

Endmember Detection using the Dirichlet Process

Alina Zare and Paul D. Gader

Department of Computer and Information Science and Engineering, University of Florida
azare@cise.ufl.edu, pgader@cise.ufl.edu

Abstract

An endmember detection algorithm for hyperspectral imagery using the Dirichlet process to determine the number of endmembers in a hyperspectral image is described. This algorithm provides an estimate of endmember spectra, proportion maps, and the number of endmembers needed for a scene. Updates to the proportion vector for a pixel are sampled using the Dirichlet process. As opposed to previous methods that prune unnecessary endmembers, the proposed algorithm is initialized with one endmember and new endmembers are added through sampling as needed. Results are shown on a two-dimensional dataset and a simulated dataset using endmembers selected from an AVIRIS hyperspectral image.

1. Introduction

A hyperspectral image is a three-dimensional data-cube with one spectral and two spatial dimensions. The spectral dimension corresponds to wavelengths in which radiance is measured for each pixel [4]. Each pixel in a hyperspectral image is a spectral vector of radiance values. The purpose of endmember detection in hyperspectral imagery is to determine the spectrally pure signatures (patterns) in a hyperspectral image that can be used to represent all the pixels in the image via a convex, linear model. Although similar to clustering, it differs in the sense that the “prototypes” (endmembers) represent vertices of a simplex that surround the pixels (at least partially) in D dimensional space, where D is the data dimensionality. Several algorithms exist for endmember detection, however, few simultaneously determine the number of endmembers [10].

The proposed algorithm provides a method of determining the number of endmembers while simultaneously estimating endmember spectra and proportion maps for an image. The number of endmembers is determined using the Dirichlet process. The algorithm provides a novel application of the Dirichlet process where, rather than determining means and variances of standard distributions, the algorithm deter-

mines vertices of a simplex which surround the pixels and the coefficients for a convex combination to describe each pixel in terms of the vertices (endmembers). This algorithm is initialized with a small number of endmembers and more endmembers are incrementally added as needed.

In Section 2, we provide a review of the endmember detection problem and the linear mixing model. Section 2 reviews the Dirichlet Process Mixture Model. The proposed algorithm is described in Section 4. Section 5 includes experiments and results followed by conclusions and future work in Section 6.

2. Linear Mixing Model

The linear mixing model for hyperspectral imagery assumes that every pixel in a scene is a convex combination of the endmembers in the scene [4, 2].

$$\mathbf{x}_i = \sum_{k=1}^M p_{ik} \mathbf{e}_k + \epsilon_i \quad i = 1, \dots, N \quad (1)$$

where N is the number of pixels, M is the number of endmembers, ϵ_i is an error term, p_{ik} is the proportion of endmember k in pixel i , and \mathbf{e}_k is the k^{th} endmember [4]. The proportions satisfy the following constraints.

$$p_{ik} \geq 0 \quad k = 1, \dots, M \quad \sum_{k=1}^M p_{ik} = 1 \quad (2)$$

Often, spectral unmixing is performed to determine the proportion values, p_{ik} , in addition to determining the spectral signatures of the endmembers, \mathbf{e}_k [4].

3. The Dirichlet Process Mixture Model

The Dirichlet Process has been previously applied to clustering problems and determining the number of clusters for a dataset [5, 8, 9, 6]. A brief description of the Dirichlet Process Mixture Model (DPMM) and Gibbs sampling for clustering is provided.

The DPMM applies a Dirichlet process prior to the mixing proportions in a mixing model. This extension allows for a countably infinite number of mixture components [3]. Consider N pixels, $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, each of which are assumed to have been independently generated by a single process, $f_i(\cdot, \phi_i)$ where ϕ_i is the parameter set that defines the process generating \mathbf{x}_i . Under the DPMM, ϕ_i is generated by some unknown distribution G [9]. Then, G is distributed according to the Dirichlet process, $\mathcal{D}(\alpha G_0)$ where G_0 is the base distribution and α is the concentration parameter [3]. The complete model is expressed as: [6, 3]

$$\begin{aligned}\mathbf{x}_i &\sim f(\cdot|\phi_i) \\ \phi_i &\sim G \\ G &\sim \mathcal{D}(\alpha G_0)\end{aligned}\quad (3)$$

. Under this model, the N ϕ_i values generated from G are members of a set of $M \leq N$ distinct values denoted as $\Theta = \{\theta_1, \dots, \theta_M\}$ corresponding to the parameters for each component. In other words, several pixels can be generated from the same mixture component [9].

