Hyperspectral Unmixing with Spatial Dependence

Chia Chye Yee*
Department of Statistics
University of Michigan
Ann Arbor, Michigan 48109-1107
Email: chye@umich.edu

Yves Atchade
Department of Statistics
University of Michigan
Ann Arbor, Michigan 48109-1107
Email: yvesa@umich.edu

Abstract

Motivated by the increasing availability of high-dimensional laboratory-generated spectral libraries, this paper introduces a library-based statistical model to unmix hyperspectral images. The model accounts for spatial correlation between the pixels via a latent Markov random field such that pixels with the same latent group label share the same mixture of pure end-members.

Keywords: Hyperspectral Unmixing; Empirical Bayes; Markov Random Fields; Monte Carlo EM.

1. Introduction

Hyperspectral cameras can capture electromagnetic radiations in a large spectral range at a very fine spectral resolution. For instance, Airborne Visible Infrared Imaging Spectrometer (AVIRIS) is able to detect wavelengths from 0.4 μm to 2.5 μm in the electromagnetic spectrum. As a comparison, the typical human eye is able to detect wavelengths from 0.4 μm to 0.7 μm in the electromagnetic spectrum. The high spectral resolution of hyperspectral images generated by these cameras gives researchers the capability to identify materials from remote sensing images by analyzing collected spectral signatures. However, due to the limited spatial resolution of the sensors, mixing of materials at the microscopic level, and scattering, the spectral signature acquired at the pixel level is a mixture of multiple spectral signatures of pure materials (end-members). In this paper, at the pixel level, the mixture is assumed to be linear and contains only few end-members. Hyperspectral unmixing is the task of discovering the material composition of hyperspectral image by analyzing spectral images. It has a wide range of applications. Examples include, but are not limited to remote geological surveys, emergency response systems, forest conservation, agricultural surveys, and toxic waste detection. Depending on the environment surveyed, remote hyperspectral unmixing is considerably cheaper and safer, and complements costly on-site surveys very well.

This paper introduces a statistical model that utilizes high-dimensional libraries and incorporate spatial correlation to unmix hyperspectral data. The spatial dependence is formulated by introducing a latent group assignment variable for each pixel. The group assignments jointly follows a Potts Markov random field that favors neighboring pixels belonging to the same group. Our latent group assignment is similar to the one described in [1]. However, our model differs from [1] in that, in our case, pixels sharing the same latent group assignment have the same pure end-members composition. This translates into a more parsimonious model that scales very well with the library size and the image size. We use the sparse Bayesian learning (SBL) framework of [2, 3] to obtain a sparse estimate of the abundances. Compared to LASSO, SBL has the advantage that no regularization parameter tuning is required.

2. The Statistical Model for Hyperspectral Unmixing

We begin by viewing the hyperspectral image as a three-dimensional data cube with pixels indexed by $p \in \{1, \ldots, P\}$ and each pixel contains an observed length L vector of observed reflectances, $\mathbf{y}_p$. At a given

pixel p with group assignment $z_p \in \{1, \ldots, K\}$, we make the following distributional assumption:

$$\mathbf{y}_p \sim N(\mathbf{X}\beta_{z_p}, s^2\mathbf{I}_L) \qquad\qquad p = 1, \ldots, P. \tag{1}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_R]$ is a known $L \times R$ matrix containing the spectral signatures of the end-members that are present in the library, L is the number of spectral bands, and $\beta_i \in \mathbb{R}^R$, $1 \leq i \leq K$ are parameters. Hence the implicit assumption made in the model is that two pixels with the same group assignment contains the same linear mixture of end-members from the library. However, the observed data at these two pixels might still be different due to the added random noise in (1).
Spatial dependence is introduced in the model through the group assignments $\{z_p,\ 1 \leq p \leq P\}$ which are assumed spatially dependent. More precisely we assume that the group assignments $\{z_p,\ 1 \leq p \leq P\}$ follows a Potts Markov random field distribution.

