Spatially Adaptive Semi-supervised Learning with Gaussian Processes for Hyperspectral Data Analysis

Goo Jun and Joydeep Ghosh

Dept. of Electrical and Computer Engineering
The University of Texas at Austin

Oct 5, 2010

Land Cover Classification with Hyperspectral Data

Hyperspectral Imagery

- A classifier is trained with high-dimensional (> 100) features.
- Each land cover type is characterized by its spectral signature.

Challenges in hyperspectral data analysis

- Spectral characteristics of a class changes over space.
- Spatially adaptive model needs nearby data.
- Class labels are expensive.

GP-EM: *Semi-supervised Learning with Spatial Adaptation.*

Spatial Variations of Hyperspectral Data
Example - Botswana data

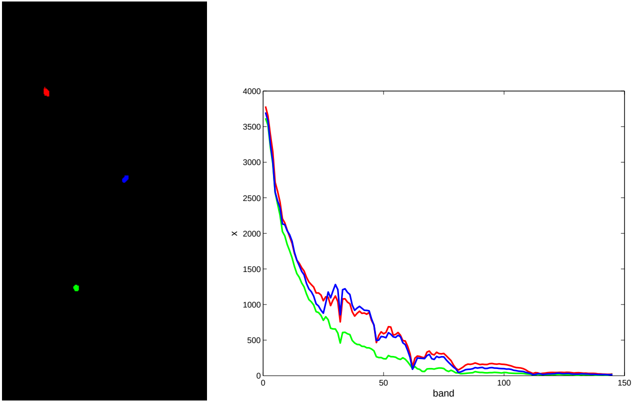
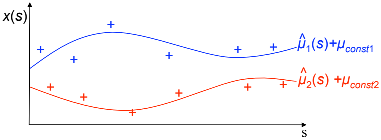
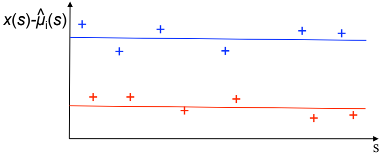


Figure: Spectral signatures of water class from different locations

GP-ML: Spatially Adaptive Classifier



(a) Data with spatial variation

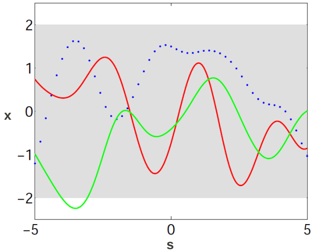


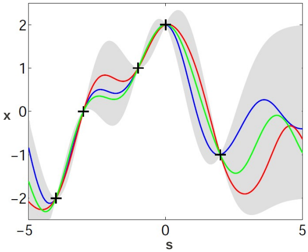
(b) Spatial variation removed

Spatially Adaptive MLC

$$p(\mathbf{x}|y = y_i; \mathbf{s}, \Theta) \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{i}(\mathbf{s}),\ \Sigma_i)$$

Gaussian Processes - Prior and Posterior



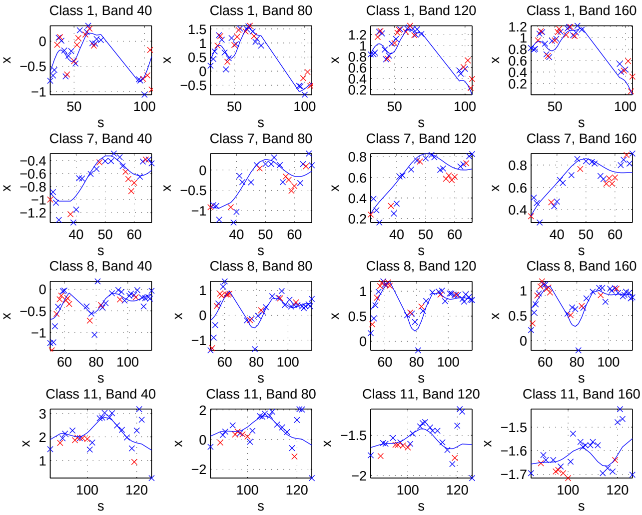


Predictive distribution:

$$p(x_*|\mathbf{s}_*, \mathbf{x}, S) = \mathcal{N}\Big(\overbrace{\mathbf{k}(\mathbf{s}_*, S)[K_{SS} + \sigma_\epsilon^2 I]^{-1}\mathbf{x}}^{mean}, \underbrace{k(\mathbf{s}_*, \mathbf{s}_*) + \sigma_\epsilon^2 - \mathbf{k}(\mathbf{s}_*, S)[K_{SS} + \sigma_\epsilon^2 I]^{-1}\mathbf{k}(S, \mathbf{s}_*)}_{variance} \Big)$$

[Figures from Rasmussen 2006]

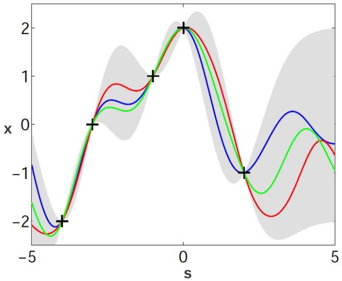
Example: Spatial Adaptation



Problem Setting

Limitations of GP-ML

- Class labels are expensive.
- GP is good for interpolation, but not for extrapolation.



Semi-supervised Approach

Mixture of Gaussians

$$p(\mathbf{x};\Theta) = \sum_{i=1}^{c} \alpha_i \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)\ , \qquad \sum_{i=1}^{c} \alpha_i = 1\ ,$$

Mixture of Spatially Adaptive Gaussians

$$p(\mathbf{x};\mathbf{s},\Theta) = \sum_{i=1}^{c} \alpha_i(\mathbf{s}) \mathcal{N}(\boldsymbol{\mu}_i(\mathbf{s}), \Sigma_i)\ , \qquad \sum_{i=1}^{c} \alpha_i(\mathbf{s}) = 1\ .$$

Semi-supervised Approach

Mixture of Gaussians

$$p(\mathbf{x};\Theta) = \sum_{i=1}^{c} \alpha_i \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i) \ , \qquad \sum_{i=1}^{c} \alpha_i = 1 \ ,$$

Mixture of Spatially Adaptive Gaussians

$$p(\mathbf{x};\mathbf{s},\Theta) = \sum_{i=1}^{c} \alpha_i(\mathbf{s}) \mathcal{N}(\boldsymbol{\mu}_i(\mathbf{s}), \Sigma_i) \ , \qquad \sum_{i=1}^{c} \alpha_i(\mathbf{s}) = 1 \ .$$

GP-EM Outline

Assume single component per class, and initialize each component using labeled samples.

E-step

Calculate membership variable, z^t

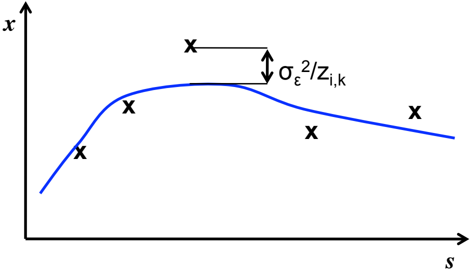
M-step

- Estimate spatially adaptive mean for each component, μ^{t+1}, by GP with soft assignments.
- Estimate spectral covariance matrix for each component, Σ^{t+1}
- Update *spatially correlated* membership variable, z^{t+1}, by another GP.