To simplify the model, G can be integrated out to express the prior of each ϕ_i in terms of the base distribution G_0 and all other parameter sets [6, 3, 9].

$$\phi_i|\phi_{-i} \sim \frac{1}{\alpha + N - 1} \sum_{j=1, i \neq j}^N \delta(\phi_j) + \frac{\alpha}{\alpha + N - 1} G_0 \quad (4)$$

where ϕ_{-i} is the set of parameters for all pixels other than i , N is the number of pixels, $\delta(\phi_i)$ is the distribution over parameter sets with a zero value at all points other than ϕ_i and G_0 is the prior distribution for the parameters.

By introducing class labels, c_i , for each pixel, the probability of a particular class label given all other class labels can be computed.

$$\begin{aligned}p(c_i = c_j \text{ for } i \neq j | \alpha, \mathbf{c}_{-i}) &= \frac{n_{-i,j}}{\alpha + N - 1} \\ p(c_i \neq c_j \forall i \neq j | \alpha, \mathbf{c}_{-i}) &= \frac{\alpha}{\alpha + N - 1}\end{aligned}\quad (5)$$

where c_{-i} is the set of class labels for all pixels other than x_i , c_i is the class label for pixel x_i , α is the concentration parameter, N is the number of pixels, and $n_{-i,j}$ is the number of pixels excluding x_i with label j [6, 3, 8].

As described in [6] and [8], the likelihood of a pixel given the parameters can be combined with (5) to define the conditional probabilities for a class label. These conditional probabilities can be used in a Gibbs sampler

to sample the class labels for each pixel.

$$\begin{aligned}P(c_i = c_j \text{ for some } j \neq i | \mathbf{c}_{-i}, \mathbf{x}_i, \theta) &= \\ C \frac{n_{-i,j}}{\alpha + N - 1} f(\mathbf{x}_i | \theta_{c_j}) \\ P(c_i \neq c_j \forall j \neq i | \mathbf{c}_{-i}, \mathbf{x}_i, \theta) &= \\ C \frac{\alpha}{\alpha + N - 1} \int f(\mathbf{x}_i | \theta) G_0(\theta) d\theta\end{aligned}\quad (6)$$

where C is a normalizing constant, $f(\mathbf{x}_i | \theta_k)$ is the likelihood of pixel \mathbf{x}_i given parameter set θ_k .

If G_0 is a conjugate prior to the likelihood distributions, $f(\cdot | \theta)$, then the integral in (6) can be analytically computed and the parameter sets can also be integrated over in the conditionals in (6) [6] [3].

4. Dirichlet Process Endmember Detection

The proposed method determines the endmembers which surround the pixels and the proportion of each endmember in every pixel. The Dirichlet process is applied to determine the number of endmembers needed. This method differs from the DPMM since each pixel has an influence on every endmember whereas the DPMM partitions the pixels causing them to only influence the distribution parameters from which they are assumed to be generated. Furthermore, the proposed method does not only determine distribution parameters which correspond to a set of pixels like the DPMM, instead, the algorithm determines a unique proportion vector for each pixel. Following the linear mixing model (1), the likelihood for a given pixel can be defined in terms of the corresponding proportion vector and set of endmembers.

$$\begin{aligned}p(\mathbf{x}_i | \mathbf{p}_i, \mathbf{E}, \sigma_X) &= \\ \frac{1}{(2\pi\sigma_X^2)^{\frac{D}{2}}} \exp \left\{ -\frac{1}{2\sigma_X^2} (\mathbf{x}_i - \mathbf{p}_i \mathbf{E})^T (\mathbf{x}_i - \mathbf{p}_i \mathbf{E}) \right\}\end{aligned}\quad (7)$$

where \mathbf{p}_i is the proportion vector for the i th pixel, \mathbf{E} is matrix containing all endmember values, D is the dimensionality of the pixel, and σ_X is the variance.

As done in [2], encouraging the endmembers to have a tight fit around the data set can be done by using a sum of squared distances term.

$$\begin{aligned}p(\mathbf{E} | \sigma_E) &= \frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}} \\ \exp \left\{ \frac{-1}{2\sigma_E^2} \sum_{k=1}^{M-1} \sum_{l=k+1}^M (\mathbf{e}_k - \mathbf{e}_l)^T (\mathbf{e}_k - \mathbf{e}_l) \right\}\end{aligned}\quad (8)$$

where D is the dimensionality of the data, \mathbf{E} is the matrix of endmembers, \mathbf{e}_k is the k th endmember, and M is the number of endmembers.