$$f(\mathbf{z}|\theta) = \frac{1}{G(\theta)} \exp\left\{ \sum_{p=1}^{P} \sum_{p' \in \mathcal{V}(p)} \theta\delta(z_p - z_{p'}) \right\}, \tag{2}$$

where $\mathcal{V}(i)$ is the neighborhood (defined here as the four nearest pixel) around pixel i, θ is the granularity constant, $G(\theta)$ is the normalizing constant, $\delta(\cdot)$ is the Kronecker function that returns 1 when the input is zero and 0 otherwise, and $\mathbf{z} = [z_1, \ldots, z_p]$.
Our model is inspired by, and shares some common features with [1]. However, our formulation is more parsimonious which is key for fast computation. Indeed, the library $\boldsymbol{X}$ in our case is a high-dimensional lab-generated library while the library in [1] is a small library extracted from the hyperspectral image.
Based on (1), the likelihood for the data is:

$$f(\mathbf{Y}|\boldsymbol{z}, \boldsymbol{\beta}, s^2) \propto \prod_{p=1}^{P} \left(\frac{1}{s^2}\right)^{L/2} \exp\left\{ -\frac{1}{2s^2} \|\boldsymbol{y}_p - \mathbf{X}\beta_{z_p}\|^2 \right\}.$$

Using the Bayesian methodology, we impose on β_k a prior distribution $\pi_{\Sigma_k}(d\beta)$ of the following form:

$$\beta_j|\sigma_{j,k} \sim \begin{cases} \delta_0 & \text{if } \sigma_{j,k} = 0 \\ N(0, \sigma_{j,k}) & \text{if } \sigma_{j,k} > 0 \end{cases}, \tag{3}$$

for some hyper-parameter $\boldsymbol{\Sigma}_k = diag(\sigma_{1,k}, \ldots, \sigma_{R,k})$, where $\sigma_{i,k} \geq 0$ and δ_0 is the Dirac measure at 0.
The joint posterior distribution for both $\boldsymbol{\beta}$ and $\mathbf{z}$ is then

$$\begin{aligned} \pi(d\boldsymbol{\beta}, \mathbf{z}|\mathbf{Y}, s^2, \boldsymbol{\Sigma}, \theta) \propto\ & f(\mathbf{z}|\theta) f(\mathbf{Y}|\boldsymbol{z}, \boldsymbol{\beta}, s^2) \\ & \times \prod_{k=1}^{K} \pi_{\boldsymbol{\Sigma}_k}(d\beta_k). \end{aligned} \tag{4}$$

According to Bayesian analysis, the estimator for the abundances which minimizes risk is the posterior mean of the abundances:

$$\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}(s, \boldsymbol{\Sigma}, \theta) = \iint \boldsymbol{\beta}\pi(d\boldsymbol{\beta}, \mathbf{z}|\mathbf{Y}, s^2, \boldsymbol{\Sigma}, \theta) d\mathbf{z}. \tag{5}$$

However, this estimator depends on unknown hyper-parameters θ, $\boldsymbol{\Sigma}$, and s^2. As in the SBL described in [3] and [2], we find these parameters by maximizing the marginal log-likelihood

$$(\hat{\theta}, \hat{\boldsymbol{\Sigma}}, \hat{s^2}) = \arg \min_{\theta, \boldsymbol{\Sigma}, s^2} l(\theta, s^2, \boldsymbol{\Sigma}; \mathbf{Y}) \tag{6}$$

where $l(\theta, s^2, \boldsymbol{\Sigma}; \mathbf{Y})$ is the log-likelihood with $\boldsymbol{\beta}$ and $\mathbf{z}$ marginalized. Solving optimization problem (6) in its original form is complex. In addition, once the hyper-parameters have been computed, the estimator (5) is not of standard form. The computation of both are covered in the next section. The formulation of

(3) in combination with the solution to the optimization problem (6) induces sparse β_k according to [3] and [2]

3. Computation and Algorithms

The computation mainly involves the EM algorithm for solving (6) and Markov Chain Monte Carlo for evaluating the posterior mean (5). The rough frame of the algorithm is as follows:

Initialize $\mathbf{z}_0$, θ_0, s_0^2, and $\mathbf{\Sigma}_{k,0}$.
For $t = 1, \ldots, T$
(Step 1) Given $\mathbf{\Sigma}_{\mathbf{z}_t,t}$ and s_t^2,
update $\mathbf{z}_{t+1}$ and $\boldsymbol{\beta}_{t+1}$
(Step 2) Given $\mathbf{z}_{t+1}$ and $\boldsymbol{\beta}_{t+1}$,
update $\mathbf{\Sigma}_{\mathbf{z}_t,t}$ and s_{t+1}^2, and θ_{t+1}
End for
$\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}_T$, $\hat{\mathbf{\Sigma}} = \mathbf{\Sigma}_T$, $\hat{s}^2 = s_T^2$, $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_T$, $\hat{z}_p = z_{p,T}$