GP-EM, M-step (1)

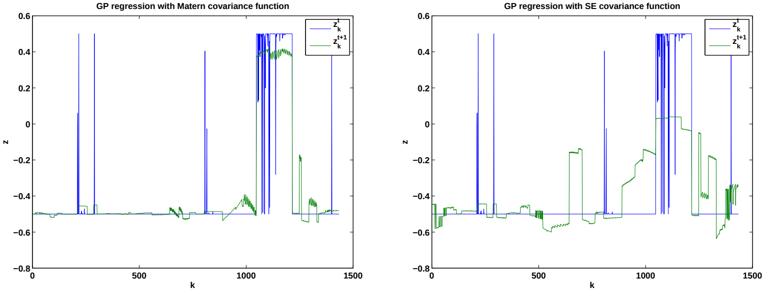
GP with Soft Membership

- Mixture of Gaussian Processes [Tresp, 2000]
- $[\sigma_{f_j}^2 K_{SS} + diag(\sigma_{\epsilon_j}^2 / z_{i,k}^t)]^{-1}$



GP-EM, M-step (2)

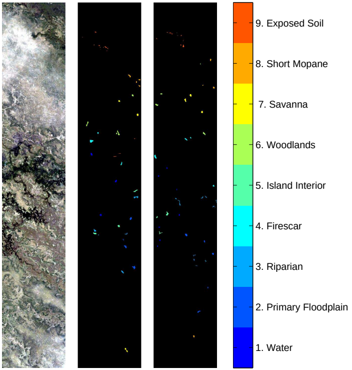
- Use Matérn covariance function for membership variable z.



(a) Matérn, $\nu = 3/2$ (b) Squared exponential

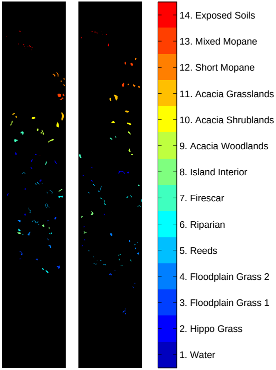
Botswana 9-class Data

	Class name	# Training	# Test
1	Water	158	139
2	Primary Floodplain	228	209
3	Riparian	237	211
4	Firescar	178	176
5	Island interior	183	154
6	Woodlands	199	158
7	Savanna	162	168
8	Short mopane	124	115
9	Exposed soil	111	104



Botswana 14-class Data

	Class name	# Training	# Test
1	Water	270	126
2	Hippo grass	101	162
3	Floodplain grasses 1	251	158
4	Floodplain grasses 2	215	165
5	Reeds	269	168
6	Riparian	269	211
7	Firescar	259	176
8	Island interior	203	154
9	Acacia woodlands	314	151
10	Acacia shrublands	248	190
11	Acacia grasslands	305	358
12	Short mopane	181	153
13	Mixed mopane	268	133
14	Exposed soils	95	89



Results

GP-EM Results

	ML	EM	GP-ML	GP-EM
9-class	87.24 %	93.72 %	90.03 %	**98.81 %**
14-class	74.30 %	85.36 %	82.76 %	**95.87 %**

Results from previous studies

	Iso-SVM [Chen et al '06]	MR-Manifold [Kim et al '07]	SkNN [Chen et al 05]
Accuracy	80.7 %	86.9 %	87.5%

(a) 9-class results

	KT-BHC [Rajan et al '06]	BH-SVM [Chen et al '04]
Overall accuracy	84.42 %	72.1 %

(b) 14-class results

Appendix

Discovering Unknown Classes

Limitations of GP-EM

- Works only if the number of components is known.
- Not realistic for large-scale remote sensing applications.

Mixture Models with Unknown Number of Components

- Parametric model: estimate number of components
- Non-parametric model: DPMM

Discovering Unknown Classes

Limitations of GP-EM

- Works only if the number of components is known.
- Not realistic for large-scale remote sensing applications.

