Spectral-Spatial Classification of Hyperspectral Images Using Approximate Sparse Multinomial Logistic Regression

KORAY KAYABOL *
Dept. of Electronics Engineering
Gebze Technical University
Kocaeli
TURKEY
koray.kayabol@gtu.edu.tr

Abstract: We propose the sparse multinomial logistic regression (SMLR) model for spectral-spatial classification of hyperspectral images. In the proposed method, the parameters of SMLR are iteratively estimated from log-posterior by using Laplace approximation. The proposed update rule provides a faster convergence compared to the state-of the-art methods used for SMLR parameter estimation. The estimated parameters are used for spectral-spatial classification of hyperspectral images using a spatial prior. The experimental results on real hyperspectral images show that the classification accuracy of proposed method is also better than those of state-of-the art methods.

Key–Words: Sparse multinomial logistic regression, softmax, hyperspectral images, spatial-spectral classification

1 Introduction

Hyperspectral image classification is a challenging problem due to high dimensionality and spatial correlation. Several deterministic and probabilistic classification methods are proposed in the literature. Multinomial logistic regression (MLR) and its sparse version is a probabilistic state-of-art-method used in hyperspectral image classification. MLR is known as softmax in machine learning and neural network literature [4]. It is also used in recently emerged deep learning studies [1]. In this study, we use SMLR for spectral-spatial classification of hyperspectral images along with a new learning rule that improves the convergence speed and classification accuracy.

Krishnapuram et al [11] propose SMLR and related parameter estimation algorithms which are obtained by using Taylor series expansion and a lower bound for Hessian matrix proposed in [5]. In [11] Laplace prior is used for the parameters of the classifier. SMLR is firstly applied to hyperspectral images in [6], [7]. In those studies, a faster version of the algorithm in [11] is proposed using the block Gauss-Siedel method [13]. In [12], a less computationally complex algorithm called LORSAL [3] is proposed for parameter estimation in MLR.

In this study, we use a new parameter estimation algorithm for SMLR [8]. Since parameter estimation in SMLR is nonlinear problem without closed-form solution, we resort to an iterative algorithm. The algorithm is based on two approximations 1) 2nd order Taylor series expansion of the log-posterior of the parameters and 2) an approximate Hessian matrix [5]. We also use a spatial model as a prior to SMLR model to achieve contextual classification. The spatial model is based on spatially varying mixture model proposed in [10], [9].

The organization of the paper is as follows. Section 2 introduces the proposed model. Section 3 gives the details of the SMLR parameter estimation algorithm. Section 4 presents the spatial model and contextual classification algorithm. The experimental results are reported in Section 5. Section 6 summarizes the conclusion.

2 Proposed Model

A pixel is denoted by the vector $\mathbf{s}_n$ of length L in a hyperspectral image which has N pixels and L spectral bands. Each element of a pixel vector comes from a spectral band, therefore a hyperspectral image can be considered as a collection of L different images.

In this study, the hyperspectral image is modeled as a mixture of multinomial logistic regression models, therefore each pixel is assumed to be generated from a different multinomial distribution. In addition to spectral modeling, an spatial model is used in order

*This work is supported by Scientific and Technological Research Council of Turkey (TUBITAK) under Project No. 114E535.

to take advantage of pixel neighborhoods for classification.

Assuming that there are K number of land cover classes in the hyperspectral image, we define a K-dimensional label vector $\mathbf{z}_n \in \{0,1\}^K$ for each pixel with the property that $\sum_{k=1}^{K} z_{n,k} = 1$. The joint density of $\mathbf{s}_n$ and $\mathbf{z}_n$ for $n = 1, 2, \ldots, N$ is given by

$$p(\mathbf{s}_{1:N}, \mathbf{z}_{1:N}|\boldsymbol{\omega}_{1:K}, \beta) = \left[\prod_{n=1}^{N}\prod_{k=1}^{K} p(\mathbf{s}_n|\boldsymbol{\omega}_k)^{z_{n,k}}\right] p(\mathbf{z}_{1:N}|\beta) \quad (1)$$

where the class density $p(\mathbf{s}_n|\boldsymbol{\omega}_k)$ and the spatial prior density of the class labels $p(\mathbf{z}_{1:N}|\beta)$ are explained in the following sections. In (1), $\boldsymbol{\omega}_k$ is the vector of regression parameters.

