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

Sezer Kutluk[1] Koray Kayabol[2] Aydin Akan[3]

[1]Electrical-Electronics Engineering Department, Istanbul University-Cerrahpasa

[2]Electronics Engineering Department, Gebze Technical University

[3]Biomedical Engineering Department, Izmir Katip Celebi University

EUSIPCO 2019

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

Outline

1. Motivation
   - Hyperspectral Image Classification
   - Convolutional Neural Networks for HSI Classification
2. CNN
3. ASMLR
   - Model
   - Training
   - Spatial Model
4. CNN+ASMLR
5. Experiments
   - Data
   - Network Architecture
   - Results
6. Conclusion

Motivation

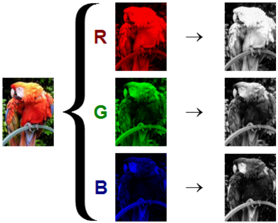
We propose a technique for training CNNs with Approximate Sparse Multinomial Logistic Regression, and using this model for classifying hyperspectral images.

The proposed model

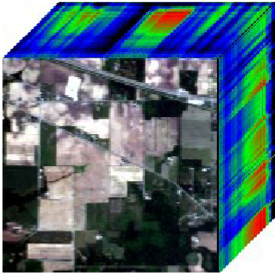
- extracts features with 1D convolutional layers
- incorporates a second order training algorithm for *the classification layer*
  - automatic step-size calculation
  - approximate calculation of Hessian
  - sparse priors on regression coefficients
- is trained *end-to-end*
- also includes a *spatial smoothing* method for hyperspectral images.

Motivation

Hyperspectral Image Classification



RGB image



HSI

- Tens or hundreds of spectral bands
- Approximately continuous spectrum
- Landcover classification: Assign each pixel a label such as crop type, asphalt, etc.

Motivation

Convolutional Neural Networks for HSI Classification

- Spectral / spatial modeling
  - 1-D, 2-D, and 3-D CNNs
  - Two channel CNN
  - Superpixels + CNN + CRF as RNN
  - MRF
- Small sample size problem
  - Sub space learning before CNN
  - Band selection
  - Data augmentation
- Different approaches
  - Spectral vector folding + 2D CNN
  - Deconvolution layers
  - Extreme learning machine
  - Transfer learning

Convolutional Neural Network

- Feature extraction layers + fully connected layer + softmax

$$p(\mathbf{z}_n|\mathbf{s}_n, \omega_{1:K}) = \prod_{k=1}^{K} \left(\frac{e^{\omega_k^T \phi(\mathbf{s}_n)}}{\sum_{j=1}^{K} e^{\omega_j^T \phi(\mathbf{s}_n)}} \right)^{z_{n,k}} \quad (1)$$

$\mathbf{z}_n$: 1-of-K coded label vector
$\mathbf{s_n}$: Spectral signature of n^{th} pixel
$\phi() = \phi_L(\phi_{L-1}(...\phi_1()...)$: Output of the *feature extraction layers*

Cross-entropy loss:

$$\mathcal{L}_{CE}(\omega) = -\sum_{n=1}^{N}\sum_{k=1}^{K} z_{n,k}\omega_k^T\phi(\mathbf{s}_n) + \ln \sum_{j=1}^{K} e^{\omega_j^T\phi(\mathbf{s}_n)} \quad (2)$$

Update rule:

$$\omega^{(t+1)} = \omega^{(t)} - \eta \cdot \mathbf{g}_L(\omega^{(t)}) \quad (3)$$

Approximate Sparse Multinomial Logistic Regression

- ASMLR (Kayabol, 2019) is a generative model with multinomial priors on pixel labels, and sparse priors on parameters:

$$p(\mathbf{s}_{1:N}, \mathbf{z}_{1:N}, \omega_{1:K} | \beta, \lambda) \tag{4}$$

$$= \left(\prod_{n=1}^{N} p(\mathbf{s}_n | \mathbf{z}_n, \omega_{1:K}) \right) p(\omega_{1:K} | \lambda) p(\mathbf{z}_{1:N} | \beta) \tag{5}$$

$$= \left[\prod_{n=1}^{N} \prod_{k=1}^{K} \left(\frac{e^{\omega_k^T \mathbf{s}_n}}{\sum_{j=1}^{K} e^{\omega_j^T \mathbf{s}_n}} \right)^{z_{n,k}} \right] \left(\prod_{k=1}^{K} \frac{\lambda}{2} e^{-\lambda ||\omega_k||_1} \right) p(\mathbf{z}_{1:N} | \beta) \tag{6}$$

