Discriminative Graphical Models for Sparsity-Based Hyperspectral Target Detection

Umamahesh Srinivas[1] Yi Chen[2]

Vishal Monga[1] Nasser Nasrabadi[3] Trac Tran[2]

[1]Pennsylvania State University, University Park, USA

[2]The Johns Hopkins University, Baltimore, USA

[3]U.S. Army Research Laboratory, Adelphi, USA

IEEE International Geoscience and Remote Sensing Symposium

July 24, 2012

Outline

1. Overview: Hyperspectral target detection
2. Sparse representation-based target detection
3. Contribution: Discriminative graphical models for target detection
4. Experiments and results

Hyperspectral Imaging (HSI)

- Materials reflect, absorb, and emit electromagnetic energy at different wavelengths in a specific manner
- Imaging spectrometer measures reflectance over many contiguous spectral wavelength bands
- Applications: military, agriculture, mineralogy, etc.

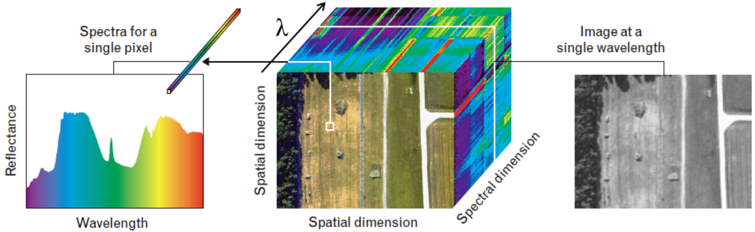


Figure : Illustration of a HSI data cube[1].

- Spectral range: 400 nm - 2500 nm (visible to near infra-red)

[1] Manolakis et al., Lincoln Lab. Journal, 2003

Hyperspectral target detection

- Binary hypothesis testing problem (at each pixel independently):

$$\begin{aligned} H_0 &: \quad \text{target absent} \\ H_1 &: \quad \text{target present.} \end{aligned}$$

- Likelihood ratio test:

$$L(\boldsymbol{x}) = \frac{p\left(\boldsymbol{x}|H_1\right)}{p\left(\boldsymbol{x}|H_0\right)} \underset{H_0}{\overset{H_1}{\gtrless}} \tau.$$

- Issues:
  - Spectral variability $\rightarrow$ atmospheric conditions, sensor noise, material composition
  - Disproportionately small number of target pixels in scene

HSI Target detection: Prior work

- Adaptive matched filter (AMF)[2]

$$
\begin{aligned}
H_0 &: \quad \boldsymbol{x} = \boldsymbol{n} \\
H_1 &: \quad \boldsymbol{x} = a\boldsymbol{s} + \boldsymbol{n}.
\end{aligned}
$$

$$L(\boldsymbol{x}) = \frac{\boldsymbol{s}^T \hat{\boldsymbol{C}}_n^{-1} \boldsymbol{x}}{\boldsymbol{s}^T \hat{\boldsymbol{C}}_n^{-1} \boldsymbol{s}},$$

where $\boldsymbol{s} \rightarrow$ target spectral signature, $\hat{\boldsymbol{C}}_n \rightarrow$ background covariance.

[2] Robey et al., IEEE Trans. Aerosp. Electron. Syst., 1992

[3] Scharf et al., IEEE Trans. Signal Process., 1994

HSI Target detection: Prior work

- Adaptive matched filter (AMF)[2]

$$\begin{aligned} H_0 &: \boldsymbol{x} = \boldsymbol{n} \\ H_1 &: \boldsymbol{x} = a\boldsymbol{s} + \boldsymbol{n}. \end{aligned}$$

$$L(\boldsymbol{x}) = \frac{\boldsymbol{s}^T \hat{\boldsymbol{C}}_n^{-1} \boldsymbol{x}}{\boldsymbol{s}^T \hat{\boldsymbol{C}}_n^{-1} \boldsymbol{s}},$$

where $\boldsymbol{s} \rightarrow$ target spectral signature, $\hat{\boldsymbol{C}}_n \rightarrow$ background covariance.

- Matched subspace detector (MSD)[3]

$$\begin{aligned} H_0 &: \boldsymbol{x} = \boldsymbol{B\zeta} + \boldsymbol{n} \\ H_1 &: \boldsymbol{x} = \boldsymbol{T\theta} + \boldsymbol{B\zeta} + \boldsymbol{n}. \end{aligned}$$

$$L(\boldsymbol{x}) = \frac{\boldsymbol{x}^T \boldsymbol{P}_B^{\perp} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{P}_{TB}^{\perp} \boldsymbol{x}},$$

where $\boldsymbol{P}_B^{\perp} = \boldsymbol{I} - \boldsymbol{P}_B$ and $\boldsymbol{P}_{SB}^{\perp} = \boldsymbol{I} - \boldsymbol{P}_{TB}$ are projection matrices.

---

[2] Robey et al., IEEE Trans. Aerosp. Electron. Syst., 1992

[3] Scharf et al., IEEE Trans. Signal Process., 1994

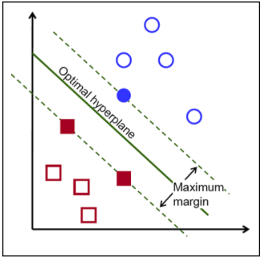
HSI Target detection and classification: Prior work

- Support vector machines (SVM)[4]

Margin-maximizing hyperplane:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{s}_i, \boldsymbol{x}) + \beta,$$

where $\boldsymbol{s}_i \rightarrow$ support vectors,
$y_i \in \{-1, +1\}$, $K \rightarrow$ kernel.