The proposed algorithm uses the Dirichlet Process as a method to update proportion values for each endmember and incrementally add endmembers as needed. The proportion update for each endmember is done iteratively by incrementally increasing the weight of elements in the proportion vector. Given an initial proportion vector \mathbf{p}_i for pixel \mathbf{x}_i , assign δ_i to the following.

$$\delta_i = \frac{(\mathbf{x}_i - \mathbf{p}_i \mathbf{E})^T (\mathbf{x}_i - \mathbf{p}_i \mathbf{E})}{\beta} \quad (9)$$

where \mathbf{E} is the set of endmembers and β is a step-size parameter. Using δ_i , a set of potential proportion vectors, $\{\mathbf{p}_i^1, \mathbf{p}_i^2, \dots, \mathbf{p}_i^{M+1}\}$, updates is computed.

$$\mathbf{p}_i^j = \begin{cases} \frac{1}{1+\delta_i} [p_{i1}, \dots, p_{ij} + \delta_i, \dots, p_{iM}] & j \leq M \\ \frac{1}{1+\delta_i} [p_{i1}, \dots, p_{iM}, \delta_i] & j = M+1 \end{cases} \quad (10)$$

The probability of selecting an update is computed using a Dirichlet Process prior and the likelihood of the datapoint given the updated proportion vector and endmember values. If the update increases the error, the probability is set to zero.

$$p(\mathbf{p}_i^j | \mathbf{x}_i, \mathbf{P}, \mathbf{E}, \sigma_X, \sigma_E) = \begin{cases} C \frac{m_j}{\alpha + N - 1} F_{1:M}(\mathbf{p}_i^j) & j \leq M \\ C \frac{\alpha}{\alpha + N - 1} F_{M+1}(\mathbf{p}_i^{M+1}) & j = M+1 \end{cases} \quad (11)$$

where

$$F_{1:M}(\mathbf{p}_i^j) = \begin{cases} p(\mathbf{x}_i | \mathbf{p}_i^j, \mathbf{E}, \sigma_x) p(\mathbf{E} | \sigma_E) & \text{if } \left\| (\mathbf{x}_i - \mathbf{p}_i^j \mathbf{E}) \right\| \leq \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

$$F_{M+1}(\mathbf{p}_i^{M+1}) = \int p(\mathbf{x}_i | \mathbf{p}_i, \mathbf{E}, \mathbf{e}_{M+1}, \sigma_X) p(\mathbf{E} | \sigma_E) p(\mathbf{e}_{M+1} | \mathbf{E}, \sigma_E) d\mathbf{e}_{M+1} \quad (13)$$

$m_j = \sum_{i=1}^N p_{ij}$, C is a normalizing constant, N is the number of pixels, and σ_X and σ_E are the variances for the data likelihood and the prior on the endmembers, respectively, M is the number of endmembers with associated proportion values greater than zero, and \mathbf{P} is the matrix of proportion values with p_{ij} the proportion value for the i th pixel of the j th endmember.

The integral in (13) can be computed analytically with

$$p(\mathbf{e}_{M+1} | \mathbf{E}, \sigma_E) = \frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}} \exp \left\{ \frac{-1}{2\sigma_E^2} \sum_{k=1}^M (\mathbf{e}_k - \mathbf{e}_{M+1})^T (\mathbf{e}_k - \mathbf{e}_{M+1}) \right\} \quad (14)$$

Updates to a proportion vector are performed by sampling an updated proportion vector using the probabilities defined by (11). However, prior to considering a new endmember, the proportion vector given only the current endmembers should be estimated. This prevents adding unnecessary endmembers when the current endmembers can adequately describe the datapoint. Therefore, estimating a proportion vector for a datapoint occurs in two stages. The first stage estimates the endmembers without considering the addition of a new endmember. This step is performed by sampling updated proportion vectors from (11) with α set to zero. After many iterations, the proportion vector is updating once with a non-zero value for α .

After updating a single proportion vector, endmember values are updated. Endmember values are determined by minimizing the likelihood of the dataset given the endmembers and the proportions. This is done by minimizing $p(\mathbf{x}_i | \mathbf{p}_i, \mathbf{E}, \sigma_X) p(\mathbf{E} | \sigma_E)$ with respect to \mathbf{E} . The solution to this minimization is similar to the endmember updates in [2] and [10].

$$\mathbf{E}_j = \left(\mathbf{P}^T \mathbf{P} + \frac{\sigma_X^2}{\sigma_E^2} (M \mathbf{I}_M - \mathbf{1} \mathbf{1}^T) \right)^{-1} \mathbf{P}^T \mathbf{X}_j \quad (15)$$

where \mathbf{E}_j is the vector of all endmember values in the j th dimension, M is the number of endmembers, \mathbf{P} is the matrix of proportion values, \mathbf{I}_M is the identity matrix with M rows, and $\mathbf{1}$ is the M -length vector of ones.