Mixture Models with Unknown Number of Components

- Parametric model: estimate number of components
- Non-parametric model: DPMM

SESSAMM: SEmi-Supervised Spatially Adaptive Mixture Model

Infinite Mixture of Gaussians

- Likelihood for each component

$$f(\mathbf{x}_i|\theta_c) \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{c}, \gamma\Sigma_c)$$

- $(\boldsymbol{\mu}_\mathbf{c}, \Sigma_c)$: posteriors with normal-inverse-Wishart priors.
- γ: approximation factor for a moment-matched Gaussian

Infinite Mixture of Spatially Adaptive Gaussians

$$f^s(\mathbf{x}|\mathbf{s}, \theta_c) \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c^s(\mathbf{s}), \gamma\Sigma_c^s)$$

Spatially Dependent DPMM

Standard DPMM

- Assumes i.i.d indicator variables.

$$\lim_{k \to \infty} p(z_i = c | \mathbf{z}_{-i}, \alpha) = \frac{n_c^{-i}}{n - \alpha - 1} \qquad \forall c, \quad n_c^{-i} > 0 \ ,$$

Spatially Dependent DPMM

- Strong spatial correlation between indicator variables.

$$p(z_i = c | \mathbf{z}_{-i}, \mathbf{s}) \sim GP(\mathbf{s})$$

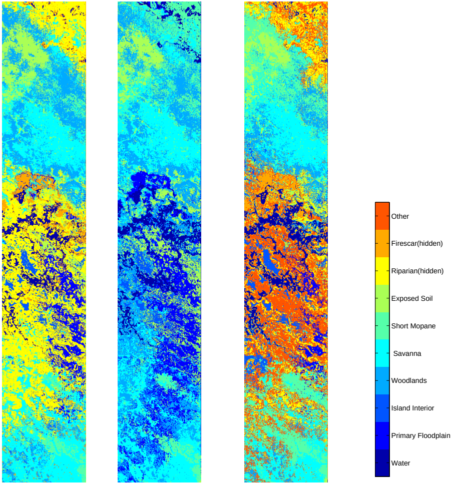
- Dependent Dirichlet Processes [MacEachern 1999]

Result: Clustering Scores, Botswana

# removed classes		# clusters	Purity (known)	Purity (other)	Purity (overall)	NMI
0	DPMM	10.1 (0.316)	0.941 (0.004)	0.717 (0.073)	0.928 (0.006)	0.871 (0.006)
	SESSAMM	12.6 (0.699)	**0.990** (0.013)	**0.881** (0.040)	**0.976** (0.021)	**0.892** (0.013)
1	DPMM	9.70 (1.12)	0.891 (0.047)	0.774 (0.134)	0.877 (0.040)	0.853 (0.019)
	SESSAMM	12.9 (1.12)	**0.975** (0.033)	**0.860** (0.096)	**0.976** (0.067)	**0.882** (0.037)
2	DPMM	8.90 (0.316)	0.856 (0.054)	0.660 (0.113)	0.815 (0.038)	0.822 (0.028)
	SESSAMM	12.3 (0.949)	**0.960** (0.057)	**0.804** (0.113)	**0.908** (0.053)	**0.852** (0.032)
3	DPMM	8.40 (0.516)	0.831 (0.079)	0.656 (0.159)	0.776 (0.032)	0.794 (0.023)
	SESSAMM	15.8 (0.919)	**0.949** (0.061)	**0.866** (0.101)	**0.965** (0.067)	**0.835** (0.046)
4	DPMM	7.00 (0.667)	0.778 (0.095)	0.562 (0.131)	0.693 (0.039)	0.774 (0.028)
	SESSAMM	14.4 (2.17)	**0.910** (0.074)	**0.884** (0.088)	**0.945** (0.079)	**0.824** (0.041)