[4] Melgani et al., IEEE Trans. Geosci. Remote Sens., 2004
[5] Camps-Valls et al., IEEE Geosci. Remote Sens. Lett, 2006

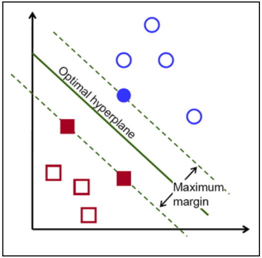
HSI Target detection and classification: Prior work

- Support vector machines (SVM)[4]

Margin-maximizing hyperplane:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{s}_i, \boldsymbol{x}) + \beta,$$

where $\boldsymbol{s}_i \rightarrow$ support vectors, $y_i \in \{-1, +1\}$, $K \rightarrow$ kernel.



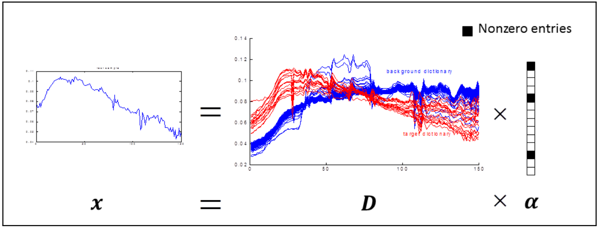
- Composite kernel SVM[5]
  - Fusion of spatial and spectral information

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mu K_s(\boldsymbol{x}_i^s, \boldsymbol{x}_j^s) + (1-\mu) K_w(\boldsymbol{x}_i^w, \boldsymbol{x}_j^w),$$

where $\mu \in [0, 1]$, $x_i^w \rightarrow$ spectral pixel, $x_i^s \rightarrow$ spatial feature from a local neighborhood.

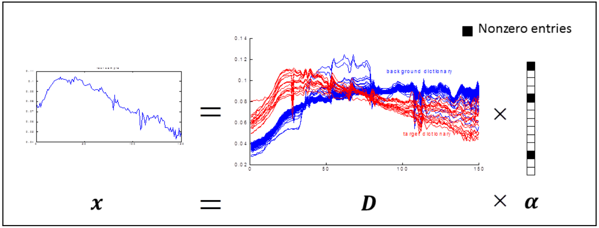
[4] Melgani et al., IEEE Trans. Geosci. Remote Sens., 2004
[5] Camps-Valls et al., IEEE Geosci. Remote Sens. Lett, 2006

Sparsity-based target detection[6]



[6] Chen et al., IEEE J. Sel. Topics Signal Process., 2011

Sparsity-based target detection[6]

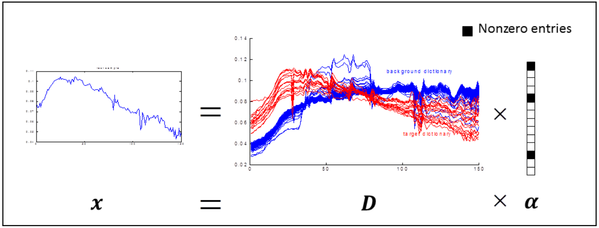


$$\boldsymbol{x} \approx \boldsymbol{D}_t\boldsymbol{\alpha}_t + \boldsymbol{D}_b\boldsymbol{\alpha}_b = \underbrace{[\boldsymbol{D}_t \quad \boldsymbol{D}_b]}_{\boldsymbol{D}} \begin{bmatrix} \boldsymbol{\alpha}_t \\ \boldsymbol{\alpha}_b \end{bmatrix} = \boldsymbol{D}\boldsymbol{\alpha}$$

- $\boldsymbol{D}_t$: target dictionary (matrix with columns as target spectra)
- $\boldsymbol{D}_b$: background dictionary (columns $\rightarrow$ background spectra)

[6] Chen et al., IEEE J. Sel. Topics Signal Process., 2011

Sparsity-based target detection[6]



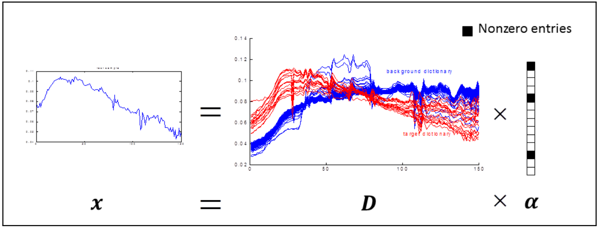
$$\boldsymbol{x} \approx \boldsymbol{D}_t\boldsymbol{\alpha}_t + \boldsymbol{D}_b\boldsymbol{\alpha}_b = \underbrace{[\boldsymbol{D}_t \quad \boldsymbol{D}_b]}_{\boldsymbol{D}} \begin{bmatrix} \boldsymbol{\alpha}_t \\ \boldsymbol{\alpha}_b \end{bmatrix} = \boldsymbol{D}\boldsymbol{\alpha}$$

- $\boldsymbol{D}_t$: target dictionary (matrix with columns as target spectra)
- $\boldsymbol{D}_b$: background dictionary (columns $\rightarrow$ background spectra)