This algorithm proceeds in an online manner, in which a single pixel's proportion vector is updated followed by an update of the endmember values. A simple initialization for \mathbf{E} is to randomly select a single datapoint from the dataset. In this case, each datapoint will only have a single proportion value of 1.

5. Experiments and Results

Initially, the algorithm was tested on a two-dimensional data set generated from 3 endmembers. The results in Figure 1 show the endmembers found using two different parameter sets for the algorithm. The results in Figure 1 (a) were found using $\beta = 5$, $\alpha = 0.01$, $\sigma_X = 1$ and $\sigma_E = 3$. For Figure 1 (b) the value for σ_E was increased to 5 and the remaining parameters were held constant. As can be seen,

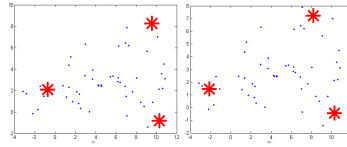


Figure 1. Results found on 2D Data. Endmember locations are starred.

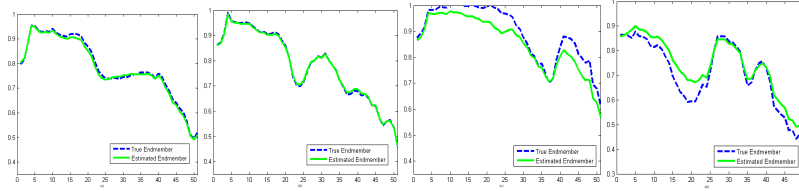


Figure 2. Results on Simulated Hyperspectral Data

the correct number of endmembers were found on both runs. Also, the endmembers generated provide a tight fit around the test set. The algorithm was run on a simulated dataset generated using four endmembers selected from the AVIRIS “Scene 4” Cuprite, Nevada dataset [1]. The dataset was generated based on the linear mixing model (1) using normalized versions of the selected endmembers. The algorithm was run 8000 iterations with $\alpha = 5000$, $\sigma_E = \sigma_X = 1$, $\beta = 5$ and 100 iterations per proportion vector update with α temporarily set to 0. Using this parameter set, the algorithm found 4 endmembers. Figure 2 shows a comparison of the endmembers found using the Dirichlet process method and the true endmembers used to generate the data set.

6. Conclusions and Future Work

The proposed algorithm estimates endmember spectra, proportion maps, and the number of endmembers for an image scene using the Dirichlet process. This method differs previous applications of the Dirichlet process for clustering since endmembers, unlike clustering prototypes, surround the image pixels. In this method, every pixel has a unique proportion vector that is determined using the Dirichlet process.

Future work includes running the proposed algorithm on full image data and a comparing results with existing endmember detection algorithms in the literature. Also, investigations into methods to determine how to set parameter values, α , σ_X and σ_E , should be conducted.

References

- [1] AVIRIS free standard data products. [Online] Available: <http://aviris.jpl.nasa.gov/html/aviris.freedata.html>,

- Sept. 2004.
- [2] M. Berman, H. Kiiveri, R. Lagerstrom, A. Ernst, R. Donne, and J. F. Huntington. ICE: A statistical approach to identifying endmembers in hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 42:2085 – 2095, Oct. 2004.
- [3] S. Jain and R. M. Neal. A split-merge Markov Chain Monte Carlo procedure for the Dirichlet process mixture model. Technical Report 2003, University of Toronto, Toronto, ON, Canada, Jul. 2000.
- [4] N. Keshava and J. F. Mustard. Spectral unmixing. *IEEE Signal Processing Magazine*, 19:44 – 57, 2002.
- [5] R. M. Neal. Bayesian mixture modeling by Monte Carlo simulation. Technical Report CRG-TR-91-2, University of Toronto, Toronto, ON, Canada, Jun. 1991.
- [6] R. M. Neal. Markov Chain sampling methods for Dirichlet process mixture models. Technical Report 9815, University of Toronto, Toronto, ON, Canada, Sep. 1998.
- [7] A. Ranganathan. The Dirichlet process mixture (DPM) model. [Online] Available: <http://www.cc.gatech.edu/~anath/docs/dirichlet.pdf>, Oct. 2006.
- [8] C. Rasmussen. The infinite Gaussian mixture model. In S. A. Solla, T. K. Leen, and K. R. Muller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 554–560. MIT Press, 2000.
- [9] M. West, P. Muller, and M. D. Escobar. Hierarchical priors and mixture models with application in regression and density estimation. In P. R. Freeman and A. F. M. Smith, editors, *Aspects of Uncertainty*, pages 363–386. John Wiley, 1994.
- [10] A. Zare and P. Gader. Sparsity promoting iterated constrained endmember detection for hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 4(3):446 – 450, July 2007.