. Therefore, this conditional mean is the LLMSE estimate of  $x_n$  from  $\mathbf{x}_{-nm}$ , which can be obtained by the Wiener-Hopf equations

$$\int_{x_n} x_n p_k(x_n/\mathbf{x}_{-nm}) dx_n = \mathbf{c}_{kn_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} (\mathbf{x}_{-nm} - \mathbf{b}_{k_{-nm}}) + b_{kn} \quad , \quad (\text{A8})$$

where vector  $\mathbf{c}_{kn_{-nm}}$  is formed by the elements of the  $n$ -th column of matrix  $\mathbf{C}_k$ , dropping rows  $n$  and  $m$ , and  $b_{kn}$  is the  $n$ -th element of vector  $\mathbf{b}_k$ . So we have

$$E[x_n/\mathbf{x}_{-nm}] = \frac{1}{P(\mathbf{x}_{-nm})} \sum_{k=1}^K P_k \cdot p_k(\mathbf{x}_{-nm}) \left[ \mathbf{c}_{kn_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} (\mathbf{x}_{-nm} - \mathbf{b}_{k_{-nm}}) + b_{kn} \right] . \quad (\text{A9})$$

But, actually, we may express  $p_k(\mathbf{x}_{-nm}) = p(\mathbf{x}_{-nm}/k)$  as a class conditional pdf, then

$$P_k \cdot p_k(\mathbf{x}_{-nm}) = P_k \cdot p(\mathbf{x}_{-nm}/k) = P(k/\mathbf{x}_{-nm}) p(\mathbf{x}_{-nm}) \quad . \quad (\text{A10})$$

Hence, finally

$$E[x_n/\mathbf{x}_{-nm}] = \sum_{k=1}^K P(k/\mathbf{x}_{-nm}) \left[ \mathbf{c}_{kn_{-nm}}^T \mathbf{C}_{k_{-nm}}^{-1} (\mathbf{x}_{-nm} - \mathbf{b}_{k_{-nm}}) + b_{kn} \right] \quad . \quad (\text{A11})$$

Therefore notice that the (nonlinear) conditional mean is a weighted sum of  $K$  components. The  $k$ -th component is the LLMSE estimate of  $x_n$  from  $\mathbf{x}_{-nm}$  assuming that the  $k$ -th mixture element is in force. So, the corresponding weight is the probability of the  $k$ -th mixture element to be in force, conditioned to the observation  $\mathbf{x}_{-nm}$ . This weighting makes the conditional mean a nonlinear function of  $\mathbf{x}_{-nm}$ .

### **Appendix B:** *A measure of separation between the GMM components*

To simplify the procedure we consider that “well separation” between  $p(\mathbf{x}/k)$  and  $p(\mathbf{x}/k')$  implies “well separation” between  $p(\mathbf{x}_{-nm}/k)$  and  $p(\mathbf{x}_{-nm}/k')$  for any  $nm$ . Then, let us define the function  $f(\mathbf{x}, k, k') = p(\mathbf{x}/k) p(\mathbf{x}/k')$   $k \neq k'$ . In our context, “well separation” means that  $\max f(\mathbf{x}, k, k') < \varepsilon$  for some small number  $\varepsilon$ . Let define  $\mathbf{x}_{\max}$  as the value which maximizes  $f(\mathbf{x}, k, k')$  for a given pair  $k, k'$ , i.e.,

$$\begin{aligned}
\mathbf{x}_{\max} &= \max_{\mathbf{x}} f(\mathbf{x}, k, k') = \max_{\mathbf{x}} \ln f(\mathbf{x}, k, k') = \max_{\mathbf{x}} \{ \ln p(\mathbf{x}/k) + \ln p(\mathbf{x}/k') \} = \\
&= \max_{\mathbf{x}} \left\{ -(\mathbf{x} - \mathbf{b}_k)^T \mathbf{C}_k^{-1} (\mathbf{x} - \mathbf{b}_k) - (\mathbf{x} - \mathbf{b}_{k'})^T \mathbf{C}_{k'}^{-1} (\mathbf{x} - \mathbf{b}_{k'}) \right\} .
\end{aligned} \tag{B1}$$

Deriving this last expression with respect to  $\mathbf{x}$  and equating to zero we obtain

$$-2\mathbf{C}_k^{-1} (\mathbf{x} - \mathbf{b}_k) - 2\mathbf{C}_{k'}^{-1} (\mathbf{x} - \mathbf{b}_{k'}) = 0 \Rightarrow \mathbf{x}_{\max} = (\mathbf{C}_k^{-1} + \mathbf{C}_{k'}^{-1})^{-1} (\mathbf{C}_k^{-1} \mathbf{b}_k + \mathbf{C}_{k'}^{-1} \mathbf{b}_{k'}) . \tag{B2}$$