3 Multinomial Logistic Regression

We assume that a pixel vector is generated from one of K multinomial distributions each of which represents a class. Probability of a pixel given a class label can be written as follows:

$$p(\mathbf{s}_n|z_{n,k} = 1, \boldsymbol{\omega}_{1:K}) = \frac{e^{\boldsymbol{\omega}_k^T \mathbf{s}_n}}{\sum_{j=1}^{K} e^{\boldsymbol{\omega}_j^T \mathbf{s}_n}} \quad (2)$$

where $z_{n,k}$ is the binary label for kth class. Using (2), the conditional probability of hyperspectral vector $\mathbf{s}_n$ given the class label vector $\mathbf{z}_n$ and the regression parameters $\boldsymbol{\omega}_{1:K}$ is a multinomial distribution such as

$$p(\mathbf{s}_n|\mathbf{z}_n, \boldsymbol{\omega}_{1:K}) = \prod_{k=1}^{K}\left(\frac{e^{\boldsymbol{\omega}_k^T \mathbf{s}_n}}{\sum_{j=1}^{K} e^{\boldsymbol{\omega}_j^T \mathbf{s}_n}}\right)^{z_{n,k}} \quad (3)$$

In order to obtain sparse regression coefficients, we may define some sparse prior distributions. Widely used sparse prior in Bayesian estimation is Laplace prior given as follow:

$$p(\boldsymbol{\omega}_{1:K}|\lambda) = \prod_{k=1}^{K} \frac{\lambda}{2} e^{-\lambda||\boldsymbol{\omega}_k||_1} \quad (4)$$

where $||\boldsymbol{\omega}_k||_1 = \sum_{l=1}^{L} |\omega_{k,l}|$ denotes the l_1 norm.

To find the maximum-a-posteriori estimate of the regression coefficient $\boldsymbol{\omega}_k$, let consider the log-posterior obtained by using (3) and (4) as follows:

$$L(\boldsymbol{\omega}) = \sum_{n=1}^{N}\left[\sum_{k=1}^{K} z_{n,k}\boldsymbol{\omega}_k^T \mathbf{s}_n - \log\sum_{j=1}^{K}\exp\left(\boldsymbol{\omega}_j^T \mathbf{s}_n\right)\right] - \lambda\sum_{k=1}^{K}||\boldsymbol{\omega}_k||_1 \quad (5)$$

where $\boldsymbol{\omega} = [\boldsymbol{\omega}_1^T, \ldots, \boldsymbol{\omega}_K^T]^T$. The second order Taylor series expansion of $L(\boldsymbol{\omega})$ around $\boldsymbol{\omega}^{(t)}$ is

$$L(\boldsymbol{\omega}) - L(\boldsymbol{\omega}^{(t)}) = (\boldsymbol{\omega} - \boldsymbol{\omega}^{(t)})^T \mathbf{g}_L(\boldsymbol{\omega}^{(t)}) + \frac{1}{2}(\boldsymbol{\omega} - \boldsymbol{\omega}^{(t)})^T \mathbf{H}_L(\boldsymbol{\omega}^{(t)})(\boldsymbol{\omega} - \boldsymbol{\omega}^{(t)}) \quad (6)$$