$$NMI(X, Y) = \frac{I(X; Y)}{\sqrt{H(X)H(Y)}}$$

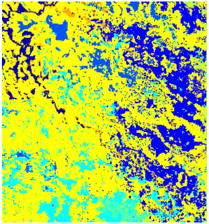
Classified map: Botswana



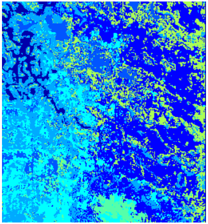
(a) 9 class (b) 7 class (c) SESSAMM

Detailed map: Botswana

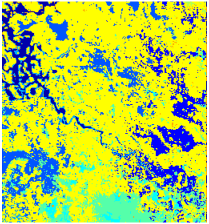
(a) RGB image



(b) 9 class



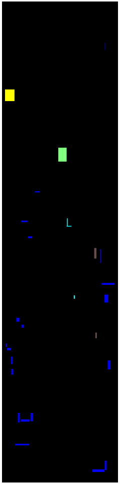
(c) 7 class



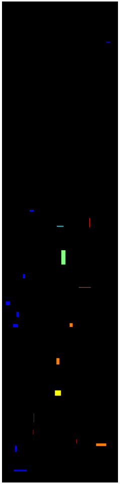
(d) SESSAMM, color adjusted

Data: DC Mall

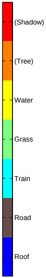
(a) Simulated IR



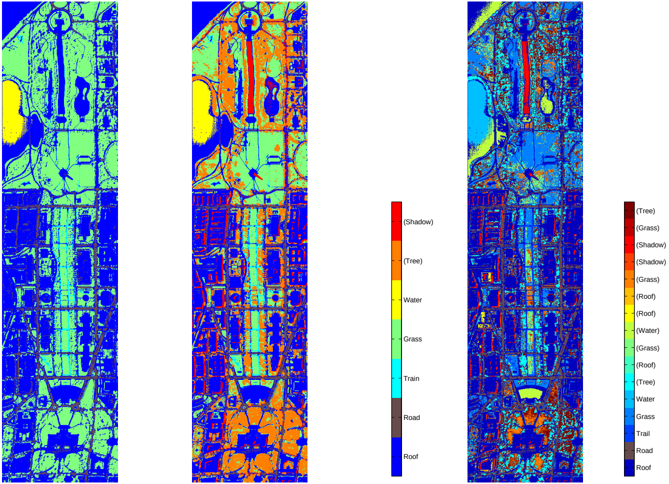
(b) Training



(c) Unlabeled



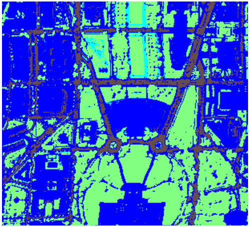
Result: DC Mall

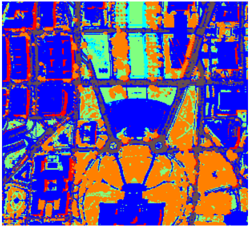


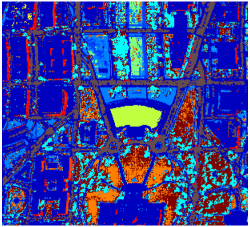
(a) Training (b)Training + Tree, Shadow (c) SESSAMM

Detailed Result: DC Mall


(a) Simulated IR


(b) Training (5 class)


(c) Training + Tree, Shadow


(d) SESSAMM

Efficient Computation of Gaussian Processes

- $(\sigma_f^2 K_{SS} + \sigma_\epsilon^2 I)^{-1}$ requires $O(n^3)$ $\times$ d GPs per class.
- K_{SS} is a positive definite matrix; hence an eigen-decomposition $K_{SS} = V \Lambda V^T$ exists.

$$
\begin{aligned}
(\sigma_f^2 K_{SS} + \sigma_\epsilon^2 I)^{-1} &= (\sigma_f^2 V \Lambda V^T + \sigma_\epsilon^2 V I V^T)^{-1} \\
&= V(\sigma_f^2 \Lambda_k + \sigma_\epsilon^2 I)^{-1} V^T \\
&= V \ diag\left(\frac{1}{\sigma_f^2 \lambda_k + \sigma_\epsilon^2}\right) \ V^T \ , \quad k = 1, ..., n \ .
\end{aligned}
$$

- $O(n^2)$ once we have the eigen-decomposition.