Approximate Sparse Multinomial Logistic Regression

- Posterior of regression coefficients given class labels:

$$\underbrace{p(\omega_{1:K}|\mathbf{s}_{1:N}, \mathbf{z}_{1:N}, \lambda)}_{\text{Posterior}} \propto \underbrace{p(\mathbf{s}_{1:N}|\mathbf{z}_{1:N}, \omega_{1:K})}_{\text{Likelihood}} \underbrace{p(\omega_{1:K}|\lambda)}_{\text{Prior}}$$

- Log-posterior:

$$\mathcal{L}(\omega) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{n,k} \omega_k^T \mathbf{s}_n - \ln \sum_{j=1}^{K} e^{\omega_j^T \mathbf{s}_n} - \lambda \sum_{k=1}^{K} ||\omega_k||_1$$

$$= -\underbrace{\mathcal{L}_{CE}(\omega)}_{\text{Cross-entropy loss}} - \lambda \sum_{k=1}^{K} ||\omega_k||_1$$

Approximate Sparse Multinomial Logistic Regression

Training

- Second order Taylor series expansion:

$$\mathcal{L}(\omega) - \mathcal{L}(\omega^{(t)}) = (\omega - \omega^{(t)})\mathbf{g}_L(\omega^{(t)}) \tag{7}$$

$$+\frac{1}{2}(\omega - \omega^{(t)})\mathbf{H}_L(\omega^{(t)})(\omega - \omega^{(t)}) \tag{8}$$

- We can write the Hessian as the sum of Hessians from likelihood and prior:

$$\mathbf{H}_L(\omega^{(t)}) = \mathbf{H}_l(\omega^{(t)}) + \lambda\Lambda(\omega^{(t)}) \tag{9}$$

- Update rule:

$$\omega^{(t+1)} = \omega^{(t)} - (\mathbf{H}_l(\omega^{(t)}) + \lambda\Lambda(\omega^{(t)}))^{-1}\mathbf{g}_L(\omega^{(t)}) \tag{10}$$

Approximate Sparse Multinomial Logistic Regression

Training

- Lower bound approximation (Böhning, 1992):

$$\mathbf{H}_L(\omega) = \mathbf{H}_l(\omega) + \lambda\Lambda(\omega) \geq \mathbf{B} + \lambda\Lambda(\omega) \tag{11}$$

With the lower bound, the update rule becomes:

$$\omega^{(t+1)} = \omega^{(t)} - (\mathbf{B} + \lambda\Lambda(\omega^{(t)}))^{-1}\mathbf{g}_L(\omega^{(t)}) \tag{12}$$

Approximate Sparse Multinomial Logistic Regression

Training

- Component-wise calculation:

$$\omega_k^{(t+1)} = \omega_k^{(t)} - [\mathbf{B}_{kk} + \lambda\Lambda(\omega_k^{(t)})]^{-1}[g_k(\omega_k^{(t)}) \tag{13}$$

$$+\frac{1}{2}\sum_{j\neq k}(\mathbf{B}_{kj} + \lambda\Lambda(\omega_j^{(t)})\mathbf{e}_j + \lambda\mathrm{sign}(\omega_k^{(t)})] \tag{14}$$

$$\mathbf{B}_{kj} = -\frac{1}{2}(\delta_{kj} - 1/K)\mathbf{S}^T\mathbf{S} \tag{15}$$

Approximate Sparse Multinomial Logistic Regression

Spatial Model

- Multinomial Autologistic Regression (Kayabol, 2013 & 2016)

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

- Classification of a new pixel:

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

We use Iterated Conditional Mode algorithm.

Proposed Model
CNN+ASMLR

- Training:
  - Forward pass
  - Loss calculation
  - Backpropagation
  - SGD for convolutional layers
  - ASMLR for the last layer
- Prediction:
  - Forward pass
  - Spatial smoothing with Multinomial Autologistic Regression using Iterated Conditional Mode

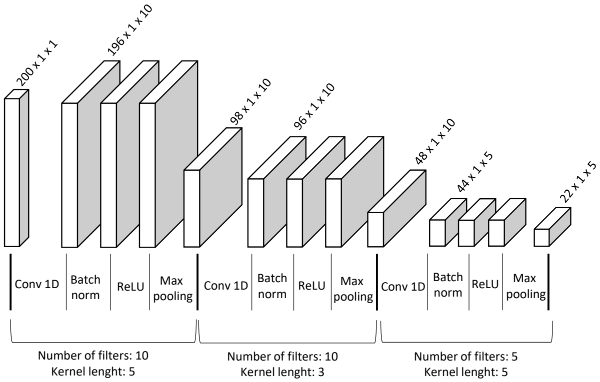
Experiments

- Datasets
  - Indian Pines
    - 145 x 145 pixels, 200 spectral bands, 16 classes
    - Training size: for each class, $\min(50, \text{pixels}/2)$
    - Training set size: 693 pixels
    - Test set size: 9556 pixels
  - Pavia University
    - 610 x 340 pixels, 103 spectral bands, 9 classes
    - Training size: for each class, $\min(100, \text{pixels}/2)$
    - Training set size: 900 pixels
    - Test set size: 41876 pixels
- Training pixels are selected randomly
- Average of 20 tests