where $\mathbf{H}_L(\boldsymbol{\omega}^{(t)})$ is the Hessian matrix and $g_L(\boldsymbol{\omega}^{(t)})$ is the gradient vector. We can decompose $\mathbf{H}_L(\boldsymbol{\omega}^{(t)})$ in two parts, i.e. that $\mathbf{H}_L(\boldsymbol{\omega}^{(t)}) = \mathbf{H}_\ell(\boldsymbol{\omega}^{(t)}) + \lambda\Lambda(\boldsymbol{\omega}^{(t)})$. The first and the second terms are the Hessian matrices obtained from the log-likelihood and log-prior, respectively. Maximizing the right-hand side of (6) yields the following lower bound iterate

$$\boldsymbol{\omega}^{(t+1)} = \boldsymbol{\omega}^{(t)} - (\mathbf{H}_\ell(\boldsymbol{\omega}^{(t)}) + \lambda\Lambda(\boldsymbol{\omega}^{(t)}))^{-1} g_L(\boldsymbol{\omega}^{(t)}) \quad (7)$$

According to Theorem 2.1 in [5], the Hessian matrix $\mathbf{H}_\ell(\boldsymbol{\omega}^{(t)})$ can be lower bounded such as

$$\mathbf{H}_L(\boldsymbol{\omega}) = \mathbf{H}_\ell(\boldsymbol{\omega}) + \lambda\Lambda(\boldsymbol{\omega}) \geq \mathbf{B} + \lambda\Lambda(\boldsymbol{\omega}) \quad (8)$$

Rather than calculating the Hessian matrix $\mathbf{H}_\ell(\boldsymbol{\omega})$ in each iterations, we can use its constant approximation. In this case, the iterations becomes as follow

$$\boldsymbol{\omega}^{(t+1)} = \boldsymbol{\omega}^{(t)} - (\mathbf{B} + \lambda\Lambda(\boldsymbol{\omega}^{(t)}))^{-1} g_L(\boldsymbol{\omega}^{(t)}) \quad (9)$$

The update equation in (9) is different from the one given in [11], because we apply the Taylor approximation to log-posterior rather than log-likelihood. To reduce the computational cost due to large scale matrix inversion in (9), component-wise update rule can be used. In each iteration one of the regression coefficients can be updated. Update rule for the kth regression vector is found as follows

$$\begin{aligned}\boldsymbol{\omega}_k^{(t+1)} &= \boldsymbol{\omega}_k^{(t)} - [\mathbf{B}_{kk} + \lambda\Lambda(\boldsymbol{\omega}_k^{(t)})]^{-1}[g_k(\boldsymbol{\omega}_k^{(t)}) \\ &+ \frac{1}{2}\sum_{j\neq k}(\mathbf{B}_{kj} + \lambda\Lambda(\boldsymbol{\omega}_j^{(t)}))\mathbf{e}_j \\ &+ \lambda sign(\boldsymbol{\omega}_k^{(t)})]\end{aligned} \quad (10)$$

where

$$\mathbf{e}_j = \boldsymbol{\omega}_j^{(t)} - \boldsymbol{\omega}_j^{(t-1)} \quad (11)$$

Each block of the approximate Hessian matrix $\mathbf{B}$ is calculated as

$$\mathbf{B}_{kj} = -\frac{1}{2}(\delta_{kj} - 1/K)\mathbf{S}^T\mathbf{S} \quad (12)$$

where $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N]^T$ and δ_{kj} is Kronecker delta function. The gradient vector with respect to $\boldsymbol{\omega}_k$ is calculated as

$$g_k(\boldsymbol{\omega}_k^{(t)}) = \sum_{n=1}^{N} (z_{n,k} - \pi_{n,k}) \mathbf{s}_n \tag{13}$$

where

$$\pi_{n,k} = \frac{e^{\boldsymbol{\omega}_k^T \mathbf{s}_n}}{\sum_{j=1}^{K} e^{\boldsymbol{\omega}_j^T \mathbf{s}_n}} \tag{14}$$

We use the same Λ function proposed in [11] such as

$$\Lambda(\boldsymbol{\omega}_k) = diag\{|\omega_{k,1}|^{-1}, |\omega_{k,2}|^{-1}, \ldots, |\omega_{k,L}|^{-1}\} \tag{15}$$