Sparse recovery: $\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_0$ subject to $\|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2 \leq \epsilon$

[6] Chen et al., IEEE J. Sel. Topics Signal Process., 2011

Sparsity-based target detection[6]



$$\boldsymbol{x} \approx \boldsymbol{D}_t\boldsymbol{\alpha}_t + \boldsymbol{D}_b\boldsymbol{\alpha}_b = \underbrace{[\boldsymbol{D}_t \quad \boldsymbol{D}_b]}_{\boldsymbol{D}} \begin{bmatrix} \boldsymbol{\alpha}_t \\ \boldsymbol{\alpha}_b \end{bmatrix} = \boldsymbol{D}\boldsymbol{\alpha}$$

- $\boldsymbol{D}_t$: target dictionary (matrix with columns as target spectra)
- $\boldsymbol{D}_b$: background dictionary (columns $\rightarrow$ background spectra)

Sparse recovery: $\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_0$ subject to $\|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2 \leq \epsilon$

- Class assignment:

$$R(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{D}_b\hat{\boldsymbol{\alpha}}_b\|_2 - \|\boldsymbol{x} - \boldsymbol{D}_t\hat{\boldsymbol{\alpha}}_t\|_2$$

$$\boldsymbol{x} = \begin{cases} \text{target}, & \text{if } R(\boldsymbol{x}) > \delta \\ \text{background}, & \text{otherwise.} \end{cases}$$

[6] Chen et al., IEEE J. Sel. Topics Signal Process., 2011

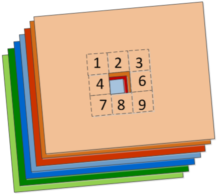
Joint sparsity model for HSI target detection[7,8]

- Homogeneous regions in HSI → neighboring pixels strongly correlated → same sparsity pattern, weighted differently

[7] Chen et al., IEEE Geosci. Remote Sens. Lett., 2011
[8] Chen et al., IEEE Trans. Geosci. Remote Sens., 2011

Joint sparsity model for HSI target detection[7,8]

- Homogeneous regions in HSI $\rightarrow$ neighboring pixels strongly correlated $\rightarrow$ same sparsity pattern, weighted differently



$$\boldsymbol{x}_i = \boldsymbol{D}\boldsymbol{\alpha}_i, i = 1, 2, \ldots, T$$

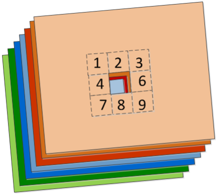
$$\begin{aligned} \boldsymbol{X} &= \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_T \end{bmatrix} \\ &= \boldsymbol{D} \underbrace{\begin{bmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \cdots & \boldsymbol{\alpha}_T \end{bmatrix}}_{\boldsymbol{S}} = \boldsymbol{D}\boldsymbol{S} \end{aligned}$$

[7] Chen et al., IEEE Geosci. Remote Sens. Lett., 2011

[8] Chen et al., IEEE Trans. Geosci. Remote Sens., 2011

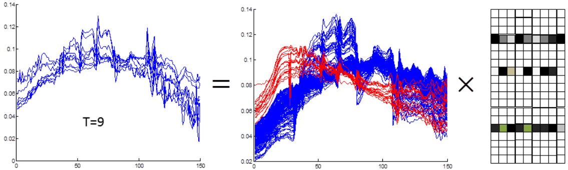
Joint sparsity model for HSI target detection[7,8]

- Homogeneous regions in HSI $\rightarrow$ neighboring pixels strongly correlated $\rightarrow$ same sparsity pattern, weighted differently



$$\boldsymbol{x}_i = \boldsymbol{D}\boldsymbol{\alpha}_i, i = 1, 2, \ldots, T$$

$$\begin{aligned} \boldsymbol{X} &= \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_T \end{bmatrix} \\ &= \boldsymbol{D} \underbrace{\begin{bmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \cdots & \boldsymbol{\alpha}_T \end{bmatrix}}_{\boldsymbol{S}} = \boldsymbol{D}\boldsymbol{S} \end{aligned}$$



$$\boldsymbol{X} = \boldsymbol{D} \times \boldsymbol{S}$$

$$\|\boldsymbol{S}\|_{\text{row},0} = 3$$

Figure : $\boldsymbol{S}$ has only a few nonzero rows.

[7] Chen et al., IEEE Geosci. Remote Sens. Lett., 2011

[8] Chen et al., IEEE Trans. Geosci. Remote Sens., 2011

Joint sparse recovery

$$\hat{\boldsymbol{S}} = \arg\min \|\boldsymbol{D}\boldsymbol{S} - \boldsymbol{X}\|_F \quad \text{subject to} \quad \|\boldsymbol{S}\|_{\text{row},0} \leq K_0$$

- $\|\boldsymbol{S}\|_{\text{row},0} \rightarrow$ number of nonzero rows in $\boldsymbol{S}$

[9] Tropp et al., Signal Processing, 2006

[10] Tropp, Signal Processing, 2006

Joint sparse recovery

$$\hat{\boldsymbol{S}} = \arg\min \|\boldsymbol{D}\boldsymbol{S} - \boldsymbol{X}\|_F \quad \text{subject to} \quad \|\boldsymbol{S}\|_{\text{row},0} \leq K_0$$