Experiments

Network Architecture

- Indian Pines

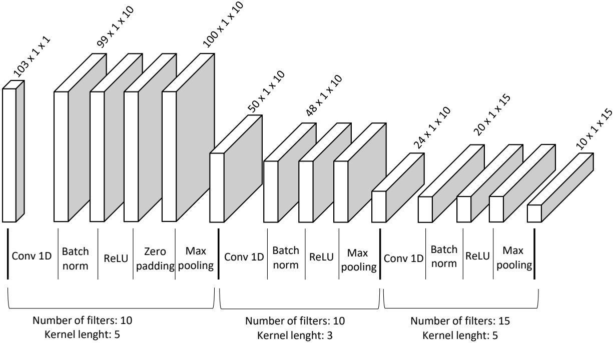


Feature length: 110

Experiments

Network Architecture

- Pavia University



Feature length: 150

Experiments

Results

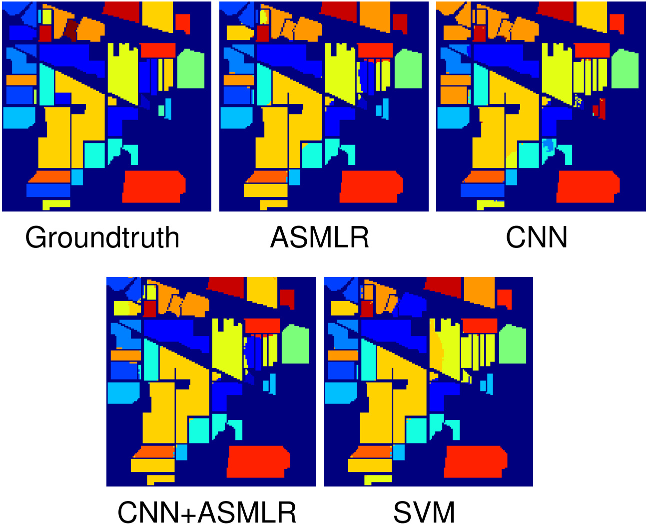
Indian Pines test results

Evaluation Metrics	**Methods**			
	ASMLR	***CNN***	***CNN+ASMLR***	***SVM***
Accuracy	0.84	0.83	0.89	0.78
Standard deviation	0.03	0.06	0.03	0.03

All models have the same spatial smoothing.

Experiments

Results



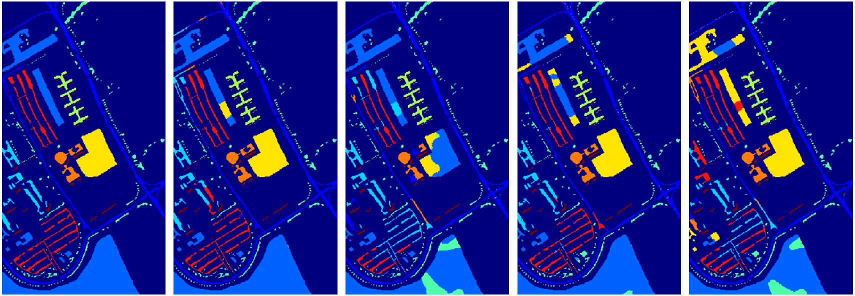
Pavia University test results

Evaluation Metrics	**Methods**			
	ASMLR	***CNN***	***CNN+ASMLR***	***SVM***
Accuracy	0.92	0.87	0.93	0.76
Standard deviation	0.02	0.11	0.05	0.02

All models have the same spatial smoothing.

Experiments

Results



Groundtruth ASMLR CNN CNN+ASMLR SVM

Conclusion

- CNN+ASMLR gives higher accuracy than ASMLR, CNN, and SVM
- Lower variance than CNN
- Future work
  - Mini-batch training
  - Proof of faster training (at least with fewer epochs)
  - Different datasets
  - Different network architectures

Thank you!

sezer.kutluk@gmail.com
koray.kayabol@gtu.edu.tr
aydin.akan@ikc.edu.tr