(Step 1) involves the use of Markov Chain Monte Carlo simulation to update both the parameters of interest and the latent group assignment, $(\boldsymbol{\beta}, \mathbf{z})$. (Step 2) uses the EM algorithm to update the hyper-parameters $(\theta, \mathbf{\Sigma}, s^2)$. This step is further complicated by the non-standard update for the granularity constant θ which requires a gradient update for each of the M-step in the EM algorithm.

3.1 Markov Chain Monte Carlo (MCMC)

(Step 1) from the algorithm uses a Gibb's sampling algorithm in order to sample from the joint posterior distribution $\pi(d\boldsymbol{\beta}, \mathbf{z}|\mathbf{Y}, s^2, \mathbf{\Sigma}, \theta)$ which samples from the following full conditional distributions iteratively.

$$\pi(\mathbf{z}|\boldsymbol{\beta}, \mathbf{Y}, s^2, \mathbf{\Sigma}, \theta) \propto f(\mathbf{Y}|\boldsymbol{z}, \boldsymbol{\beta}, s^2) f(\mathbf{z}|\theta) \tag{7}$$

$$\pi(\beta_k|\mathbf{z}, \boldsymbol{\beta}_{-k}, \mathbf{Y}, s^2, \mathbf{\Sigma}, \theta) \sim N(\gamma_k, \Gamma_k)$$

where $\gamma_k = \left[|\mathcal{I}_k|\mathbf{X}^T\mathbf{X} + s^2\mathbf{\Sigma}_k^{-1}\right]^{-1}\mathbf{X}^T\left(\sum_{p\in\mathcal{I}_k}\mathbf{y}_p\right)$ and $\Gamma_k = \left[|\mathcal{I}_k|\mathbf{X}^T\mathbf{X} + s^2\mathbf{\Sigma}_k^{-1}\right]^{-1}$ with $|\mathcal{I}_k|$ being the number of pixels belonging to group k. Sampling from (7) involves the use of Wolff Clustering algorithm. Further details on the Wolff Clustering algorithm is available from [4]. Once $\mathbf{z}_{t+1}$ has been generated, $\beta_{k,t+1}$ is set to the mean γ_k.

3.2 EM Algorithm

(Step 2) uses the EM algorithm to approximate the solution to (6). This algorithm simplifies (6) by iteratively maximizing a surrogate function which provides a lower bound for the original objective function. The E-step finds the surrogate function to be maximized and the M-step maximizes the surrogate function.

- E-step:

$$Q_t = \mathbb{E}\left\{l_c(\theta_t, s_t^2, \mathbf{\Sigma}_t; \mathbf{Y}, \boldsymbol{\beta}, \mathbf{z})|\mathbf{Y})\right\}$$

  where $l_c(\theta_t, s_t^2, \mathbf{\Sigma}_t; \mathbf{Y}, \boldsymbol{\beta}, \mathbf{z})|\mathbf{Y})$ is the complete log-likelihood: $\log f(\mathbf{Y}|\boldsymbol{\beta}, s_t^2)\pi(\boldsymbol{\beta}|\mathbf{z}, \mathbf{\Sigma}_t) f(\mathbf{z}|\theta_t)$. It should be noted that the expectation in the E-step is taken with respect to the posterior distribution.