4 Spatial Smoothing

For spatial smoothing of classification map, we use spatially varying mixture model proposed in [10], [9]. Spatial prior is given by

$$p(\mathbf{z}_{1:N}|\beta) = \frac{\prod_{k=1}^{K} \exp\left\{\beta \sum_{n=1}^{N} z_{n,k} \left(1 + \frac{1}{2} \sum_{m \in \tilde{n}} z_{m,k}\right)\right\}}{\mathcal{Z}(\beta)} \tag{16}$$

where $\mathcal{Z}(\beta)$ is the normalization term and $\tilde{n}$ denotes the set of pixels around the nth pixel.

After learning the parameters $\boldsymbol{\omega}_{1:K}$, we can perform the classification step by maximizing the posterior of the class labels $\mathbf{z}_n$, $n \in \mathcal{B}$ where $\mathcal{B}$ is the set of test data indices. Since the joint maximization of the posterior of the class labels is not possible, we resort to iterated conditional mode (ICM) algorithm [2]. From (3) and (16), we can write the conditional of $\mathbf{z}_n$ to be maximized as

$$\begin{aligned} p(\mathbf{z}_n|\mathbf{z}_{\overline{n}}, \mathbf{s}_{\mathcal{B}}, \hat{\theta}_{1:K}, \beta) &\propto p(\mathbf{s}_n|\mathbf{z}_n, \hat{\theta}_{1:K}) p(\mathbf{z}_n|\mathbf{z}_{\tilde{n}}, \beta) \\ &= \prod_{k=1}^{K} \left[\frac{e^{\boldsymbol{\omega}_k^T \mathbf{s}_n}}{\sum_{j=1}^{K} e^{\boldsymbol{\omega}_j^T \mathbf{s}_n}} \frac{e^{\beta v_{n,k}}}{\sum_{j=1}^{K} e^{\beta v_{n,j}}} \right]^{z_{n,k}} \end{aligned} \tag{17}$$

where $\overline{n} = \{1, 2, \ldots, N\} \backslash \{n\}$.

5 Experimental Results

For experiments, we use four well-known HSI data sets which are Indian Pines, Pavia Centre, Pavia University, and Salinas. Indian Pines data set is obtained by Airborne Visible Infrared Imaging Spectrometer (AVIRIS) over Northern Indiana on June 12, 1992. The data set contains a 145×145 pixels and 220 bands HSI, and a 16-class ground-truth map. We remove the 20 noisy bands and use 200 spectral bands in our experiments. Pavia Centre and Pavia University data sets are obtained by the ROSIS sensor over Pavia, Italy. Pavia Centre data set consists of an HSI that has 1096×715 pixels and 102 spectral bands, and Pavia University data set contains a 610×340 pixels and 103 spectral bands. Both Pavia Centre and Pavia University data sets have ground truth maps of 9 classes. Salinas data set is an HSI image of 224 spectral bands that was acquired by the AVIRIS sensor over Salinas Valley, California. It contains 512×217 pixels within 16 classes, and we remove the 20 noisy bands in our experiments.

We compare the performance of the proposed learning rule APSMLR with its two predecessors, namely component-wise SMLR (CWSMLR) [11] and LORSAL [3]. We use Markov random fields prior as a spatial model for CWSMLR and LORSAL.

We construct the training set by randomly choosing $N_k = 50$ pixels from each class, and use the rest of the pixels as test sets. Since some of the classes have small sample size, we use following rule to assign N_k i.e. $N_k = \min\{N_k, N_c/2\}$ where N_c is total number of the ground-truth pixels. The initial values of $\boldsymbol{\omega}_{1:K}$ are set to $\mathbf{1}/1000$. In order to completely remove the intervention of the training samples, we do not use the training samples at initialization of $\mathbf{z}_n$. Otherwise, training samples affect the results diffusely due to spatial smoothing model.