- $\|\boldsymbol{S}\|_{\text{row},0} \rightarrow$ number of nonzero rows in $\boldsymbol{S}$
- Recovery algorithms:
  - Simultaneous versions of greedy pursuit algorithms (SOMP) [9]
  - Convex relaxation of $\|\boldsymbol{S}\|_{\text{row},0}$ to mixed ℓ_1/ℓ_2-norm[10]

[9] Tropp et al., Signal Processing, 2006

[10] Tropp, Signal Processing, 2006

Joint sparse recovery

$$\hat{\boldsymbol{S}} = \arg\min \|\boldsymbol{D}\boldsymbol{S} - \boldsymbol{X}\|_F \quad \text{subject to} \quad \|\boldsymbol{S}\|_{\text{row},0} \leq K_0$$

- $\|\boldsymbol{S}\|_{\text{row},0} \rightarrow$ number of nonzero rows in $\boldsymbol{S}$
- Recovery algorithms:
  - Simultaneous versions of greedy pursuit algorithms (SOMP) [9]
  - Convex relaxation of $\|\boldsymbol{S}\|_{\text{row},0}$ to mixed ℓ_1/ℓ_2-norm[10]
- Class decision using total reconstruction residual:

$$r(\boldsymbol{x}) = \|\boldsymbol{X} - \boldsymbol{D}_b\hat{\boldsymbol{S}}_b\|_F - \|\boldsymbol{X} - \boldsymbol{D}_t\hat{\boldsymbol{S}}_t\|_F$$

---

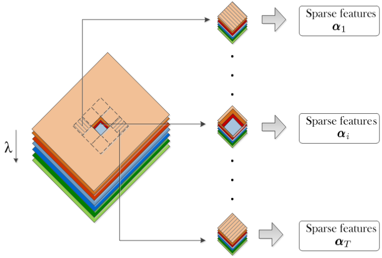
[9] Tropp et al., Signal Processing, 2006

[10] Tropp, Signal Processing, 2006

Summary: Opportunities and challenges

1. Feature representations of pixels in a local neighborhood are statistically correlated
   - How to mine the class-conditional correlations among distinct feature representations?
2. Practical concern: Availability of very few target pixels
   - How to learn class-conditional models under limited training?
3. Use of reconstruction residual for detection
   - Design of truly discriminative classifiers

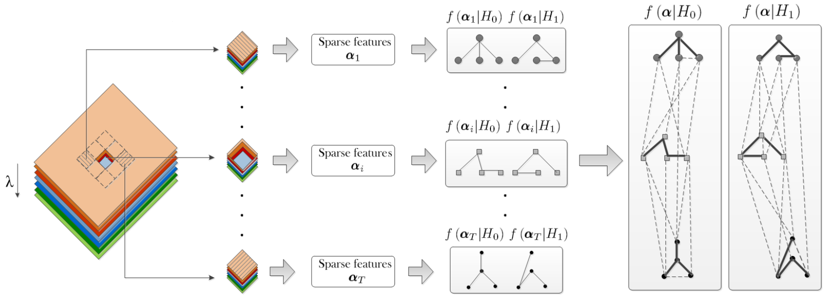
Spatio-spectral sparsity via discriminative graphical models



Two-stage framework:

1. Extract multiple local sparse feature representations, which are conditionally correlated per class

Spatio-spectral sparsity via discriminative graphical models



Two-stage framework:

1. Extract multiple local sparse feature representations, which are conditionally correlated per class
2. Mine dependencies between different features via boosting on discriminative graphs

Probabilistic graphical models: A brief review

- Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ defined by a set of nodes $\mathcal{V} = \{1, \ldots, n\}$, and a set of edges $\mathcal{E} \subset \binom{\mathcal{V}}{2}$.
- Graphical model: Random vector defined on a graph; nodes represent random variables, edges reveal conditional dependencies.
- Graph structure defines factorization of joint probability distribution

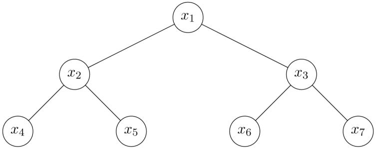


Figure : Tree - connected acyclic graph.

$$f(\boldsymbol{x}) = f(x_1)f(x_2|x_1)f(x_3|x_1)f(x_4|x_2)f(x_5|x_2)f(x_6|x_3)f(x_7|x_3).$$

$$\Uparrow$$

$$\boxed{f\left(\boldsymbol{\alpha}_i|H_0\right),\ f\left(\boldsymbol{\alpha}_i|H_1\right)}$$

Learning graphical models

- Generative learning: Single graph to minimize approximation error[11]

$$\text{Given } p, \text{ find } \hat{p} = \arg \min_{\hat{p} \text{ is a tree}} D(p||\hat{p}).$$

$$\left(D(p||\hat{p}) := \int p(\boldsymbol{x}) \log \left(\frac{p(\boldsymbol{x})}{\hat{p}(\boldsymbol{x})} \right) d\boldsymbol{x} \rightarrow \text{ KL-divergence.} \right)$$

- Discriminative learning: Simultaneously learn a pair of graphs to approximately minimize classification error[12]

[11] Chow et al., IEEE Trans. Inf. Theory, 1968

[12] Tan et al., IEEE Trans. Signal Process., 2010

Learning graphical models

- Generative learning: Single graph to minimize approximation error[11]

$$\text{Given } p,\ \text{find } \hat{p} = \arg \min_{\hat{p} \text{ is a tree}} D(p||\hat{p}).$$