- M-step:

$$\begin{aligned} \theta_{t+1} &= \arg\max_{\theta_t} Q_t \\ s_{t+1}^2 &= \arg\max_{s_t^2} Q_t \\ \mathbf{\Sigma}_{t+1} &= \arg\max_{\mathbf{\Sigma}_t} Q_t \end{aligned} \tag{8}$$

The M-step involves solving the three optimization problems which will yield the following updates:

$$s^2_{t+1} = \frac{\iint \sum_{p=1}^{P} \|\mathbf{y}_p - \mathbf{X}\beta_{z_p}\|^2 \pi(\boldsymbol{\beta}, \mathbf{z}|\mathbf{Y}, s^2_t, \boldsymbol{\Sigma}_t, \theta_t) d\mathbf{z} d\boldsymbol{\beta}}{PL}$$

$$\sigma_{r,k,t+1} = \iint \beta^2_{r,k} \pi(\boldsymbol{\beta}, \mathbf{z}|\mathbf{Y}, s^2_t, \boldsymbol{\Sigma}_t, \theta_t) d\mathbf{z} d\boldsymbol{\beta}$$

The updates require MCMC sampling from the joint posterior distribution $\pi(d\boldsymbol{\beta}, \mathbf{z}|\mathbf{Y}, s^2, \boldsymbol{\Sigma}, \theta)$ with the hyper-parameters set to values from the previous iteration. Note that update (8) is not available in closed form due to difficulty in evaluating the normalizing constant from the Potts Markov random field distribution $G(\theta)$. This update is done using a gradient algorithm outlined in [5]. The gradient algorithm used in this case is slightly different from the one outlined in [5] because the first and second derivatives of Q_t with respect to θ are approximated via MCMC as opposed to directly evaluated.

3.3 (Optional) Thresholding of $\hat{\beta}$

After the parameter of interest β is evaluated, we further applied a thresholding scheme outlined in [3] in order to induce strict sparsity:

$$\hat{\beta}_{k,p,r} = 0 \qquad if \qquad \hat{\sigma}_{k,p,r} \leq \frac{\hat{s}^2(1 + 2|\hat{\rho}|)}{\boldsymbol{X}_r^T \boldsymbol{X}_r}$$

where $\boldsymbol{X}_r$ is the r-th column of $\boldsymbol{X}$ and $\hat{\rho}$ is the coherence of $\boldsymbol{X}$. This further simplifies the composition of end-members for each pixel.

4. Simulation and Results

A simulation study using synthetic data is used to evaluate the performance of the method. The metrics used to evaluate the performance are pixel group misclassification rate and the relative error rate of the abundances $\frac{\|\hat{\beta} - \beta^*\|}{\|\beta^*\|}$.

The simulation is conducted in R using synthetic data. The library $\boldsymbol{X}$ used in the simulation study is generated from the following model:

$$\boldsymbol{X}_l \sim N(0_R, \boldsymbol{\Psi}) \qquad \forall l = 1, \ldots L$$

where $\boldsymbol{\Psi}_{ij} = 0.8^{|i-j|}$. The noise level in the observed image is set to $s^2 = 1$ according to model stated in (1). The granularity constant of the images used in the simulation is set to $\theta = 0.3142$. With $\boldsymbol{\Psi}_{ij} = 0.8^{|i-j|}$, the average coherence between the columns of $\boldsymbol{X}$ in 100 repetitions is approximately 0.8664.

In the simulation, we varied the library size $R = \{100, 150, 200, 300, 1000\}$ and the number of spectral bands is set to $L = 200$ while the image size is set to 100×100 pixels. Approximately 1% of the entries in $\boldsymbol{\beta}_k$ are non-zero with magnitude generated from $U(1, 2)$. The number of classes for this simulation is set to $K = 5$. The relative error rate is presented in Table 1 while the misclassification rate is presented in Figure 1. The simulation results using synthetic data are promising with both the pixel misclassification rate and relative error rate for abundances converging to zero. The results in this section are significant because in field applications, the size of the spectral libraries grow much faster than the spectral resolution of hyperspectral sensors.

In addition to the simulation using synthetic data, we applied the method using an agricultural scene with corn, grass-trees, oats and soybean. The image is a subset of the Indian Pines data set obtained from the website: `http://www.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes`. For this particular scene, the unmixing is performed using the vegetation and man-made section of USGS hyperspectral library obtained from `http://speclab.cr.usgs.gov/spectral.lib06/`. In short, the library has $L = 200$ rows and $R = 310$ after the water absorption bands have been removed from the library and hyperspectral image. Figure 2a shows the classification plot for both the ground-truth and the recovered classification.