GP-EM, M-step

Spatially Adaptive Mean

$$\hat{\mathbf{x}}_k = \mathbf{x}_k - \boldsymbol{\mu}_i^c, \quad \text{where} \quad \boldsymbol{\mu}_i^c = \frac{\sum_{k=1}^n z_{i,k}^t \mathbf{x}_k}{\sum_{k=1}^n z_{i,k}^t} .$$

$$\boldsymbol{\mu}_{i,\cdot}^j = \sigma_{f_j}^2 K_{SS} [\sigma_{f_j}^2 K_{SS} + diag(\sigma_{\epsilon_j}^2 / z_{i,k}^t)]^{-1} \hat{\mathbf{x}}^j \ , \quad 1 \le j \le d$$

Updates for mean and covariance

$$\boldsymbol{\mu}_{i,k}^{t+1} = \boldsymbol{\mu}_{i,k} + \boldsymbol{\mu}_i^c \ ,$$

$$\Sigma_i^{t+1} = \frac{\sum_{k=1}^n z_{i,k}^t (\hat{\mathbf{x}}_k - \boldsymbol{\mu}_{i,k}^{t+1})(\hat{\mathbf{x}}_k - \boldsymbol{\mu}_{i,k}^{t+1})^T}{\sum_{k=1}^n z_{i,k}^t} .$$

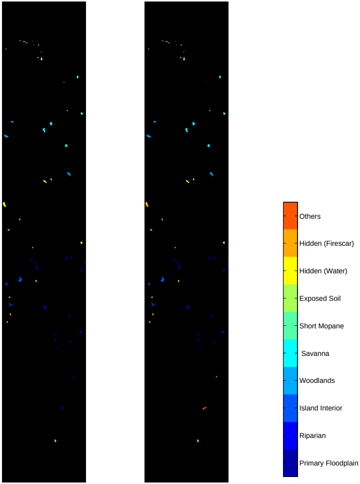
Gibbs Sampling for SESSAMM

- k_0 components for labeled data, $k = (k_0 + 1)$-th component for all unlabeled data.
  1. For each $\mathbf{x}_i \in X_u$, do
     1. Update parameters for each component with $\mathbf{x}_i$ removed.
     2. Calculate the likelihood of each component:

$$
\begin{aligned}
l_c &= f^s(\mathbf{x}|\mathbf{s}, \theta_c) \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c^s(\mathbf{s}), \gamma\Sigma_c^s), \quad 1 \leq c \leq k_0 \ , \\
l_c &= f(\mathbf{x}|\theta_c) \sim \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}_c, \gamma\Sigma_c\right), \qquad k_0 < c \leq k \ .
\end{aligned}
$$

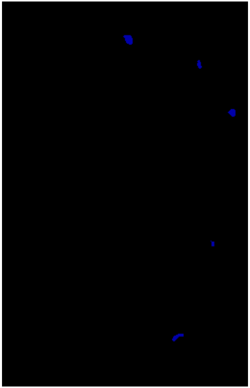
     3. Calculate the likelihood of an unpopulated component, $l_{k+1} = f(\mathbf{x}_i|\theta_0)$.
     4. Calculate spatially adjusted posteriors of z_i, p_c, and normalize.
     5. Draw $z_i \sim \text{Multi}(\frac{1}{Z}p_1 l_1, \ldots, \frac{1}{Z}p_{k+1}l_{k+1})$, where $Z = \sum_{c=1}^{k+1} p_c l_c$.
     6. If $z_i = k+1$, $k \leftarrow k+1$.
  2. Re-sample α and repeat.

Result: Botswana Test Data

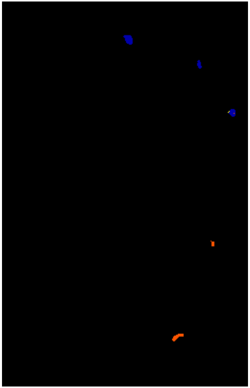


(a) Groundtruth (b) SESSAMM

Result: Primary Floodplain



(a) Groundtruth



(b) SESSAMM

Result: Water

(a) Groundtruth

(b) SESSAMM