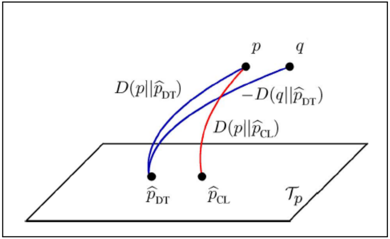
$$\left(D(p||\hat{p}) := \int p(\boldsymbol{x}) \log \left(\frac{p(\boldsymbol{x})}{\hat{p}(\boldsymbol{x})} \right) d\boldsymbol{x} \rightarrow \text{ KL-divergence.} \right)$$

- Discriminative learning: Simultaneously learn a pair of graphs to approximately minimize classification error[12]

Tree-approximate J-divergence:

$$\widehat{J}(\widehat{p}, \widehat{q}; p, q) := \int_{\Omega \subset \mathcal{X}^n} (p(\boldsymbol{x}) - q(\boldsymbol{x})) \log \left(\frac{\widehat{p}(\boldsymbol{x})}{\widehat{q}(\boldsymbol{x})} \right) d\boldsymbol{x}.$$

$$(\widehat{p}, \widehat{q}) = \arg \max_{\widehat{p} \in \mathcal{T}_p, \widehat{q} \in \mathcal{T}_q} \widehat{J}(\widehat{p}, \widehat{q}; p, q).$$



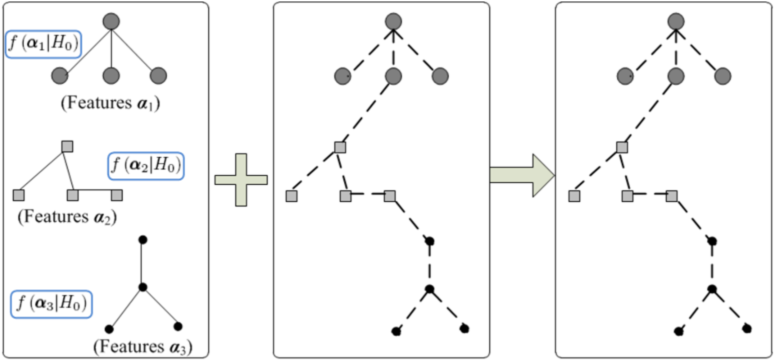
(Figure courtesy Tan et al.)

[11] Chow et al., IEEE Trans. Inf. Theory, 1968

[12] Tan et al., IEEE Trans. Signal Process., 2010

Learning discriminative graphs: An illustration[13]

Iteration 1:

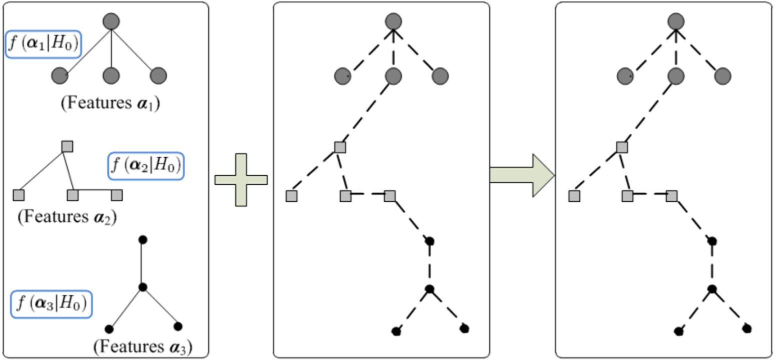


(a) Initial graph (b) Newly-learned tree (c) Augmented graph

[13] Shown for distribution p; graph for q learnt analogously.

Learning discriminative graphs: An illustration[13]

Iteration 1:



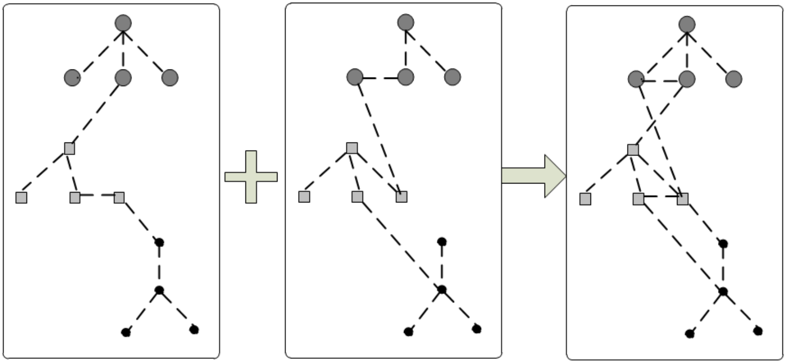
(a) Initial graph (b) Newly-learned tree (c) Augmented graph

Re-weighting of training samples (boosting) $\rightarrow$ learn another tree ...

[13] Shown for distribution p; graph for q learnt analogously.