Then

$$\begin{aligned}
\ln f(\mathbf{x}_{\max}, k, k') &= \\
&= -\ln(2\pi)^N \sqrt{|\mathbf{C}_k| |\mathbf{C}_{k'}|} - \frac{1}{2} (\mathbf{x}_{\max} - \mathbf{b}_k)^T \mathbf{C}_k^{-1} (\mathbf{x}_{\max} - \mathbf{b}_k) - \frac{1}{2} (\mathbf{x}_{\max} - \mathbf{b}_{k'})^T \mathbf{C}_{k'}^{-1} (\mathbf{x}_{\max} - \mathbf{b}_{k'})
\end{aligned} \tag{B3}$$

So  $\max \ln f(\mathbf{x}, k, k')$  can be obtained by computing (B3) for all pairs  $k, k'$  and retaining the maximum.

Notice that if  $\mathbf{C}_k = \mathbf{C}_{k'}$ , then

$$\begin{aligned}
\mathbf{x}_{\max} &= \frac{1}{2} (\mathbf{b}_k + \mathbf{b}_{k'}) \\
\ln f(\mathbf{x}_{\max}, k, k') &= -\ln(2\pi)^N |\mathbf{C}_k| - \frac{1}{2} (\mathbf{b}_{k'} - \mathbf{b}_k)^T \mathbf{C}_k^{-1} (\mathbf{b}_{k'} - \mathbf{b}_k)
\end{aligned} , \tag{B4}$$

which is closely related with the well-known Mahalanobis distance [26] between two multivariate normal densities having the same covariance matrix. Finally let us call  $s = \max \ln f(\mathbf{x}, k, k')$  so that we can define an ‘‘ad hoc’’ normalized index of separability  $I_s$  as

$$I_s = \frac{e^{-s}}{1 + e^{-s}} . \tag{B5}$$

Clearly,  $0 \leq I_s \leq 1$  and, as separation increases,  $s \rightarrow -\infty$  and  $I_s \rightarrow 1$ .

## Acknowledgements

This work was supported by Spanish Administration (Ministerio de Economía y Competitividad) and European Union (FEDER) under grant TEC2014-58438-R, and Generalitat Valenciana under grant PROMETEO II/2014/032.

## References

- [1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains, *IEEE Signal Process. Magazine* 30 (2013) 83–98.

- [2] A. Sandryhaila, J. M. F. Moura, Discrete signal processing on graphs, *IEEE Trans. Signal Processing* 61 (2013) 1644–1656.
- [3] A. Sandryhaila, J. M. F. Moura, Discrete signal processing on graphs: Frequency analysis, *IEEE Trans. Signal Processing* 62 (2014) 3042–3054.
- [4] R. Merris, Laplacian matrices of a graph: a survey, *Linear Alg. Applicat.* 197 (1994) 143–176.
- [5] D. Spielman, Spectral graph theory, in: U. Naumann, O. Schnek (Eds), *Combinatorial Scientific Computing*, Chapter 16, Chapman and Hall/CRC Press, 2012.
- [6] C. Hu, L. Cheng, J. Sepulcre, G. E. Fakhri, Y. M. Lu, Q. Li, A graph theoretical regression model for brain connectivity learning of Alzheimer’s disease, 2013 IEEE 10<sup>th</sup> International Symposium on Biomedical Imaging (ISBI2013), San Francisco (USA) (2013) 616-619.
- [7] X. Dong, D. Thanou, P. Frossard, P. Vandergheynst, Learning Laplacian matrix in smooth graph signal representations, *IEEE Trans. on Signal Processing* 64 (2016) 1660–1673.
- [8] Ch. Zhang, D. Florencio, P. A Chou, Graph signal processing—a probabilistic framework, Microsoft Research Technical Report MSR-TR-2015-31 (2015) 1-10.
- [9] E. Pávez, A. Ortega, Generalized precision matrix estimation for graph signal processing, *Proc. of 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)*, Sanghai (China) (2016) 6350-6354.
- [10] J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso, *Biostatistics* 9 (2008) 432-441.
- [11] R. Mazumder, T. Hastie, The graphical lasso: New insights and alternatives, *Electronic Journal of Statistics* 6 (2012) 2125-2149.
- [12] C.J. Hsieh, M.A. Sustik, I.S. Dhillon, P. Ravikumar, Sparse inverse covariance matrix estimation using quadratic approximation, *Advances in Neural Information Processing Systems* 24 (2011) 2330-2338.
- [13] X. Chen, M. Xu, W. B. Wu, Covariance and precision matrix estimation for high-dimensional time series, *The Annals of Statistics* 41 (2013), 2994-3021.
- [14] V. Öllerer, C. Croux, Robust high-dimensional precision matrix estimation, in K. Nordhausen, S.Taskinen (Eds), *Modern Multivariate and Robust Methods*, Springer, New York, 2015.
- [15] J. Peng J, P. Wang, N. Zhou N, J. Zhu, Partial correlation estimation by joint sparse regression model, *Journal of the American Statistical Association* 104 (2009) 735–746.
- [16] J. Igual, A. Salazar, G. Safont, L. Vergara, Semi-Supervised Bayesian Classification of Materials with Impact-Echo Signals, *Sensors* 15 (2015) 11528-11550.
- [17] S. Aminikhanghahi, W. Wang, S. I. Jeon, S. H. Son, A new fuzzy Gaussian mixture model (FGMM) based algorithm for mammography tumor image classification, *Multimedia Tools and Applications* 75 (2016) 1-15.
- [18] M. Marczyk, Improved Detection of 2D Gel Electrophoresis Spots by Using Gaussian Mixture Model, 12-th International Symposium on Bioinformatics Research and Applications, Minks (Belarus) (2016) 284-294.