Table 1-4 lists the average overall accuracies (OAs) calculated using the results of 20 random runs of the algorithms. As seen from Table 1-4, proposed APSMLR algorithm gives better classification results than other two algorithms according to OA and κ measures. Considering the standard deviations calculated over 20 random runs, APSMLR algorithm yields better results as well. While the slowest algorithm is CWSMLR, LORSAL is the fastest one. Although LORSAL is more or less 5 times faster than APSMLR, its OA and κ values are worse than APSMLR.

Table 1: Average OAs and κ measures along with standard deviations and computation time for Indian Pines.

	OA	std.	κ	std.	time
CW-SMLR	55.96	$\pm$ 8.96	0.55	$\pm$ 0.0912	6.4276
LORSAL	55.92	$\pm$ 7.29	0.55	$\pm$ 0.0742	0.0854
APSMLR	75.92	$\pm$ 7.02	0.75	$\pm$ 0.0719	0.6339

As seen from Fig. 1, CWSMLR and LORSAL algorithms are not converged to a stable point after 100 iterations. However, APSMLR algorithm converges after a few iterations. According to our experiment,

Table 2: Average OAs and κ measures along with standard deviations and computation time for Pavia Centre.

	OA	std.	κ	std.	time
CW-SMLR	93.56	$\pm$ 3.53	0.93	$\pm$ 0.0253	1.2166
LORSAL	76.38	$\pm$ 8.77	0.76	$\pm$ 0.0882	0.0354
APSMLR	98.08	$\pm$ 0.34	0.98	$\pm$ 0.0035	0.1167

Table 3: Average OAs and κ measures along with standard deviations and computation time for Pavia University.

	OA	std.	κ	std.	time
CW-SMLR	65.96	$\pm$ 7.77	0.65	$\pm$ 0.0785	1.2142
LORSAL	59.91	$\pm$ 4.91	0.59	$\pm$ 0.0498	0.0333
APSMLR	76.91	$\pm$ 4.38	0.76	$\pm$ 0.0439	0.1020

Table 4: Average OAs and κ measures along with standard deviations and computation time for Salinas.

	OA	std.	κ	std.	time
CW-SMLR	70.40	$\pm$ 3.91	0.70	$\pm$ 0.0839	9.0545
LORSAL	66.67	$\pm$ 8.30	0.66	$\pm$ 0.0394	0.1018
APSMLR	93.72	$\pm$ 0.74	0.93	$\pm$ 0.0075	0.8127

5-10 iterations are adequate for APSMLR.

Fig. 2 shows the classification maps of Indian Pines data obtained by three algorithms along with the ground-truth.

6 Conclusion

We propose a new iterative algorithm for sparse MLR (or softmax) parameter estimation. The proposed algorithm converges to a solution faster than its predecessors. Its classification performance is better. In this study, we apply the algorithm to hyperspectral image classification problem but it can be used in any application area for classification.

Acknowledgements: The author would like to thank David Landgrebe and Paolo Gamba for providing hyperspectral data sets, and Jun Li and Jose M. Bioucas-Dias for providing their codes online.

References:

[1] Ian Goodfellow Yoshua Bengio and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016.

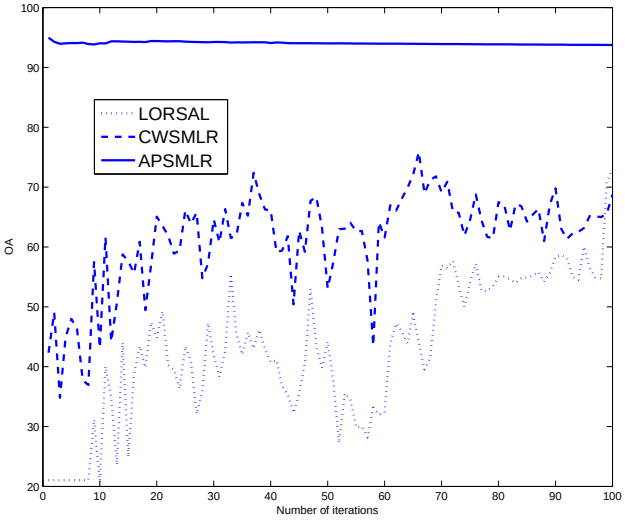


Figure 1: Iteration number versus overall accuracy (OA) for Salinas data.