Learning discriminative graphs: An illustration

Iteration 2:

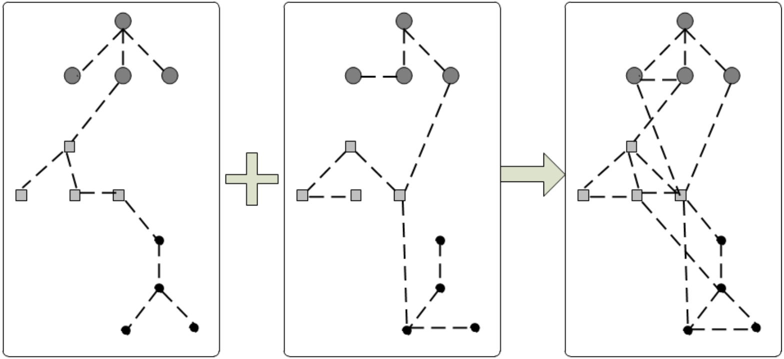


(a) Initial graph (b) Newly-learned tree (c) Augmented graph

Newly introduced edges crucial for capturing correlations amongst distinct signal representations.

Learning discriminative graphs: An illustration

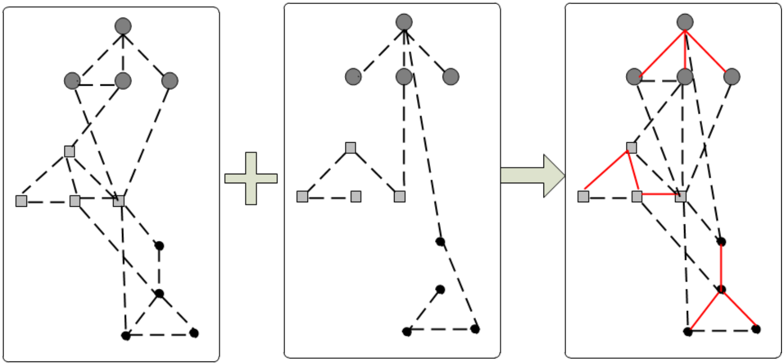
Iteration 3:



(a) Initial graph (b) Newly-learned tree (c) Augmented graph

Learning discriminative graphs: An illustration

Iteration 4:



(a) Initial graph (b) Newly-learned tree (c) Augmented graph

Stopping criterion and class assignment

How many edges to learn?

1. Cross-validation

2. Using the J-divergence:

$$\widehat{J}(\widehat{p},\widehat{q};p,q) := \int_{\Omega\subset\mathcal{X}^n} (p(\boldsymbol{x}) - q(\boldsymbol{x}))\log\left(\frac{\widehat{p}(\boldsymbol{x})}{\widehat{q}(\boldsymbol{x})}\right) d\boldsymbol{x}.$$

Stopping criterion:

Stop after i boosting iterations if:

$$\frac{\widehat{J}^{(i+1)}(\widehat{p},\widehat{q};p,q) - \widehat{J}^{(i)}(\widehat{p},\widehat{q};p,q)}{\widehat{J}^{(i)}(\widehat{p},\widehat{q};p,q)} < \epsilon$$

Stopping criterion and class assignment

How many edges to learn?

1. Cross-validation

2. Using the J-divergence:

$$\widehat{J}(\widehat{p}, \widehat{q}; p, q) := \int_{\Omega \subset \mathcal{X}^n} (p(\boldsymbol{x}) - q(\boldsymbol{x})) \log\left(\frac{\widehat{p}(\boldsymbol{x})}{\widehat{q}(\boldsymbol{x})}\right) d\boldsymbol{x}.$$

Stopping criterion:

Stop after i boosting iterations if:

$$\frac{\widehat{J}^{(i+1)}(\widehat{p}, \widehat{q}; p, q) - \widehat{J}^{(i)}(\widehat{p}, \widehat{q}; p, q)}{\widehat{J}^{(i)}(\widehat{p}, \widehat{q}; p, q)} < \epsilon$$

Class label assignment:

$$\text{Class}(\boldsymbol{x}) = \begin{cases} \text{Target} & \text{if } \log\left(\frac{\hat{f}(\boldsymbol{\alpha}|H_1)}{\hat{f}(\boldsymbol{\alpha}|H_0)}\right) \geq \tau \\ \text{Background} & \text{if } \log\left(\frac{\hat{f}(\boldsymbol{\alpha}|H_1)}{\hat{f}(\boldsymbol{\alpha}|H_0)}\right) < \tau. \end{cases}$$

Experiment I

- HYDICE desert radiance II (DR-II) data collection[14]

- 150 spectral bands: 400nm - 2500nm; 6 targets
- Target dictionary $\boldsymbol{D}_t$: 18 training spectra from leftmost target
- Background dictionary $\boldsymbol{D}_b$: 216 training spectra

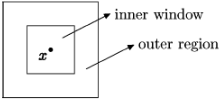


Figure : Dual window for adaptive background selection[15]

[14] Basedow et al., SPIE Conf. Algorithms Technol. Multispectral, Hyperspectral, Ultraspectral Imagery XV, 1995

[15] Chen et al., IEEE J. Sel. Topics Signal Process., 2011

Results: Confusion matrix

Methods compared:

1. MSD: matched subspace detector[16]
2. SVM-CK: composite kernel SVM[17]
3. SOMP: joint sparsity model[18]
4. LSGM: proposed local-sparsity-graphical-model approach

Table : Confusion matrix for DR-II hyperspectral image.