	Class 1	Class 2	Class 3	Class 4	Class 5
R = 100	0	0	0.0000	0.0000	0.000
R = 150	0	0	0.0000	0.0000	0.000
R = 200	0	0	0.0000	0.0000	0.000
R = 300	0	0	0.0000	0.0000	0.000
R = 1000	0	0	0.0032	0.0043	0.067

Table 1: Relative Error Evaluated via l_2 norm for different library sizes

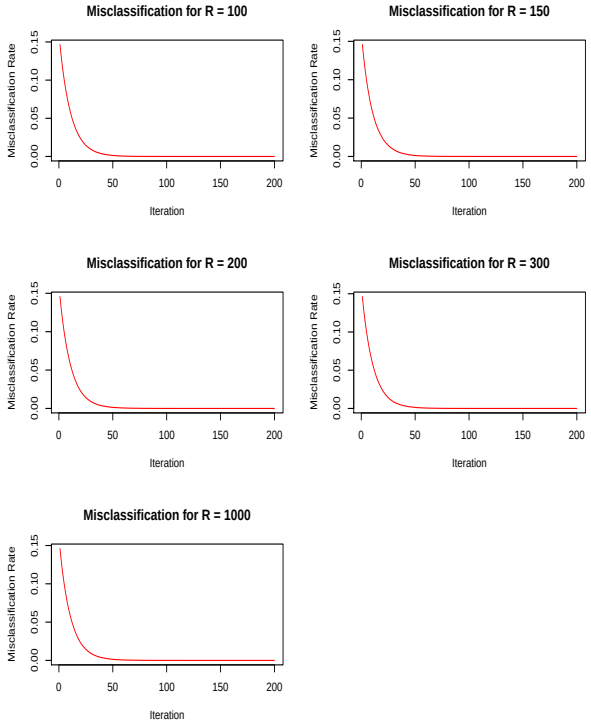

Figure 1: Plots of the misclassification rate for the images as a function of the EM iteration

Although the classification plot is reasonable, the abundances recovered are less informative. The vegetation section of the library provided by the USGS does not include spectral signatures of agricultural products such as soybean, corn and oats. The library is ill-matched to the scene. It is difficult to draw any conclusion from the abundances because the algorithm merely attempts to reconstruct the observed spectral signature using linear combination of non-related end-members in the library. In addition, the method shows poor performance when it comes to differentiating different crops that are not in the library. However, the method works well in terms of separating the crops and non-crops in the scene. In this context, it is encouraging that the algorithm is able to recover the pixel classes even if the library is not related to the image. In this case, there might be some merit to reducing the scope of the estimation by reducing the number of pixel labels to $K = 3$. Figure 2b shows the result for the classification plot of this simulation. It is clear that in this case the method works well when it comes to differentiating the pixels containing crop vegetation (green) vs. the non-crop vegetation (pink).

5. Conclusions

We applied an empirical Bayes method to unmix hyperspectral images. A Markov random field of latent

(a) Classification with $K = 4$

(b) Classification with $K = 3$

Figure 2: For each subfigure (a) and (b), the recovered classification plot is on the left and the plot of the ground-truth is on the right

pixel labels is used to model the spatial correlation between neighboring pixels. The pixels sharing the same label shares the same composition of end-members in order to improve the scalability of the method and reduce the complexity of the model used in [1]. The Monte Carlo EM algorithm is used to compute the hyper-parameters and MCMC is used to sample from the joint posterior distribution which is needed for estimating the pixel labels as well as composition for each label. The results obtained for synthetic data, is promising. In real data simulations, the method performs reasonably well even if the library is ill-matched to the scene.

References

[1] Eches O., Dobigeon N. and Tourneret J-Y. (2011). Enhancing Hyperspectral Image Unmixing with Spatial Correlations with Spatial Correlations *IEEE Trans. on Geosc. and Remote Sensing*, 4239–4247.

[2] Tipping M. (2001). Sparse Bayesian Learning and the Relevance Vector Machine *J. Mach. Learn. Res.*, 211–244.

[3] Atchade Y. and Yee C. C. (2014). On the Sparse Bayesian Learning of Linear Models *Tech. Report* .

[4] Liu J. S. (2001). Cluster Algorithms for the Ising Model *Monte Carlo Strategies in Scientific Computing*, 153–159.

[5] Lange K. (1995) A Gradient Algorithm Locally Equivalent to the EM Algorithm *Journal of the Royal Statistical Society*, 425–437.