- [19] J. Wang, M. R. Taaffe, Multivariate Mixtures of Normal Distributions: Properties, Random Vector Generation, Fitting, and as Models of Market Daily Changes, *INFORMS Journal on Computing* 27 (2015) 193-203.
- [20] M. U. Hassan, P. Stockhammar, Fitting probability distributions to economic growth: a maximum likelihood approach, *Journal of Applied Statistics* 43 (2016) 1583-1603.
- [21] A. P. Igoshev, S. B. Popov, Gaussian mixture models and the population synthesis of radio pulsars, *Monthly Notices of the Royal Astronomical Society* 434 (2013) 2229-2237.
- [22] P. Salvo Rossi, D. Ciounzo, K. Kansanen, T. Ekman, Performance Analysis of Energy Detection for MIMO Decision Fusion in Wireless Sensor Networks Over Arbitrary Fading Channels, *IEEE Transactions on Wireless Communications* 15 (2016) 7794-7806.
- [23] P. Salvo Rossi, D. Ciounzo, K. Kansanen, T. Ekman, On energy detection for MIMO decision fusion in wireless sensor networks over NLOS fading, *IEEE Communications Letters* 19 (2015) 303-306.
- [24] S. Kotz, S. Nadarajah, Local dependence functions for elliptically symmetric distributions, *Sankhya: The Indian Journal of Statistics* 65 (2003) 207–223.
- [25] K. Baba, R. Shibata, M. Sibuya, Partial correlation and conditional correlation as measures of conditional independence, *Australian & New Zealand Journal of Statistics* 46 (2004) 657-664.
- [26] E. Nowakowska, J. Koronacki, S. Lipovetsky, Tractable measure of component overlap for gaussian mixture models, *ArXiv:1407.7172v1*. (2014).
- [27] D. Leung, J. Romagnoli, Dynamic probabilistic model-based expert system for fault diagnosis, *Journal of Computers & Chemical Engineering* 24 (2000) 2473-2492.
- [28] A. Salazar, L. Vergara, R. Miralles, On including sequential dependence in ICA mixture models, *Signal Processing*, 90 (2010) 2314-2318.
- [29] E.W. Lang, A. Tomé, I.R. Keck, J. Górriz-Sáez, C. Puntonet, Brain connectivity analysis: a short survey. *Computational intelligence and neuroscience*, 2012 (2012) 1-21.
- [30] A. Salazar, L. Vergara, A. Serrano, J. Igual, A General Procedure for Learning Mixtures of Independent Component Analyzers, *Pattern Recognition*, 43 (2010) 69-85.
- [31] G., Safont, A. Salazar, L. Vergara, E. Gomez, V. Villanueva, Probabilistic Distance for Mixtures of Independent Component Analyzers, *IEEE Transactions on Neural Networks and Learning Systems*, Article in Press, DOI: 10.1109/TNNLS.2017.2663843 (2017).