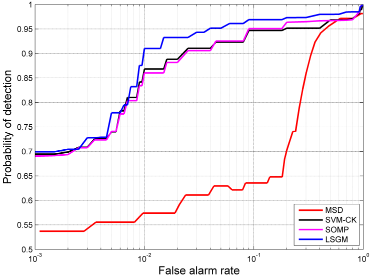
Class	Target	Background	Method
Target	0.6356	0.3644	MSD
	0.9469	0.0531	SVM-CK
	0.9508	0.0492	SOMP
	0.9690	**0.0310**	**LSGM**
Background	0.0157	0.9843	MSD
	0.0075	0.9925	SVM-CK
	0.0077	0.9923	SOMP
	0.0074	**0.9926**	**LSGM**

[16] Scharf et al., IEEE Trans. Signal Process., 1994

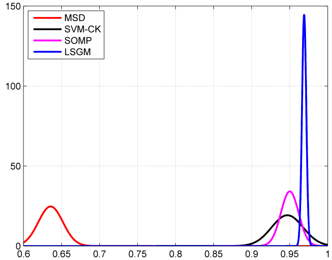
[17] Camps-Valls et al., IEEE Geosci. Remote Sens. Lett, 2006

[18] Chen et al., IEEE Trans. Geosci. Remote Sens., 2011

Results: Receiver operating characteristic (ROC)



(a) ROC for DR-II.



(b) Density function of detection rates (multiple training runs).

Experiment II

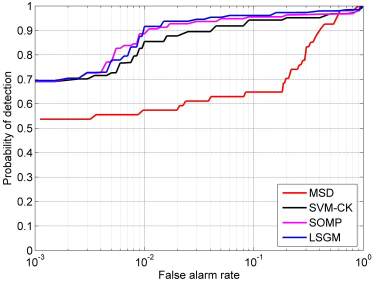
- HYDICE forest radiance I (FR-I) data collection

- 150 spectral bands: 400nm - 2500nm; 14 targets

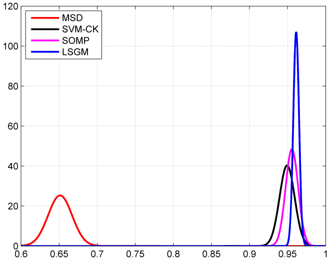
Table : Confusion matrix for FR-I hyperspectral image.

Class	Target	Background	Method
Target	0.6512	0.3488	MSD
	0.9493	0.0507	SVM-CK
	0.9556	0.0444	SOMP
	0.9612	**0.0388**	**LSGM**
Background	0.0239	0.9761	MSD
	0.0090	0.9910	SVM-CK
	0.0097	0.9903	SOMP
	0.0086	**0.9914**	**LSGM**

Results: ROC



(a) ROC for FR-I.



(b) Density function of detection rates (multiple training runs).

Conclusions

- Probabilistic graphical model framework for hyperspectral target detection
  - Incorporates spatio-spectral notion of sparsity via joint sparsity model
  - Explicitly captures conditional correlations between local sparse features (instead of using reconstruction residuals) for better discrimination.

* Journal version (HSI classification) accepted to IEEE Geosci. Remote Sens. Lett., June 2012

Thank you

Questions?

Backup Slides

Edge weights:

$$\psi_{i,j}^{p} := \mathbb{E}_{\widetilde{p}_{i,j}}\left[\log \frac{\widetilde{p}_{i,j}}{\widetilde{p}_i \widetilde{p}_j}\right] - \mathbb{E}_{\widetilde{q}_{i,j}}\left[\log \frac{\widetilde{p}_{i,j}}{\widetilde{p}_i \widetilde{p}_j}\right]$$

$$\psi_{i,j}^{q} := \mathbb{E}_{\widetilde{q}_{i,j}}\left[\log \frac{\widetilde{q}_{i,j}}{\widetilde{q}_i \widetilde{q}_j}\right] - \mathbb{E}_{\widetilde{p}_{i,j}}\left[\log \frac{\widetilde{q}_{i,j}}{\widetilde{q}_i \widetilde{q}_j}\right].$$

Algorithm 1 Discriminative trees (DT)

Given: Training sets $\mathcal{T}_p$ and $\mathcal{T}_q$.

1: Estimate pairwise statistics $\widetilde{p}_{i,j}(x_i, x_j)$, $\widetilde{q}_{i,j}(x_i, x_j)$ for all edges (i, j).

2: Compute edge weights $\psi_{i,j}^{p}$ and $\psi_{i,j}^{q}$ for all edges (i, j).

3: Find $\mathcal{E}_{\widehat{p}} =$ MWST$(\psi_{i,j}^{p})$ and $\mathcal{E}_{\widehat{q}} =$ MWST$(\psi_{i,j}^{q})$.

4: Get $\widehat{p}$ by projection of $\widetilde{p}$ onto $\mathcal{E}_{\widehat{p}}$; likewise $\widehat{q}$.

5: LRT using $\widehat{p}$ and $\widehat{q}$.

Boosting

Algorithm 2 AdaBoost learning algorithm

1: Input data $(x_i, y_i),\ i = 1, 2, \ldots, N$, where $x_i \in S,\ y_i \in \{-1, +1\}$
2: Initialize $D_1(i) = \frac{1}{N}, i = 1, 2, \ldots, N$
3: For $t = 1, 2, \ldots, T$:

- Train weak learner using distribution D_t
- Determine weak hypothesis $h_t : S \mapsto \mathbb{R}$ with error ϵ_t
- Choose $\beta_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$
- $D_{t+1}(i) = \frac{1}{Z_t} \{D_t(i) \exp(-\beta_t y_i h_t(x_i))\}$, where Z_t is a normalization factor

4: Output soft decision $H(x) = \text{sign} \left[\sum_{t=1}^{T} \beta_t h_t(x) \right]$.