[2] J. Besag. On the statistical analysis of dirty pictures. *J. R. Stat. Soc. B*, 48(3):259302, 1986.

[3] J. Bioucas-Dias and M. Figueiredo. Logistic regression via variable splitting and augmented Lagrangian tools. Technical report, Instituto Superior Tecnico, TULisbon, Lisbon, Portugal, August 2009.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

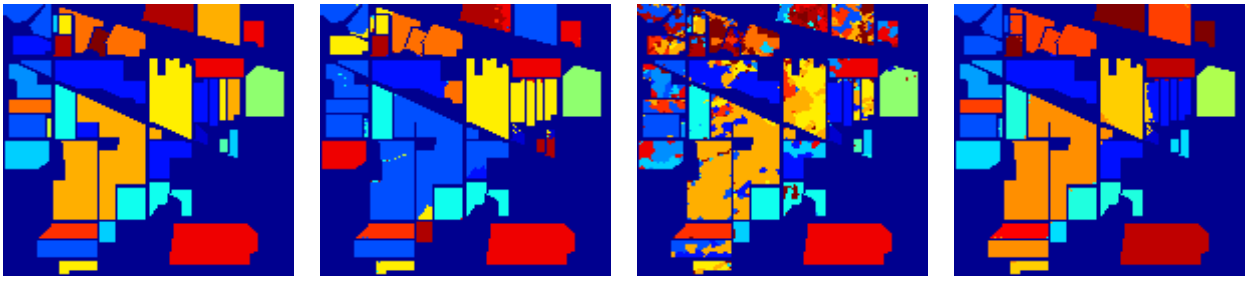
[5] D. Bohning. Multinomial logistic regression algorithm. *Annals of the Inst. of Statistical Math.*, 44:197–200, 1992.

[6] J. Borges, J. Bioucas-Dias, and A. Marcal. Fast sparse multinomial regression applied to hyperspectral data. In *Int. Conf. Image Analysis and Recognition, ICIAR*, Povoa de Varzim, Portugal, 2006.

[7] J. Borges, J. Bioucas-Dias, and A. Marcal. Bayesian hyperspectral image segmentation with discriminative class learning. *IEEE Trans. Geosci. Remote Sens.*, 49(6):2151–2164, 2011.

[8] K. Kayabol. Approximate sparse multinomial logistic regression for classification. *IEEE Trans. Pattern Anal. Machine Intell.*, Submitted, 2016.

[9] K. Kayabol and S. Kutluk. Bayesian classification of hyperspectral images using spatially-varying Gaussian mixture model. *Digital Signal Processing*, 59:106–114, 2016.



(a) Ground-truth (b) CWSMLR 68.80 (c) LORSAL 67.89 (d) APSMLR 84.41

Figure 2: Indian Pines ground-truth 2(a) and classification maps obtained by CWSMLR, LORSAL and APSMLR methods. The numbers under the maps represent the corresponding overall accuracies.

[10] K. Kayabol and J. Zerubia. Unsupervised amplitude and texture classification of SAR images with multinomial latent model. *IEEE Trans. Image Process.*, 22(2):561–572, 2013.

[11] B. Krishnapuram, L. Carin, M. A. T. Figueiredo, and A. J. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(6):957–968, 2016.

[12] J. Li, J. Bioucas-Dias, and A. Plaza. Hyperspectral image segmentation using a new Bayesian approach with active learning. *IEEE Trans. Geosci. Remote Sens.*, 49(10):3947–3960, 2011.

[13] A. Quarteroni, R. Sacco, and F.Saleri. *Numerical Mathematics*. Springer-Verlag, New York, 2000.