- Distribution of weights over the training set
- In each iteration, weak learner h_t minimizes weighted training error
- Weights on incorrectly classified samples increased $\rightarrow$ slow learners penalized for harder examples.

Learning thicker graphical models

- Final boosted classifier:

$$
\begin{aligned}
H_T(\boldsymbol{x}) &= \operatorname{sgn}\left[\sum_{t=1}^{T} \alpha_t \log\left(\frac{\widehat{p}_t(\boldsymbol{x})}{\widehat{q}_t(\boldsymbol{x})}\right)\right] = \operatorname{sgn}\left[\log \prod_{t=1}^{T} \left(\frac{\widehat{p}_t(\boldsymbol{x})}{\widehat{q}_t(\boldsymbol{x})}\right)^{\alpha_t}\right] \\
&= \operatorname{sgn}\left[\log\left(\frac{\prod_{t=1}^{T}(\widehat{p}_t(\boldsymbol{x}))^{\alpha_t}}{\prod_{t=1}^{T}(\widehat{q}_t(\boldsymbol{x}))^{\alpha_t}}\right)\right] = \operatorname{sgn}\left[\log\left(\frac{\widehat{p}(\boldsymbol{x})}{\widehat{q}(\boldsymbol{x})}\right)\right]
\end{aligned}
$$

Define:

$$
Z_p(\boldsymbol{\alpha}) = Z_p(\alpha_1, \ldots, \alpha_T) = \sum_{\boldsymbol{x}} \hat{p}(\boldsymbol{x}); Z_q(\boldsymbol{\alpha}) = \sum_{\boldsymbol{x}} \hat{q}(\boldsymbol{x})
$$

- Normalized distributions for inference: $\frac{\hat{p}(\boldsymbol{x})}{Z_p(\boldsymbol{\alpha})}, \frac{\hat{q}(\boldsymbol{x})}{Z_q(\boldsymbol{\alpha})}$

$\rightarrow$ Thicker graphical models learnt.

Local Sparsity Graphical Models

Algorithm 3 LSGM (Steps 1-4 offline)

1: **Feature extraction (training):** Compute sparse representations $\boldsymbol{\alpha}_l, l = 1, \ldots, T$ for neighboring pixels of the training data
2: **Initial disjoint graphs:**
Discriminatively learn T pairs of N-node tree graphs $\mathcal{G}_l^t$ and $\mathcal{G}_l^b$ on $\{\boldsymbol{\alpha}_l\}$, for $l = 1, \ldots, T$, obtained from training data
3: Separately concatenate nodes corresponding to the two classes, to generate initial graphs
4: **Boosting on disjoint graphs**: Iteratively thicken initial disjoint graphs via boosting to obtain final graphs $\mathcal{G}^t$ and $\mathcal{G}^b$

{**Online process**}
5: **Feature extraction (test):** Obtain sparse representations $\boldsymbol{\alpha}_l, l = 1, \ldots, T$ in $\mathbb{R}^N$ from test image
6: **Inference**: Classify based on output of the resulting classifier.

Simultaneous Orthogonal Matching Pursuit

Algorithm 4 SOMP

Input: $B \times N$ dictionary matrix $\boldsymbol{D} = [\boldsymbol{d}_1 \ \cdots \ \boldsymbol{d}_N]$, $B \times T$ signal matrix $\boldsymbol{X} = [\boldsymbol{x}_1 \ \cdots \ \boldsymbol{x}_T]$, and number of iterations K

Initialization: residual $\boldsymbol{R}_0 = \boldsymbol{X}$, index set $\Lambda_0 = \phi$, iteration counter $k = 1$

while $k \leq K$ **do**

(1) Find the index of the atom that best approximates all residuals: $\lambda_k = \arg \max_{i=1,\ldots,N} \left\| \boldsymbol{R}_{k-1}^T \boldsymbol{d}_i \right\|_p, p \geq 1$

(2) Update the index set $\Lambda_k = \Lambda_{k-1} \bigcup \{\lambda_k\}$

(3) Compute the orthogonal projector $\boldsymbol{P}_k = \left(\boldsymbol{D}_{\Lambda_k}^T \boldsymbol{D}_{\Lambda_k} \right)^{-1} \boldsymbol{D}_{\Lambda_k}^T \boldsymbol{X} \in \mathbb{R}^{k \times T}$ where $\boldsymbol{D}_{\Lambda_k} \in \mathbb{R}^{B \times k}$ consists of the k atoms in $\boldsymbol{D}$ indexed in Λ_k

(4) Update the residual matrix $\boldsymbol{R}_k = \boldsymbol{X} - \boldsymbol{D}_{\Lambda_k} \boldsymbol{P}_k$

(5) Increment k: $k \leftarrow k + 1$

end while

Output: Index set $\Lambda = \Lambda_K$, the sparse representation $\boldsymbol{S}$ whose nonzero rows index by Λ are the K rows of the matrix $\left(\boldsymbol{D}_{\Lambda_K}^T \boldsymbol{D}_{\Lambda_K} \right)^{-1} \boldsymbol{D}_{\Lambda_K}^T \boldsymbol{X}$